

Manual de procesos y automatización  
de IBM SPSS Modeler 15



*Nota:* Antes de utilizar esta información y el producto que admite, lea la información general en Avisos el p. .

Esta edición se aplica a IBM SPSS Modeler 15 y a todas las versiones y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Capturas de pantalla de productos de Adobe reimpresas con permiso de Adobe Systems Incorporated.

Capturas de pantalla de productos de Microsoft reimpresas con permiso de Microsoft Corporation.

Materiales bajo licencia: Propiedad de IBM

**© Copyright IBM Corporation 1994, 2012.**

Derechos restringidos para los usuarios del gobierno de Estados Unidos: Uso, duplicación o revelación restringidos por GSA ADP Schedule Contract con IBM Corp.

---

# Prefacio

IBM® SPSS® Modeler es el conjunto de programas de minería de datos de IBM Corp. orientado a las empresas. SPSS Modeler ayuda a las organizaciones a mejorar la relación con sus clientes y los ciudadanos a través de la comprensión profunda de los datos. Las organizaciones utilizan la comprensión que les ofrece SPSS Modeler para retener a los clientes más rentables, identificar las oportunidades de venta cruzada, atraer a nuevos clientes, detectar el fraude, reducir el riesgo y mejorar la prestación de servicios del gobierno.

La interfaz visual de SPSS Modeler invita a la pericia empresarial específica de los usuarios, lo que deriva en modelos predictivos más eficaces y la reducción del tiempo necesario para encontrar soluciones. SPSS Modeler ofrece muchas técnicas de modelado tales como pronósticos, clasificaciones, segmentación y algoritmos de detección de asociaciones. Una vez que se crean los modelos, IBM® SPSS® Modeler Solution Publisher permite su distribución en toda la empresa a los encargados de tomar las decisiones o a una base de datos.

## ***Acerca de IBM Business Analytics***

El software IBM Business Analytics ofrece información completa, coherente y precisa en la que los órganos de toma de decisiones confían para mejorar el rendimiento comercial. Un conjunto integral de [inteligencia empresarial](#), [análisis predictivo](#), [rendimiento comercial y gestión de estrategias](#), así como de [aplicaciones de análisis](#) le ofrece una información clara, inmediata e interactiva del rendimiento actual y la capacidad para predecir resultados futuros. En combinación con extensas soluciones sectoriales, prácticas probadas y servicios profesionales, las organizaciones de cualquier tamaño pueden conseguir el máximo de productividad y alcanzar mejores resultados.

Como parte de esta familia, el software de análisis predictivo de IBM SPSS ayuda a las organizaciones a predecir eventos futuros y actuar proactivamente según esa información para lograr mejores resultados comerciales. Los clientes comerciales, gubernamentales y académicos de todo el mundo confían en la tecnología de IBM SPSS como ventaja ante la competencia para atraer, retener y hacer crecer los clientes, reduciendo al mismo tiempo el fraude y reduciendo el riesgo. Al incorporar el software de IBM SPSS en sus operaciones diarias, las organizaciones se convierten en empresas predictivas, capaces de dirigir y automatizar decisiones para alcanzar los objetivos comerciales y lograr una ventaja considerable sobre la competencia. Para obtener más información o contactar con un representante, visite <http://www.ibm.com/spss>.

## ***Asistencia técnica***

La asistencia técnica está disponible para el mantenimiento de los clientes. Los clientes podrán ponerse en contacto con el servicio de asistencia técnica si desean recibir ayuda sobre la utilización de los productos de IBM Corp. o sobre la instalación en los entornos de hardware admitidos. Para ponerse en contacto con el servicio de asistencia, visite el IBM Corp. sitio Web en <http://www.ibm.com/support>. Prepárese para identificarse, identificar a su organización y su acuerdo de asistencia al solicitar asistencia.

---

# Contenido

## **1 Acerca de IBM SPSS Modeler 1**

Productos IBM SPSS Modeler . . . . .	1
IBM SPSS Modeler . . . . .	1
IBM SPSS Modeler Server . . . . .	2
IBM SPSS Modeler Administration Console . . . . .	2
IBM SPSS Modeler Batch . . . . .	2
IBM SPSS Modeler Solution Publisher . . . . .	3
IBM SPSS Modeler Server Adaptadores para IBM SPSS Collaboration and Deployment Services . . . . .	3
Ediciones de IBM SPSS Modeler . . . . .	3
Documentación de IBM SPSS Modeler . . . . .	4
Documentación de SPSS Modeler Professional . . . . .	4
Documentación de SPSS Modeler Premium . . . . .	5
Ejemplos de aplicaciones . . . . .	6
Carpeta Demos . . . . .	7

## **Parte I: Procesamientos y el lenguaje de procesamiento**

## **2 Conceptos básicos del procesamiento 9**

Tipo de procesos . . . . .	9
Procesos de ruta . . . . .	10
Ejemplo de proceso de ruta: Entrenamiento de una red neuronal . . . . .	11
Procesos independientes . . . . .	12
Ejemplo de proceso independiente: Guardar y cargar un modelo . . . . .	13
Ejemplo de proceso independiente: Generar un modelo de selección de características . . . . .	14
Procesos de Supernodo . . . . .	15
Proceso de supernodo de ejemplo . . . . .	16
Ejecución e interrupción de procesos . . . . .	16
Buscar y reemplazar . . . . .	17

## **3 Lenguaje de procesamiento 20**

Conceptos básicos del lenguaje de procesamiento . . . . .	20
Sintaxis de procesos . . . . .	20
Referencia a nodos . . . . .	21

Recuperación de objetos . . . . .	23
Establecimiento del objeto actual . . . . .	23
Apertura de rutas y otros objetos . . . . .	24
Cómo trabajar con varias rutas . . . . .	24
VARIABLES DE PROCESO LOCALES . . . . .	25
Parámetros de ruta, sesión y Supernodo . . . . .	26
Control de la ejecución del proceso . . . . .	28
Operadores en los procesos . . . . .	28
Expresiones CLEM en los procesos . . . . .	28
Inserción de comentarios y continuaciones . . . . .	29
Bloques de texto literal . . . . .	30

## **4 Comandos de procesos**

**31**

Comandos de procesos generales . . . . .	31
execute_all . . . . .	31
execute_script . . . . .	31
exit . . . . .	31
for...endfor . . . . .	32
if...then...else.... . . . .	33
Comando set . . . . .	33
Comando var . . . . .	36
Objetos de nodo . . . . .	37
create NODO . . . . .	37
connect NODO . . . . .	38
delete NODO . . . . .	38
disable NODO . . . . .	38
disconnect NODO . . . . .	39
duplicate NODO . . . . .	39
enable NODO . . . . .	39
execute NODO . . . . .	39
export NODO as ARCHIVO . . . . .	40
flush NODO . . . . .	40
get node NODO . . . . .	41
load node NOMBREARCHIVO . . . . .	41
position NODO . . . . .	41
rename NODO as NUEVONOMBRE . . . . .	42
retrieve node RUTA_REPOSITORIO . . . . .	42
save node NODO as NOMBREARCHIVO . . . . .	42
store node NODO as RUTA_REPOSITORIO . . . . .	43

Objetos del modelo . . . . .	43
Nombres de nugget de modelo . . . . .	43
Evitar nombres duplicados del modelo . . . . .	45
delete model MODELO . . . . .	46
export model MODELO as ARCHIVO . . . . .	46
insert model MODELO . . . . .	47
load model NOMBREARCHIVO . . . . .	48
retrieve model RUTA_REPOSITORIO . . . . .	48
save model MODELO as NOMBREARCHIVO . . . . .	48
store model MODELO as RUTA_REPOSITORIO . . . . .	48
Objetos de ruta . . . . .	49
create stream NOMBREARCHIVO_PORDEFECTO . . . . .	49
close RUTA . . . . .	49
clear stream . . . . .	49
get stream RUTA . . . . .	50
load stream NOMBREARCHIVO . . . . .	50
open stream NOMBREARCHIVO . . . . .	50
retrieve stream RUTA_REPOSITORIO . . . . .	50
save RUTA as NOMBREARCHIVO . . . . .	51
store stream as RUTA_REPOSITORIO . . . . .	51
with stream RUTA . . . . .	52
Objetos del proyecto . . . . .	52
execute_project . . . . .	52
load project NOMBREARCHIVO . . . . .	53
retrieve project RUTA_REPOSITORIO . . . . .	53
save project as NOMBREARCHIVO . . . . .	53
store project as RUTA_REPOSITORIO . . . . .	53
Objetos state . . . . .	53
load state NOMBREARCHIVO . . . . .	53
Objetos de resultado . . . . .	54
value RESULTADO . . . . .	54
Objetos de archivo . . . . .	54
close ARCHIVO . . . . .	54
open ARCHIVO . . . . .	54
write ARCHIVO . . . . .	55
Objetos de resultados . . . . .	55
Nombres de tipo de resultados . . . . .	55
delete output RESULTADO . . . . .	56
export output RESULTADO . . . . .	56
get output RESULTADO . . . . .	56
load output NOMBREARCHIVO . . . . .	57
retrieve output RUTA_REPOSITORIO . . . . .	57

save output SALIDA as NOMBREARCHIVO .....	57
store output SALIDA as RUTA_REPOSITORIO .....	57

## **5 Sugerencias sobre procesos 58**

Modificación de la ejecución de una ruta .....	58
Recorrido en bucle de nodos .....	58
Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository .....	59
Generación de una contraseña codificada .....	61
Comprobación del proceso .....	61
Procesamientos desde la línea de comandos .....	62
Compatibilidad con versiones anteriores .....	62

## **6 Ejemplos de procesos 64**

Informe del nodo Tipo .....	64
Informe de rutas .....	67

## **7 Argumentos de la línea de comandos 70**

Invocación del software .....	70
Uso de argumentos en la línea de comandos .....	70
Combinación de varios argumentos .....	71
Argumentos de conexión con el servidor .....	71
Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository .....	73
Argumentos del sistema .....	73
Argumentos de parámetros .....	75

## **8 CLEM Referencia del lenguaje 76**

Conceptos básicos de la referencia de CLEM .....	76
Tipos de datos de CLEM .....	76
Enteros .....	77
Reales .....	77
Caracteres .....	78
Cadenas .....	78

Listas . . . . .	78
Fields . . . . .	78
Fechas . . . . .	79
Hora . . . . .	80
Operadores de CLEM . . . . .	80
Referencia de funciones . . . . .	82
Convenciones en las descripciones de las funciones . . . . .	83
Funciones de información . . . . .	84
Funciones de conversión . . . . .	85
Funciones de comparación . . . . .	86
Funciones lógicas . . . . .	88
Funciones numéricas . . . . .	89
Funciones trigonométricas . . . . .	90
Funciones de probabilidad . . . . .	91
Operaciones de enteros a nivel de bit . . . . .	91
Funciones aleatorias . . . . .	92
Funciones de cadena . . . . .	93
Funciones SoundEx . . . . .	98
Funciones de fecha y hora . . . . .	98
Funciones de secuencia . . . . .	103
Funciones globales . . . . .	108
Funciones para gestionar los valores vacíos y nulos . . . . .	109
Campos especiales . . . . .	110

## ***Parte II: Referencia de propiedades***

### **9 Referencia de propiedades 114**

Conceptos básicos de referencia de propiedades . . . . .	114
Sintaxis de las propiedades . . . . .	114
Ejemplos de propiedades de nodos y rutas . . . . .	116
Conceptos básicos de las propiedades de nodos . . . . .	117
Propiedades de nodos comunes . . . . .	117



**10 Propiedades de ruta 119**

**11 Propiedades de proyecto 122**

**12 Propiedades de nodos de fuente 123**

Propiedades comunes de nodos de fuente . . . . .	123
Propiedades de cognosimportnode. . . . .	125
Propiedades de databasenode . . . . .	127
Propiedades de datacollectionimportnode . . . . .	129
Propiedades de excelimportnode . . . . .	131
Propiedades de evimportnode . . . . .	132
Propiedades de fixedfilenode . . . . .	133
Propiedades de sasimportnode. . . . .	135
Propiedades de statisticsimportnode . . . . .	136
Propiedades de userinputnode . . . . .	136
Propiedades de variablefilenode. . . . .	137
Propiedades de xmlimportnode. . . . .	140

**13 Propiedades de nodos de operaciones con registros 142**

Propiedades de appendnode . . . . .	142
Propiedades de aggregatenode . . . . .	142
Propiedades de balancenode . . . . .	143
Propiedades de distinctnode . . . . .	144
Propiedades de mergenode . . . . .	145
Propiedades rfmaggagatenode . . . . .	146
Propiedades de samplenode. . . . .	148
Propiedades de selectnode . . . . .	150
Propiedades de sortnode . . . . .	150

## **14 Propiedades de nodos de operaciones con campos 152**

Propiedades de anonymizenode . . . . .	152
Propiedades autodatapreprenode . . . . .	153
Propiedades de binningnode . . . . .	156
Propiedades de derivenode . . . . .	158
Propiedades de ensemblenode . . . . .	160
Propiedades de fillernode . . . . .	161
Propiedades de filternode . . . . .	162
Propiedades de historynode . . . . .	163
Propiedades de partitionnode . . . . .	164
Propiedades de reclassifynode . . . . .	165
Propiedades de reordernode . . . . .	166
Propiedades de restructurenode . . . . .	167
Propiedades de rfanalysisnode . . . . .	168
Propiedades de settoflagnode . . . . .	169
Propiedades de statisticstransformnode . . . . .	170
Propiedades de timeintervalnode . . . . .	170
Propiedades de transposenode . . . . .	174
Propiedades de typenode . . . . .	175

## **15 Propiedades de nodos de gráficos 180**

Propiedades comunes de nodos de gráficos . . . . .	180
Propiedades de collectionnode . . . . .	181
Propiedades de distributionnode . . . . .	182
Propiedades de evaluationnode . . . . .	183
Propiedades de graphboardnode . . . . .	185
Propiedades de histogramnode . . . . .	187
Propiedades de multiplotnode . . . . .	188
Propiedades de plotnode . . . . .	189
Propiedades de timeplotnode . . . . .	191
Propiedades de webnode . . . . .	192

## **16 Propiedades de nodos de modelado**

**195**

Propiedades comunes de nodos de modelado . . . . .	195
Propiedades de anomalydetectionnode . . . . .	196
Propiedades de apriorinode . . . . .	197
Propiedades de autoclassifiernode. . . . .	199
Definición de propiedades de algoritmo. . . . .	200
Propiedades de nodo de autoconglomeración . . . . .	201
Propiedades de autonumericnode . . . . .	203
Propiedades de bayesnetnode . . . . .	204
Propiedades de c50node . . . . .	206
Propiedades de carmanode . . . . .	207
Propiedades de cartnode . . . . .	208
Propiedades de chaidnode . . . . .	211
Propiedades de coxregnode . . . . .	213
Propiedades de decisionlistnode . . . . .	215
Propiedades de discriminantnode. . . . .	216
Propiedades de factornode. . . . .	218
Propiedades de featureselectionnode . . . . .	220
Propiedades de genlinnode . . . . .	222
Propiedades de glmnode . . . . .	225
Propiedades de kmeansnode . . . . .	228
Propiedades de knnnode . . . . .	229
Propiedades de kohonennode . . . . .	231
Propiedades de linearnode. . . . .	232
Propiedades de logregnode . . . . .	233
Propiedades de neuralnetnode. . . . .	238
Propiedades de neuralnetworknode . . . . .	240
Propiedades de questnode . . . . .	242
Propiedades de regressionnode . . . . .	244
Propiedades de sequencenode. . . . .	246
Propiedades de slrmnode . . . . .	247
Propiedades de statisticsmodelnode . . . . .	248
Propiedades de svmnode . . . . .	248
Propiedades de timeseriesnode . . . . .	249
Propiedades de twostepnode . . . . .	252

## **17 Propiedades de nodos de nugget de modelo**

**253**

Propiedades de applyanomalydetectionnode . . . . .	253
Propiedades de applyapriorinode . . . . .	253
Propiedades de applyautoclassifiernode . . . . .	254
Propiedades de applyautoclusternode . . . . .	254
Propiedades de applyautonumericnode . . . . .	255
Propiedades de applybayesnetnode . . . . .	255
Propiedades de applyc50node . . . . .	255
Propiedades de applycarmanode . . . . .	256
Propiedades de applycartnode . . . . .	256
Propiedades de applychaidnode . . . . .	256
Propiedades de applycoxregnode . . . . .	257
Propiedades de applydecisionlistnode . . . . .	257
Propiedades de applydiscriminantnode . . . . .	257
Propiedades de applyfactornode . . . . .	258
Propiedades de applyfeatureselectionnode . . . . .	258
Propiedades de applygeneralizedlinearnode . . . . .	258
Propiedades de applykmeansnode . . . . .	258
Propiedades de applyknnnode . . . . .	259
Propiedades de applykohonennode . . . . .	259
Propiedades de applylinearnode . . . . .	259
Propiedades de applylogregnode . . . . .	259
Propiedades de applyneuralnetnode . . . . .	260
Propiedades de applyneuralnetworknode . . . . .	260
Propiedades de applyquestnode . . . . .	261
Propiedades de applyregressionnode . . . . .	261
Propiedades de applyselflearningnode . . . . .	261
Propiedades de applysequencenode . . . . .	262
Propiedades de applysvmnode . . . . .	262
Propiedades de applytimeseriesnode . . . . .	262
Propiedades de applytwostepnode . . . . .	262

## **18 Propiedades de nodos de modelado de base de datos**

**263**

Propiedades de nodos de modelado de Microsoft . . . . .	264
Propiedades de nodos de modelado de Microsoft . . . . .	264
Propiedades de nugget de modelo de Microsoft . . . . .	266

Propiedades de nodos de modelado de Oracle . . . . .	268
Propiedades de nodos de modelado de Oracle . . . . .	268
Propiedades de nugget de modelo de Oracle . . . . .	274
Propiedades de nodos para modelado de IBM DB2. . . . .	275
Propiedades de nodos de modelado de IBM DB2 . . . . .	275
Propiedades de nugget de modelo de IBM DB2. . . . .	280
Propiedades de nodos de modelado de IBM Netezza Analytics. . . . .	281
Propiedades de nodos de modelado de Netezza . . . . .	281
Propiedades de nugget de modelo de Netezza . . . . .	288

## **19 Propiedades de los nodos de resultados 290**

Propiedades de analysisnode . . . . .	290
Propiedades de dataauditnode . . . . .	291
Propiedades de matrixnode . . . . .	293
Propiedades de meansnode . . . . .	295
Propiedades de reportnode . . . . .	296
Propiedades de setglobalsnode . . . . .	297
Propiedades de statisticsnode . . . . .	298
Propiedades de statisticsoutputnode . . . . .	299
Propiedades de tablenode . . . . .	300
Propiedades de transformnode. . . . .	302

## **20 Propiedades de nodos Exportar 304**

Propiedades de nodos Exportar comunes . . . . .	304
Propiedades del nodo de exportación Cognos . . . . .	304
Propiedades de databaseexportnode . . . . .	305
Propiedades de datacollectionexportnode . . . . .	310
Propiedades de excelexportnode . . . . .	311
Propiedades de outputfilenode . . . . .	312
Propiedades de sasexportnode. . . . .	312
Propiedades de statisticsexportnode . . . . .	313
Propiedades de xmlexportnode. . . . .	313

**21 Propiedades de nodos de IBM SPSS Statistics 315**

Propiedades de statisticsimportnode ..... 315  
Propiedades de statisticstransformnode..... 315  
Propiedades de statisticsmodelnode ..... 316  
Propiedades de statisticsoutputnode ..... 317  
Propiedades de statisticsexportnode ..... 317

**22 Propiedades de Supernodos 319**

**Apéndice**

**A Avisos 321**

**Índice 324**

# ***Acerca de IBM SPSS Modeler***

IBM® SPSS® Modeler es un conjunto de herramientas de minería de datos que permite desarrollar rápidamente modelos predictivos mediante técnicas empresariales y utilizarlos en operaciones empresariales para mejorar la toma de decisiones. Con un diseño que sigue el modelo CRISP-DM, estándar del sector, SPSS Modeler admite el proceso completo de minería de datos, desde los propios datos hasta obtener los mejores resultados empresariales.

SPSS Modeler ofrece una gran variedad de métodos de modelado procedentes del aprendizaje automático, la inteligencia artificial y el estadístico. Los métodos disponibles en la paleta de modelado permiten derivar nueva información procedente de los datos y desarrollar modelos predictivos. Cada método tiene ciertos puntos fuertes y es más adecuado para determinados tipos de problemas.

SPSS Modeler puede adquirirse como producto independiente o utilizarse como cliente junto con SPSS Modeler Server. También hay disponible cierto número de opciones adicionales que se resumen en las siguientes secciones. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/modeler/>.

## ***Productos IBM SPSS Modeler***

La familia de productos IBM® SPSS® Modeler y su software asociado se componen de lo siguiente:

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adaptadores para IBM SPSS Collaboration and Deployment Services

## ***IBM SPSS Modeler***

SPSS Modeler es una versión con todas las funcionalidades del producto que puede instalar y ejecutar en su ordenador personal. Puede ejecutar SPSS Modeler en modo local como un producto independiente o utilizarla en modo distribuido junto con IBM® SPSS® Modeler Server para mejorar el rendimiento a la hora de trabajar con grandes conjuntos de datos.

Con SPSS Modeler, puede crear modelos predictivos precisos de forma rápida e intuitiva sin necesidad de programación. Mediante su exclusiva interfaz visual, podrá visualizar fácilmente el proceso de minería de datos. Con ayuda del análisis avanzado incrustado en el producto podrá

detectar patrones y tendencias en sus datos que anteriormente estaban ocultos. Podrá modelar los resultados y comprender los factores que influyen en ellos, lo que le permitirá aprovechar oportunidades comerciales y mitigar los riesgos.

SPSS Modeler está disponible en dos ediciones: SPSS Modeler Professional y SPSS Modeler Premium. [Si desea obtener más información, consulte el tema Ediciones de IBM SPSS Modeler en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

## ***IBM SPSS Modeler Server***

SPSS Modeler utiliza una arquitectura de cliente/servidor para distribuir peticiones de cliente para operaciones que requieren un uso intensivo de los recursos a un software de servidor de gran potencia, lo que proporciona un rendimiento más rápido con conjuntos de datos de mayor volumen.

SPSS Modeler Server es un producto con licencia independiente que se ejecuta de manera continua en modo de análisis distribuido en un host de servidor junto con una o más instalaciones de IBM® SPSS® Modeler. De este modo, SPSS Modeler Server ofrece un mejor rendimiento cuando se trabaja con grandes conjuntos de datos, ya que las operaciones que requieren un uso intensivo de memoria se pueden realizar en el servidor sin tener que descargar datos al equipo cliente. IBM® SPSS® Modeler Server también ofrece asistencia para las capacidades de optimización de SQL y modelado interno de la base de datos, lo que proporciona mayores ventajas en cuanto al rendimiento y la automatización.

## ***IBM SPSS Modeler Administration Console***

Modeler Administration Console es una aplicación gráfica para administrar muchas de las opciones de configuración de SPSS Modeler Server, las cuales también pueden configurarse a través de un archivo de opciones. La aplicación proporciona una interfaz de usuario de la consola para supervisar y configurar las instalaciones de SPSS Modeler Server y está disponible de forma completamente gratuita para los clientes actuales de SPSS Modeler Server. La aplicación solamente se puede instalar en los ordenadores con Windows; sin embargo, puede administrar un servidor que esté instalado en cualquier plataforma compatible.

## ***IBM SPSS Modeler Batch***

Aunque la minería de datos suele ser un proceso interactivo, también es posible ejecutar SPSS Modeler desde una línea de comandos, sin necesidad de la interfaz gráfica del usuario. Por ejemplo, puede que tenga tareas repetitivas o cuya ejecución sea de larga duración que quiera realizar sin intervención por parte del usuario. SPSS Modeler Batch es una versión especial del producto que ofrece asistencia para todas las capacidades analíticas de SPSS Modeler sin acceder a la interfaz de usuario habitual. Es necesario disponer de una licencia de SPSS Modeler Server para utilizar SPSS Modeler Batch.



## **IBM SPSS Modeler Solution Publisher**

SPSS Modeler Solution Publisher es una herramienta que le permite crear una versión empaquetada de una ruta de SPSS Modeler que se puede ejecutar en un motor de tiempo de ejecución externo o incrustado en una aplicación externa. De este modo, podrá publicar y distribuir rutas completas de SPSS Modeler para utilizarlas en entornos que no tengan SPSS Modeler instalado. SPSS Modeler Solution Publisher se distribuye como parte del servicio IBM SPSS Collaboration and Deployment Services - Scoring, para el que se necesita una licencia independiente. Con esta licencia, recibirá SPSS Modeler Solution Publisher Runtime, que le permite ejecutar las rutas publicadas.

## **IBM SPSS Modeler Server Adaptadores para IBM SPSS Collaboration and Deployment Services**

Tiene a su disposición un determinado número de adaptadores para IBM® SPSS® Collaboration and Deployment Services que permiten que SPSS Modeler y SPSS Modeler Server interactúen con un repositorio de IBM SPSS Collaboration and Deployment Services. De este modo, varios usuarios podrán compartir una ruta de SPSS Modeler distribuida en el repositorio, o bien se podrá acceder a ella desde la aplicación cliente de baja intensidad IBM SPSS Modeler Advantage. Debe instalar el adaptador en el sistema donde se aloje el repositorio.

## **Ediciones de IBM SPSS Modeler**

SPSS Modeler está disponible en las siguientes ediciones.

### **SPSS Modeler Professional**

SPSS Modeler Professional proporciona todas las herramientas que necesita para trabajar con la mayoría de los tipos de datos estructurados, como los comportamientos e interacciones registrados en los sistemas de CRM, datos demográficos, comportamientos de compra y datos de ventas.

### **SPSS Modeler Premium**

SPSS Modeler Premium es un producto con licencia independiente que amplía SPSS Modeler Professional para poder trabajar con datos especializados, como los utilizados para el análisis de entidades o las redes sociales, así como con datos de texto no estructurados. SPSS Modeler Premium está formado por los siguientes componentes:

**IBM® SPSS® Modeler Entity Analytics** incorpora una dimensión completamente nueva al análisis predictivo de IBM® SPSS® Modeler. Mientras que el análisis predictivo trata de predecir comportamientos futuros a partir de datos del pasado, el análisis de entidades se centra en mejorar la coherencia de los datos actuales mediante la resolución de conflictos de identidades dentro de los propios registros. La identidad de un individuo, una organización, un objeto o cualquier otra entidad puede estar expuesta a ambigüedades. La resolución de identidades puede ser vital en diversos campos, entre los que se incluyen la gestión de la relación con el cliente, la detección de fraudes, la lucha contra el blanqueo de dinero y la seguridad nacional e internacional.

**IBM SPSS Modeler Social Network Analysis** transforma la información sobre relaciones en campos que caracterizan el comportamiento social de individuos y grupos. Mediante el uso de datos que describen las relaciones subyacentes de las redes sociales, IBM® SPSS® Modeler Social Network Analysis identifica a los líderes sociales que influyen en el comportamiento de otros en la red. Además, puede determinar qué personas se ven más afectadas por otros participantes de la red. Al combinar estos resultados con otras medidas, puede crear perfiles completos de individuos en los que basar sus modelos predictivos. Los modelos que incluyan esta información social tendrán un mejor rendimiento que los modelos que no la incluyan.

**Text Analytics for IBM® SPSS® Modeler** utiliza tecnologías de lingüística avanzada y Procesamiento del lenguaje natural (PLN) para procesar con rapidez una gran variedad de datos de texto sin estructurar, extraer y organizar los conceptos clave y agruparlos en categorías. Las categorías y conceptos extraídos se pueden combinar con los datos estructurados existentes, como pueden ser datos demográficos, y se pueden aplicar para modelar utilizando el conjunto completo de herramientas de minería de datos de SPSS Modeler para tomar decisiones mejores y más certeras.

## ***Documentación de IBM SPSS Modeler***

Tiene a su disposición documentación en formato de ayuda en línea desde el menú Ayuda de SPSS Modeler. Se incluye documentación para SPSS Modeler, SPSS Modeler Server y SPSS Modeler Solution Publisher, así como el Manual de aplicaciones y otros materiales de apoyo.

La documentación completa de cada producto (incluidas las instrucciones de instalación) en formato PDF está disponible en la carpeta *Documentation* en cada DVD del producto. También es posible descargar los documentos de instalación en Internet en <http://www-01.ibm.com/support/docview.wss?uid=swg27023172>.

La documentación en ambos formatos también está disponible desde el centro de información de SPSS Modeler en <http://publib.boulder.ibm.com/infocenter/spssmodl/v15r0m0/>.

## ***Documentación de SPSS Modeler Professional***

El conjunto de documentación de SPSS Modeler Professional (excluidas las instrucciones de instalación) es el siguiente.

- **Manual del usuario de IBM SPSS Modeler.** Introducción general sobre cómo usar SPSS Modeler, incluyendo cómo crear rutas de datos, tratar valores perdidos, crear expresiones CLEM, trabajar con proyectos e informes y empaquetar rutas para su distribución en IBM SPSS Collaboration and Deployment Services, Predictive Applications o IBM SPSS Modeler Advantage.
- **Nodos Fuente, Proceso y Resultado de IBM SPSS Modeler.** Descripciones de todos los nodos utilizados para leer, procesar y dar salida a datos en diferentes formatos. En la práctica, esto implica todos los nodos que no sean nodos de modelado.
- **Nodos de modelado de IBM SPSS Modeler.** Descripciones de todos los nodos utilizados para crear modelos de minería de datos. IBM® SPSS® Modeler ofrece una variedad de métodos de modelado tomados del aprendizaje de las máquinas, la inteligencia artificial y

la estadística. [Si desea obtener más información, consulte el tema Conceptos básicos sobre nodos de modelado en el capítulo 3 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

- **Manual de algoritmos de IBM SPSS Modeler.** Descripciones de los fundamentos matemáticos de los métodos de modelado que se utilizan en SPSS Modeler. Esta guía está disponible únicamente en formato PDF.
- **Manual de aplicaciones de IBM SPSS Modeler.** Los ejemplos de esta guía ofrecen introducciones breves y concisas a métodos y técnicas de modelado específicos. También tiene a su disposición una versión en línea de este manual en el menú Ayuda. [Si desea obtener más información, consulte el tema Ejemplos de aplicaciones en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)
- **Procesos y automatización de IBM SPSS Modeler.** Información sobre la automatización del sistema a través de procesos, incluidas las propiedades que se pueden utilizar para manipular nodos y rutas.
- **IBM SPSS Modeler Manual de distribución.** Información sobre la ejecución de rutas y escenarios de SPSS Modeler como pasos en trabajos de procesamiento en IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **Guía del desarrollador de IBM SPSS Modeler CLEF.** CLEF permite integrar programas de otros fabricantes, como rutinas de procesamiento de datos o algoritmos de modelado como nodos en SPSS Modeler.
- **Manual de minería interna de bases de datos de IBM SPSS Modeler.** Este manual incluye información sobre cómo utilizar la potencia de su base de datos, tanto para mejorar su rendimiento como para ampliar su oferta de capacidades analíticas a través de algoritmos de terceros.
- **Guía de administración y rendimiento de IBM SPSS Modeler Server.** Información sobre la configuración y administración de IBM® SPSS® Modeler Server.
- **Manual del usuario de IBM SPSS Modeler Administration Console.** Información sobre cómo instalar y utilizar la interfaz de usuario de la consola para supervisar y configurar SPSS Modeler Server. La consola se implementa como complemento de la aplicación Deployment Manager.
- **Manual de IBM SPSS Modeler Solution Publisher.** SPSS Modeler Solution Publisher es un componente complementario que permite a las organizaciones publicar rutas para su uso fuera del entorno estándar de SPSS Modeler.
- **Manual de CRISP-DM de IBM SPSS Modeler.** Manual que explica paso a paso cómo utilizar la metodología de CRISP-DM en la minería de datos con SPSS Modeler.
- **Manual del usuario de IBM SPSS Modeler Batch.** Guía completa de cómo utilizar IBM SPSS Modeler en modo por lotes, incluida información detallada sobre la ejecución del modo por lotes y argumentos de línea de comandos. Esta guía está disponible únicamente en formato PDF.

## ***Documentación de SPSS Modeler Premium***

El conjunto de documentación de SPSS Modeler Premium (excluidas las instrucciones de instalación) es el siguiente.

- **IBM SPSS Modeler Entity Analytics Manual del usuario.** Información sobre cómo utilizar el análisis de entidades con SPSS Modeler, que cubre la instalación y configuración de repositorios, nodos de análisis de entidades y tareas administrativas.
- **IBM SPSS Modeler Social Network Analysis Manual del usuario.** Una guía para realizar análisis de redes sociales con SPSS Modeler, incluido el análisis de grupos y el análisis de difusión.
- **Text Analytics for SPSS Modeler Manual del usuario.** Información sobre cómo utilizar el análisis de texto con SPSS Modeler, que cubre los nodos de minería de texto, programa interactivo, plantillas y otros recursos.
- Manual del usuario de **Text Analytics for IBM SPSS Modeler Administration Console.** Información sobre cómo instalar y utilizar la interfaz de usuario de la consola para supervisar y configurar IBM® SPSS® Modeler Server para su uso con Text Analytics for SPSS Modeler. La consola se implementa como complemento de la aplicación Deployment Manager.

## ***Ejemplos de aplicaciones***

Mientras que las herramientas de minería de datos de SPSS Modeler pueden ayudar a resolver una amplia variedad de problemas organizativos y empresariales, los ejemplos de la aplicación ofrecen introducciones breves y adaptadas de técnicas y métodos de modelado específicos. Los conjuntos de datos utilizados aquí son mucho más pequeños que los enormes almacenes de datos gestionados por algunos analizadores de datos, pero los conceptos y métodos implicados deberían ser escalables a las aplicaciones reales.

Para acceder a los ejemplos pulsando Ejemplos de aplicación en el menú Ayuda de SPSS Modeler. Los archivos de datos y rutas de muestra se instalan en la carpeta *Demos* en el directorio de instalación del producto. [Si desea obtener más información, consulte el tema Carpeta Demos en Manual de usuario de IBM SPSS Modeler 15.](#)

**Ejemplos de modelado de base de datos.** Consulte los ejemplos que figuran en el Manual de minería interna de bases de datos de *IBM SPSS Modeler*.

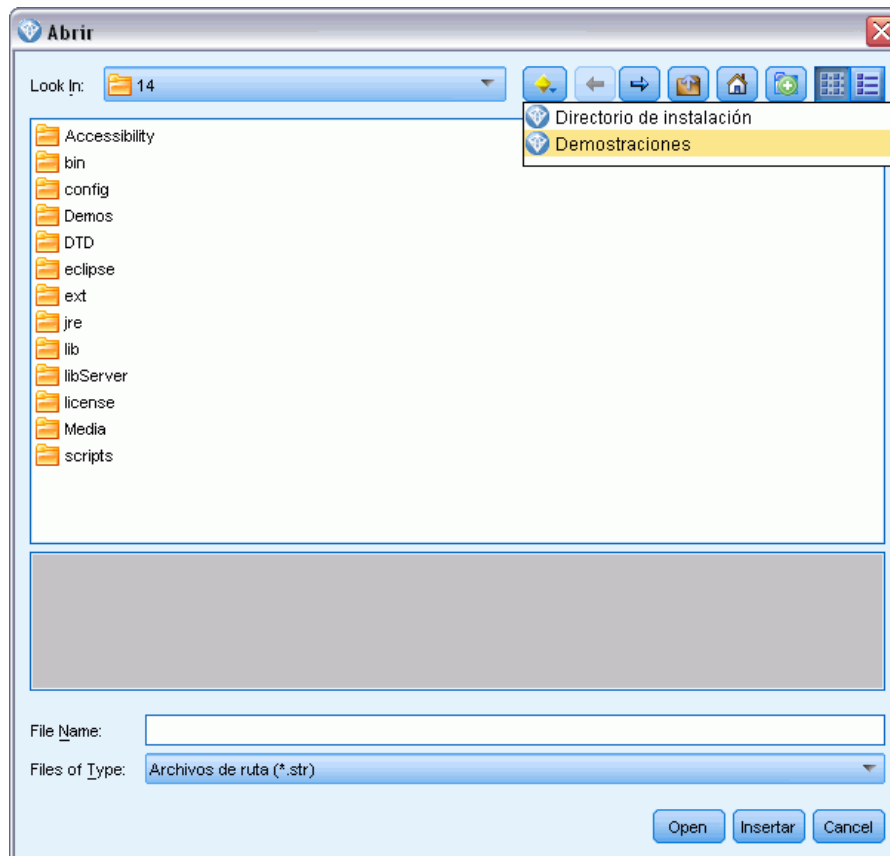
**Ejemplos de procesos.** Consulte los ejemplos que figuran en la Guía de procesos y automatización de *IBM SPSS Modeler*.

## Carpeta Demos

Los archivos de datos y rutas de muestra utilizados con los ejemplos de la aplicación se instalan en la carpeta *Demos* en el directorio de instalación del producto. También puede acceder a esta carpeta desde el grupo de programas IBM SPSS Modeler 15 del menú Inicio de Windows o pulsando *Demos* de la lista de directorios recientes en el cuadro de diálogo Abrir archivo.

Figura 1-1

*Selección de la carpeta Demos desde la lista de directorios utilizados recientemente*



***Parte I:***  
***Procesamientos y el lenguaje de***  
***procesamiento***

# Conceptos básicos del procesamiento

El procesamiento en IBM® SPSS® Modeler es una herramienta potente para automatizar procesos en la interfaz de usuario. Los procesos pueden realizar los mismos tipos de acciones que se realizan con el ratón o el teclado y se utilizan para automatizar tareas que resultarían extremadamente repetitivas o llevarían mucho tiempo si se realizaran manualmente.

Puede utilizar los procesos para:

- Imponer un orden concreto para la ejecución de nodos en una ruta.
- Establecer propiedades de los nodos y realizar derivaciones usando un subconjunto de CLEM (Control Language for Expression Manipulation).
- Especificar una secuencia automática de acciones que normalmente implican la interacción del usuario (por ejemplo, puede generar un modelo y comprobarlo a continuación).
- Configurar procesos complejos que requieren una interacción sustancial del usuario, como los procedimientos de validación cruzada que requieren una repetitiva generación y comprobación de modelo.
- Configurar procesos que manipulen rutas; por ejemplo, puede tomar una ruta de entrenamiento de modelo, ejecutarla y producir la ruta de comprobación del modelo automáticamente.

El capítulo de esta proporciona descripciones de alto nivel y ejemplos de procesos de nivel, procesos independientes y procesos en Supernodos en la interfaz de SPSS Modeler. Los capítulos de las siguientes recogen más información sobre lenguaje de procesos, sintaxis y comandos.

*Nota:* no puede importar y ejecutar procesos creados en IBM® SPSS® Statistics de SPSS Modeler.

## Tipo de procesos

IBM® SPSS® Modeler utiliza tres tipos de procesos:

- Los **procesos de la ruta** se guardan como una propiedad de ruta y se guardan y se cargan con una ruta específica. Por ejemplo, puede escribir un proceso de ruta que automatice el proceso de entrenamiento y aplicación de un nugget de modelo. También puede especificar que cuando se ejecute una ruta particular, se ejecute el proceso, en lugar del contenido del lienzo de la ruta.
- Los **procesos independientes** no están asociados a ninguna ruta en particular y se guardan en archivos de texto externos. Puede utilizar un proceso independiente, por ejemplo, para manipular varias rutas a la vez.
- Los **procesos Supernodos** se guardan como una propiedad de ruta de supernodo. Los procesos Supernodos sólo están disponibles en supernodos terminales. Puede utilizar un proceso de supernodo para controlar la secuencia de ejecución del contenido del supernodo. En supernodos no terminales (origen o proceso), puede definir propiedades del supernodo o los nodos que contiene en su proceso de ruta directamente.

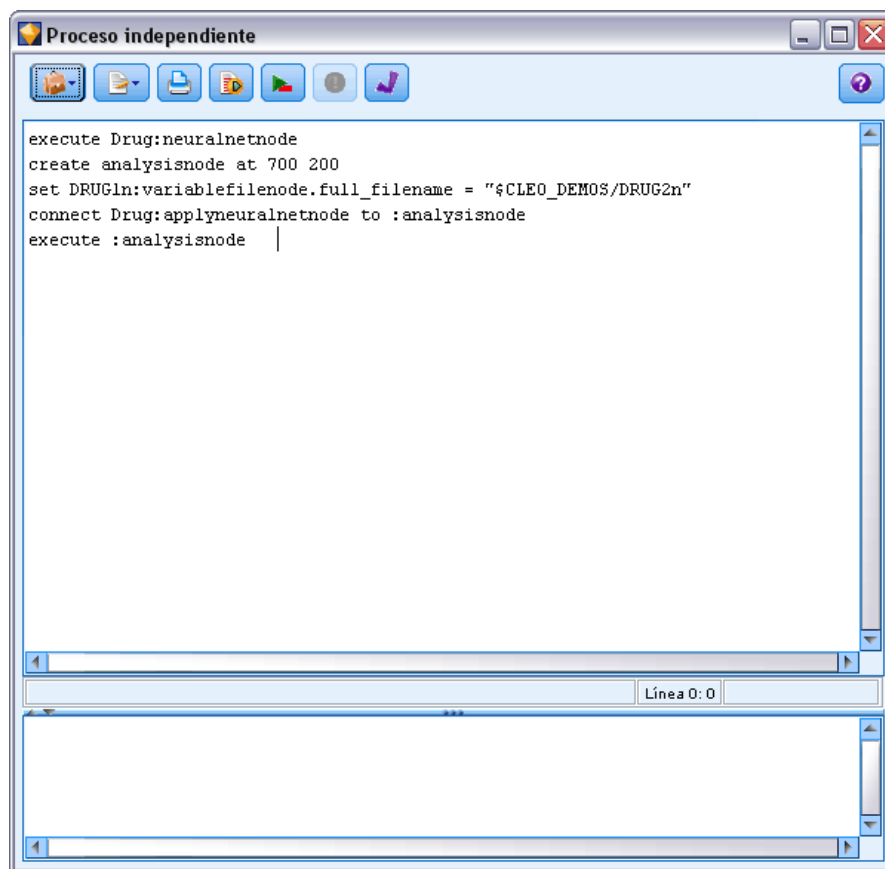
## Procesos de ruta

Los procesos se pueden utilizar para personalizar operaciones dentro de una ruta particular y se guardan con esa ruta. Los procesos de la ruta se pueden utilizar para especificar un orden de ejecución particular para los nodos terminales de una ruta. El cuadro de diálogo del proceso de la ruta se utiliza para editar el proceso que está guardado con la ruta actual.

**Para acceder a la pestaña de procesos de ruta en el cuadro de diálogo Propiedades de ruta:**

- ▶ En el menú Herramientas, seleccione:  
Propiedades de ruta > Proceso...
- ▶ Pulse en la pestaña Proceso para trabajar con procesos en la ruta actual.

Figura 2-1  
Cuadro de diálogo del proceso de la ruta



Los iconos de la barra de herramientas de la parte superior del cuadro de diálogo le permiten realizar las siguientes operaciones:

- Importar el contenido de una ruta independiente preexistente en la ventana.
- Guardar un proceso como archivo de texto.
- Imprimir un proceso.



- Añadir proceso por defecto.
- Ejecutar el proceso completo actual.
- Ejecutar líneas concretas de un proceso.
- Comprobar la sintaxis del proceso y, si se detectan errores, abrirlos en el panel inferior del cuadro de diálogo para su revisión.

Además, puede especificar si este proceso se debe ejecutar o no cuando se ejecuta la ruta. Puede seleccionar Ejecutar este proceso para que se ejecute cada vez que se ejecute la ruta y se use el orden de ejecución especificado en el proceso. De este modo se proporciona una automatización a nivel de ruta para acelerar la generación del modelo. Sin embargo, la configuración por defecto es omitir el proceso durante la ejecución de la ruta. Incluso si selecciona la opción Omitir este proceso, siempre puede ejecutar la ruta directamente desde este cuadro de diálogo.

### **Ejemplo de proceso de ruta: Entrenamiento de una red neuronal**

Una ruta se puede usar para entrenar un modelo de red neuronal cuando se ejecute. Normalmente, para comprobar el modelo, se inserta el nodo de modelado para agregar el modelo a la ruta, realizar las conexiones adecuadas y ejecutar el nodo Análisis.

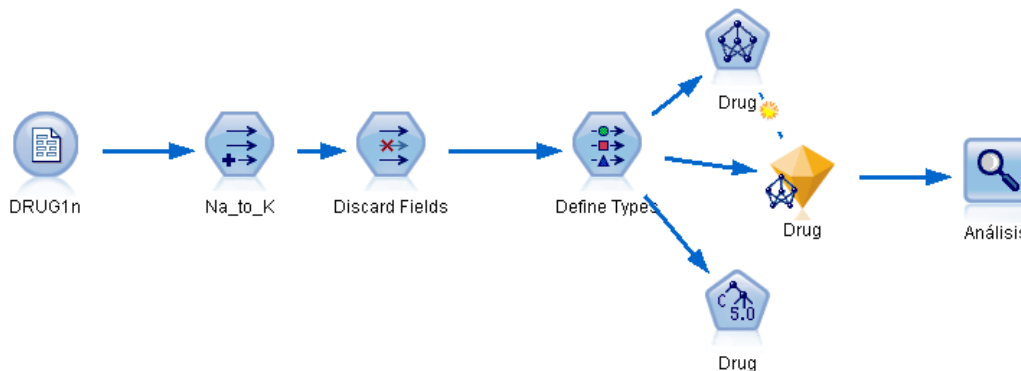
Mediante un proceso de IBM® SPSS® Modeler se puede automatizar el proceso de comprobar el nugget de modelo tras crearlo. Por ejemplo, el siguiente proceso de ruta para la ruta de demostración *druglearn.str* (disponible en la carpeta */Demos/streams/* de su instalación de SPSS Modeler) se puede ejecutar desde el cuadro de diálogo de propiedades de ruta (Herramientas > Propiedades de ruta > Ruta):

```
execute Drug:neuralnetworknode
create analysisnode at 700 200
set DRUG1n:variablefilenode.full_filename = "$CLEO_DEMOS/DRUG2n"
connect :applyneuralnetworknode to :analysisnode
execute :analysisnode
```

Los puntos siguientes describen cada línea de este ejemplo de proceso.

- La primera línea ejecuta el nodo Red neuronal denominado Drug ya encontrado en la ruta de demostración para generar un nugget de modelo y colocarlo en el lienzo de rutas, conectado al nodo Tipo ya en el sistema.
- En la línea 2, el proceso crea un nodo Análisis y lo coloca en la posición del lienzo 700 x 200.
- En la línea 3, el origen de datos original utilizado en la ruta cambia a un conjunto de datos de prueba denominado DRUG2n.
- En la línea 4, el nugget de modelo Red neuronal está conectado al nodo Análisis. Tenga en cuenta que no se utilizan nombres para denotar el nugget de modelo Red neuronal o Análisis ya que no existen nodos similares en la ruta.
- Finalmente, el nodo Análisis se ejecuta para producir el informe Análisis.

Figura 2-2  
Ruta resultante



Este proceso está diseñado para ejecutarse con una ruta existente, ya que asume que el nodo Red neuronal llamado *Medicamento* ya existe. Sin embargo, también es posible utilizar un proceso para crear y ejecutar una ruta desde cero, comenzando con un lienzo vacío. Para obtener más información acerca de lenguaje de procesos en general, consulte *Conceptos básicos del lenguaje de procesamiento* el p. 20. Para obtener más información acerca de los comandos de procesos de forma específica, consulte *Comandos de procesos* el p. 31.

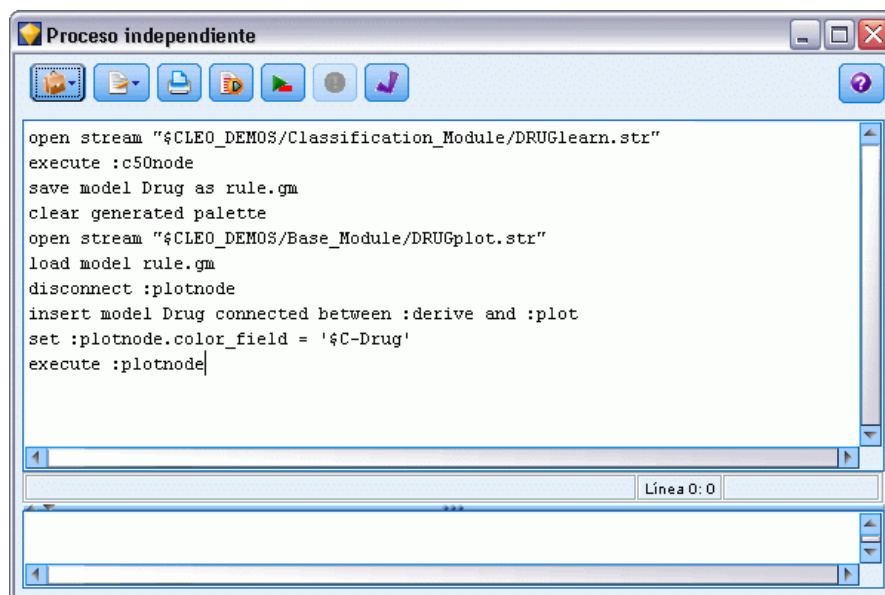
## Procesos independientes

El cuadro de diálogo Proceso independiente se usa para crear o editar un proceso que se ha guardado como archivo de texto. En él se muestra el nombre del archivo y se proporcionan funcionalidades para la carga, almacenamiento, importación y ejecución de procesos.

### **Para acceder al cuadro de diálogo del proceso independiente:**

- ▶ En el menú principal, elija:  
Herramientas > Proceso independiente

Figura 2-3  
Cuadro de diálogo Proceso independiente



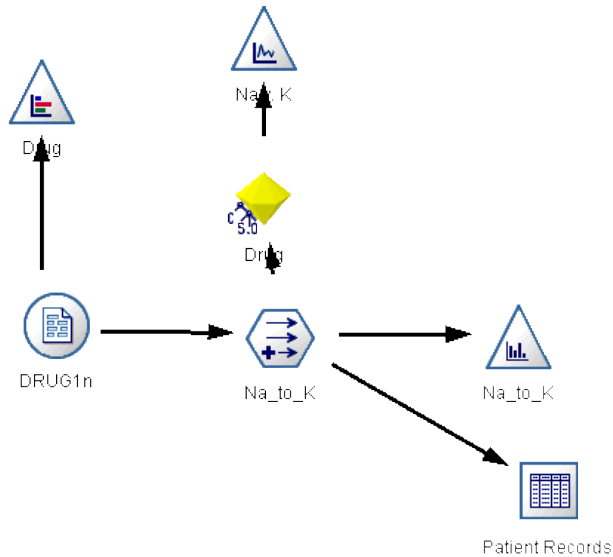
Los procesos independientes y los de ruta comparten las mismas opciones de comprobación de sintaxis de procesos y barra de herramientas. [Si desea obtener más información, consulte el tema Procesos de ruta el p. 10.](#)

### ***Ejemplo de proceso independiente: Guardar y cargar un modelo***

Los procesos independientes son útiles para la manipulación de rutas. Imagine que tiene dos rutas, una que crea un modelo y otra que usa diagramas para examinar el conjunto de reglas generado a partir de la primera ruta con campos de datos existentes. Un proceso independiente para este escenario tendría un aspecto similar a éste:

```
open stream "$CLEO_DEMOS/streams/druglearn.str"
execute :c50node
save model Drug as rule.gm
clear generated palette
open stream "$CLEO_DEMOS/streams/drugplot.str"
load model rule.gm
disconnect :plotnode
insert model Drug connected between :derive and :plot
set :plotnode.color_field = '$C-Drug'
execute :plotnode
```

Figura 2-4  
Ruta resultante



*Nota:* para obtener más información acerca de lenguaje de procesos en general, consulte Conceptos básicos del lenguaje de procesamiento el p. 20. Para obtener más información acerca de los comandos de procesos de forma específica, consulte Comandos de procesos el p. 31.

### **Ejemplo de proceso independiente: Generar un modelo de selección de características**

Comenzando con un lienzo vacío, este ejemplo crea una ruta que genera un modelo de selección de características, aplica el modelo y crea una tabla que indica los 15 campos más importantes respecto al objetivo especificado.

```
create stream 'featureselection'
create statisticsimportnode
position :statisticsimportnode at 50 50
set :statisticsimportnode.full_filename = "$CLEO_DEMOS/customer_dbase.sav"
```

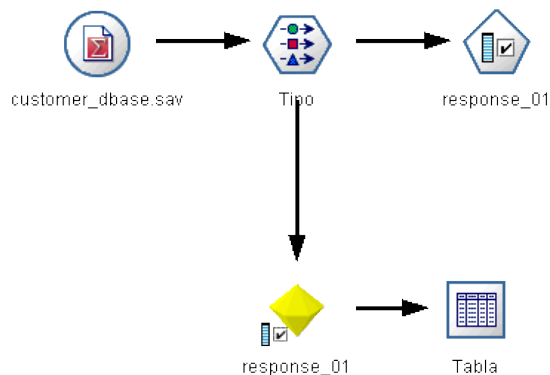
```
create typenode
position :typenode at 150 50
set :typenode.direction.'response_01' = Target
connect :statisticsimportnode to :typenode
```

```
create featureselectionnode
position :featureselectionnode at 250 50
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.important_label = "Check Me Out!"
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
connect :typenode to :featureselectionnode
execute :featureselectionnode
```

```
create tablenode
position :tablenode at 250 250
connect response_01:applyfeatureselectionnode to :tablenode
execute :tablenode
```

El proceso crea un nodo de origen para leer en los datos, utiliza un nodo Tipo para definir el papel (dirección) del campo *response\_01* hacia Target y, a continuación, crea y ejecuta un nodo Selección de características. Este proceso también conecta cada nodo y posiciones en el lienzo de la ruta para producir un diseño legible. El nugget de modelo resultante se conecta al nodo Tabla, que indica los 15 campos más importantes, tal y como determinan las propiedades *selection\_mode* y *top\_n*. [Si desea obtener más información, consulte el tema Propiedades de featureselectionnode en el capítulo 16 el p. 220.](#)

Figura 2-5  
Ruta resultante



## Procesos de Supernodo

Puede crear y guardar procesos con cualquier supernodo terminal utilizando el lenguaje de procesos de IBM® SPSS® Modeler. Estos procesos sólo están disponibles para supernodos terminales y se suelen utilizar cuando crea rutas de plantilla o para imponer un orden de ejecución especial del contenido del supernodo. Los procesos de supernodo también permiten ejecutar más de un proceso en una ruta.

Por ejemplo, supongamos que necesita especificar el orden de ejecución de una ruta compleja y su supernodo contiene varios nodos, incluyendo un nodo Val. globales, que se debe ejecutar antes de derivar un nuevo campo utilizado en un nodo Gráfico. En este caso, puede crear un proceso de supernodo que ejecute el nodo Val. globales en primer lugar. Los valores calculados por este nodo, como la media o la desviación estándar, se pueden usar posteriormente cuando se ejecute el nodo Gráfico.

En un proceso Supernodo, puede especificar las propiedades del nodo de la misma manera que otros procesos. También puede cambiar y definir las propiedades de cualquier supernodo o sus nodos encapsulados directamente desde un proceso de ruta. [Si desea obtener más información, consulte el tema Propiedades de Supernodos en el capítulo 22 el p. 319.](#) Este método funciona para supernodos de origen y proceso y supernodos terminales.

*Nota:* como sólo los supernodos terminales pueden ejecutar sus propios procesos, la pestaña Procesos del cuadro de diálogo Supernodo sólo está disponible para supernodos terminales.

***Para abrir el cuadro de diálogo de proceso de supernodo desde el lienzo principal:***

- ▶ Seleccione un supernodo terminal en el lienzo de ruta y, en el menú de supernodo, seleccione: Proceso de Supernodo...

***Para abrir el cuadro de diálogo de proceso de supernodo desde el lienzo de supernodo aumentado:***

- ▶ Pulse con el botón derecho en el lienzo de supernodo y en el menú contextual seleccione: Proceso de Supernodo...

Si desea obtener más información, consulte el tema [Supernodos y procesos en el capítulo 9 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*](#).

## ***Proceso de supernodo de ejemplo***

El siguiente proceso de supernodo establece el orden en que se ejecutarán los nodos terminales del supernodo. Este orden garantiza que el nodo Val. globales se ejecuta primero para que los valores que calcula este nodo se puedan utilizar cuando se ejecute otro nodo.

```
execute 'Set Globals'
execute 'gains'
execute 'profit'
execute 'age v. $CC-pep'
execute 'Table'
```

## ***Ejecución e interrupción de procesos***

Existen diversas formas de ejecutar procesos. Por ejemplo, en el proceso de ruta o en el cuadro de diálogo del proceso, el botón “Ejecutar este proceso” ejecuta el proceso completo:

Figura 2-6  
*Botón Ejecutar este proceso*



El botón “Ejecutar líneas seleccionadas” ejecuta una única línea, o un bloque de líneas adyacentes, que ha seleccionado en el proceso\_

Figura 2-7  
*Botón Ejecutar líneas seleccionadas*



Un proceso se puede ejecutar mediante cualquiera de los siguientes métodos:

- Pulse en el botón “Ejecutar este proceso” o “Ejecutar líneas seleccionadas” dentro de un proceso de ruta o un cuadro de diálogo de proceso.

- Ejecutando una ruta donde Ejecutar este proceso esté establecido como el método de ejecución por defecto.
- Utilizando la marca `-execute` al inicio en modo interactivo. [Si desea obtener más información, consulte el tema Uso de argumentos en la línea de comandos en el capítulo 7 el p. 70.](#)

*Nota:* cuando se ejecuta el Supernodo, se ejecuta un proceso de Supernodo siempre que se haya seleccionado Ejecutar este proceso en el cuadro de diálogo del proceso Supernodo.

### ***Interrupción de la ejecución del proceso***

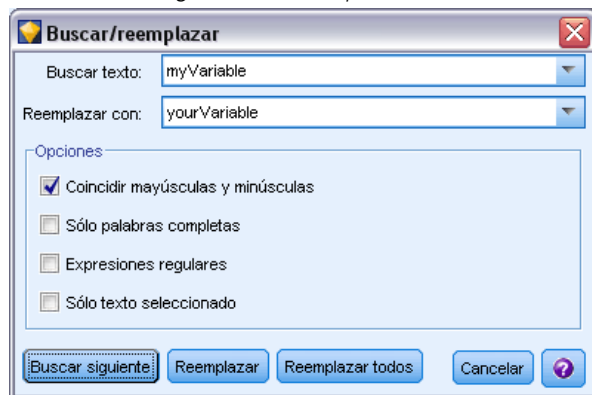
En el cuadro de diálogo de procesos de ruta, se activará el botón rojo de detención durante la ejecución de procesos. Pulsando este botón, puede abandonar la ejecución del proceso y de cualquier ruta actual.

## ***Buscar y reemplazar***

El cuadro de diálogo Buscar/reemplazar está disponible en lugares donde edita procesos o texto de expresión, incluido el editor de procesos, el generador de expresiones CLEM o cuando define una plantilla en el nodo Informe. Al editar texto en cualquiera de estas áreas, pulse Ctrl-F para acceder al cuadro de diálogo, asegurándose de que el cursor está centrado en un área de texto. Por ejemplo, si trabaja en un nodo Rellenar, puede acceder al cuadro de diálogo desde cualquiera de las áreas de texto de la ficha Configuración o desde el campo de texto del generador de expresiones.

Figura 2-8

*Cuadro de diálogo Buscar/reemplazar*



- ▶ Con el cursor en un área de texto, pulse Ctrl+F para acceder al cuadro de diálogo Buscar/reemplazar.
- ▶ Introduzca el texto que desee buscar o selecciónelo de la lista desplegable de elementos buscados recientemente.
- ▶ Introduzca el texto de reemplazo, si lo tiene.
- ▶ Pulse en Buscar siguiente para iniciar la búsqueda.
- ▶ Pulse en Reemplazar para reemplazar la sección actual o en Reemplazar todos para actualizar todas las instancias o sólo las seleccionadas.

- El cuadro de diálogo se cierra después de cada operación. Pulse F3 desde cualquier área de texto para repetir la operación de búsqueda más reciente o pulse Ctrl+F para volver a acceder al cuadro de diálogo.

### **Opciones de búsqueda**

**Coincidir mayúsculas y minúsculas.** Especifica si la operación de búsqueda hace distinción entre mayúsculas y minúsculas; por ejemplo, si *mivar* es igual que *miVar*. El texto de reemplazo siempre se introduce exactamente como se ha introducido, independientemente de este ajuste.

**Sólo palabras completas.** Especifica si la operación de búsqueda tiene en cuenta el texto incluido dentro de las palabras. Por ejemplo, si se selecciona, la búsqueda de *fuego* no será igual que *cortafuegos* ni que *corta-fuegos*.

**Expresiones regulares.** Especifica si se utiliza la sintaxis de expresiones regulares (consulte la sección siguiente). Si está seleccionado, la opción Sólo palabras completas está desactivada y su valor se ignora.

**Sólo texto seleccionado.** Controla el ámbito de la búsqueda al utilizar la opción Reemplazar todos.

### **Sintaxis de expresiones regulares**

Las expresiones regulares le permiten buscar caracteres especiales como caracteres de tabulador o de nueva línea, clases o rangos de caracteres como de la *a* a la *d*, cualquier dígito o no dígito y límites como el principio o el final de una línea. Se admiten los siguientes tipos de expresiones.

### **Coincidencias de caracteres**

Caracteres	Coincidencias
x	El carácter x
\\	El carácter de barra invertida
\0n	El carácter con valor octal 0n (0 <= n <= 7)
\0nn	El carácter con valor octal 0nn (0 <= n <= 7)
\0mnn	El carácter con valor octal 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	El carácter con valor hexadecimal 0xhh
\uhhhh	El carácter con valor hexadecimal 0xhhhh
\t	El carácter de tabulador (' <u>0009'</u> )
\n	El carácter de nueva línea (avance de línea) (' <u>000A'</u> )
\r	El carácter de retorno de carro (' <u>000D'</u> )
\f	El carácter de avance de página (' <u>000C'</u> )
\a	El carácter de alerta (campana) (' <u>0007'</u> )
\e	El carácter de escape (' <u>001B'</u> )
\cx	El carácter de control correspondiente a x



**Clases de caracteres coincidentes**

Clases de caracteres	Coincidencias
[abc]	a, b o c (clase simple)
[^abc]	Cualquier carácter excepto a, b o c (resta)
[a-zA-Z]	De la a a la z o de la A a la Z, ambas inclusive (rango)
[a-d[m-p]]	De la a a la d o de la m a la p (unión). Esto también puede especificarse como [a-dm-p].
[a-z&&[def]]	De la a a la z y d, e o f (intersección)
[a-z&&[^bc]]	De la a a la z, excepto b y c (resta). Esto también puede especificarse como [ad-z].
[a-z&&[^m-p]]	De la a a la z y no de la m a la p (resta). Esto también puede especificarse como [a-lq-z].

**Clases de caracteres predefinidas**

Clases de caracteres predefinidas	Coincidencias
.	Cualquier carácter (puede o no coincidir con los terminadores de línea)
\d	Cualquier dígito: [0-9]
\D	Un carácter que no es un dígito: [^0-9]
\s	Un carácter de espacio en blanco: [ \t\n\r\f]
\S	Un carácter de espacio que no está en blanco: [^\s]
\w	Un carácter alfabético: [a-zA-Z_0-9]
\W	Un carácter no alfabético: [^\w]

**Coincidencias de límite**

Reconocedores de límite	Coincidencias
^	El comienzo de una línea
\$	El final de una línea
\b	Un límite alfabético
\B	Un límite no alfabético
\A	El comienzo de la entrada
\Z	El final de la entrada pero para el terminador final, si lo hay
\z	El final de la entrada

# Lenguaje de procesamiento

## Conceptos básicos del lenguaje de procesamiento

El lenguaje de procesamiento de IBM® SPSS® Modeler consiste en:

- Un formato para hacer referencia a nodos, rutas, proyectos, resultados y otros objetos de SPSS Modeler.
- Un conjunto de instrucciones o comandos de procesamiento que se puede utilizar para manipular tales objetos.
- Un lenguaje de expresión de procesamiento para establecer los valores de las variables, los parámetros y otros objetos.
- Compatibilidad con comentarios, continuaciones y bloques de texto literal.

Esta sección describe la sintaxis básica para utilizar el lenguaje de procesamiento. Las secciones siguientes recogen información sobre comandos y propiedades específicas.

## Sintaxis de procesos

Para mejorar la claridad durante el análisis, se deben seguir las siguientes reglas cuando se trabaja con procesos en IBM® SPSS® Modeler:

- Los nombres de variables, como `income` o `referrerID`, deben aparecer entrecomillados.
- Los nombres de variables, como `^mystream`, están precedidos por un símbolo de intercalación (^) cuando se refieren a una variable existente cuyo valor ya se ha establecido. Este símbolo no se utiliza al declarar o establecer el valor de la variable. [Si desea obtener más información, consulte el tema Referencia a nodos el p. 21.](#)
- Las referencias a la sesión, la ruta y los parámetros del Supernodo, como `'$P-Maxvalue'`, se deben acotar con comillas simples.
- Si se utilizan comillas dobles, una expresión se tratará como una cadena literal, por ejemplo, `"Web graph of BP and Drug"`. Si no se utilizan con cuidado las comillas (simples y dobles), pueden producirse resultados inesperados; por ejemplo, `"$P-Maxvalue"` será una cadena en lugar de una referencia en el valor almacenado en un parámetro.
- Los nombres de archivos, como `"druglearn.str"`, se deben acotar con comillas dobles.
- Los nombres de nodos, como `datanode` o `Na_to_K`, pueden llevar comillas simples o no llevarlas. *Nota:* los nombres deben aparecer entrecomillados si incluyen espacios o caracteres especiales. Sin embargo, no puede utilizar el nombre de un nodo en un proceso si comienza por un número, como `'2a_referrerID'`.
- Las propiedades de marcas se deben leer o definir mediante los valores `true` y `false` (escritos en minúsculas, como se muestra). Las variaciones que contengan `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `False`, `FALSE` o `0` también se reconocen al configurar los valores, pero pueden provocar

errores al leer los valores de propiedad en algunos casos. El resto de valores se consideran como true. El uso de minúsculas en true y false de forma consistente evitará confusiones.

- Las cadenas o bloques literales que contienen saltos de línea, espacios, comillas simples o dobles dentro del bloque se pueden acotar con comillas triples. [Si desea obtener más información, consulte el tema Bloques de texto literal el p. 30.](#)
- Las expresiones CLEM, como "Age >= 55", se deben acotar con comillas dobles, por ejemplo:  

```
set :derivenode.flag_expr = "Age >= 55"
```
- Si utiliza comillas en una expresión CLEM, asegúrese de que una barra invertida (\) precede a cada comilla. Por ejemplo:  

```
set :node.parameter = "BP = \"HIGH\""
```

Aunque no es necesario en todas las instancias, se recomienda seguir estas directrices para aumentar la claridad. El comprobador de procesos, disponible en todos los cuadros de diálogo de procesamiento, marca los casos de sintaxis ambigua.

## Referencia a nodos

En los procesos se puede hacer referencia a los nodos de diversas maneras:

- Puede especificar los nodos por su nombre; por ejemplo, DRUG1n. Puede calificar el nombre por tipo. Por ejemplo, Drug:neuralnetworknode hace referencia al nodo Red neuronal llamado Drug y no a ninguna otra clase de nodo.
- Es posible especificar los nodos sólo por tipo, por ejemplo, :neuralnetworknode se refiere a todos los nodos Red neuronal. Se puede utilizar cualquier tipo de nodo válido, por ejemplo, samplenode, neuralnetworknode y kmeansnode. El sufijo node es opcional y se puede omitir, pero se recomienda incluirlo, ya que facilita la identificación de errores en los procesos.
- Puede hacer referencia a cada nodo por su ID único, que se muestra en la ficha Anotaciones de cada nodo. Utilice un símbolo "@" seguido del ID; por ejemplo, @id5E5GJK23L.custom\_name = "My Node". [Si desea obtener más información, consulte el tema Anotaciones en el capítulo 5 en Manual de usuario de IBM SPSS Modeler 15.](#)

**Modelos generados.** Las mismas reglas se aplican a los nodos de los modelos generados. Se puede utilizar el nombre del nodo como se muestra en la paleta de modelos generados de la ventana de administradores, o bien se puede hacer referencia a los nodos de modelo generados por tipo. Tenga en cuenta que los nombres que se usan para hacer referencia a los modelos generados en el administrador son diferentes de los utilizados para los modelos que se han añadido a una ruta con fines de puntuación (en este último se utiliza un prefijo "apply"). [Si desea obtener más información, consulte el tema Nombres de nugget de modelo en el capítulo 4 el p. 43.](#)

### Referencia a nodos mediante variables

Puede proporcionar nombres y tipos de nodos como valores de variables de proceso locales utilizando la sintaxis de intercalación (^). Por ejemplo, cuando se necesita el nombre de un nodo, ^n evoca el nodo cuyo nombre está almacenado en la variable n, mientras que Drug:^t evoca el nodo llamado Drug cuyo tipo está almacenado en la variable t.

Las referencias a los nodos se pueden almacenar en variables de proceso locales (declaradas mediante una instrucción `var`), aunque no en parámetros de ruta, sesión o Supernodo. Para garantizar que las referencias no sean ambiguas en los nodos, asigne un ID de nodo exclusivo a una variable en el momento de crear el nodo.

```
var x
set x = create typenode
set ^x.custom_name = "Mi nodo"
```

- La primera línea crea una variable llamada `x`.
- La segunda línea crea un nuevo nodo Tipo y almacena una referencia al nodo en `x`. Observe que `x` almacena una referencia al propio nodo y no el nombre del nodo.
- La tercera línea establece el valor de la propiedad `custom_name` para el nodo de "My Node". El símbolo de intercalación se utiliza para indicar que `x` es el nombre de una variable en lugar de un nodo. (Sin el símbolo de intercalación `^`, el sistema buscaría un nodo llamado `x`. El símbolo de intercalación no es necesario ni al declarar ni al configurar la variable, porque el objeto de un comando `var`, por ejemplo, sólo puede ser una variable. Sin embargo, en la tercera línea, `x` lógicamente podría ser el nombre de un nodo en lugar de una variable, por lo que es necesario el símbolo de intercalación para diferenciarlos.)

Un problema común suele ser tratar de almacenar una referencia a un nodo en una variable sin haberla declarado primero.

```
set x = create typenode
set ^x.custom_name = "Mi nodo"
```

En este caso, el comando `SET` intenta crear `x` como un parámetro de ruta, sesión o Supernodo, en lugar de como una variable, y devuelve un error porque no se puede almacenar una referencia a un nodo en un parámetro.

### **Referencia a nodos mediante ID**

También puede guardar un ID de nodo único en una variable. Por ejemplo:

```
var n
set n = "id5E5GJK23L"
set @^n.custom_name = "Mi nodo"
```

**Recorrido en bucle por los nodos de una ruta.** También puede utilizar la propiedad `stream.nodes` para devolver una lista de todos los nodos de una ruta y, a continuación, iterar por la lista para acceder a nodos individuales. [Si desea obtener más información, consulte el tema Informe de rutas en el capítulo 6 el p. 67.](#)

### **Ejemplos**

NOMBRE:TIPO

NAME es el nombre de un nodo y TYPE es su tipo. Como mínimo, debe incluir NAME o TYPE. Se puede omitir uno, pero no los dos. Por ejemplo, el siguiente comando crea un nuevo nodo Derivar entre un nodo Archivo variable existente llamado drug1n y un nodo Gráfico existente (los nodos nuevos no utilizan dos puntos):

```
create derivenode connected between medicamento1n and :plotnode
```

También es posible introducir un símbolo de intercalación (^) antes de NAME o TYPE para indicar la presencia de un parámetro; por ejemplo:

```
Medicamento:^t
```

Esta referencia evoca un nodo llamado Drug, donde t es un parámetro que especifica el tipo de nodo. Por ejemplo, si el valor de ^t es c50node, la referencia anterior se puede traducir como:

```
Medicamento:c50node
```

De igual manera, un parámetro puede utilizarse para el nombre del nodo. Por ejemplo, las dos opciones siguientes se pueden utilizar en un contexto donde un nombre de nodo es necesario:

```
^n:derivnode
^n
```

## ***Recuperación de objetos***

El comando `get` devuelve una referencia a una ruta, nodo u objeto de resultados, permitiendo manipular estos objetos mediante procesos. Por ejemplo:

```
var mynode
set mynode = get node flag1:derivnode
position ^mynode at 400 400

var mytable = get output :tableoutput
export output ^mytable as c:/mytable.htm format html

set stream = get stream 'Ruta1'
set ^stream.execute_method = "Proceso"
```

## ***Establecimiento del objeto actual***

Las siguientes variables especiales se pueden utilizar para hacer referencia a objetos actuales:

- node
- stream
- output
- project

A excepción de `project`, se pueden restablecer para cambiar el contexto actual. A diferencia de otras variables de procesos, no es necesario declararlas primero con el comando `var`, ya que están predefinidas.

```
set node = create typenode
rename ^node as "mytypenode"

set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

Como estas variables especiales coinciden con los nombres de los objetos a los que hacen referencia, es posible que en ocasiones sea difícil distinguir entre variable y objeto, lo que da lugar a sutiles diferencias de uso. [Si desea obtener más información, consulte el tema Comando set en el capítulo 4 el p. 33.](#)

### **Comentarios**

La asignación de un valor de tipo erróneo a una variable especial (como configurar un objeto de nodo a la variable stream) origina un error de tiempo de ejecución.

En los casos en que pueda utilizarse la variable especial, se puede usar cualquier variable. Por ejemplo, el almacenamiento de la ruta actual puede desarrollarse con:

```
save stream as 'C:/Mis rutas/Churn.str'
```

Es también correcto decir:

```
save mi_ruta as 'C:/Mis rutas/Churn.str'
```

donde a my\_stream se le ha asignado con anterioridad un valor de ruta.

## **Apertura de rutas y otros objetos**

En un proceso independiente, puede abrir una ruta especificando el nombre de archivo y la ubicación de éste; por ejemplo:

```
open stream "c:/demos/druglearn.str"
```

Otros tipos de objetos se pueden abrir mediante el comando load; por ejemplo:

```
load node c:/minodo.nod
```

```
load model c:/mimodelo.gm
```

**Open stream frente a load stream.** El comando load stream añade una ruta especificada al lienzo sin eliminar los nodos de la ruta actual. Este comando se utilizó más en versiones nuevas y se ha sustituido de forma extendida en versiones posteriores por la capacidad para abrir, administrar y copiar nodos entre varias rutas.

## **Cómo trabajar con varias rutas**

Aparte de los comandos utilizados para acceder a las rutas desde el sistema de archivos o desde IBM® SPSS® Collaboration and Deployment Services Repository (open, load y retrieve), la mayoría de los comandos de proceso se aplican automáticamente a la ruta actual. Sin embargo, en los procesos independientes puede que prefiera abrir y manipular varias rutas desde el mismo

proceso. Esto es posible si configura una referencia a cualquier ruta abierta, o a través del comando `with... endwith` para volver a asignar temporalmente una ruta actual.

Por ejemplo, para cerrar una ruta distinta a la actual, se puede utilizar el comando `get stream` para hacer referencia a la ruta que se desee:

```
set stream = get stream "druglearn"
close stream
```

El proceso vuelve a asignar la ruta de la variable especial a la ruta `druglearn` (convirtiéndola básicamente en la ruta actual) y, a continuación, cierra la ruta.

Si lo prefiere, puede volver a asignar la ruta actual temporalmente a través de la instrucción `with stream`; por ejemplo:

```
with stream 'druglearn'
  create typenode
  execute_script
endwith
```

Las instrucciones anteriores ejecutan la acción `create` y ejecutan el proceso de la ruta con la ruta especificada definida como ruta actual. La ruta actual original se vuelve a almacenar una vez que se ha ejecutado cada instrucción. Las instrucciones condicionales y las construcciones de bucle también pueden incluirse; por ejemplo:

```
with stream 'druglearn'
  create tablenode at 500 400
  create selectnode connected between :typenode and :tablenode
  for l from 1 to 5
    set :selectnode.condition = 'Edad > ' >> (l * 10)
    execute :selectnode
  endfor
endwith
```

Las instrucciones anteriores establecerán la ruta actual en `STREAM` para todas las expresiones dentro del bucle y restaurarán el valor original cuando el bucle haya finalizado.

## ***Variables de proceso locales***

Las variables de proceso locales se declaran con el comando `var` y se establecen sólo para el proceso actual. Las variables se distinguen de los parámetros, que se pueden establecer para una sesión, una ruta o un Supernodo y sólo pueden contener cadenas o números.

```
var mi_nodo
set mi_nodo = create nododistribución
rename ^mi_nodo as "Distribución de marca"
```

Cuando haga referencia a variables existentes, asegúrese de escribir el símbolo de intercalación (^) antes del nombre del parámetro. Por ejemplo, en el proceso anterior:

- La primera línea declara la variable.
- La segunda línea establece su valor.

- La tercera línea vuelve a nombrar el nodo al que hace referencia la variable (no la propia variable). El símbolo de intercalación indica que `^my_node` es el nombre de una variable en lugar del nombre literal del nodo. (Sin el símbolo de intercalación `^`, `rename` buscaría un nodo llamado `my_node`. El símbolo de intercalación no es necesario ni en la primera ni en la segunda línea, ya que el objeto de un comando `var` sólo puede ser una variable. Este símbolo sólo se utiliza cuando se hace referencia a una variable ya establecida, en cuyo caso su eliminación podría provocar una referencia ambigua.)
- Al resolver las referencias de la variable, se busca la lista de variables locales antes que los parámetros de la lista de la sesión, la ruta o el Supernodo. Por ejemplo, si una variable `x` existía como variable local y como parámetro de sesión, la utilización de la sintaxis `'$P-X'` en un parámetro de procesamiento aseguraría que se utiliza el parámetro de sesión en lugar de la variable local.

*Nota:* En términos prácticos, si define una variable sin haberla declarado primero con un comando `var`, se crea un parámetro de ruta, sesión o Supernodo, dependiendo del contexto del proceso actual. Por ejemplo, el siguiente código crea una variable de proceso local llamada `z` y establece su valor en `[1 2 3]`:

```
var z
set z = [1 2 3]
```

Si el comando `var` se omite (y se da por hecho que no existe una variable o nodo con ese nombre), se creará `z` como un parámetro en lugar de como una variable.

## ***Parámetros de ruta, sesión y Supernodo***

Se pueden definir los parámetros para utilizarlos en procesos y expresiones CLEM. Son, de hecho, variables definidas por el usuario que se guardan y conservan con la ruta actual, sesión, o Supernodo, y a los que se puede acceder tanto desde la interfaz de usuario como a través de procesos. Si, por ejemplo, se guarda una ruta, cualquier conjunto de parámetros para esa ruta también se guarda. (Así se distinguen de las variables de proceso local, que sólo se pueden utilizar en el proceso en que se declaran.) Generalmente los parámetros se utilizan en el procesamiento como parte de una expresión CLEM en la que el valor del parámetro se especifica en el proceso.

El ámbito de un parámetro depende de dónde se establezca:

- los parámetros de ruta se pueden establecer en un proceso de ruta o en el cuadro de diálogo de propiedades de ruta, y están disponibles para todos los nodos de la ruta. Se muestran en la lista Parámetros del generador de expresiones.
- Los parámetros de sesión se pueden establecer en un proceso independiente o en el cuadro de diálogo de parámetros de sesión. Están disponibles para todas las rutas utilizadas en la sesión actual (todas las rutas enumeradas en la ficha Rutas del panel de administradores).

También se pueden configurar los parámetros para Supernodos, donde sólo estarán visibles para los nodos encapsulados dentro del Supernodo. [Si desea obtener más información, consulte el tema \*Definición de los parámetros de Supernodos en el capítulo 9 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.\*](#)



### **Configuración de parámetros en los procesos**

Los parámetros se pueden configurar en los procesos mediante el comando `set` y la siguiente sintaxis:

```
set tipocomida= pizza
```

Si no existen nodos o variables con el nombre `foodtype` declarados en el proceso actual, este comando crea un parámetro llamado `foodtype` con un valor por defecto `pizza`.

**Interfaz de usuario.** Si lo prefiere, los parámetros también se pueden configurar o visualizar mediante la interfaz de usuario seleccionando Propiedades de ruta o Definir parámetros de sesión en el menú Herramientas. Estos cuadros de diálogo también le permiten especificar opciones adicionales, como tipo de almacenamiento, que no están disponibles a través de los procesos. [Si desea obtener más información, consulte el tema \*Cómo configurar parámetros de sesión y ruta\* en el capítulo 5 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

**Línea de comandos.** Los parámetros también se pueden configurar desde la línea de comandos, en cuyo caso se crean como parámetros de sesión.

### **Referencia a parámetros en los procesos**

Es posible referirse a parámetros creados con anterioridad acotándolos con comillas simples, precedidos de la cadena `$P`, por ejemplo, `'$P-minvalue'`. También es posible referirse únicamente al nombre del parámetro, como `minvalue`. El valor de un parámetro es siempre una cadena o un número. Por ejemplo, puede referirse al parámetro `foodtype` y configurar un nuevo valor mediante la siguiente sintaxis:

```
set tipocomida = pasta
```

También puede referirse a los parámetros dentro del contexto de la expresión CLEM utilizada en un proceso. El siguiente proceso es un ejemplo. Dicho proceso define las propiedades de un nodo Seleccionar para incluir registros donde el valor de `Age` sea mayor que el especificado en el parámetro de ruta llamado `cutoff`. El parámetro se utiliza en una expresión CLEM con la sintaxis adecuada para CLEM—`'$P-cutoff'`:

```
set :selectnode {  
  mode = "Incluir"  
  condition = "Edad >= '$P-corte'"  
}
```

En el proceso anterior se utiliza el valor por defecto para el parámetro de ruta llamado `cutoff`. Es posible especificar un nuevo valor del parámetro añadiendo la siguiente sintaxis por encima de las especificaciones del nodo Seleccionar:

```
set corte = 50
```

El proceso resultante selecciona todos los registros en los que el valor de `Age` es mayor que 50.

[Si desea obtener más información, consulte el tema \*Parámetros de ruta, sesión y Supernodo\* en el capítulo 7 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

## Control de la ejecución del proceso

La ejecución del proceso generalmente procesa una instrucción detrás de otra. Sin embargo, puede omitir este orden de ejecución utilizando una instrucción condicional if y distintas variedades de los bucles for, por ejemplo:

```
if s.maxsize > 10000 then
s.maxsize = 10000
connect s to :derive
endif
```

El bucle for dispone una variedad de formas; por ejemplo:

```
for PARÁMETRO in LISTA
INSTRUCCIONES
endfor
```

El proceso anterior ejecuta STATEMENTS una vez para cada valor en LIST asignado a PARAMETER, siguiendo el orden de la lista. La lista no incluye corchetes circundantes y el contenido son constantes. También hay disponibles otras formas. [Si desea obtener más información, consulte el tema Comandos de procesos generales en el capítulo 4 el p. 31.](#)

## Operadores en los procesos

Además de los operadores CLEM habituales, también puede manipular variables de procesos locales (declaradas a través del comando var) mediante los operadores “+” y “-”. El operador + añade un elemento a la lista y el operador - elimina un elemento. A continuación se muestra un ejemplo:

```
var z # crear una variable local nueva
set z = [1 2 3] # llevar a la lista con los conjuntos 1, 2 y 3
set z = z + 4 # añadir un elemento; z ahora es igual a [1 2 3 4]
```

Estos operadores no pueden utilizarse con la ruta, el Supernodo o los parámetros de sesión (definidos en procesos a través del comando set), ni tampoco fuera de los procesos de las expresiones generales CLEM (como una fórmula del nodo Derivar).

## Expresiones CLEM en los procesos

Es posible emplear expresiones, funciones y operadores CLEM en los procesos de IBM® SPSS® Modeler; sin embargo, la expresión de procesamiento no puede contener llamadas a ninguna función @, función de fecha/hora ni operaciones a nivel de bit. Además, las siguientes reglas se aplican a las expresiones CLEM en el procesamiento:

- Los parámetros se deben escribir entre comillas simples y con el prefijo \$P-.
- Las expresiones CLEM se deben escribir entrecomilladas. Si la propia expresión CLEM contiene cadenas entrecomilladas o nombres de campos entrecomillados, las comillas incrustadas han de estar precedidas por una barra invertida (\). [Si desea obtener más información, consulte el tema Sintaxis de procesos el p. 20.](#)

Se pueden utilizar valores globales, como GLOBAL\_MEAN(Age), en el procesamiento; sin embargo, no es posible utilizar la propia función @GLOBAL dentro del entorno de procesamiento.

Ejemplos de expresiones CLEM utilizadas en el procesamiento:

```
set :balancenode.directives = {{1.3 "Edad > 60"}}
```

```
set :fillernode.condition = "(Edad > 60) and (PS = \"Alta\")"
```

```
set :derivenode.formula_expr = "substring(5, 1, Medicamento)"
```

```
set Flag:derivenode.flag_expr = "Medicamento = X"
```

```
set :selectnode.condition = "Edad >= '$P-cutoff'"
```

```
set :derivenode.formula_expr = "Edad - GLOBAL_MEAN(Edad)"
```

## Inserción de comentarios y continuaciones

En los procesos se utilizan los siguientes caracteres para denotar comentarios y continuaciones:

Carácter	Uso	Ejemplo
#	El signo almohadilla presenta un comentario. El resto de la línea se omite.	#Éste es un comentario en una sola línea.
\	Una línea que termina con una barra invertida indica que la instrucción continúa en la línea siguiente.	Vea el ejemplo que se muestra a continuación.
/*	La secuencia /* indica el inicio de un comentario. Se omite todo hasta que se encuentra el marcador de final de comentario: */.	Vea el ejemplo que se muestra a continuación.
"""	Las cadenas o bloques literales que contienen saltos de línea, espacios, comillas simples o dobles dentro del bloque se pueden acotar con comillas triples. <a href="#">Si desea obtener más información, consulte el tema Bloques de texto literal el p. 30.</a>	

### Ejemplos

```
/* Éste es un
comentario en
varias líneas
*/
```

```
#a continuación se recoge una instrucción
en varias líneas set :fixedfilenode.fields = {"Edad" 1 3}\
{"Sexo" 5 7} {"PS" 9 10} {"Colesterol" 12 22}\
{"Na" 24 25} {"K" 27 27} {"Droga" 29 32}
```

## ***Bloques de texto literal***

Los bloques de texto literales que incluyen espacios, tabulaciones y saltos de línea pueden incluirse en los procesos acotándolos con comillas triples. Cualquier texto dentro del bloque entrecomillado se conserva como texto literal, incluidos espacios, saltos de línea y comillas simples y dobles incrustadas. No se necesitan caracteres de continuación de línea ni de escape.

Por ejemplo, puede utilizar esta técnica para incrustar un conjunto de directivas de desarrollo de árboles en un proceso, de la siguiente manera:

```
set :cartnode.tree_directives = ""  
Create Root_Node  
Grow Node Index 0 Children 1 2 SplitOn ("DROGA",  
Group ( "drogaA", "drogaB", "drogaC" )  
Group ( "drogaY", "drogaX" ))  
End Tree  
""
```

Esto también resulta útil para rutas de acceso y anotaciones como, por ejemplo:

```
set :node.annotation = ""Este nodo se generó para ayudar a identificar cuáles de los siguientes indicadores  
Lácteos  
Pescado  
Verduras  
Carne  
Repostería  
Pastelería  
muestra un comportamiento de ventas inusual""
```

IBM® SPSS® Modeler omitirá un salto de línea situado después del marcador literal de apertura. Por ejemplo, lo siguiente equivale al ejemplo anterior:

```
set :node.annotation = ""  
Este nodo se generó para ayudar a identificar cuáles de los siguientes indicadores  
Etc.  
""
```

# Comandos de procesos

En esta sección se resumen los comandos que se pueden utilizar en los procesos de IBM® SPSS® Modeler, organizados por tipo de objeto. Si desea obtener más información sobre el lenguaje de procesos, consulte [el capítulo 3](#). Si desea obtener más información sobre las propiedades del nodo, la ruta, el proyecto y el Supernodo, consulte el capítulo 9 en el capítulo 22.

## Comandos de procesos generales

Excepto que se indique de otra manera, los comandos siguientes están disponibles en todos los procesos independientes, de ruta y de Supernodo.

### ***execute\_all***

```
execute_all
```

Ejecuta todos los nodos terminales de la ruta actual.

```
open stream "c:/demos/druglearn.str"
execute_all
```

### ***execute\_script***

```
execute_script
```

*Sólo procesos independientes.* Ejecuta el proceso de ruta asociado a la ruta actual. (Restringido a procesos independientes ya que de otra manera el proceso de ruta se llamaría a sí mismo.)

```
open stream "c:/demos/mysample.str"
execute_script
```

### ***exit***

```
exit CÓDIGO
```

Sale del proceso actual. — El código de salida se puede utilizar para evaluar el proceso o condición de una ruta o nodo, por ejemplo:

```
create tablenode
create variablefilenode
connect :variablefilenode to :tablenode

set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"
execute 'Tabla'

set param = value :tablenode.output at 1 1
```

```

if ^param = 23 then
  create derivenode
else exit 2
endif

```

### ***for...endfor***

El comando `for...endfor` crea un bucle a través de un conjunto de instrucciones basado en una condición. El comando puede tomar varias formas, pero todas siguen la misma estructura general.

```

for PARÁMETRO in LISTA
  INSTRUCCIONES
endfor

```

**for PARÁMETRO in LISTA.** Ejecuta STATEMENTS una vez para cada valor en LIST asignado a PARAMETER, siguiendo el orden de la lista. Por ejemplo, la propiedad `Filter.include` se podría establecer en `true` para múltiples campos del siguiente modo:

```

for f in Age Sex
  set Filter.include.^f=true
endfor

```

**for PARÁMETRO from N to M** Ejecuta STATEMENTS una vez para cada entero comprendido entre N y M, incluidos, por ejemplo:

```

for l from 1 to 5
  set :selectnode.condition = 'Age >' >< (l * 10)
  execute :selectnode
endfor

```

**for PARÁMETRO in \_fields\_to NODO.** Ejecuta STATEMENTS una vez para cada campo en la parte superior de NODO. Por ejemplo, a continuación se establece la propiedad `include` en `true` para todos los campos, incluidos aquellos previamente establecidos en `false`:

```

for f in _fields_to Filter
  set Filter.include.^f = "true"
endfor

```

*Nota:* En los casos en los que un nodo puede tener varios campos de entrada con el mismo nombre, como en `Fundir` o `Añadir`, este método devuelve una lista de campos en la parte posterior de la ruta, en lugar de la anterior, con el fin de evitar posibles conflictos.

**for PARÁMETRO in \_fields\_at NODO.** Ejecuta STATEMENTS una vez para cada campo procedente (o posterior en la ruta de) del NODO especificado. Así, si se trata de un nodo `Filtro`, sólo se incluyen los campos que pasan a través de éste, y no debe ser un nodo terminal, ya que no se devolvería ningún campo. Por ejemplo, al contrario que en el anterior, el proceso siguiente no tendría efecto ya que el bucle solo se ejecutaría para aquellos campos ya establecidos en `true`:

```

for f in _fields_at Filter
  set Filter.include.^f = "true"
endfor

```

**for PARÁMETRO in\_models.** Ejecuta STATEMENTS una vez para cada nugget de modelo en la paleta Modelos. Por ejemplo, el proceso siguiente inserta cada modelo desde la paleta a la ruta actual. (La variable xpos se usa para evitar que los nodos se apilen, uno encima de otro, en el lienzo de rutas.)

```
var xpos
set xpos = 100
for m in_models
  set xpos = xpos + 100
  insert model ^m at ^xpos 100
endfor
```

**for PARÁMETRO in\_streams** *Sólo procesos independientes.* Ejecuta STATEMENTS una vez para cada ruta cargada (como se indica en la paleta de rutas). Si PARAMETER es la ruta de la variable especial, la ruta actual se establece para STATEMENTS en el bucle. Cuando finaliza el bucle, se vuelve a almacenar el valor original de la ruta.

### ***if...then...else...***

```
if EXPR then
  INSTRUCCIONES 1
else
  INSTRUCCIONES 2
endif
```

Ejecuta STATEMENTS 1 si la expresión especificada es verdadera y STATEMENTS 2 si la expresión es falsa. La cláusula else es opcional.

```
if :sampler.use_max_size = true then
  set x = "yes"
else
  set x = "no"
endif
```

### ***Comando set***

```
set VARIABLE = EXPRESIÓN
set PARÁMETRO = EXPRESIÓN
set PROPIEDAD = EXPRESIÓN
```

Establece los valores de una variable de proceso local, variable especial, parámetro, o propiedad.

### ***Configuración de variables***

Antes de establecer el valor de una variable de proceso local, hay que declarar la variable a través del comando var, por ejemplo:

```
var xpos
var ypos
set xpos = 100
set ypos = 100
```

El valor de la variable puede ser una expresión CLEM válida en procesos, un comando de proceso que devuelve un valor (como `load`, `create` o `get`) o un valor literal.

```
set xpos = ^xpos + 50
```

```
var x
set x = create typenode
```

```
var s
set s = get stream 'Druglearn'
```

### **Configuración de variables especiales para hacer referencia a objetos**

Las variables especiales `node`, `stream`, `output` y `project` se utilizan para hacer referencia al objeto “actual” en cada contexto. A excepción de `project`, se pueden restablecer para cambiar el contexto actual. A diferencia de otras variables de procesos, no es necesario declararlas primero con el comando `var`, ya que están predefinidas.

```
set node = create typenode
rename ^node as "mytypenode"

set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

Aunque útiles, estas variables muestran algunas diferencias sutiles en el uso, como se demuestra en el siguiente ejemplo:

```
set stream = get stream 'Stream7'
set ^stream.execute_method = "Script"
save stream as c:/sample7.str
close stream
```

- La primera línea restablece la ruta actual, o literalmente, establece el valor de la variable especial `stream`. (En otras palabras, `stream` es una variable en lugar de ser parte de un comando.)
- La segunda línea utiliza esta variable para definir una propiedad para la ruta actual (consulte más adelante para obtener más detalles sobre las propiedades). El símbolo de intercalación se utiliza para indicar que `^stream` es el nombre de una variable en lugar del nombre de un objeto, como un nodo. (Sin el símbolo de intercalación `^`, el comando `set` buscaría un nodo llamado *ruta*.)
- Las dos últimas líneas guardan y cierran la ruta actual. Como antes, `stream` es una variable, pero en este caso no se utiliza ningún símbolo ya que los comandos `save` y `close` tal como se utilizan en este ejemplo solo se pueden aplicar a una ruta. (El símbolo de intercalación sólo se utiliza en casos donde su omisión pudiera originar una referencia ambigua.)

**Referencias del proyecto actual.** La variable especial `project` se puede utilizar como referencia hacia el proyecto actual (consulte un ejemplo de la configuración de las propiedades de proyecto más abajo). El valor de `project` no se puede restablecer ya que solo se puede abrir un proyecto (es decir, el actual) cada vez.



### **Configuración de los parámetros**

Los parámetros para la ruta, la sesión y el Supernodo se pueden definir de la misma manera que las variables, pero sin utilizar el comando `var`.

```
set p = 1
set minvalue = 21
```

*Nota:* En términos prácticos, si el objeto de un comando `set` no coincide con el nombre de una variable declarada, una variable especial o un objeto existente como un nodo, se creará un parámetro. [Si desea obtener más información, consulte el tema Parámetros de ruta, sesión y Supernodo en el capítulo 3 el p. 26.](#)

### **Configuración de las propiedades de nodo, ruta y proyecto.**

Las propiedades de nodos, rutas y proyectos también se pueden configurar, por ejemplo:

```
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"

set ^stream.execute_method = "Proceso"

load project "C:/myproject.cpj"
set ^project.structure = Phase
```

Para obtener una lista completa de las propiedades disponibles para nodos, rutas y proyectos, consulte [Referencia de propiedades](#) el p. 114.

**Configuración de varias propiedades.** Se pueden asignar varias expresiones a las propiedades para los nodos u otros objetos en una sola operación. Este método se utiliza cuando es necesario realizar varios cambios al nodo antes de determinar el modelo de los datos. El formato que se usa para establecer múltiples propiedades es:

```
set NODO {
  PROPIEDADNODO1 = EXPRESIÓN1
  PROPIEDADNODO2 = EXPRESIÓN2
}
```

Por ejemplo:

```
set :samplenode {
  max_size = 200
  mode = "Include"
  sample_type = "First"
}

set ^project {
  summary = "Trabajo de modelado inicial sobre los datos más recientes"
  ordering = NameAddedType
}
```

**Configuración de valores de marcas (verdadero y falso).** Al leer o escribir las propiedades de tipo de marca, los valores `true` y `false` deben escribirse en minúscula; por ejemplo:

```
set :variablefilenode.read_field_names = true
```

*Nota:* Las variaciones que contengan Off, OFF, off, No, NO, no, n, N, f, F, false, False, FALSE o 0 también se reconocen al configurar los valores, pero pueden provocar errores al leer los valores de propiedad en algunos casos. El resto de valores se consideran verdaderos. El uso de minúsculas en true y false de forma consistente evitará confusiones.

### **Ejemplo: Configuración de propiedades de los nodos**

Existen muchas propiedades específicas de los nodos (a veces denominados parámetros de propiedades) que se utilizan para establecer las opciones que se encuentran en los cuadros de diálogo de la interfaz de usuario para cada nodo. Por ejemplo, para crear una ruta y especificar las opciones de cada nodo, podría utilizar un proceso similar al que aquí se detalla. Si desea obtener más información sobre las propiedades del nodo, la ruta, el proyecto y el Supernodo, consulte el capítulo 9 en el capítulo 22.

```
create varfilenode at 100 100
set :varfilenode {
full_filename = "demos/drug1n"
read_field_names = true
}
create tablenode at 400 100
create samplenode connected between :varfilenode and :tablenode
set :samplenode {
max_size = 200
mode = "Incluir"
sample_type = "Primero"
}
create plotnode at 300 300
create derivenode connected between drug1n and :plotnode
set :derivnode {
new_name = "Cociente sodio-potasio"
formula_expr = "'Na' / 'K'"
}
set :plotnode {
x_field = 'Cociente sodio-potasio'
y_field = 'Edad'
color_field = 'BP'
}
```

### **Comando var**

```
var NOMBREVAR
```

Declara una variable de proceso local.

```
var mi_nodo
set mi_nodo = create nododistribución
rename ^mi_nodo as "Distribución de marca"
```

Las variables se distinguen de los parámetros, que se pueden establecer para una sesión, una ruta o un Supernodo y sólo pueden contener cadenas o números. En términos prácticos, si define una variable sin haberla declarado primero con un comando VAR, se crea un parámetro de ruta, sesión

o Supernodo, dependiendo del contexto del proceso actual. [Si desea obtener más información, consulte el tema Variables de proceso locales en el capítulo 3 el p. 25.](#)

## Objetos de nodo

Existen los siguientes comandos de procesos disponibles para objetos de nodo.

### create NODO

```
create NODO
create NODO at X Y
create NODO between NODO1 and NODO2
create NODO connected between NODO1 and NODO2
```

Crea un nodo del tipo especificado, por ejemplo:

```
create statisticsimportnode
```

Si lo desea, también puede especificar las opciones de posición y conexión:

```
create featureselectionnode at 400 100
```

```
create typenode between :statisticsimportnode and :featureselectionnode
```

```
create selectnode connected between :typenode and :featureselectionnode
```

Asimismo, puede crear un nodo utilizando variables para evitar la ambigüedad. Por ejemplo, en el supuesto que se detalla a continuación, se creará un nodo Tipo y la variable de referencia *x* quedará configurada para contener una referencia a ese nodo Tipo. A continuación, puede utilizar la variable *x* para devolver el objeto al que *x* hace referencia (en este caso, el nodo Tipo) y realizar operaciones adicionales, como el cambio de nombre, el posicionamiento o la conexión de un nodo nuevo.

```
var x
set x = create typenode
rename ^x as "mitipodenodo"
position ^x at 200 200
var y
set y = create varfilenode
rename ^y as "miorigendedatos"
position ^y at 100 200
connect ^y to ^x
```

El ejemplo anterior crea dos nodos, vuelve a nombrar cada uno, los posiciona y, por último, los conecta en el lienzo de las rutas.

Figura 4-1  
Nodos creados mediante variables



Si lo prefiere, el `node` de la variable especial (por defecto) se puede usar de la misma manera que las variables `x` e `y` del ejemplo anterior. En este caso, no es necesario declarar la variable a través del comando `var` (ya que está declarado por defecto) y el proceso resultante puede ser más fácil de leer.

```
set node = create typenode
rename ^node as "mitipodenodo"
position ^node at 200 200
set node = create varfilenode
rename ^node as "miorigendedatos"
position ^node at 100 200
connect miorigendedatos to mitipodenodo
```

*Nota:* las variables especiales, como `node`, pueden reutilizarse para hacer referencia a nodos múltiples. Simplemente utilice el comando `set` para volver a configurar el objeto al que hace referencia la variable. [Si desea obtener más información, consulte el tema Establecimiento del objeto actual en el capítulo 3 el p. 23.](#)

**Duplicación de nodos.** También puede utilizar el comando `duplicate` para duplicar un nodo existente. [Si desea obtener más información, consulte el tema `duplicate NODO` el p. 39.](#)

### ***connect NODO***

```
connect NOD01 to NOD02
connect NOD01 between NOD02 and NOD03
```

Conecta `NODE1` a otros nodos según lo especificado.

```
connect :statisticsimportnode to :typenode
```

```
connect :selectnode between :typenode and :featureselectionnode
```

### ***delete NODO***

```
delete NODO
```

Elimina el nodo especificado desde la ruta actual.

```
delete :statisticsimportnode
```

```
delete DRUG1N:variablefilenode
```

### ***disable NODO***

```
disable NODO
```

Desactiva el nodo especificado de la ruta actual, lo que tiene como consecuencia que se ignore el nodo durante la ejecución de la ruta. De esta forma se evita que tenga que eliminar u omitir el nodo y podrá dejarlo conectado al resto de nodos. Podrá editar la configuración del nodo; sin embargo, las modificaciones no surtirán efecto hasta que vuelva a activar el nodo.

```
disable :statisticsimportnode
```

```
disable DRUG1N:variablefilenode
```

### ***disconnect NODO***

```
disconnect NODO  
disconnect NODO1 from NODO2  
disconnect NODO1 between NODO2 and NODO3
```

Desconecta el nodo especificado del resto de nodos (por defecto) o desde nodos específicos según lo indicado.

```
disconnect :typenode
```

```
disconnect :typenode from :selectnode
```

### ***duplicate NODO***

```
duplicate NODO as NUEVONOMBRE
```

Crea un nuevo nodo como un duplicado del nodo especificado. Si lo desea, también se puede especificar la posición en términos absolutos o relativos.

```
duplicate :derivenode as flag1 at 100 400
```

```
duplicate flag1 as flag2 connected between flag1 and flag3
```

### ***enable NODO***

```
enable NODO
```

Activa el nodo desactivado anteriormente en la ruta actual, lo que tiene como consecuencia que se incluya el nodo durante la ejecución de la ruta. Si ha editado la configuración del nodo mientras estaba desactivado, las modificaciones surtirán efecto ahora.

```
enable :statisticsimportnode
```

```
enable DRUG1N:variablefilenode
```

### ***execute NODO***

```
execute NODO
```

Ejecuta el nodo especificado, por ejemplo:

```
execute :neuralnetworknode
```

Si el nodo no es terminal, la ejecución equivale a seleccionar la opción Ejecutar desde aquí del menú emergente.

Para ejecutar todos los nodos terminales de la ruta actual:

```
execute_all
```

*Sólo procesos independientes.* Para ejecutar el proceso de ruta asociado a la ruta actual:

```
execute_script
```

*Nota:* los procesos asociados con rutas diferentes pueden ejecutarse configurando la ruta como la actual o usando el comando `with`. [Si desea obtener más información, consulte el tema \*Cómo trabajar con varias rutas en el capítulo 3\* el p. 24.](#)

### ***export NODO as ARCHIVO***

```
export node NODO in DIRECTORIO format FORMATO  
export node NODO as ARCHIVO format FORMATO
```

**Exportar PMML.** Para exportar un modelo generado en formato PMML:

```
export Medicamento as c:/mimodelo.txt format pmml
```

**Exportar SQL.** Para exportar un modelo generado en formato SQL:

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

**Detalles del nodo.** Para exportar los detalles del nodo en formato HTML o de texto:

```
export Medicamento as c:\mimodelo.htm format html
```

```
export Medicamento as c:\mimodelo.txt format text
```

**Resumen del nodo.** Para exportar el resumen del nodo en formato HTML o de texto:

```
export Medicamento summary in c:/mismodelos format html
```

```
export Medicamento summary as c:/mimodelo.txt format text
```

```
export 'assocapriori' as 'C:/temp/assoc_apriori' format html
```

### ***flush NODO***

```
flush NODO
```

Vacía la caché del nodo especificado o de todos los nodos de la ruta. Si la caché no está activada o no está completa para un nodo determinado, esta operación no realiza ninguna función.

```
flush :mergenode
```

Para ejecutar todos los nodos de la ruta actual:

```
flush_all
```

### ***get node NODO***

```
get node NODO
```

Obtiene una referencia a un nodo existente. Puede ser una manera útil de garantizar referencias no ambiguas a los nodos.

```
var mynode  
set mynode = get node flag1:derivenode  
position ^mynode at 400 400
```

### ***load node NOMBREARCHIVO***

```
load node NOMBREARCHIVO
```

Carga un nodo guardado a la ruta actual.

```
load node c:/minodo.nod
```

### ***position NODO***

```
position NODO at X Y  
position NODO between NODO1 and NODO2  
position NODO connected between NODO1 and NODO2
```

Posiciona un nodo en el lienzo de ruta en términos absolutos o relativos. Si lo desea, también puede especificar las opciones de posición:

```
position DRUG1n:variablefilenode at 100 100
```

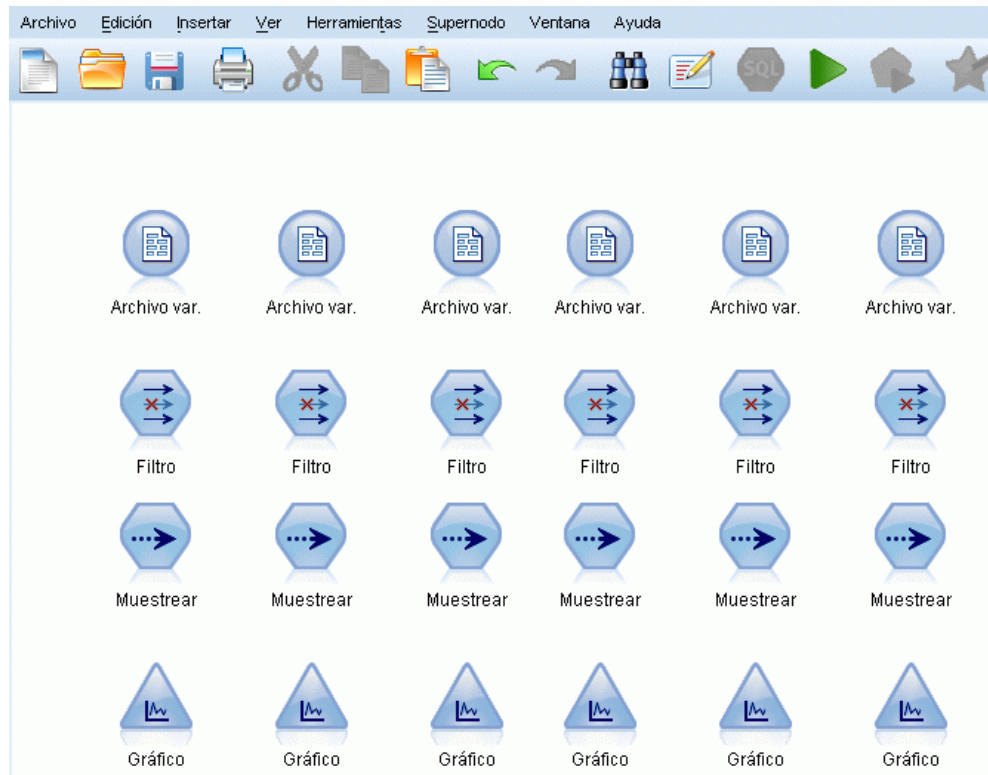
```
position Drug:net between DRUG2n and analysis
```

```
position :typenode connected between :variablefilenode and :tablenode
```

#### ***Posicionamiento de coordenadas***

El posicionamiento de los nodos sobre el lienzo de rutas utiliza una cuadrícula invisible  $x$ - $y$ . Puede utilizar la imagen que se muestra a continuación como referencia para las coordenadas de cuadrícula  $x$ - $y$ .

Figura 4-2  
Nodos creados en la posición especificada con las coordenadas x-y



### ***rename NODO as NUEVONOMBRE***

```
rename NODO as NUEVONOMBRE
```

Cambia el nombre del nodo especificado.

```
rename :derivenode as 'Flag1'
```

```
rename :varfilenode as 'testdata'
```

### ***retrieve node RUTA\_REPOSITORIO***

```
retrieve node RUTA_REPOSITORIO {label ETIQUETA | version VERSIÓN}
```

Recupera el nodo especificado desde IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
retrieve node "/samples/drugtypenode"
```

### ***save node NODO as NOMBREARCHIVO***

```
save node NODO as NOMBREARCHIVO
```



Guarda el nodo especificado.

```
save node :statisticsimportnode as c:/minode.nod
```

### **store node NODO as RUTA\_REPOSITORIO**

```
store node NODO as RUTA_REPOSITORIO {label ETIQUETA}
```

Almacena un nodo en IBM® SPSS® Collaboration and Deployment Services Repository. Si desea obtener más información, consulte el tema [Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59](#).

```
store node MEDICAMENTO1n as "/samples/drug1ntypenode"
```

```
store node :typenode as "/samples/drugtypenode"
```

## **Objetos del modelo**

Existen los siguientes comandos de procesos disponibles para objetos del modelo.

### **Nombres de nugget de modelo**

Se puede hacer referencia a los nugget de modelo (también denominados modelos generados) según el tipo, como con los objetos de nodo y de resultado. Las siguientes tablas muestran los nombres de referencia de los objetos del modelo.

Tenga en cuenta que estos nombres se utilizan específicamente para hacer referencia a los nugget de modelo en la paleta Modelos (en la esquina superior derecha de la ventana de IBM® SPSS® Modeler). Para hacer referencia a los nodos de modelo que se han añadido a una ruta para la puntuación, se utiliza un conjunto diferente de nombres con el prefijo *apply...* Si desea obtener más información, consulte el tema [Propiedades de nodos de nugget de modelo en el capítulo 17 el p. 253](#).

Por ejemplo, el proceso siguiente añade un nugget de modelo a la ruta actual, lo conecta a un nodo Tipo y crea y ejecuta un nodo Tabla. Observe que el nombre utilizado para insertar el modelo desde la paleta es diferente del nombre utilizado para hacer referencia al nodo del modelo “apply” cuando se ha añadido a la ruta (:featureselection versus :applyfeatureselectionnode).

```
insert model :featureselection at 150 250
connect Type to :applyfeatureselectionnode
create tablenode at 250 250
connect :applyfeatureselectionnode to :tablenode
execute :tablenode
```

*Nota:* Esto sólo es un ejemplo. En circunstancias normales, se recomienda hacer referencia a los modelos por nombre y tipo para evitar la confusión (por ejemplo, `response_01:featureselection`).

**Nombres de nugget de modelo (paleta de modelado)**

Nombre del modelo	Modelo
anomalydetection	Anomalia
a priori	A priori
autoclassifier	Clasificador automático
autocluster	Autoconglomeración
autonumeric	Autonumérico
bayesnet	Red bayesiana
c50	C5.0
carma	Carma
cart	Árbol C&R
chaid	CHAID
coxreg	Regresión de Cox
decisionlist	Lista de decisiones
discriminant	Discriminante
factor	PCA/Factorial
featureselection	Selección de funciones
genlin	Regresión lineal generalizada
kmeans	K-medias
knn	$k$ : vecino más cercano
kohonen	Kohonen
lineal	Lineal
logreg	Regresión logística
neuralnetwork	Red neuronal
quest	QUEST
regresión	Regresión lineal
secuencia	Secuencia
slrm	Modelo de respuesta de autoaprendizaje
statisticsmodel	Modelo de IBM® SPSS® Statistics
svm	Máquina de vectores de soporte
timeseries	Serie temporal
twostep	Dos fases

**Nombres de nugget de modelo (paleta de modelado de la base de datos)**

Nombre del modelo	Modelo
db2imassoc	Asociación IBM ISW
db2imcluster	Conglomerado de IBM ISW
db2imreg	Regresión de IBM ISW
db2imsequence	Secuencia IBM ISW

Nombre del modelo	Modelo
db2imtree	Árbol de decisión de IBM ISW
msassoc	Reglas de asociación de MS
msbayes	Naive bayesiano de MS
mscluster	Conglomerado de MS
mslogistic	Regresión logística de MS
msneuralnetwork	Red neuronal de MS
msregression	Regresión lineal de MS
mssequencecluster	Conglomeración de secuencias de MS
mstimeseries	Series temporales de MS
mstree	Árbol de decisión de MS
oraabn	Bayesiano adaptativo de Oracle
oraai	Oracle AI
oraapriori	Apriori de Oracle
oradecisiontree	Árbol de decisión de Oracle
oraglm	Oracle GLM
orakmeans	K-medias de Oracle
oramdl	LMD de Oracle
oranb	Bayesiano Naive de Oracle
oranmf	NMF de Oracle
oracluster	O-conglomerado de Oracle
orasvm	SVM de Oracle

### ***Evitar nombres duplicados del modelo***

Al utilizar los procesos para manipular los modelos generados, debe tener en cuenta que el hecho de permitir nombres de modelo duplicados puede originar referencias ambiguas. Para evitarlo, resulta útil utilizar nombres únicos para los modelos generados en el procesamiento.

Para configurar las opciones de los nombres de modelo duplicados:

- ▶ Elija en los menús:  
Herramientas > Opciones de usuario
- ▶ Pulse en la pestaña Notificaciones.
- ▶ Seleccione Sustituir modelo anterior para restringir los nombres duplicados de los modelos generados.

El comportamiento de la ejecución de procesos puede variar entre SPSS Modeler y IBM SPSS Collaboration and Deployment Services cuando haya referencias de modelo ambiguas. El cliente de SPSS Modeler incluye la opción “Reemplazar modelo anterior”, que reemplaza automáticamente los modelos que tengan el mismo nombre (por ejemplo, cuando un proceso se itera a través de un bucle para producir un modelo diferente cada vez). Sin embargo, esta opción no está disponible cuando el mismo proceso se ejecuta en IBM SPSS Collaboration and Deployment Services. Puede evitar esta situación cambiando el nombre del modelo generado en cada iteración para evitar referencias ambiguas a los modelos o borrando el modelo actual (por ejemplo, añadiendo una instrucción `clear generated palette`) antes del final del bucle.

### ***delete model MODELO***

```
delete model MODELO
```

Elimina un modelo especificado (o borra todos los modelos) desde la paleta de nugget de modelo.

```
delete model Medicamento
```

```
delete model Medicamento:c50
```

Para eliminar el último modelo insertado por el proceso actual:

```
delete last model
```

Para que funcione esta última instrucción, se debe de haber ejecutado la instrucción `insert model` al menos una vez durante la ejecución del proceso actual.

Para eliminar todos los nugget de modelo desde la paleta de modelos:

```
clear generated palette
```

### ***export model MODELO as ARCHIVO***

```
export model MODELO in DIRECTORIO format FORMATO  
export model MODELO as ARCHIVO format FORMATO
```

**Exportar PMML.** Para exportar el modelo generado en formato PMML:

```
export model Medicamento in c:/mismodelos format pmml
```

```
export model Medicamento as c:/mimodelo.xml format pmml
```

Si desea obtener más información, consulte el tema [Cómo importar y exportar modelos como PMML en el capítulo 10 en \*Manual de usuario de IBM SPSS Modeler 15\*](#).

**Exportar SQL.** Para exportar un modelo generado en formato SQL:

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

*Nota:* La exportación de SQL sólo está disponible para ciertos tipos de modelos. Si desea obtener más información, consulte el tema [Exploración de nugget de modelo en el capítulo 3 en \*Nodos de modelado de IBM SPSS Modeler 15\*](#).

**Detalles del modelo.** Para exportar detalles del modelo (como aparecen en la pestaña Modelo cuando se examina el nugget de modelo) en HTML o formato de texto:

```
export model Medicamento as c:\mimodelo.htm format html
```

```
export model Medicamento as c:\mimodelo.txt format text
```

*Nota:* Estos formatos no están disponibles para los modelos que no tienen una pestaña Modelo.

**Resumen del modelo.** Para exportar el resumen del modelo (la pestaña Resumen cuando se examina el nugget de modelo) en HTML o formato de texto:

```
export Medicamento summary in c:/mismodelos format html
```

```
export Medicamento summary as c:/mimodelo.txt format text
```

```
export 'assocapriori' as 'C:/temp/assoc_apriori' format html
```

Figura 4-3  
Pestaña del modelo de asociación exportada como HTML

	Consecuente	Antecedente	% de soporte	% de confianza
1	frozenmeal	beer and cannedveg	16,7	87,425
2	cannedveg	beer and frozenmeal	17,0	85,882
3	beer	frozenmeal and cannedveg	17,3	84,393
4	frozenmeal	beer	29,3	58,02
5	cannedveg	frozenmeal	30,2	57,285
6	frozenmeal	cannedveg	30,3	57,096
7	cannedveg	beer	29,3	56,997
8	beer	frozenmeal	30,2	56,291
9	beer	cannedveg	30,3	55,116
10	wine	confectionery	27,6	52,174
11	confectionery	wine	28,7	50,174

## ***insert model MODELO***

```
insert model MODELO
```

```
insert model MODELO at X Y
```

```
insert model MODELO between NOD01 and NOD02
```

```
insert model MODELO connected between NOD01 and NOD02
```

Añade el modelo a la ruta actual. Si lo desea, también puede especificar las opciones de posición y conexión.

```
insert model Kohonen between :typenode and :analysisnode
```

```
insert model Drug:neuralnetwork connected between 'Definir tipos' and 'Análisis'
```

### ***load model NOMBREARCHIVO***

```
load model NOMBREARCHIVO
```

Carga un modelo guardado en la paleta de modelos.

```
load model c:/mimodelo.gm
```

### ***retrieve model RUTA\_REPOSITORIO***

```
retrieve model RUTA_REPOSITORIO {label ETIQUETA | version VERSIÓN}
```

Recupera el modelo guardado desde IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
retrieve model "/my folder/Kohonen.gm"
```

### ***save model MODELO as NOMBREARCHIVO***

```
save model MODELO as NOMBREARCHIVO
```

Guarda el modelo especificado como un archivo de un modelo generado.

```
save model Medicamento as c:/mimodelo.gm
```

### ***store model MODELO as RUTA\_REPOSITORIO***

```
store model MODELO as RUTA_REPOSITORIO {label ETIQUETA}
```

Almacena un modelo especificado en IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
store model Kohonen as "/my folder/Kohonen.gm"
```

La extensión (\*.gm) es opcional, pero se debe utilizar de forma coherente al almacenar y recuperar el modelo. Por ejemplo, si se guarda simplemente como “Kohonen”, el modelo debería recuperarse por el mismo nombre. (Es decir, la extensión, si se utiliza, es simplemente parte del nombre del modelo.)

## **Objetos de ruta**

Existen los siguientes comandos de procesos disponibles para objetos de ruta.

### ***create stream NOMBREARCHIVO\_PORDEFECTO***

```
create stream NOMBREARCHIVO_PORDEFECTO
```

*Sólo procesos independientes.* Crea una ruta nueva en memoria con el nombre especificado. La ruta no se guarda automáticamente.

```
create stream 'Druglearn'
```

### ***close RUTA***

```
close RUTA
```

*Sólo procesos independientes.* Cierra la ruta especificada.

Para cerrar la ruta actual, escriba el comando con todos los caracteres en minúscula, como se detalla a continuación:

```
close stream
```

#### ***Procesos independientes***

Si se trabaja con varias rutas, debe tener en cuenta que `stream` (en minúsculas tal como se muestra) es realmente una variable especial que se utiliza para hacer referencia a la ruta actual. Para cerrar una ruta diferente, el valor de esta variable se puede restablecer:

```
set stream = get stream 'Stream5'  
close stream
```

También se puede especificar cualquier variable declarada que haga referencia a una ruta, por ejemplo:

```
var s  
set s = get stream 'Ruta2'  
save s as c:/ruta2.str  
close s
```

Por último, la ruta actual se puede volver a asignar temporalmente a través del comando `with stream`:

```
with stream 'Stream1'  
close stream  
endwith
```

### ***clear stream***

```
clear stream
```

Elimina todos los nodos de la ruta actual.

### ***get stream RUTA***

```
get stream RUTA
```

*Sólo procesos independientes.* Se utiliza para obtener una referencia a la ruta especificada, que se puede asignar a una variable local (o a la stream de la variable especial). La ruta especificada debe estar ya abierta.

```
var s
set s = get stream 'Druglearn'
close s
```

### ***load stream NOMBREARCHIVO***

```
load stream NOMBREARCHIVO
```

*Sólo procesos independientes.* Añade una ruta especificada al lienzo sin eliminar los nodos de la ruta actual.

```
load stream "c:/demos/druglearn.str"
```

**Open stream frente a load stream.** El comando `load stream` añade una ruta especificada al lienzo sin eliminar los nodos de la ruta actual. Este comando se utilizó más en versiones anteriores de IBM® SPSS® Modeler y se ha sustituido de forma extendida en versiones posteriores por la capacidad para abrir, administrar y copiar nodos entre varias rutas.

### ***open stream NOMBREARCHIVO***

```
open stream NOMBREARCHIVO
```

*Sólo procesos independientes.* Abre la ruta especificada.

```
open stream "c:/demos/druglearn.str"
```

### ***retrieve stream RUTA\_REPOSITORIO***

```
retrieve stream RUTA_REPOSITORIO {label ETIQUETA | version VERSIÓN}
retrieve stream URI [{#m.marker | #l.label}]
```

Recupera el nodo especificado desde IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
retrieve stream "/micarpeta/druglearn.str"
```

```
retrieve stream "spsscr:///modelos/drug%20model.gm#m.0:2005-10-12%2014:15:41.281"
```



## ***save RUTA as NOMBREARCHIVO***

```
save RUTA
save RUTA as NOMBREARCHIVO
```

Para guardar cambios en la ruta actual (asumiendo que se ha guardado antes), escriba el comando utilizando todos los caracteres en minúsculas, como se detalla a continuación:

```
save stream
```

Para guardar una ruta por primera vez con un nombre de archivo nuevo:

```
create stream nifty
create featureselectionnode
save stream as c:/nifty.str
```

### ***Procesos independientes***

Si se trabaja con varias rutas, debe tener en cuenta que `stream` (en minúsculas tal como se muestra) es realmente una variable especial que se utiliza para hacer referencia a la ruta actual. Para guardar una ruta diferente, el valor de esta variable se puede restablecer:

```
set stream = get stream 'Ruta5'
save stream
```

También se puede especificar cualquier variable declarada que haga referencia a una ruta, por ejemplo:

```
var s
set s = get stream 'Ruta2'
save s as c:/ruta2.str
close s
```

Por último, la ruta actual se puede volver a asignar temporalmente a través del comando `with stream`:

```
with stream 'Ruta1'
  save stream
endwith
```

[Si desea obtener más información, consulte el tema \*Cómo trabajar con varias rutas en el capítulo 3\* el p. 24.](#)

## ***store stream as RUTA\_REPOSITORIO***

```
store stream as RUTA_REPOSITORIO {label ETIQUETA}
store stream as URI [#l.label]
```

```
store stream as "/carpeta_1/carpeta_2/miruta.str"
```

Almacena la ruta actual en IBM® SPSS® Collaboration and Deployment Services Repository. Si desea obtener más información, consulte el tema [Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository](#) en el capítulo 5 el p. 59.

```
store stream as "/carpeta_1/carpeta_2/druglearn.str"
store stream as "spsscr:///carpeta_1/carpeta_2/mystream.str"
```

### **Procesos independientes**

Si se trabaja con varias rutas, debe tener en cuenta que `stream` (en minúsculas tal como se muestra) es realmente una variable especial que se utiliza para hacer referencia a la ruta actual. Para guardar una ruta diferente, el valor de esta variable se puede restablecer:

```
set stream = get stream 'Ruta5'
store stream as "/carpeta_1/miruta.str"
```

También se puede especificar cualquier variable declarada que haga referencia a una ruta, o se puede volver a asignar la ruta actual temporalmente a través del comando `with stream`:

```
with stream 'Ruta6'
store stream as "/carpeta_1/miruta.str"
endwith
```

### **with stream RUTA**

```
with stream RUTA
INSTRUCCIONES
endwith
```

*Sólo procesos independientes.* Ejecuta `STATEMENTS` con la `STREAM` especificada como ruta actual. La ruta actual original se vuelve a almacenar una vez que se han ejecutado las instrucciones.

```
with stream 'druglearn'
create typenode
execute_script
endwith
```

## **Objetos del proyecto**

Existen los siguientes comandos de procesos disponibles para objetos de nodos.

La extensión (`*.cpj`) es opcional, pero se debe utilizar de forma coherente al almacenar y recuperar un proyecto determinado.

### **execute\_project**

```
execute_project
```

Genera el informe por defecto del proyecto.

### ***load project NOMBREARCHIVO***

```
load project NOMBREARCHIVO
```

Abre la ruta especificada.

```
load project "C:/clemdata/DrugData.cpj"  
set ^project.summary="Initial modeling work on the latest data."  
set ^project.ordering=NameAddedType  
execute_project
```

### ***retrieve project RUTA\_REPOSITORIO***

```
retrieve project RUTA_REPOSITORIO {label ETIQUETA | version VERSIÓN}
```

Recupera un proyecto desde IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
retrieve project "/CRISPDM/DrugExample.cpj"
```

### ***save project as NOMBREARCHIVO***

```
save project  
save project as NOMBREARCHIVO
```

Guarda el proyecto actual.

### ***store project as RUTA\_REPOSITORIO***

```
store project as RUTA_REPOSITORIO {label ETIQUETA}
```

Almacena el proyecto actual en el IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
store project as "/CRISPDM/DrugExample.cpj"
```

## ***Objetos state***

Un estado guardado se puede cargar a través del comando `load state`.

### ***load state NOMBREARCHIVO***

```
load state NOMBREARCHIVO
```

Carga el estado especificado.

```
load state "c:/data/myproject.cst"
```

## Objetos de resultado

Se puede acceder a los resultados a través del comando `value`.

### *value RESULTADO*

`value RESULTADO at FILA COLUMNA`

Los nodos terminales contienen un parámetro de solo lectura llamado `output` que puede utilizarse para acceder al objeto generado más recientemente. En el caso de los nodos que producen resultados tabulares en filas y columnas, permite acceder al valor de la casilla especificada, por ejemplo:

```
execute :tablenode
set last_row = :tablenode.output.row_count
set last_column = :tablenode.output.column_count
set last_value = value :tablenode.output at ^last_row ^last_column
var myresults
set myresults = open create 'C:/myresults.txt'
write myresults 'El valor de la última casilla es ' >< ^last_value
```

La fila y la columna están desplazadas desde 1. Si el objeto de resultados no existe, se devuelve un error.

#### **Propiedades de objetos de resultado**

Las siguientes propiedades son comunes a todos los objetos de resultado (como los resultados de Tabla y Matriz) que incluyen datos en filas y columnas:

Propiedad	Descripción
<code>row_count</code>	Devuelve el número de filas de los datos.
<code>column_count</code>	Devuelve el número de columnas de los datos.

## Objetos de archivo

Existen los siguientes comandos de procesos para objetos de archivo.

### **close ARCHIVO**

`close ARCHIVO`

La instrucción anterior cierra el archivo especificado.

### **open ARCHIVO**

```
open create NOMBREARCHIVO
open append NOMBREARCHIVO
```

Las instrucciones anteriores cierran el archivo especificado.

- **create.** Crea el archivo si éste no existe o lo sobrescribe en caso contrario.
- **append.** Añade a un archivo existente. Genera un error si el archivo no existe.

Esto devuelve el identificador del archivo abierto.

```
var file
set file = open create 'C:/script.out'
for I from 1 to 3
  write file 'Stream ' >< I
endfor
close file
```

## ***write ARCHIVO***

```
write ARCHIVO EXPRESIÓN_TEXTO
writeln ARCHIVO EXPRESIÓN_TEXTO
```

Las expresiones anteriores escriben la expresión de texto en el archivo. La primera instrucción escribe el texto tal cual, mientras que la segunda escribe además una nueva línea después de haber escrito la expresión. Genera un error si FILE no es un objeto de archivo abierto.

```
var file
set file = open create 'C:/hola.txt'
writeln file 'Hola'
writeln file 'a todos'
write file '¿Queréis jugar a un juego?'
close file
```

## ***Objetos de resultados***

Existen los siguientes comandos de procesos disponibles para objetos de resultados.

### ***Nombres de tipo de resultados***

La siguiente tabla indica los tipos de objetos de resultados y los nodos que los crean. Para obtener una lista completa de formatos disponibles para exportar cada tipo de objeto de resultado, consulte la descripción de las propiedades del nodo que crea dicho tipo de resultado, disponible en [el capítulo 15, Propiedades de nodos de gráficos](#), y [el capítulo 19, Propiedades de los nodos de resultados](#).

<b>Tipo de objeto de resultado</b>	<b>Nodo</b>
analysisoutput	Análisis
collectionoutput	Colección
dataauditoutput	Auditar datos
distributionoutput	Distribución
evaluationoutput	Evaluación
histogramoutput	Histograma
matrixoutput	Matriz

Tipo de objeto de resultado	Nodo
meansoutput	Medias
multiplotoutput	G. múltiple
plotoutput	Gráfico
qualityoutput	Calidad
reportdocumentoutput	Este tipo de objeto no es de un nodo, es un resultado creado por un informe de proyecto
reportoutput	Informe
statisticsprocedureoutput	Resultado de Statistics
statisticsoutput	Estadísticos
tableoutput	Tabla
timeplotoutput	Gráfico de tiempo
weboutput	Web

### ***delete output RESULTADO***

```
delete output RESULTADO
```

Elimina el resultado especificado desde la paleta del administrador. Por ejemplo:

```
delete output :statisticsoutput
```

Para eliminar todos los elementos del resultado desde la paleta del administrador:

```
clear outputs
```

### ***export output RESULTADO***

```
export output RESULTADO as ARCHIVO format FORMATO
```

Exporta el resultado en el formato especificado. Tenga en cuenta que los formatos disponibles dependen del tipo de resultado, pero deberían reflejar aquellos disponibles en el menú Exportar al examinar el resultado especificado.

```
export output :statisticsoutput as "C:/output/statistics.html" format html
export output :matrixoutput as "C:/output/matrix.csv" format delimited
export output :tableoutput as "C:/output/table.tab" format transposed formatted
```

### ***get output RESULTADO***

```
get output RESULTADO
```

Obtiene una referencia al resultado especificado. Por ejemplo, se podría utilizar un bucle para obtener una serie de objetos de resultado y exportar cada uno de estos por turnos.

```
execute_all
for item in statisticsoutput matrixoutput tableoutput
var theoutput
set theoutput = get output :^item
```

```
set filename = 'c:/><^item ><'.htm'  
export output ^theoutput as ^filename format html  
endfor
```

### ***load output NOMBREARCHIVO***

```
load output NOMBREARCHIVO
```

Carga el resultado especificado.

```
load output 'c:/matrix.cou'
```

### ***retrieve output RUTA\_REPOSITORIO***

```
retrieve output RUTA_REPOSITORIO {label ETIQUETA | version VERSIÓN}
```

Recupera el resultado especificado desde IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
retrieve output "/results/mytable"
```

### ***save output SALIDA as NOMBREARCHIVO***

```
save output as NOMBREARCHIVO
```

Guarda el resultado especificado.

```
save output :matrixoutput as 'c:/matrix.cou'
```

### ***store output SALIDA as RUTA\_REPOSITORIO***

```
store output SALIDA as RUTA_REPOSITORIO {label ETIQUETA}
```

Almacena el resultado especificado en IBM® SPSS® Collaboration and Deployment Services Repository. [Si desea obtener más información, consulte el tema Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59.](#)

```
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

```
store output :tableoutput as "/results/mytable"
```

## ***Sugerencias sobre procesos***

Esta sección proporciona una visión general de las técnicas y sugerencias para utilizar procesos, incluida la modificación de la ejecución de la ruta, la utilización de una contraseña codificada en un proceso y el acceso a objetos en IBM® SPSS® Collaboration and Deployment Services Repository.

### ***Modificación de la ejecución de una ruta***

Cuando se ejecuta una ruta, sus nodos terminales se ejecutan en un orden optimizado para la situación por defecto. En algunos casos, es posible que prefiera un orden de ejecución diferente. Para modificar el orden de ejecución de una ruta, lleve a cabo los siguientes pasos desde la pestaña Proceso del cuadro de diálogo Propiedades de ruta:

- ▶ Comience con un proceso vacío.
- ▶ Pulse en el botón Añadir proceso por defecto de la barra de herramientas y añada el proceso de ruta por defecto.
- ▶ Cambie el orden de las instrucciones del proceso de ruta por defecto por el orden en que desee que se ejecuten las instrucciones.

### ***Recorrido en bucle de nodos***

Se puede utilizar un bucle for en combinación con la propiedad `^stream.nodes` para recorrer en bucle todos los nodos de una ruta. Por ejemplo, el siguiente proceso recorre en bucle todos los nodos y cambia los nombres de campos de cualquier nodo Filtro a mayúsculas.

El proceso puede emplearse en cualquier ruta que tenga un nodo Filtro, incluso si no hay campos filtrados. Simplemente, añada un nodo Filtro que recorra todos los campos para cambiar todos los nombres de campo a mayúsculas.

```
var my_node
var loop_me
var var_name

for my_node in ^stream.nodes
  if ^my_node.node_type = filternode then
    for loop_me in _fields_to ^my_node:filternode
      set var_name = lowertoupper(^my_node:filternode.new_name.^loop_me)
      set ^my_node:filternode.new_name.^loop_me = ^var_name
    endfor
  else
  endif
endfor
```



El proceso recorre en bucle los nodos de la ruta actual, cuando es devuelto por la propiedad `^stream.nodes` y comprueba cada nodo para ver si es un Filtro. Si es así, el proceso recorre en bucle cada campo del nodo y utiliza la función `lowertoupper()` para cambiar el nombre a mayúsculas.

*Sugerencia:* Para cambiar los nombres de campo a minúsculas, utilice `uppertolower()` en su lugar.

## **Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository**

*Nota:* Es necesario disponer de una licencia independiente para acceder a un repositorio IBM® SPSS® Collaboration and Deployment Services. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Si ha obtenido una licencia para IBM® SPSS® Collaboration and Deployment Services Repository, puede almacenar, recuperar, bloquear y desbloquear objetos del repositorio mediante comandos de procesos. El repositorio permite administrar el ciclo vital de los modelos de minería de datos y los objetos predictivos relacionados dentro del contexto de las aplicaciones, herramientas y soluciones empresariales. [Si desea obtener más información, consulte el tema Acerca de IBM SPSS Collaboration and Deployment Services Repository en el capítulo 9 en Manual de usuario de IBM SPSS Modeler 15.](#)

### **Conexión con IBM SPSS Collaboration and Deployment Services Repository**

Para acceder al repositorio, primero debe establecer una válida conexión con el mismo, mediante el menú Herramientas de la interfaz de usuario de IBM® SPSS® Modeler o mediante la línea de comandos. [\(Si desea obtener más información, consulte el tema Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository en el capítulo 7 el p. 73.\)](#)

### **Almacenamiento y recuperación de objetos**

En un proceso, los comandos `retrieve` y `store` permiten acceder a varios objetos, entre los que se incluyen rutas, modelos, resultados, nodos y proyectos. La sintaxis es la siguiente:

```
store object as REPOSITORY_PATH {label ETIQUETA}
store object as URI [#l.label]
```

```
retrieve object REPOSITORY_PATH {label ETIQUETA | version VERSIÓN}
retrieve object URI [{#m.marker | #l.label}]
```

`REPOSITORY_PATH` proporciona la ubicación del objeto en el repositorio. La ruta debe estar entre comillas y utilizar barras diagonales como delimitadores. No distingue entre mayúsculas y minúsculas.

```
store stream as "/folder_1/folder_2/mystream.str"
store model Drug as "/myfolder/drugmodel"
store model Drug as "/myfolder/drugmodel.gm" label "final"
store node DRUG1n as "/samples/drug1ntypenode"
store project as "/CRISPD/DrugExample.cpj"
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

Si lo desea, se puede incluir una extensión como *.str* o *.gm* en el nombre de objeto, aunque no es necesario siempre que el nombre sea coherente. Por ejemplo, si un modelo se almacena sin una extensión, deberá recuperarse por el mismo nombre:

```
store model "/myfolder/drugmodel"
retrieve model "/myfolder/drugmodel"
```

frente a:

```
store model "/myfolder/drugmodel.gm"
retrieve model "/myfolder/drugmodel.gm" version "0:2005-10-12 14:15:41.281"
```

Tenga en cuenta que cuando recupera objetos, la versión más reciente del objeto siempre se devuelve a menos que especifique una versión o etiqueta. Al recuperar un objeto de nodo, el nodo se introduce automáticamente en la ruta actual. Al recuperar un objeto de ruta, debe utilizar un proceso independiente. No puede recuperar un objeto de ruta de un proceso de ruta.

### ***Bloqueo y desbloqueo de objetos***

Puede bloquear un objeto desde un proceso para evitar que otros usuarios actualicen cualquiera de las versiones existentes o creen nuevas versiones. También puede desbloquear un objeto que haya bloqueado.

La sintaxis para bloquear y desbloquear un objeto es:

```
lock REPOSITORY_PATH
lock URI
```

```
unlock REPOSITORY_PATH
unlock URI
```

Como cuando se almacenan y se recuperan objetos, `REPOSITORY_PATH` le ofrece la ubicación del objeto en el repositorio. La ruta debe estar entre comillas y utilizar barras diagonales como delimitadores. No distingue entre mayúsculas y minúsculas.

```
lock "/myfolder/Stream1.str"
```

```
unlock "/myfolder/Stream1.str"
```

Si lo prefiere, puede utilizar un identificador de recursos uniforme (URI) en lugar de una ruta de repositorio para proporcionar la ubicación del proyecto. El URI debe incluir el prefijo `spsscr:` y debe estar entre comillas. Sólo se puede utilizar barras diagonales como delimitadores, y los espacios deben estar codificados. Es decir, utilizar `%20` en lugar de un espacio en la ruta. El URI no distingue entre mayúsculas y minúsculas. A continuación aparecen algunos ejemplos:

```
lock "spsscr:///myfolder/Stream1.str"
```

```
unlock "spsscr:///myfolder/Stream1.str"
```

Tenga en cuenta que el bloqueo de objetos se aplica a todas las versiones de un objeto: no puede bloquear o desbloquear versiones por separado.

## Generación de una contraseña codificada

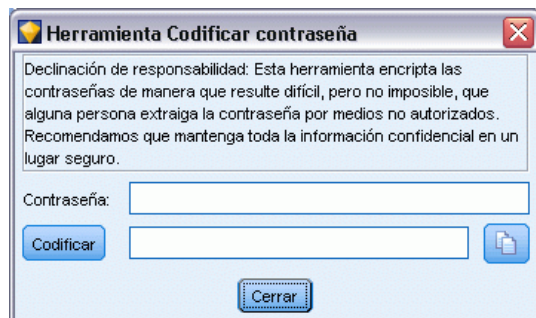
En algunos casos, puede que necesite incluir una contraseña en un proceso. Por ejemplo, es posible que desee acceder a un origen de datos protegido con contraseña. Las contraseñas codificadas pueden utilizarse en:

- Propiedades de los nodos para un origen de base de datos y nodos de resultado
- Argumentos de línea de comando para conectarse al servidor
- Propiedades de conexión con la base de datos almacenadas en un archivo *.par* (archivo de parámetro generado desde la ficha Publicar de un nodo de exportación)

En la interfaz de usuario se proporciona una herramienta para generar contraseñas codificadas en función del algoritmo Blowfish (si desea obtener más información, consulte <http://www.schneier.com/blowfish.html>). Una vez codificada, puede copiar y almacenar la contraseña en archivos de proceso y argumentos de líneas de comando. La propiedad del nodo *epassword* utilizada para *database*node y *database*exportnode almacena la contraseña codificada.

- ▶ Para generar una contraseña codificada, en el menú Herramientas seleccione: Codificar contraseña...

Figura 5-1  
Herramienta Codificar contraseña



- ▶ Especifique una contraseña en el cuadro de texto Contraseña.
- ▶ Pulse en Codificar para generar una codificación aleatoria de la contraseña.
- ▶ Pulse en el botón Copiar para copiar la contraseña codificada al Portapapeles.
- ▶ Pegue la contraseña en el proceso o parámetro deseado.

## Comprobación del proceso

Puede comprobar rápidamente la sintaxis de todos los tipos de procesos pulsando en el botón de comprobación de la barra de herramientas del cuadro de diálogo del proceso de la ruta.

Figura 5-2  
Iconos de barra de herramientas del proceso de ruta



En la comprobación de errores se avisa de cualquier error que se detecte en el código y se sugieren recomendaciones de mejora. Para ver la línea con errores, pulse en los comentarios, en la mitad inferior del cuadro de diálogo. Los errores se señalan en rojo.

## ***Procesamientos desde la línea de comandos***

El procesamiento permite ejecutar operaciones típicamente desarrolladas en la interfaz de usuario. Simplemente especifique y ejecute una ruta independiente en la línea de comandos cuando ejecute IBM® SPSS® Modeler. Por ejemplo:

```
client -script scores.txt -execute
```

La marca `-script` carga el proceso especificado, mientras que la marca `-execute` ejecuta todos los comandos del archivo de proceso.

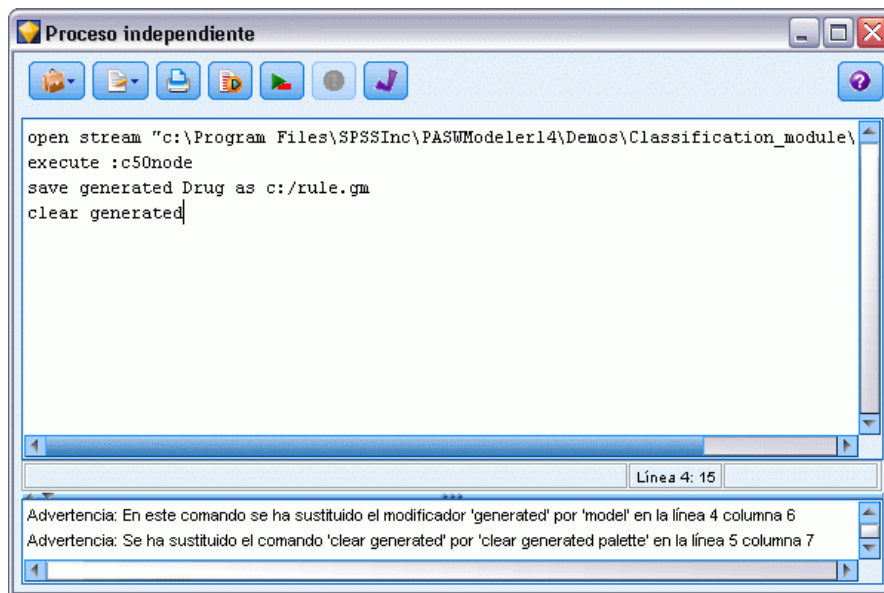
## ***Compatibilidad con versiones anteriores***

Los procesos creados en versiones anteriores de IBM® SPSS® Modeler deberían funcionar normalmente sin cambios en la versión actual. Sin embargo, los nuggets de modelos podrán ahora insertarse en la ruta automáticamente (es el comportamiento por defecto) y podrán sustituir o complementar un nugget existente del tipo en la ruta. El que esto ocurra depende de la configuración de las opciones Añadir modelo a ruta y Sustituir modelo anterior (Herramientas > Opciones > Opciones de usuario > Notificaciones). Por ejemplo, es posible que tenga que modificar un proceso de una versión anterior en el que la sustitución del nugget se trate borrando el nugget existente e insertando uno nuevo.

Es posible que los procesos creados en esta versión no funcionen en versiones anteriores.

Si un proceso creó una liberación antigua utiliza un comando que se ha sustituido desde entonces (o desaprobado), la forma antigua se seguirá admitiendo, pero aparecerá un mensaje de advertencia. Por ejemplo, la antigua palabra clave `generated` ha sido sustituida por `model`, y `clear generated` ha sido sustituida por `clear generated palette`. Los procesos que utilizan las formas antiguas se seguirán ejecutando, pero se mostrará una advertencia.

Figura 5-3  
Ejecución de un proceso que utiliza un comando deprecated



## Ejemplos de procesos

Esta sección proporciona varios ejemplos que muestran cómo se pueden utilizar los procesos en IBM® SPSS® Modeler.

### Informe del nodo Tipo

Este proceso crea un informe en formato HTML que incluye información sobre los campos de la ruta actual. El proceso se puede utilizar con cualquier ruta que tenga un nodo Tipo instanciado y se puede extender fácilmente para cubrir propiedades o nodos adicionales.

- Se utilizan etiquetas HTML estándar para dar formato a los resultados con el fin de mostrarlos en un explorador estándar.
- Se utiliza un nodo Tipo de IBM® SPSS® Modeler para acceder a las propiedades de cada campo. El proceso se puede ampliar fácilmente para cubrir propiedades adicionales expuestas a través del nodo Tipo, como los valores perdidos o el papel del campo. [Si desea obtener más información, consulte el tema Propiedades de typenode en el capítulo 14 el p. 175.](#)
- Se utilizan comandos de procesamiento de SPSS Modeler para escribir el resultado en un archivo y recorrer los campos en bucle con el fin de acceder a las propiedades de cada uno. [Si desea obtener más información, consulte el tema Comandos de procesos en el capítulo 4 el p. 31.](#)

Figura 6-1  
*Proceso de muestra del informe del nodo Tipo*

```
# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"
writeln myreport "<header>Información del nodo Tipo de IBM SPSS Modeler</header>"
writeln myreport "<body><br><br>"

#create the table and write out the headers
writeln myreport "<table border='1'"
writeln myreport "<tr bgcolor='C0C0C0'"
writeln myreport "<td>Campo</td><td>Tipo</td><td>Valores</td>"
writeln myreport "</tr>"

# loop through fields and add a row for each
var current_field
for current_field in _fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >> ^current_field >> "</td>"
  write myreport "<td>" >> Type:typenode.type.^current_field >> "</td>"
```

```

# add values for numeric fields
if Type:typenode.type.^current_field = Range then
  writeln myreport "<td>" << Type:typenode.values.^current_field << "</td>"
endif

# add values for flag fields
if Type:typenode.type.^current_field = Flag then
  writeln myreport "<td>" << Type:typenode.values.^current_field << "</td>"
endif

# add values for nominal fields
if Type:typenode.type.^current_field = Set then
  writeln myreport "<td>"
  var current_value
  for current_value in Type:typenode.values.^current_field
    writeln myreport ^current_value << "<BR/>"
  endfor
  writeln myreport "</td>"
endif

  writeln myreport "</tr>"
endfor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport

```

### **Creación del archivo de resultados**

En primer lugar, el proceso crea un nuevo archivo HTML y añade las etiquetas necesarias para crear una tabla con una fila de encabezado que incluya los títulos de columna *Campo*, *Tipo* y *Valores*. (Cada par de etiquetas <td> </td> crea una casilla dentro de una fila de la tabla.) Estas columnas se rellenarán para cada campo según las propiedades del nodo Tipo.

```

# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"
writeln myreport "<header>Información del nodo Tipo de IBM SPSS Modeler</header>"
writeln myreport "<body><br><br>"

#create the table and write out the headers
writeln myreport "<table border=\"1\">"
writeln myreport "<tr bgcolor=\"COCOC0\">"
writeln myreport "<td>Campo</td><td>Tipo</td><td>Valores</td>"
writeln myreport "</tr>"

```

**Recorrido en bucle de los campos**

A continuación, el proceso recorre en bucle todos los campos del nodo Tipo y añade una fila para cada campo, que indica el nombre y el tipo de campo.

```
# loop through fields and add a row for each
var current_field
for current_field in _fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >> ^current_field >> "</td>"
  write myreport "<td>" >> Type:typenode.type.^current_field >> "</td>"
```

**Valores para campos marca y continuo**

Para los campos continuos (rango numérico), la propiedad `typenode.values` devuelve los valores altos y bajos con el formato `[0.500517, 0.899774]`, que se muestra en la tabla. Para los campos de marcas, se muestran los valores verdadero/falso con un formato similar.

```
# add values for numeric fields
if Type:typenode.type.^current_field = Range then
  writeln myreport "<td>" >> Type:typenode.values.^current_field >> "</td>"
endif

# add values for flag fields
if Type:typenode.type.^current_field = Flag then
  writeln myreport "<td>" >> Type:typenode.values.^current_field >> "</td>"
endif
```

**Valores para campos nominales**

Para los campos nominales, la propiedad `typenode.values` devuelve la lista completa de valores definidos. El proceso recorre en bucle los campos de la lista para insertar por turnos cada valor, con un salto de línea (etiqueta `<br/>`) entre cada uno.

```
# add values for nominal fields
if Type:typenode.type.^current_field = Set then
  writeln myreport "<td>"
  var current_value
  for current_value in Type:typenode.values.^current_field
    writeln myreport ^current_value >> "<BR/>"
  endfor
  writeln myreport "</td>"
endif
```

**Cierre del archivo**

Finalmente, el proceso cierra la fila, cierra las etiquetas `<table>`, `<body>` y `<html>` y cierra el archivo de resultados.



```
writeln myreport "</tr>"
endfor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport
```

## Informe de rutas

Este proceso crea un informe en formato HTML que incluye el nombre, el tipo y la notación para cada nodo de la ruta actual. Además de crear un archivo HTML y acceder a las propiedades del nodo y la ruta, muestra cómo crear un bucle que ejecute un conjunto de instrucciones específico para cada nodo dentro de una ruta. Se puede utilizar con cualquier ruta.

Figura 6-2

*Proceso de muestra del informe de rutas*

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile " <BODY>"
writeln myfile " <HEAD>Report for stream " >> ^stream.name >> ".str</HEAD>"
writeln myfile " <p>" >> ^stream.annotation >> "</p>"

#Create the table with header row
writeln myfile "<TABLE border=\\"1\" width=\\"90%\">"
writeln myfile " <tr bgcolor=\\"lightgrey\" colspan=\\"3\">"
writeln myfile " <th>Node Name</th>"
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile " </tr>"

# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.name
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.node_type
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.annotation >> "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
endfor
```

```
writeln myfile "</TABLE>"
writeln myfile "</BODY>"
writeln myfile "</HTML>"
close myfile
```

### **Creación del informe**

En primer lugar, el proceso crea un nuevo archivo HTML con los elementos <BODY> y <HEAD>. La propiedad `^stream.name` devuelve el nombre de la ruta actual, que se introduce en el encabezado. El operador `><` se utiliza para concatenar cadenas.

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile " <BODY>"
writeln myfile " <HEAD>Report for stream " >< ^stream.name >< ".str</HEAD>"
writeln myfile " <p>" >< ^stream.annotation >< "</p>"
```

A continuación, el proceso crea una tabla HTML con una fila de encabezado que incluye los títulos de columna *Nombre de nodo*, *Tipo* y *Anotación*. (Cada par de etiquetas `<td></td>` crea una casilla dentro de una fila de la tabla.)

```
#Create the table with header row
writeln myfile "<TABLE border=\"1\" width=\"90%\">"
writeln myfile " <tr bgcolor=\"lightgrey\" colspan=\"3\">"
writeln myfile " <th>Node Name</th>"
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile " </tr>"
```

A continuación, el proceso recorre en bucle todos los nodos de la ruta actual. Se añade una fila a la tabla para cada nodo, que incluye el nombre, el tipo y la anotación. Tras la anotación, se inserta un espacio “sin ruptura” (`&nbsp;`) para evitar que se cree una casilla vacía en los casos en que no se especifica la anotación para un nodo determinado. (Las casillas vacías pueden generar un formato inesperado en la representación de la tabla.)

```
# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.name
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.node_type
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.annotation >< "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
```

```
endfor
```

Finalmente, el proceso añade las etiquetas de HTML necesarias para cerrar el documento y cierra el archivo.

```
writeln myfile "</TABLE>"  
writeln myfile "</BODY>"  
writeln myfile "</HTML>"  
close myfile
```

# Argumentos de la línea de comandos

## Invocación del software

Puede utilizar la línea de comandos del sistema operativo para iniciar IBM® SPSS® Modeler de la siguiente manera:

- ▶ En un ordenador en el que se haya instalado IBM® SPSS® Modeler, abra una ventana de DOS o del símbolo del sistema.
- ▶ Para iniciar la interfaz de SPSS Modeler en modo interactivo, escriba el comando `modelerclient` seguido de los argumentos deseados, por ejemplo:

```
modelerclient -stream report.str -execute
```

Los argumentos disponibles (modificadores) permiten conectar con un servidor, cargar rutas, ejecutar procesos o especificar otros parámetros, según sea necesario.

## Uso de argumentos en la línea de comandos

Se pueden añadir argumentos de línea de comandos (también denominados **marcas**) al comando inicial `modelerclient` o para alterar la invocación de IBM® SPSS® Modeler.

Por ejemplo, puede utilizar las marcas `-server`, `-stream` y `-execute` para conectar con un servidor y, a continuación, cargar y ejecutar una ruta, de la siguiente forma:

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

Tenga en cuenta que al ejecutarse en una instalación cliente local, no se necesitan argumentos de conexión con el servidor.

Los valores de parámetros que contienen espacios se pueden poner entre comillas, por ejemplo:

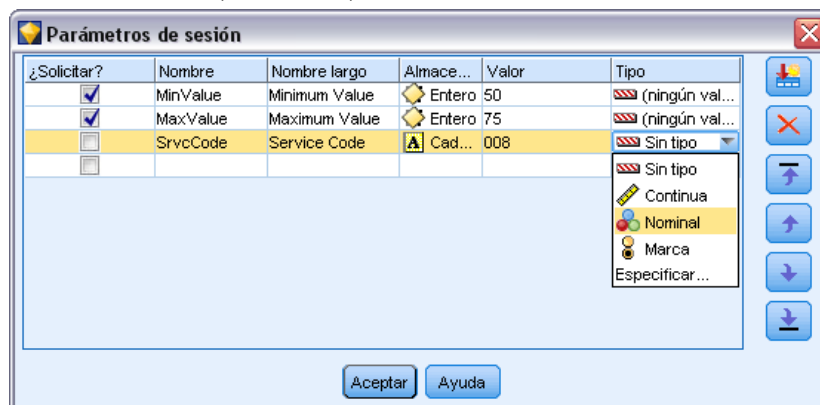
```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

También puede ejecutar procesos y estados de SPSS Modeler de esta forma, utilizando las marcas `-state` y `-script` respectivamente.

### Depuración de argumentos de la línea de comandos

Para depurar una línea de comandos, utilice el comando `modelerclient` para iniciar SPSS Modeler con los argumentos deseados. Esto permite comprobar que los comandos se ejecutarán como se espera. También puede confirmar los valores de cualquier parámetro pasado desde la línea de comandos en el cuadro de diálogo Parámetros de sesión (menú Herramientas, Definir parámetros de sesión).

Figura 7-1  
Establecimiento de parámetros para la sesión



## Combinación de varios argumentos

Es posible combinar varios argumentos en un único archivo de comandos especificado en la invocación utilizando el símbolo @ seguido del nombre de archivo. De este modo podrá acortar la invocación de la línea de comandos y superar cualquier limitación del sistema operativo en la longitud del comando. Por ejemplo, el siguiente comando de inicio utiliza todos los argumentos especificados en el archivo de referencia <commandFileName>.

```
modelerclient @<commandFileName>
```

Ponga el nombre del archivo y la ruta del archivo de comandos entre comillas si hay que incluir espacios, de la siguiente forma:

```
modelerclient @"C:\Archivos de programa\IBM\SPSS\Modeler\...\scripts\my_command_file.txt"
```

El archivo de comandos puede contener todos los argumentos especificados previamente a nivel individual en el inicio. Por ejemplo:

```
-stream report.str
-Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt
-execute
```

Cuando escriba y referencie archivos de comandos, asegúrese de cumplir estas restricciones:

- Utilice sólo un comando por línea.
- No incruste un argumento @CommandFile en un archivo de comandos.

## Argumentos de conexión con el servidor

La marca -server indica a IBM® SPSS® Modeler que debe conectar con un servidor público, y las marcas -hostname, -use\_ssl, -port, -username, -password, y -domain se utilizan para indicar a SPSS Modeler cómo conectar con el servidor público. Si no se especifica ningún argumento -server, se utilizará el servidor por defecto o local.

## Ejemplos

Para conectarse con un servidor público:

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

Para conectarse con un conglomerado de servidores:

```
modelerclient -server -cluster "Ordenadores de garantía de calidad" \
-spsscr_hostname pes_host -spsscr_port 8080 \
-spsscr_username asmith -spsscr_epassword xyz
```

Tenga en cuenta que para conectarse a un conglomerado de servidores necesita Coordinator of Processes a través de IBM® SPSS® Collaboration and Deployment Services, de modo que debe utilizar el argumento `-cluster` junto con las opciones de conexión de repositorio (`spsscr_*`).

[Si desea obtener más información, consulte el tema Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository el p. 73.](#)

Argumento	Comportamiento/Descripción
<code>-server</code>	Ejecuta SPSS Modeler en el modo servidor, conectándose a un servidor público utilizando las marcas <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> , y <code>-domain</code> .
<code>-hostname &lt;name&gt;</code>	Nombre de host del equipo servidor. Disponible en el modo servidor solamente.
<code>-use_ssl</code>	Especifica que la conexión debería utilizar SSL (secure socket layer). La marca es opcional, el parámetro por defecto <i>no</i> utiliza SSL.
<code>-port &lt;número&gt;</code>	Número de puerto del servidor especificado. Disponible en el modo servidor solamente.
<code>-cluster &lt;name&gt;</code>	Especifica una conexión a un conglomerado de servidores en lugar de un servidor especificado; este argumento es una alternativa a los argumentos <code>hostname</code> , <code>port</code> y <code>use_ssl</code> . El nombre es el del conglomerado o un URI exclusivo que identifica el conglomerado en IBM® SPSS® Collaboration and Deployment Services Repository. Coordinator of Processes gestiona el conglomerado de servidores a través de IBM SPSS Collaboration and Deployment Services. <a href="#">Si desea obtener más información, consulte el tema Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository el p. 73.</a>
<code>-username &lt;name&gt;</code>	Nombre de usuario con el que iniciar sesión en el servidor. Disponible en el modo servidor solamente.
<code>-password &lt;password&gt;</code>	Contraseña con la que iniciar sesión en el servidor. Disponible en el modo servidor solamente. <i>Nota:</i> si no se utiliza el argumento <code>-password</code> , se le solicitará una contraseña.
<code>-epassword &lt;encodedpasswordstring&gt;</code>	Contraseña codificada con la que iniciar sesión en el servidor. Disponible en el modo servidor solamente. <i>Nota:</i> Las contraseñas codificadas se pueden generar desde el menú Herramientas de la aplicación SPSS Modeler.
<code>-domain &lt;name&gt;</code>	Dominio utilizado para iniciar sesión en el servidor. Disponible en el modo servidor solamente.
<code>-P &lt;nombre&gt;=&lt;valor&gt;</code>	Se utiliza para definir un parámetro de inicio. También se puede utilizar para definir propiedades de nodos (parámetros de propiedades).

## Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository

*Nota:* Es necesario disponer de una licencia independiente para acceder a un repositorio IBM® SPSS® Collaboration and Deployment Services. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Si desea almacenar o recuperar objetos de IBM SPSS Collaboration and Deployment Services a través de la línea de comandos, debe especificar una conexión válida con IBM® SPSS® Collaboration and Deployment Services Repository. Por ejemplo:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

La siguiente tabla enumera los argumentos que pueden utilizarse para configurar la conexión:

Argumento	Comportamiento/Descripción
-spsscr_hostname <nombre de host o dirección IP>	El nombre del host o la dirección IP del servidor en que se ha instalado IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <número>	Número de puerto en el que el IBM SPSS Collaboration and Deployment Services Repository acepta las conexiones (normalmente 8080 por defecto).
-spsscr_use_ssl	Especifica que la conexión debería utilizar SSL (secure socket layer). La marca es opcional, el parámetro por defecto <i>no</i> utiliza SSL.
-spsscr_username <name>	Nombre de usuario con el que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <password>	Contraseña con la que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <contraseña codificada>	Contraseña cifrada con la que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_domain <name>	Dominio utilizado para iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository. Esta marca es opcional, no la utilice a menos que inicie sesión mediante LDAP o Active Directory.

## Argumentos del sistema

En la siguiente tabla se describen los argumentos del sistema disponibles para la invocación de la línea de comandos de la interfaz de usuario:

Argumento	Comportamiento/Descripción
@ <archivo de comandos>	El carácter @ seguido de un nombre de archivo especifica una lista de comandos. Cuando modelerclient encuentra un argumento que comienza por @, opera en los comandos de este archivo como si hubieran estado en la línea de comandos. <a href="#">Si desea obtener más información, consulte el tema Combinación de varios argumentos el p. 71.</a>
-directory <dir>	Define el directorio de trabajo por defecto. En el modo local, este directorio se utiliza tanto para datos como para resultados.

Argumento	Comportamiento/Descripción
-server_directory <dir>	Define el directorio de servidor por defecto para datos. El directorio de trabajo, especificado con la marca -directory, se utiliza para resultados.
-execute	Después del inicio, ejecuta cualquier ruta, estado o proceso que se haya cargado en el inicio. Si se carga un proceso además de una ruta o un estado, el proceso se ejecutará solo.
-stream <ruta>	Carga en el inicio la ruta especificada. Se pueden especificar varias rutas, pero la última se definirá como la actual.
-script <script>	Carga en el inicio el proceso independiente especificado. Se puede especificar además de una ruta o un estado, tal y como se describe a continuación, pero sólo se puede cargar un único proceso en el inicio.
-model <modelo>	En el inicio, carga el modelo generado (archivo de formato .gm) especificado.
-state <estado>	Carga en el inicio el estado especificado guardado.
-project <proyecto>	Carga el proyecto especificado. Sólo se puede cargar un único proyecto en el inicio.
-output <resultados>	Carga en el inicio el objeto de resultados guardado (archivo de formato .cou).
-help	Muestra una lista de argumentos de la línea de comandos. Cuando se especifica esta opción, todos los demás argumentos se ignoran y se muestra la pantalla Ayuda.
-P <nombre>=<valor>	Se utiliza para definir un parámetro de inicio. También se puede utilizar para definir propiedades de nodos (parámetros de propiedades).

*Nota:* los directorios por defecto también se pueden definir en la interfaz de usuario. Para acceder a las opciones en el menú Archivo, seleccione Definir directorio o Definir directorio de servidor.

### **Carga de varios archivos**

Desde la línea de comandos puede cargar varias rutas, estados y resultados en el inicio repitiendo el argumento relevante para cada objeto cargado. Por ejemplo, para cargar y ejecutar dos rutas llamadas *report.str* y *train.str*, utilizaría el siguiente comando:

```
modelerclient -stream report.str -stream train.str -execute
```

### **Carga de objetos desde IBM SPSS Collaboration and Deployment Services Repository**

Como puede cargar determinados objetos de un archivo o desde IBM® SPSS® Collaboration and Deployment Services Repository (si dispone de licencia), el prefijo de nombre de archivo *spsscr:* y, si lo desea, *file:* (para objetos en disco) indica a IBM® SPSS® Modeler donde buscar el objeto. El prefijo funciona con las siguientes marcas:

- -stream
- -script
- -output
- -model
- -project

El prefijo se utiliza para crear un URI que especifique la ubicación del objeto, por ejemplo,



-stream "spsscr:///folder\_1/scoring\_stream.str". La presencia del prefijo spsscr: necesita que se haya especificado una conexión válida a IBM SPSS Collaboration and Deployment Services Repository en el mismo comando. Así, por ejemplo, el comando completo sería:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

Para obtener más detalles sobre URI para objetos en el IBM SPSS Collaboration and Deployment Services Repository, consulte [Acceso a objetos en el IBM SPSS Collaboration and Deployment Services Repository en el capítulo 5 el p. 59](#). Recuerde que en la línea de comandos *debe* utilizar un URI. El REPOSITORY\_PATH más simple no se admite. (Sólo funciona en procesos.)

## Argumentos de parámetros

Los parámetros se pueden utilizar como marcas durante la ejecución de la línea de comandos de IBM® SPSS® Modeler. En los argumentos de la línea de comandos, la marca -P se utiliza para denotar un parámetro del tipo -P <nombre>=<valor>.

Los parámetros pueden ser:

- **Parámetros simples** (o parámetros utilizados directamente en expresiones CLEM).
- **Parámetros de propiedades**, también denominados **propiedades de nodos**. Estos parámetros se utilizan para modificar la configuración de los nodos en la ruta. [Si desea obtener más información, consulte el tema Conceptos básicos de las propiedades de nodos en el capítulo 9 el p. 117.](#)
- **Parámetros de la línea de comandos**, que son parámetros utilizados para alterar la invocación de SPSS Modeler.

Por ejemplo, puede proporcionar contraseñas y nombres de usuario de orígenes de datos como una marca de la línea de comandos, por ejemplo:

```
modelerclient -stream response.str -P:databasenode.datasource={"ORA 10gR2", user1, mypsw, true}
```

El formato es el mismo que el del parámetro datasource de la propiedad de nodo databasenode. [Si desea obtener más información, consulte el tema Propiedades de databasenode en el capítulo 12 el p. 127.](#)

# ***CLEM Referencia del lenguaje***

## ***Conceptos básicos de la referencia de CLEM***

Esta sección describe el lenguaje para la manipulación de expresiones de control de (CLEM), una potente herramienta que se utiliza para analizar y manipular los datos utilizados en las rutas de IBM® SPSS® Modeler. Se puede utilizar CLEM en los nodos para realizar tareas que abarcan desde la evaluación de condiciones o la derivación de valores a la inserción de datos en informes. [Si desea obtener más información, consulte el tema Acerca de CLEM en el capítulo 7 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

Los subconjuntos del lenguaje CLEM se pueden utilizar también cuando se usan procesos tanto en la interfaz de usuario. Esto permite realizar muchas de las manipulaciones de datos que son posibles en un procesamiento automático. [Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.](#)

Las expresiones CLEM se componen de valores, nombres de campos, operadores y funciones. Utilizando la sintaxis correcta, puede crear una amplia variedad de eficaces operaciones de datos. [Si desea obtener más información, consulte el tema Ejemplos de CLEM en el capítulo 7 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

## ***Tipos de datos de CLEM***

Los tipos de datos de CLEM pueden estar compuestos por lo siguiente:

- Enteros
- Reales
- Caracteres
- Cadenas
- Listas
- Fields
- Fecha/hora

### ***Reglas de entrecomillado***

Aunque IBM® SPSS® Modeler es flexible a la hora de determinar los campos, los valores, los parámetros y las cadenas que se utilizan en una expresión CLEM, las siguientes reglas generales ofrecen una lista de “procedimientos recomendables” que se deben seguir cuando se creen expresiones:

- Cadenas: utilice siempre comillas dobles cuando escriba cadenas, como por ejemplo "Type 2". También se pueden utilizar las comillas simples, pero con el riesgo de crear confusión con los campos entrecomillados.

- Campos: utilice comillas simples, sólo cuando sea necesario, para poner espacios u otros caracteres especiales entre comillas, como por ejemplo 'Order Number'. los campos entrecorillados pero no definidos en el conjunto de datos se confundirán con cadenas.
- Parámetros: utilice siempre comillas simples cuando utilice parámetros, como por ejemplo '\$P-threshold'.
- Caracteres: utilice siempre comillas inversas simples (^), como por ejemplo stripchar(`d`, "drugA").

Si desea obtener más información, consulte el tema [Valores y tipos de datos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15](#). Estas reglas también se tratan con mayor detalle en los siguientes temas.

## Enteros

Los enteros se representan como una secuencia de dígitos decimales. Si lo desea, puede colocar un signo menos (–) delante del entero para indicar un número negativo; por ejemplo, 1234, 999, –77.

El lenguaje CLEM gestiona enteros de precisión arbitraria. El tamaño máximo de los enteros depende de cada plataforma. Si los valores son demasiado grandes para mostrarlos en un campo de entero, la acción de cambiar el tipo de campo a Real suele restaurar el valor.

## Reales

*Real* hace referencia a un número de coma flotante. Los reales se representan con uno o más dígitos, seguidos por una coma decimal y, después, por uno o más dígitos. Los reales CLEM se guardan en doble precisión.

Si lo desea, puede colocar un signo menos (–) delante del número real para denotar un número negativo; por ejemplo, 1.234, 0.999, –77.001. Utilice el formato <número> e <exponente> para expresar un número real en notación exponencial; por ejemplo, 1234.0e5, 1.7e–2. Cuando la aplicación IBM® SPSS® Modeler lee cadenas numéricas de archivos y las convierte automáticamente en números, se aceptan números sin dígitos antes o después de la coma decimal; por ejemplo, 999. o .11. Sin embargo, estos formatos no son válidos en las expresiones CLEM.

*Nota:* Al hacer referencia a números reales en expresiones CLEM, se debe utilizar el punto como separador decimal, independientemente de lo que se haya definido en la configuración regional o ruta actual. Por ejemplo, especifique:

Na > 0,6

en vez de

Na > 0.6

Todas las opciones anteriores se aplican aunque se haya seleccionado una coma como símbolo decimal en el cuadro de diálogo de propiedades de la ruta y es coherente con la norma general de que la sintaxis de código debe ser independiente de cualquier convención o configuración regional específica.

## Caracteres

Los caracteres (mostrados normalmente como CHAR) se suelen utilizar dentro de una expresión CLEM para realizar comprobaciones en cadenas. Por ejemplo, puede utilizar la función `isuppercode` para determinar si el primer carácter de una cadena va en mayúsculas. La siguiente expresión CLEM utiliza un carácter para indicar que la comprobación se debe realizar en el primer carácter de la cadena:

```
isuppercode(subscrs(1,"MiCadena"))
```

Para expresar el código (en lugar de la ubicación) de un carácter determinado en una expresión CLEM, utilice comillas inversas simples del tipo ``<carácter>``; por ejemplo, ``A``, ``Z``.

*Nota:* no hay ningún tipo de almacenamiento CHAR para un campo, por lo que si un campo se deriva o rellena con una expresión que da como resultado un CHAR, este resultado se convertirá en una cadena.

## Cadenas

Por lo general, debe encerrar las cadenas entre comillas dobles. `"c35product2"` y `"referrerID"` son ejemplos de cadenas. Para indicar caracteres especiales en una cadena, utilice una barra invertida; por ejemplo, `"\$65443"`. (Para indicar un carácter de barra invertida, utilice una barra invertida doble, `\\`.) Puede utilizar comillas simples para las cadenas, pero el resultado no se distinguirá de un campo entrecomillado (`'referrerID'`). [Si desea obtener más información, consulte el tema Funciones de cadena en Manual de usuario de IBM SPSS Modeler 15.](#)

## Listas

Una lista es una secuencia ordenada de elementos, que pueden ser de distintos tipos. Las listas se encierran entre corchetes (`[]`). `[1 2 4 16]` y `["abc" "def"]` son ejemplos de listas. Las listas no se utilizan como valores de los campos de IBM® SPSS® Modeler. Se utilizan para proporcionar argumentos a funciones, como por ejemplo `member` y `oneof`.

## Fields

Los nombres de las expresiones CLEM que no son nombres de funciones se consideran nombres de campos. Puede escribirlos simplemente como `Power`, `val27`, `state_flag`, etc., pero si el nombre comienza por un dígito o incluye caracteres no alfabéticos, como espacios (a excepción del guión bajo), debe poner el nombre entre comillas simples; por ejemplo, `'Power Increase'`, `'2nd answer'`, `'#101'`, `'$P-NextField'`.

*Nota:* los campos entrecomillados pero no definidos en el conjunto de datos se confundirán con cadenas.

## Fechas

Los cálculos de fecha se basan en una fecha de “línea base”, la cual se especifica en el cuadro de diálogo de propiedades de la ruta. La fecha de línea base por defecto es el 1 de enero de 1900. [Si desea obtener más información, consulte el tema Configuración de opciones generales de las rutas en el capítulo 5 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

El lenguaje CLEM admite los siguientes formatos de fecha.

Formato	Ejemplos
DDMMYY	150163
MMDDYY	011563
YYMMDD	630115
YYYYMMDD	19630115
YYYYDDD	Año de cuatro dígitos, seguido de un número de tres dígitos que representa el día del año; por ejemplo, 2000032 representa el día 32 de 2000 o el 1 de febrero de 2000.
DAY	Día de la semana en la configuración regional actual; por ejemplo, Monday, Tuesday, etc., en español.
MONTH	Mes en la configuración regional actual; por ejemplo, January, February, ....
DD/MM/YY	15/01/63
DD/MM/YYYY	15/01/1963
MM/DD/YY	01/15/63
MM/DD/YYYY	01/15/1963
DD-MM-YY	15-01-63
DD-MM-YYYY	15-01-1963
MM-DD-YY	01-15-63
MM-DD-YYYY	01-15-1963
DD.MM.YY	15.01.63
DD.MM.YYYY	15.01.1963
MM.DD.YY	01.15.63
MM.DD.YYYY	01.15.1963
DD-MON-YY	15-JAN-63, 15-jan-63, 15-Jan-63
DD/MON/YY	15/JAN/63, 15/jan/63, 15/Jan/63
DD.MON.YY	15.JAN.63, 15.jan.63, 15.Jan.63
DD-MON-YYYY	15-JAN-1963, 15-jan-1963, 15-Jan-1963
DD/MON/YYYY	15/JAN/1963, 15/jan/1963, 15/Jan/1963
DD.MON.YYYY	15.JAN.1963, 15.jan.1963, 15.Jan.1963
MON YYYY	Jan 2004

Formato	Ejemplos
q Q YYYY	Fecha en forma de dígito (1–4), que representa el trimestre seguido de la letra <i>T</i> y un año de cuatro dígitos; por ejemplo, la fecha 25 Dic 2004 se representaría como 4 Q 2004.
ww WK YYYY	Número de dos dígitos que representa la semana del año, seguido de las letras <i>SE</i> y, a continuación, un año de cuatro dígitos. La semana del año se calcula asumiendo que el primer día de la semana es lunes y que hay al menos un día en la primera semana.

## Hora

El lenguaje CLEM admite los siguientes formatos de hora.

Formato	Ejemplos
HHMMSS	120112, 010101, 221212
HHMM	1223, 0745, 2207
MMSS	5558, 0100
HH:MM:SS	12:01:12, 01:01:01, 22:12:12
HH:MM	12:23, 07:45, 22:07
MM:SS	55:58, 01:00
(H)H:(M)M:(S)S	12:1:12, 1:1:1, 22:12:12
(H)H:(M)M	12:23, 7:45, 22:7
(M)M:(S)S	55:58, 1:0
HH.MM.SS	12.01.12, 01.01.01, 22.12.12
HH.MM	12.23, 07.45, 22.07
MM.SS	55.58, 01.00
(H)H.(M)M.(S)S	12.1.12, 1.1.1, 22.12.12
(H)H.(M)M	12.23, 7.45, 22.7
(M)M.(S)S	55.58, 1.0

## Operadores de CLEM

Los siguientes operadores están disponibles:

Operación	Comentarios	Prioridad (consulte la sección siguiente)
or	Se utiliza entre dos expresiones CLEM. Devuelve un valor de verdadero si una de ellas es verdadera o si ambas lo son.	10
and	Se utiliza entre dos expresiones CLEM. Devuelve un valor de verdadero si ambas son verdaderas.	9

Operación	Comentarios	Prioridad (consulte la sección siguiente)
=	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 NO es igual al ELEMENTO2.	7
==	Idéntico a =	7
/=	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 <i>no</i> es igual al ELEMENTO2.	7
/==	Idéntico a /=	7
>	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 es estrictamente mayor que el ELEMENTO2.	6
>=	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 es mayor o igual que el ELEMENTO2.	6
<	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 es estrictamente menor que el ELEMENTO2.	6
<=	Se utiliza entre dos elementos comparables. Devuelve verdadero si el ELEMENTO1 es menor o igual que el ELEMENTO2.	6
&&=_0	Se utiliza entre dos enteros. Equivalente a la expresión booleana ENT1 && ENT2 = 0.	6
&&/=_0	Se utiliza entre dos enteros. Equivalente a la expresión booleana ENT1 && ENT2 /= 0.	6
+	Suma dos números: NÚM1 + NÚM2.	5
>>	Concatena dos cadenas, por ejemplo, STRING1 >> STRING2.	5
-	Resta un número a otro: NÚM1 - NÚM2. También se puede utilizar delante de un número: - NÚM.	5
*	Se utiliza para multiplicar dos números: NÚM1 * NÚM2.	4
&&	Se utiliza entre dos enteros. El resultado es el 'y' a nivel de bit de los enteros ENT1 y ENT2.	4
&&~~	Se utiliza entre dos enteros. El resultado es el 'y' a nivel de bit de ENT1 y el complemento a nivel de bit de ENT2.	4
	Se utiliza entre dos enteros. El resultado es el 'o inclusivo' a nivel de bit de ENT1 y ENT2.	4
~~	Se utiliza delante de un entero. Genera el complemento a nivel de bit de ENT.	4

Operación	Comentarios	Prioridad (consulte la sección siguiente)
&	Se utiliza entre dos enteros. El resultado es el 'o exclusivo' a nivel de bit de ENT1 y ENT2.	4
INT1 << N	Se utiliza entre dos enteros. Genera el patrón de bits de ENT desplazado a la izquierda N posiciones.	4
INT1 >> N	Se utiliza entre dos enteros. Genera el patrón de bits de ENT desplazado a la derecha N posiciones.	4
/	Se utiliza para dividir un número por otro: NÚM1 / NÚM2.	4
**	Se utiliza entre dos números: BASE ** POTENCIA. Devuelve BASE elevado a la potencia POTENCIA.	3
rem	Se utiliza entre dos enteros: ENT1 rem ENT2. Devuelve el resto, ENT1 - (ENT1 div ENT2) * ENT2.	2
div	Se utiliza entre dos enteros: ENT1 div ENT2. Realiza la división de enteros.	2

### **Prioridad del operador**

Las prioridades determinan el análisis de expresiones complejas, especialmente aquellas que no van entre paréntesis y tienen más de un operador infijo. Por ejemplo:

$3 + 4 * 5$

se analiza como  $3 + (4 * 5)$  en lugar de  $(3 + 4) * 5$  porque las prioridades relativas dictan que \* debe analizarse antes que +. Cada operador del lenguaje CLEM tiene un valor de prioridad asociado y, mientras más bajo sea éste, más importante será en la lista de análisis, lo que significa que será procesado antes que otros operadores con valores de prioridad superiores.

## **Referencia de funciones**

Las siguientes funciones de CLEM están disponibles para trabajar con datos en IBM® SPSS® Modeler. Puede introducir estas funciones como código en diversos cuadros de diálogo, como los de los nodos Derivar y Marcas, o puede utilizar el generador de expresiones para crear expresiones CLEM válidas sin memorizar listas de funciones ni nombres de campos.

Tipo de función	Descripción
Información	Se utiliza para comprender mejor los valores de campos. Por ejemplo, la función is_string devuelve verdadero para todos los registros cuyo tipo es una cadena.
Conversión	Se utiliza para construir nuevos campos o convertir el tipo de almacenamiento. Por ejemplo, la función to_timestamp convierte el campo seleccionado en una marca de tiempo.
Comparación	Se utiliza para comparar valores de campos entre ellos o con una cadena especificada. Por ejemplo, <= se utiliza para comparar y ver si los valores de dos campos son menores o iguales.



Tipo de función	Descripción
Lógicas	Se utilizan para realizar operaciones lógicas como, por ejemplo, operaciones if, then, else.
Numérico	Se utilizan para realizar cálculos numéricos como, por ejemplo, el logaritmo natural de valores de campos.
Trigonómicas	Se utilizan para realizar cálculos trigonométricos como, por ejemplo, el arcocoseno de un ángulo especificado.
Probability	Probabilidades basadas en distintas distribuciones, como probabilidad de que un valor de una distribución <i>t</i> de Student sea menor que un valor específico.
A nivel de bit	Se utiliza para manipular enteros como patrones de bits.
Random	Se utilizan para seleccionar elementos al azar o generar números.
Cadena	Se utiliza para realizar numerosas operaciones en cadenas como, por ejemplo, stripchar, que permite eliminar un carácter especificado.
SoundEx	Se utiliza para encontrar cadenas cuando no se conoce la ortografía exacta, se basa en supuestos fonéticos sobre cómo se pronuncian ciertas letras.
La fecha y la hora	Se utiliza para realizar diversas operaciones en los campos de fecha, hora y marca de tiempo.
Secuencia	Se utiliza para comprender mejor la secuencia de registros de un conjunto de datos o para realizar operaciones en función de dicha secuencia.
Globales	Se utilizan para acceder a los valores globales creados por un nodo Val. globales. Por ejemplo, @MEAN se utiliza para hacer referencia a la media de todos los valores de un campo en todo el conjunto de datos.
Vacios y nulos	Se utiliza para acceder, marcar y, con frecuencia, rellenar elementos vacíos especificados por el usuario y valores perdidos por el sistema. Por ejemplo, @BLANK(FIELD) se utiliza para obtener una marca verdadera para los registros con elementos vacíos.
Campos especiales	Se utilizan para denotar los campos específicos que están siendo examinados. Por ejemplo, @FIELD se utiliza cuando se derivan varios campos.

### **Convenciones en las descripciones de las funciones**

En este manual se utilizan las siguientes convenciones para referirse a los elementos de una función.

Convención	Descripción
<i>BOOL</i>	Un valor booleano, o una marca, como por ejemplo verdadero o falso.
<i>NÚM, NÚM1, NÚM2</i>	Cualquier número.
<i>REAL, REAL1, REAL2</i>	Cualquier número real, como por ejemplo 1.234 o -77.01.
<i>ENT, ENT1, ENT2</i>	Cualquier número entero, como por ejemplo 1 o -77.
<i>CAR</i>	Un código de carácter, como por ejemplo `A`.
<i>STRING</i>	Una cadena, como por ejemplo "referrerID".
<i>LIST</i>	Una lista de elementos, como por ejemplo ["abc" "def"].
<i>ELEMENTO</i>	Un campo, como por ejemplo Customer o extract_concept.

Convención	Descripción
<i>DATE</i>	Un campo de fecha, como por ejemplo <i>start_date</i> , donde los valores tienen formatos tales como DD-MON-YYYY.
<i>TIME</i>	Un campo de hora, como por ejemplo <i>power_flux</i> , donde los valores tienen formatos como HHMMSS.

Las funciones de este manual se enumeran con la función en una columna, el tipo de resultado (entero, cadena, etc.) en otra y una descripción (si está disponible) en una tercera columna. Por ejemplo, a continuación se muestra la descripción de la función *rem*.

Función	Resultado	Descripción
INT1 rem INT2	<i>Number</i>	Devuelve el resto de dividir <i>ENT1</i> entre <i>ENT2</i> . Por ejemplo, INT1 – (INT1 div INT2) * INT2.

En otra sección se ofrecen detalles sobre el uso de las convenciones, como por ejemplo cómo crear una lista de elementos o especificar caracteres en una función. [Si desea obtener más información, consulte el tema Tipos de datos de CLEM en Manual de usuario de IBM SPSS Modeler 15.](#)

## Funciones de información

Las funciones de información se utilizan para comprender mejor los valores de un campo en particular. Se suelen utilizar para derivar campos de marcas. Por ejemplo, puede utilizar la función *@BLANK* para crear un campo de marca que indique los registros cuyos valores están vacíos para el campo seleccionado. De igual modo, puede comprobar el tipo de almacenamiento de un campo utilizando cualquiera de las funciones de tipo de almacenamiento, como por ejemplo *is\_string*.

Función	Resultado	Descripción
<i>@BLANK</i> (FIELD)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyos valores están vacíos de acuerdo con el conjunto de reglas de tratamiento de los elementos vacíos en un nodo Tipo o de origen (pestaña Tipos) anterior de la ruta. Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.</a>
<i>@NULL</i> (ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyos valores no están definidos. Los valores no definidos son valores nulos del sistema y se muestran en IBM® SPSS® Modeler como \$null\$. Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.</a>
<i>is_date</i> (ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es una fecha.
<i>is_datetime</i> (ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es una fecha, hora o marca de tiempo.
<i>is_integer</i> (ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es un entero.
<i>is_number</i> (ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es un número.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
is_real(ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es un número real.
is_string(ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es una cadena.
is_time(ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es una hora.
is_timestamp(ITEM)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyo tipo es una marca de tiempo.

## ***Funciones de conversión***

Las funciones de conversión permiten construir nuevos campos y convertir el tipo de almacenamiento de los archivos existentes. Por ejemplo, puede formar cadenas nuevas uniendo o desmontando otras. Para unir dos cadenas, utilice el operador ><. Por ejemplo, si el campo Site tiene el valor "BRAMLEY", "xx" >< Site devuelve "xxBRAMLEY". El resultado de ><< siempre es una cadena, aun cuando los argumentos no sean cadenas. Por lo tanto, si el campo V1 es 3 y el campo V2 es 5, entonces V1 >< V2 devuelve "35" (una cadena, no un número).

Las funciones de conversión (y otras muchas funciones que requieren un tipo específico de entrada, como un valor de fecha u hora) dependen de los formatos actuales especificados en el cuadro de diálogo de opciones de ruta. Por ejemplo, si desea convertir un campo de cadena con los valores *Ene 2003*, *Feb 2003*, etc., seleccione el formato de fecha coincidente MES AAAA como formato de fecha por defecto de la ruta. [Si desea obtener más información, consulte el tema Configuración de opciones generales de las rutas en el capítulo 5 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
ITEM1 >< ITEM2	<i>Cadena</i>	Concatena los valores de dos campos y devuelve la cadena resultante como <i>ELEMENTO1ELEMENTO2</i> .
to_integer(ITEM)	<i>Entero</i>	Convierte el almacenamiento del campo especificado en un entero.
to_real(ITEM)	<i>Real</i>	Convierte el almacenamiento del campo especificado en un número real.
to_number(ITEM)	<i>Number</i>	Convierte el almacenamiento del campo especificado en un número.
to_string(ITEM)	<i>Cadena</i>	Convierte el almacenamiento del campo especificado en una cadena.
to_time(ITEM)	<i>Hora</i>	Convierte el almacenamiento del campo especificado en una hora.
to_date(ITEM)	<i>Fecha</i>	Convierte el almacenamiento del campo especificado en una fecha.
to_timestamp(ITEM)	<i>Marca de tiempo</i>	Convierte el almacenamiento del campo especificado en una marca de tiempo.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
to_datetime(ITEM)	<i>Mo-mento_fecha</i>	Convierte el almacenamiento del campo especificado en una fecha, hora o marca de tiempo.
datetime_date(ITEM)	<i>Fecha</i>	Devuelve el valor de fecha de un <i>número</i> , <i>cadena</i> o <i>marca de tiempo</i> . Tenga en cuenta que es la única función que le permite volver a convertir un número (en segundos) en una fecha. Si ITEM es una cadena, crea una fecha analizando una cadena en el formato de fecha actual. El formato de fecha especificado en el cuadro de diálogo de propiedades de la ruta debe ser correcto para que esta función se ejecute correctamente. Si ITEM es un número, se interpreta como un número de segundos desde la fecha de base (o época). Las fracciones de un día se truncan. Si ITEM es una marca de tiempo, se devuelve la parte de la fecha de la marca de tiempo. Si ITEM es una fecha, se devuelve sin modificar.

### **Funciones de comparación**

Las funciones de comparación se utilizan para comparar valores de campos entre ellos o con una cadena especificada. Por ejemplo, puede comprobar la igualdad de cadenas utilizando =. Un ejemplo de verificación de igualdad de cadenas es: `Class = "class 1"`.

En la comparación numérica, *mayor* significa más cerca del infinito positivo y *menor*, más cerca del infinito negativo. Es decir, que todos los números negativos son menores que cualquier número positivo.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
count_equal(ITEM1, LIST)	<i>Entero</i>	Devuelve el número de valores de una lista de campos que son iguales a <i>ELEMENTO1</i> o nulos si <i>ELEMENTO1</i> es nulo. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
count_greater_than(ITEM1, LIST)	<i>Entero</i>	Devuelve el número de valores de una lista de campos que son mayores que <i>ELEMENTO1</i> o nulos si <i>ELEMENTO1</i> es nulo.
count_less_than(ITEM1, LIST)	<i>Entero</i>	Devuelve el número de valores de una lista de campos que son menores que <i>ELEMENTO1</i> o nulos si <i>ELEMENTO1</i> es nulo.
count_not_equal(ITEM1, LIST)	<i>Entero</i>	Devuelve el número de valores de una lista de campos que no son iguales a <i>ELEMENTO1</i> o son nulos si <i>ELEMENTO1</i> es nulo.
count_nulls(LIST)	<i>Entero</i>	Devuelve el número de valores nulos de una lista de campos.
count_non_nulls(LIST)	<i>Entero</i>	Devuelve el número de valores no nulos de una lista de campos.
date_before(ITEM1, ITEM2)	<i>Booleana</i>	Se utiliza para comprobar el orden de los valores de fecha. Devuelve un valor verdadero si <i>FECHA1</i> es anterior a <i>FECHA2</i> .

Función	Resultado	Descripción
first_index(ITEM, LIST)	Entero	Devuelve el índice del primer campo que contiene ELEMENTO de una LISTA de campos o 0 si no se encuentra el valor. Compatible con los tipos cadena, entero y real únicamente. <a href="#">Si desea obtener más información, consulte el tema Trabajo con datos de respuestas múltiples en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
first_non_null(LIST)	Cualquiera	Devuelve el primer valor que no sea nulo en la lista de campos. Se admiten todos los tipos de almacenamiento.
first_non_null_index(LIST)	Entero	Devuelve el índice del primer campo en la LISTA especificada con un valor no nulo o 0 si todos los valores son nulos. Se admiten todos los tipos de almacenamiento.
ITEM1 = ITEM2	Booleana	Devuelve verdadero para los registros donde ELEMENTO1 es igual a ELEMENTO2.
ITEM1 /= ITEM2	Booleana	Devuelve verdadero si las dos cadenas no son idénticas o 0 si son idénticas.
ITEM1 < ITEM2	Booleana	Devuelve verdadero para los registros donde ELEMENTO1 es menor que ELEMENTO2.
ITEM1 <= ITEM2	Booleana	Devuelve verdadero para los registros donde ELEMENTO1 es menor o igual que ELEMENTO2.
ITEM1 > ITEM2	Booleana	Devuelve verdadero para los registros donde ELEMENTO1 es mayor que ELEMENTO2.
ITEM1 >= ITEM2	Booleana	Devuelve verdadero para los registros donde ELEMENTO1 es mayor o igual que ELEMENTO2.
last_index(ITEM, LIST)	Entero	Devuelve el índice del último campo que contiene ELEMENTO de una LISTA de campos o 0 si no se encuentra el valor. Compatible con los tipos cadena, entero y real únicamente. <a href="#">Si desea obtener más información, consulte el tema Trabajo con datos de respuestas múltiples en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
last_non_null(LIST)	Cualquiera	Devuelve el último valor que no sea nulo en la lista de campos. Se admiten todos los tipos de almacenamiento.
last_non_null_index(LIST)	Entero	Devuelve el índice del último campo en la LISTA especificada con un valor no nulo o 0 si todos los valores son nulos. Se admiten todos los tipos de almacenamiento.
max(ITEM1, ITEM2)	Cualquiera	Devuelve el mayor de los dos elementos: ELEMENTO1 o ELEMENTO2.
max_index(LIST)	Entero	Devuelve el índice del campo con el valor máximo de una lista de campos numéricos o 0 si todos los valores son nulos. Por ejemplo, si el tercer campo de la lista contiene el máximo, se devuelve el valor de índice 3. Si varios campos contienen el valor máximo, se devuelve el valor que aparece primero (el primero por la izquierda). <a href="#">Si desea obtener más información, consulte el tema Trabajo con datos de respuestas múltiples en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
max_n(LIST)	Number	Devuelve el valor máximo de una lista de campos numéricos o es nulo si todos los valores de campo son nulos. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>

Función	Resultado	Descripción
member(ITEM, LIST)	Booleana	Devuelve verdadero si <i>ELEMENTO</i> es un miembro de la <i>LISTA</i> especificada. En el resto de los casos, devuelve falso. También se puede especificar una lista de nombres de campo. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
min(ITEM1, ITEM2)	Cualquiera	Devuelve el menor de los dos elementos: <i>ELEMENTO1</i> o <i>ELEMENTO2</i> .
min_index(LIST)	Entero	Devuelve el índice del campo con el valor mínimo de una lista de campos numéricos o 0 si todos los valores son nulos. Por ejemplo, si el tercer campo de la lista contiene el mínimo, se devuelve el valor de índice 3. Si varios campos contienen el valor mínimo, se devuelve el valor que aparece primero (el primero por la izquierda). <a href="#">Si desea obtener más información, consulte el tema Trabajo con datos de respuestas múltiples en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
min_n(LIST)	Number	Devuelve el valor mínimo de una lista de campos numéricos o es nulo si todos los valores de campo son nulos.
time_before(TIME1, TIME2)	Booleana	Se utiliza para comprobar el orden de los valores de hora. Devuelve un valor verdadero si <i>HORA1</i> es anterior a <i>HORA2</i> .
value_at(INT, LIST)		Devuelve el valor de cada campo de la lista con un desplazamiento INT o NULL si el desplazamiento está fuera del rango de valores válidos (es decir, menor que 1 o mayor que el número de campos de la lista). Se admiten todos los tipos de almacenamiento.

## Funciones lógicas

Las expresiones CLEM se pueden utilizar para realizar operaciones lógicas.

Función	Resultado	Descripción
COND1 and COND2	Booleana	Esta operación es una conjunción lógica y devuelve un valor verdadero si <i>COND1</i> y <i>COND2</i> son verdaderos. Si <i>COND1</i> es falsa, <i>COND2</i> no se evalúa; esto permite tener conjunciones donde <i>COND1</i> comprueba primero si una operación en <i>COND2</i> es legal. Por ejemplo, <code>length(Label) &gt;=6</code> y <code>Label(6) = 'x'</code> .
COND1 or COND2	Booleana	Esta operación es una disyunción (inclusiva) lógica y devuelve verdadero si <i>COND1</i> o <i>COND2</i> son verdaderas, o bien si ambas lo son. Si <i>COND1</i> es verdadera, <i>COND2</i> no se evalúa.
not(COND)	Booleana	Esta operación es una negación lógica y devuelve un valor verdadero si <i>COND</i> es falsa. En el resto de los casos, esta operación devuelve un valor de 0.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
if COND then EXPR1 else EXPR2 endif	<i>Cualquiera</i>	Esta operación es una evaluación condicional. Si <i>COND</i> es verdadera, esta operación devuelve el resultado de <i>EXPR1</i> . En el resto de los casos, devuelve el resultado de evaluar <i>EXPR2</i> .
if COND1 then EXPR1 elseif COND2 then EXPR2 else EXPR_N endif	<i>Cualquiera</i>	Esta operación es una evaluación condicional de varias ramas. Si <i>COND1</i> es verdadera, esta operación devuelve el resultado de <i>EXPR1</i> . En caso contrario, si <i>COND2</i> es verdadera, esta operación devuelve el resultado de evaluar <i>EXPR2</i> . En caso contrario, devuelve el resultado de evaluar <i>EXPR_N</i> .

## Funciones numéricas

CLEM contiene varias funciones numéricas de uso común.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
-NUM	<i>Number</i>	Se utiliza para negar <i>NÚM</i> . Devuelve el número correspondiente con el signo opuesto.
NUM1 + NUM2	<i>Number</i>	Devuelve la suma de <i>NÚM1</i> y <i>NÚM2</i> .
código -NUM2	<i>Number</i>	Devuelve el valor de <i>NÚM1</i> menos <i>NÚM2</i> .
NUM1 * NUM2	<i>Number</i>	Devuelve el valor de <i>NÚM1</i> multiplicado por <i>NÚM2</i> .
NUM1 / NUM2	<i>Number</i>	Devuelve el valor de <i>NÚM1</i> dividido por <i>NÚM2</i> .
INT1 div INT2	<i>Number</i>	Se utiliza para realizar una división entera. Devuelve el valor de <i>ENT1</i> dividido por <i>ENT2</i> .
INT1 rem INT2	<i>Number</i>	Devuelve el resto de dividir <i>ENT1</i> entre <i>ENT2</i> . Por ejemplo, $INT1 - (INT1 \text{ div } INT2) * INT2$ .
INT1 mod INT2	<i>Number</i>	Esta función está desaprobada. Utilice la función <i>rem</i> en su lugar.
BASE ** POWER	<i>Number</i>	Devuelve <i>BASE</i> elevada a la potencia <i>POTENCIA</i> , donde ambas pueden ser cualquier número (con la excepción de que <i>BASE</i> no debe ser cero si <i>POTENCIA</i> es cero de cualquier tipo, menos el entero 0). Si <i>POTENCIA</i> es un entero, el cálculo se realiza multiplicando sucesivamente potencias de <i>BASE</i> . Así, si <i>BASE</i> es un entero, el resultado será un entero. Si <i>POTENCIA</i> es el entero 0, el resultado siempre será un 1 del mismo tipo que <i>BASE</i> . De lo contrario, si <i>POTENCIA</i> no es un entero, el resultado se calculará como $\exp(\text{POWER} * \log(\text{BASE}))$ .
abs(NUM)	<i>Number</i>	Devuelve el valor absoluto de <i>NÚM</i> , que siempre será un número del mismo tipo.
exp(NUM)	<i>Real</i>	Devuelve <i>e</i> elevado a la potencia de <i>NÚM</i> , donde <i>e</i> es la base de los logaritmos naturales.
fracof(NUM)	<i>Real</i>	Devuelve la parte fraccional de <i>NÚM</i> , definida como $\text{NUM} - \text{intof}(\text{NUM})$ .
intof(NUM)	<i>Entero</i>	Trunca su argumento a un entero. Devuelve el entero del mismo signo de <i>NÚM</i> y con la mayor magnitud que $\text{abs}(\text{INT}) \leq \text{abs}(\text{NUM})$ .
log(NUM)	<i>Real</i>	Devuelve el logaritmo natural (en base <i>e</i> ) de <i>NÚM</i> , que no puede ser un cero de ningún tipo.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
log10(NUM)	<i>Real</i>	Devuelve el logaritmo en base 10 de <i>NÚM</i> , que no puede ser un cero de ningún tipo. Esta función se define como $\log(\text{NUM}) / \log(10)$ .
negate(NUM)	<i>Number</i>	Se utiliza para negar <i>NÚM</i> . Devuelve el número correspondiente con el signo opuesto.
round(NUM)	<i>Entero</i>	Se utiliza para redondear <i>NÚM</i> a un número entero tomando $\text{intof}(\text{NUM}+0.5)$ si <i>NÚM</i> es positivo o $\text{intof}(\text{NUM}-0.5)$ si <i>NÚM</i> es negativo.
sign(NUM)	<i>Number</i>	Se utiliza para determinar el signo de <i>NÚM</i> . Esta operación devuelve -1, 0 ó 1 si <i>NÚM</i> es un entero. Si <i>NÚM</i> es un real, devuelve -1,0, 0,0 ó 1,0 dependiendo de si <i>NÚM</i> es negativo, cero o positivo.
sqrt(NUM)	<i>Real</i>	Devuelve la raíz cuadrada de <i>NÚM</i> . <i>NÚM</i> debe ser positivo.
sum_n(LIST)	<i>Number</i>	Devuelve la suma de valores de una lista de campos numéricos o es nulo si todos los valores de campo son nulos. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
mean_n(LIST)	<i>Number</i>	Devuelve el valor medio de una lista de campos numéricos o es nulo si todos los valores de campo son nulos.
sdev_n(LIST)	<i>Number</i>	Devuelve la desviación estándar de una lista de campos numéricos o es nulo si todos los valores de campo son nulos.

### **Funciones trigonométricas**

Todas las funciones de esta sección toman un ángulo como argumento o devuelven uno como resultado. En ambos casos, las unidades del ángulo (radianes o grados) se controlan con la configuración de la opción de ruta correspondiente.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
arccos(NUM)	<i>Real</i>	Calcula el arcocoseno del ángulo especificado.
arccosh(NUM)	<i>Real</i>	Calcula el arcocoseno hiperbólico del ángulo especificado.
arcsin(NUM)	<i>Real</i>	Calcula el arcoseno del ángulo especificado.
arcsinh(NUM)	<i>Real</i>	Calcula el arcoseno hiperbólico del ángulo especificado.
arctan(NUM)	<i>Real</i>	Calcula la arcotangente del ángulo especificado.
arctan2(NUM_Y, NUM_X)	<i>Real</i>	Calcula la arcotangente de $\text{NUM\_Y} / \text{NUM\_X}$ y utiliza los signos de los dos números para derivar la información sobre el cuadrante. El resultado es un número real en el rango $-\pi < \text{ANGLE} \leq \pi$ (radians) – $-180 < \text{ANGLE} \leq 180$ (degrees)
arctanh(NUM)	<i>Real</i>	Calcula la arcotangente hiperbólica del ángulo especificado.
cos(NUM)	<i>Real</i>	Calcula el coseno del ángulo especificado.
cosh(NUM)	<i>Real</i>	Calcula el coseno hiperbólico del ángulo especificado.
pi	<i>Real</i>	Esta constante es la mejor aproximación real de pi.
sin(NUM)	<i>Real</i>	Calcula el seno del ángulo especificado.
sinh(NUM)	<i>Real</i>	Calcula el seno hiperbólico del ángulo especificado.
tan(NUM)	<i>Real</i>	Calcula la tangente del ángulo especificado.
tanh(NUM)	<i>Real</i>	Calcula la tangente hiperbólica del ángulo especificado.



## Funciones de probabilidad

Las funciones de probabilidad devuelven probabilidades basadas en distintas distribuciones, como probabilidad de que un valor de una distribución  $t$  de Student sea menor que un valor específico.

Función	Resultado	Descripción
<code>cdf_chisq(NUM, DF)</code>	<i>Real</i>	Devuelve la probabilidad de que un valor de la distribución chi-cuadrado con los grados especificados de libertad sea menor que el número especificado.
<code>cdf_f(NUM, DF1, DF2)</code>	<i>Real</i>	Devuelve la probabilidad de que un valor de la distribución $F$ con los grados de libertad $DF1$ y $DF2$ sea menor que el número especificado.
<code>cdf_normal(NUM, MEAN, STDDEV)</code>	<i>Real</i>	Devuelve la probabilidad de que un valor de la distribución normal con la desviación media y estándar especificada sea menor que el número especificado.
<code>cdf_t(NUM, DF)</code>	<i>Real</i>	Devuelve la probabilidad de que un valor de la distribución $t$ de Student con los grados especificados de libertad sea menor que el número especificado.

## Operaciones de enteros a nivel de bit

Estas funciones permiten manipular los enteros como patrones de bits que representan valores en complemento a dos, donde la posición del bit  $N$  tiene la ponderación  $2^{*N}$ . Los bits se numeran de 0 en adelante. Estas operaciones actúan como si el bit de signo de un entero se extendiera indefinidamente hacia la izquierda. Así, por encima de su bit más significativo, un entero positivo tiene 0 bits y un entero negativo tiene 1 bit.

*Nota:* Las funciones a nivel de bit no se pueden llamar desde procesos. [Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.](#)

Función	Resultado	Descripción
<code>~~ INT1</code>	<i>Entero</i>	Genera el complemento a nivel de bit del entero $ENT1$ . Es decir, hay un 1 en el resultado de cada posición de bits para la que $ENT1$ tiene 0. Siempre es verdadero que <code>~~ INT = -(INT + 1)</code> . Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.</a>
<code>INT1    INT2</code>	<i>Entero</i>	El resultado de esta operación es el “o inclusivo” a nivel de bit de $ENT1$ y $ENT2$ . Es decir, hay un 1 en el resultado de cada posición de bits para la que hay un 1 en $ENT1$ o $ENT2$ , o en ambos.
<code>INT1   /&amp; INT2</code>	<i>Entero</i>	El resultado de esta operación es el “o exclusivo” a nivel de bit de $ENT1$ y $ENT2$ . Es decir, hay un 1 en el resultado de cada posición de bits para la que hay un 1 en $ENT1$ o $ENT2$ , pero no en ambos.
<code>INT1 &amp;&amp; INT2</code>	<i>Entero</i>	Genera el “y” a nivel de bit de los enteros $ENT1$ y $ENT2$ . Es decir, hay un 1 en el resultado de cada posición de bits para la que hay un 1 en $ENT1$ y $ENT2$ .

Función	Resultado	Descripción
INT1 &&~~ INT2	Entero	Genera el “y” a nivel de bit de <i>ENT1</i> y el complemento a nivel de bit de <i>ENT2</i> . Es decir, hay un 1 en el resultado de cada posición de bits para la que hay un 1 en <i>ENT1</i> y un 0 en <i>ENT2</i> . Es lo mismo que INT1&&(~~INT2) y resulta útil para borrar bits de <i>ENT1</i> definidos en <i>ENT2</i> .
INT << N	Entero	Genera el patrón de bits de <i>ENT1</i> desplazado a la izquierda <i>N</i> posiciones. Un valor negativo para <i>N</i> genera un desplazamiento a la derecha.
INT >> N	Entero	Genera el patrón de bits de <i>ENT1</i> desplazado a la derecha <i>N</i> posiciones. Un valor negativo para <i>N</i> genera un desplazamiento a la izquierda.
INT1 &&=_0 INT2	Booleana	Equivalente a la expresión booleana INT1 && INT2 /== 0, pero es más eficaz.
INT1 &&/=_0 INT2	Booleana	Equivalente a la expresión booleana INT1 && INT2 == 0, pero es más eficaz.
integer_bitcount(INT)	Entero	Cuenta el número de bits 1 ó 0 de la representación en complemento a dos de <i>ENT</i> . Si <i>ENT</i> no es negativo, <i>N</i> es el número de bits 1. Si <i>ENT</i> es negativo, es el número de bits 0. Debido a la extensión del signo, hay un número infinito de bits 0 en un entero no negativo, o de bits 1 en un entero negativo. Siempre se da el caso de que integer_bitcount(INT) = integer_bitcount(-(INT+1)).
integer_leastbit(INT)	Entero	Devuelve la posición de bits <i>N</i> correspondiente al bit menos significativo definido en el entero <i>ENT</i> . <i>N</i> es la potencia más elevada de 2 por la que <i>ENT</i> tiene una división exacta.
integer_length(INT)	Entero	Devuelve la longitud en bits de <i>ENT</i> como un entero en complemento a dos. Es decir, <i>N</i> es el número entero más pequeño, tal que INT < (1 << N) if INT >= 0 INT >= (-1 << N) if INT < 0. Si <i>ENT</i> no es negativo, la representación de <i>ENT</i> como entero sin signo requiere un campo de al menos <i>N</i> bits. También se requiere un mínimo de <i>N+1</i> bits para representar <i>ENT</i> como un entero con signo, independientemente de su signo.
testbit(INT, N)	Booleana	Comprueba el bit situado en la posición <i>N</i> del entero <i>ENT</i> y devuelve el estado del bit <i>N</i> como un valor booleano, que es verdadero para 1 y falso para 0.

## Funciones aleatorias

Las siguientes funciones se utilizan para seleccionar elementos o generar números al azar.

Función	Resultado	Descripción
oneof(LIST)	Cualquiera	Devuelve un elemento seleccionado al azar de <i>LISTA</i> . Los elementos de la lista se deben introducir como [ITEM1,ITEM2,...,ITEM_N]. Recuerde que también se puede especificar una lista de nombres de campo. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
random(NUM)	Number	Devuelve un número aleatorio del mismo tipo ( <i>ENT</i> o <i>REAL</i> ) distribuido de modo uniforme, desde 1 a <i>NUM</i> . Si utiliza un entero, sólo se devuelven enteros. Si utiliza un número real

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
		(decimal), se devuelven números reales (precisión decimal determinada por las opciones de la ruta). El número aleatorio más grande que devuelve la función puede ser igual a <i>NÚM</i> .
random0(NUM)	<i>Number</i>	Tiene las mismas propiedades que random(NUM), pero comenzando desde 0. El número aleatorio más grande que devuelve la función nunca puede ser igual a <i>X</i> .

## Funciones de cadena

En CLEM, puede realizar las siguientes operaciones con cadenas:

- Comparar cadenas
- Crear cadenas
- Acceder a caracteres

En CLEM, una cadena es cualquier secuencia de caracteres que se encuentre entre un par de comillas dobles ("string quotes"). Los caracteres (CHAR) pueden ser cualquier carácter alfanumérico simple. Se especifican en las expresiones de CLEM mediante comillas inversas simples del tipo `<carácter>`, como ``z``, ``A`` o ``2``. Los caracteres que se encuentran fuera de los límites o los índices negativos de una cadena darán como resultado un comportamiento indefinido.

*Nota:* Las comparaciones entre cadenas que utilizan y no utilizan retrotracción SQL pueden generar diferentes resultados donde existen espacios precedentes.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
allbutfirst(N, STRING)	<i>Cadena</i>	Devuelve una cadena, que es <i>CADENA</i> sin los <i>N</i> primeros caracteres.
allbutlast(N, STRING)	<i>Cadena</i>	Devuelve una cadena, que es <i>CADENA</i> sin los últimos caracteres.
alphabefore(STRING1, STRING2)	<i>Booleana</i>	Se utiliza para comprobar el orden alfabético de las cadenas. Devuelve verdadero si <i>CADENA1</i> precede a <i>CADENA2</i> .
endstring(LENGTH, STRING)	<i>Cadena</i>	Extrae los últimos <i>N</i> caracteres de la cadena especificada. Si la longitud de la cadena es menor o igual que la longitud especificada, no hay cambios.
hasendstring(STRING, SUBSTRING)	<i>Entero</i>	Esta función es la misma que <code>isendstring(SUBSTRING, STRING)</code> .
hasmidstring(STRING, SUBSTRING)	<i>Entero</i>	Esta función es la misma que <code>ismidstring(SUBSTRING, STRING)</code> (subcadena incrustada).
hasstartstring(STRING, SUBSTRING)	<i>Entero</i>	Esta función es la misma que <code>isstartstring(SUBSTRING, STRING)</code> .
hassubstring(STRING, N, SUBSTRING)	<i>Entero</i>	Esta función es la misma que <code>issubstring(SUBSTRING, N, STRING)</code> , donde <i>N</i> toma el valor 1 por defecto.
count_substring(STRING, SUBSTRING)	<i>Entero</i>	Devuelve el número de veces que aparece la subcadena especificada en la cadena. Por ejemplo: <code>count_substring("foooo.txt", "oo")</code> devuelve 3.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
hassubstring( <i>STRING</i> , <i>SUBSTRING</i> )	<i>Entero</i>	Esta función es la misma que <code>issubstring(<i>SUBSTRING</i>, 1, <i>STRING</i>)</code> , donde <i>N</i> toma el valor 1 por defecto.
isalphacode( <i>CHAR</i> )	<i>Booleana</i>	Devuelve verdadero si <i>CAR</i> es un carácter en la cadena especificada (normalmente un nombre de campo) cuyo código de carácter es una letra. En el resto de los casos, esta función devuelve un valor de 0. Por ejemplo, <code>isalphacode(produce_num(1))</code> .
isendstring( <i>SUBSTRING</i> , <i>STRING</i> )	<i>Entero</i>	Si la cadena <i>CADENA</i> finaliza con la subcadena <i>SUBCADENA</i> , esta función devuelve el subíndice entero de <i>SUBCADENA</i> en <i>CADENA</i> . En el resto de los casos, esta función devuelve 0.
islowercode( <i>CHAR</i> )	<i>Booleana</i>	Devuelve verdadero si <i>CAR</i> es un carácter de una letra minúscula para la cadena especificada (normalmente un nombre de campo). En el resto de los casos, esta función devuelve un valor de 0. Por ejemplo, tanto <code>islowercode('')</code> como <code>islowercode(country_name(2))</code> son expresiones válidas.
ismidstring( <i>SUBSTRING</i> , <i>STRING</i> )	<i>Entero</i>	Si <i>SUBCADENA</i> es una subcadena de <i>CADENA</i> pero no comienza en el primer carácter de <i>CADENA</i> ni termina en el último, esta función devuelve el subíndice donde dicha subcadena comienza. En el resto de los casos, esta función devuelve un valor de 0.
isnumbercode( <i>CHAR</i> )	<i>Booleana</i>	Devuelve verdadero si <i>CAR</i> , para la cadena especificada (normalmente un nombre de campo), es un carácter cuyo código de carácter es un número. En el resto de los casos, esta función devuelve un valor de 0. Por ejemplo, <code>isnumbercode(product_id(2))</code> .
isstartstring( <i>SUBSTRING</i> , <i>STRING</i> )	<i>Entero</i>	Si la cadena <i>CADENA</i> comienza con la subcadena <i>SUBCADENA</i> , esta función devuelve el subíndice 1. En el resto de los casos, esta función devuelve un valor de 0.
issubstring( <i>SUBSTRING</i> , <i>N</i> , <i>STRING</i> )	<i>Entero</i>	Busca la cadena <i>CADENA</i> , a partir del carácter número <i>N</i> , para una subcadena igual a la cadena <i>SUBCADENA</i> . Si la encuentra, esta función devuelve el subíndice entero donde comienza el subíndice coincidente. En el resto de los casos, esta función devuelve 0. Si no se proporciona <i>N</i> , esta función toma el valor 1 por defecto.
issubstring( <i>SUBSTRING</i> , <i>STRING</i> )	<i>Entero</i>	Busca la cadena <i>CADENA</i> , a partir del carácter número <i>N</i> , para una subcadena igual a la cadena <i>SUBCADENA</i> . Si la encuentra, esta función devuelve el subíndice entero donde comienza el subíndice coincidente. En el resto de los casos, esta función devuelve 0. Si no se proporciona <i>N</i> , esta función toma el valor 1 por defecto.

Función	Resultado	Descripción
issubstring_count(SUBSTRING, N, STRING):	<i>Entero</i>	Devuelve el índice de la instancia número <i>N</i> de la <i>SUBCADENA</i> en la <i>CADENA</i> especificada. Si hay menos de <i>N</i> instancias de <i>SUBCADENA</i> , se devuelve 0.
issubstring_lim(SUBSTRING, N, STARTLIM, ENDLIM, STRING)	<i>Entero</i>	Esta función es la misma que <i>issubstring</i> , pero la coincidencia se limita de manera que se inicia en el subíndice <i>LÍMINICIAL</i> (o antes) y finaliza en el subíndice <i>LÍMFINAL</i> (o antes). Las restricciones <i>STARTLIM</i> o <i>ENDLIM</i> se pueden desactivar proporcionando un valor de falso para cada argumento; por ejemplo, <i>issubstring_lim</i> (SUBSTRING, N, false, false, STRING) es lo mismo que <i>issubstring</i> .
isuppercode(CHAR)	<i>Booleana</i>	Devuelve verdadero si <i>CAR</i> es un carácter de una letra mayúscula. En el resto de los casos, esta función devuelve un valor de 0. Por ejemplo, tanto <i>isuppercode</i> ('') como <i>isuppercode</i> (country_name(2)) son expresiones válidas.
last(CHAR)	<i>Cadena</i>	Devuelve el último carácter <i>CAR</i> de <i>CADENA</i> (que debe tener una longitud de al menos un carácter).
length(STRING)	<i>Entero</i>	Devuelve la longitud de la cadena <i>CADENA</i> ; es decir el número de caracteres que hay en ella.
locchar(CHAR, N, STRING)	<i>Entero</i>	Se utiliza para identificar la ubicación de los caracteres en los campos simbólicos. La función busca en la cadena <i>CADENA</i> el carácter <i>CAR</i> , comenzando la búsqueda en el carácter número <i>N</i> de la <i>CADENA</i> . Esta función devuelve un valor que indica la ubicación (comenzando por <i>N</i> ) en la que se encuentra el carácter. Si no se encuentra el carácter, esta función devuelve 0. Si la función tiene un desplazamiento no válido ( <i>N</i> ) (por ejemplo, un desplazamiento que supera la longitud de la cadena), esta función devuelve \$null\$. Por ejemplo, <i>locchar</i> ('`', 2, web_page) busca el campo denominado <i>web_page</i> para el carácter '`' empezando por el segundo carácter en el valor del campo. <i>Nota:</i> Asegúrese de utilizar comillas inversas simples para entrecomillar el carácter especificado.

Función	Resultado	Descripción
locchar_back(CHAR, N, STRING)	Entero	Similar a locchar, con la diferencia de que la búsqueda se realiza hacia atrás, comenzando a partir del carácter número <i>N</i> . Por ejemplo, locchar_back('n', 9, web_page) realiza búsquedas en el campo <i>página_web</i> comenzando por el noveno carácter y yendo hacia atrás, hacia el inicio de la cadena. Si la función tiene un desplazamiento no válido (por ejemplo, un desplazamiento superior a la longitud de la cadena), esta función devuelve \$null\$. Lo ideal sería usar locchar_back junto con la función length(<field>) para utilizar dinámicamente la longitud del valor actual del campo. Por ejemplo, locchar_back('n', (length(web_page)), web_page).
lowertoupper(CHAR) lowertoupper (STRING)	CAR o Cadena	La entrada puede ser una cadena o un carácter, y se utiliza en esta función para devolver un nuevo elemento del mismo tipo, con los caracteres en minúscula convertidos a sus equivalentes en mayúscula. Por ejemplo, lowertoupper('a'), lowertoupper("My string") y lowertoupper(field_name(2)) son expresiones válidas.
matches	Booleana	Devuelve verdadero si una cadena coincide con un patrón determinado. El patrón debe ser una cadena literal, no puede ser un nombre de campo que contenga un patrón. Se puede incluir una interrogación (?) en el patrón para que coincida exactamente con un carácter, un asterisco (*) coincide con cero o más caracteres. Para que coincida con un asterisco o una interrogación literal (en lugar de utilizarlos como comodines), se puede utilizar una barra invertida (\) como carácter de escape.
replace(SUBSTRING, NEWSUBSTRING, STRING)	Cadena	En la CADENA especificada, sustituye todas las instancias de SUBCADENA con NUEVASUBCADENA.
replicate(COUNT, STRING)	Cadena	Devuelve una cadena que consta de la cadena original copiada el número específico de veces.
stripchar(CHAR,STRING)	Cadena	Permite eliminar determinados caracteres de una cadena o campo. Por ejemplo, puede utilizar esta función para eliminar de los datos símbolos adicionales, como notaciones de moneda, con el fin de obtener un nombre o número simple. Por ejemplo, la sintaxis stripchar('\$', 'Cost') devuelve un nuevo campo con el signo de dólar eliminado en todos los valores. <i>Nota:</i> Asegúrese de utilizar comillas inversas simples para entrecomillar el carácter especificado.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
skipchar(CHAR, N, STRING)	Entero	Busca en la cadena <i>CADENA</i> cualquier carácter que no sea <i>CAR</i> , comenzando por el carácter número <i>N</i> . Esta función devuelve una subcadena de enteros que indica el punto en que se encuentra cada uno, o 0 si todos los caracteres a partir del <i>N</i> son <i>CAR</i> . Si la función tiene un desplazamiento no válido (por ejemplo, un desplazamiento superior a la longitud de la cadena), esta función devuelve \$null\$. locchar se suele utilizar junto con las funciones skipchar para determinar el valor de <i>N</i> (el punto en que comenzar la búsqueda en la cadena). Por ejemplo, skipchar('s', (locchar('s', 1, "MyString")), "MyString").
skipchar_back(CHAR, N, STRING)	Entero	Similar a skipchar, con la diferencia de que la búsqueda se realiza <b>hacia atrás</b> , comenzando a partir del carácter número <i>N</i> .
startstring(LENGTH, STRING)	Cadena	Extrae los primeros <i>N</i> caracteres de la cadena especificada. Si la longitud de la cadena es menor o igual que la longitud especificada, no hay cambios.
strmember(CHAR, STRING)	Entero	Equivalente a locchar(CHAR, 1, STRING). Devuelve una subcadena de enteros que indica el punto en que <i>CHAR</i> aparece por primera vez o bien 0. Si la función tiene un desplazamiento no válido (por ejemplo, un desplazamiento superior a la longitud de la cadena), esta función devuelve \$null\$.
subscrs(N, STRING)	CAR	Devuelve el carácter <i>CAR</i> número <i>N</i> de la cadena de entrada <i>CADENA</i> . Esta función también se puede escribir de forma abreviada como STRING( <i>N</i> ). Por ejemplo, lowertoupper("name"(1)) es una expresión válida.
substring(N, LEN, STRING)	Cadena	Devuelve una cadena <i>SUBCADENA</i> , que consiste en los caracteres <i>LON</i> de la cadena <i>CADENA</i> , a partir del carácter situado en el subíndice <i>N</i> .
substring_between(N1, N2, STRING)	Cadena	Devuelve la subcadena de <i>CADENA</i> que comienza en el subíndice <i>N1</i> y termina en el subíndice <i>N2</i> .
trim(STRING)	Cadena	Extrae los espacios en blanco anteriores y posteriores a la cadena especificada.
trim_start(STRING)	Cadena	Extrae los espacios en blanco anteriores a la cadena especificada.
trimend(STRING)	Cadena	Extrae los espacios en blanco posteriores a la cadena especificada.
unicode_char(NUM)	CAR	Devuelve el carácter con el valor Unicode <i>NÚM</i> .

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
unicode_value(CHAR)	NUM	Devuelve el valor Unicode de <i>CAR</i>
uppertolower(CHAR) uppertolower (STRING)	<i>CAR</i> o <i>Cadena</i>	La entrada puede ser una cadena o un carácter y se utiliza en esta función para devolver un nuevo elemento del mismo tipo, con los caracteres en mayúscula convertidos a sus equivalentes en minúscula. <i>Nota:</i> No olvide especificar las cadenas con comillas dobles y los caracteres con comillas inversas simples. Los nombres de campos simples deben especificarse sin comillas.

### **Funciones SoundEx**

SoundEx es un método utilizado para buscar cadenas cuando se conoce el sonido pero no la ortografía exacta. Desarrollado en 1918, busca palabras con sonidos similares según ciertos supuestos fonéticos sobre cómo se pronuncian ciertas letras. Se puede utilizar para buscar nombres en una base de datos, por ejemplo, cuando la ortografía o la pronunciación de ciertos nombres puede variar. El algoritmo SoundEx básico se documenta en varios orígenes y, a pesar de las limitaciones conocidas (por ejemplo, combinaciones de letras precedentes como ph y f no coincidirán aunque el sonido sea el mismo), es compatible en cierto modo con la mayoría de las bases de datos.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
soundex(STRING)	<i>Entero</i>	Devuelve el código SoundEx de cuatro caracteres para la <i>CADENA</i> especificada.
soundex_difference(STRING1, STRING2)	<i>Entero</i>	Devuelve un entero entre 0 y 4 que indica el número de caracteres que son iguales en la codificación SoundEx para las dos cadenas, donde 0 indica que no hay similitud y 4 indica una fuerte similitud o cadenas idénticas.

### **Funciones de fecha y hora**

CLEM incluye un grupo de funciones para la gestión de campos con almacenamiento de fecha y hora de variables de cadena que representan fechas y horas. Los formatos de fecha y hora utilizados son específicos de cada ruta y se especifican en el cuadro de diálogo de propiedades de la ruta. Las funciones de fecha y hora analizan las cadenas de fecha y hora en función del formato seleccionado actualmente.

Cuando se especifica un año en una fecha que utiliza sólo dos dígitos (es decir, el siglo no se especifica), IBM® SPSS® Modeler utiliza el siglo por defecto que se ha especificado en el cuadro de diálogo de propiedades de la ruta.



*Nota:* Las funciones de fecha y hora no se pueden llamar desde procesos. [Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.](#)

Función	Resultado	Descripción
@TODAY	Cadena	Si selecciona Admitir fecha/mín. negativos en el cuadro de diálogo de propiedades de la ruta, esta función devuelve la fecha actual, como una cadena, con el formato de fecha actual. Si utiliza un formato de fecha de dos dígitos y no selecciona Admitir fecha/mín. negativos, esta función devuelve \$null\$ en el servidor actual. Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.</a>
to_time(ITEM)	Hora	Convierte el almacenamiento del campo especificado en una hora.
to_date(ITEM)	Fecha	Convierte el almacenamiento del campo especificado en una fecha.
to_timestamp(ITEM)	Marca de tiempo	Convierte el almacenamiento del campo especificado en una marca de tiempo.
to_datetime(ITEM)	Momento_fecha	Convierte el almacenamiento del campo especificado en una fecha, hora o marca de tiempo.
datetime_date(ITEM)	Fecha	Devuelve el valor de fecha de un número, cadena o marca de tiempo. Tenga en cuenta que es la única función que le permite volver a convertir un número (en segundos) en una fecha. Si ITEM es una cadena, crea una fecha analizando una cadena en el formato de fecha actual. El formato de fecha especificado en el cuadro de diálogo de propiedades de la ruta debe ser correcto para que esta función se ejecute correctamente. Si ITEM es un número, se interpreta como un número de segundos desde la fecha de base (o época). Las fracciones de un día se truncan. Si ITEM es una marca de tiempo, la parte de la fecha de la marca de tiempo se devuelve. Si ITEM es una fecha, se devuelve sin modificar.
date_before(DATE1, DATE2)	Booleana	Devuelve un valor de verdadero si FECHA1 representa una fecha o marca de tiempo anterior a la representada por FECHA2. En el resto de los casos, esta función devuelve 0.
date_days_difference(DATE1, DATE2)	Entero	Devuelve el tiempo en días desde la fecha o marca de tiempo representada por FECHA1 hasta la fecha representada por FECHA2, como un entero. Si FECHA2 es anterior a FECHA1, esta función devuelve un número negativo.
date_in_days(ITEM)	Entero	Devuelve el tiempo en días desde la fecha de línea base hasta la fecha o marca de tiempo representada por FECHA, como un entero. Si FECHA es anterior a la fecha de línea base, esta función devuelve un número negativo. Debe incluir una fecha válida para que el cálculo se realice correctamente. Por ejemplo, no debe especificar 29 de febrero de 2001 como fecha. Dado que 2001 no es un año bisiesto, esta fecha no existe.

Función	Resultado	Descripción
date_in_months(DATE)	Real	Devuelve el tiempo en meses desde la fecha de línea base hasta la fecha o marca de tiempo representada por <i>FECHA</i> , como un número real. Esta cifra es aproximada; se basa en un mes de 30.4375 días. Si <i>FECHA</i> es anterior a la fecha de línea base, esta función devuelve un número negativo. Debe incluir una fecha válida para que el cálculo se realice correctamente. Por ejemplo, no debe especificar 29 de febrero de 2001 como fecha. Dado que 2001 no es un año bisiesto, esta fecha no existe.
date_in_weeks(DATE)	Real	Devuelve el tiempo en semanas desde la fecha de línea base hasta la fecha o marca de tiempo representada por <i>FECHA</i> , como un número real. Éste se basa en una semana de 7,0 días. Si <i>FECHA</i> es anterior a la fecha de línea base, esta función devuelve un número negativo. Debe incluir una fecha válida para que el cálculo se realice correctamente. Por ejemplo, no debe especificar 29 de febrero de 2001 como fecha. Dado que 2001 no es un año bisiesto, esta fecha no existe.
date_in_years(DATE)	Real	Devuelve el tiempo en años desde la fecha de línea base hasta la fecha o marca de tiempo representada por <i>FECHA</i> , como un número real. Esta cifra es aproximada; se basa en un año de 365.25 días. Si <i>FECHA</i> es anterior a la fecha de línea base, esta función devuelve un número negativo. Debe incluir una fecha válida para que el cálculo se realice correctamente. Por ejemplo, no debe especificar 29 de febrero de 2001 como fecha. Dado que 2001 no es un año bisiesto, esta fecha no existe.
date_months_difference (DATE1, DATE2)	Real	Devuelve el tiempo en meses desde la fecha o marca de tiempo representada por <i>FECHA1</i> hasta la fecha representada por <i>FECHA2</i> , como un número real. Esta cifra es aproximada; se basa en un mes de 30.4375 días. Si <i>FECHA2</i> es anterior a <i>FECHA1</i> , esta función devuelve un número negativo.
datetime_date(YEAR, MONTH, DAY)	Fecha	Crea un valor de fecha para los valores <i>AÑO</i> , <i>MES</i> y <i>DÍA</i> especificados. Los argumentos deben ser enteros.
datetime_day(DATE)	Entero	Devuelve el día del mes de una <i>FECHA</i> o marca de tiempo especificada. El resultado es un entero comprendido entre 1 y 31.
datetime_day_name(DAY)	Cadena	Devuelve el nombre completo del <i>DÍA</i> especificado. El argumento debe ser un entero comprendido entre 1 (domingo) y 7 (sábado).
datetime_hour(TIME)	Entero	Devuelve la hora de una <i>HORA</i> o marca de tiempo. El resultado es un entero comprendido entre 0 y 23.
datetime_in_seconds(TIME)	Real	Devuelve la segunda parte almacenada en <i>TIEMPO</i> .
datetime_in_seconds(DATE), datetime_in_seconds(DATE-TIME)	Real	Devuelve el número acumulado, convertido en segundos, desde la diferencia entre la <i>FECHA</i> o <i>MOMENTO DE FECHA</i> actual y la fecha de línea base (01-01-1900).
datetime_minute(TIME)	Entero	Devuelve el minuto de una <i>HORA</i> o marca de tiempo. El resultado es un entero comprendido entre 0 y 59.
datetime_month(DATE)	Entero	Devuelve el mes de una <i>FECHA</i> o marca de tiempo. El resultado es un entero comprendido entre 1 y 12.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
datetime_month_name(MONTH)	<i>Cadena</i>	Devuelve el nombre completo del <i>MES</i> especificado. El argumento debe ser un entero comprendido entre 1 y 12.
datetime_now	<i>Marca de tiempo</i>	Devuelve la hora actual como una marca de tiempo.
datetime_second(TIME)	<i>Entero</i>	Devuelve el segundo de una <i>HORA</i> o marca de tiempo. El resultado es un entero comprendido entre 0 y 59.
datetime_day_short_name(DAY)	<i>Cadena</i>	Devuelve el nombre abreviado del <i>DÍA</i> especificado. El argumento debe ser un entero comprendido entre 1 (domingo) y 7 (sábado).
datetime_month_short_name(MONTH)	<i>Cadena</i>	Devuelve el nombre abreviado del <i>MES</i> especificado. El argumento debe ser un entero comprendido entre 1 y 12.
datetime_time(HOUR, MINUTE, SECOND)	<i>Hora</i>	Devuelve el valor de tiempo para los valores <i>HORA</i> , <i>MINUTO</i> y <i>SEGUNDO</i> especificados. Los argumentos deben ser enteros.
datetime_time(ITEM)	<i>Hora</i>	Devuelve el valor de tiempo del <i>ELEMENTO</i> especificado.
datetime_timestamp(YEAR, MONTH, DAY, HOUR, MINUTE, SECOND)	<i>Marca de tiempo</i>	Devuelve el valor de marca de tiempo para los valores <i>AÑO</i> , <i>MES</i> , <i>DÍA</i> , <i>HORA</i> , <i>MINUTO</i> y <i>SEGUNDO</i> especificados.
datetime_timestamp(DATE, TIME)	<i>Marca de tiempo</i>	Devuelve el valor de marca de tiempo para los valores <i>FECHA</i> y <i>HORA</i> especificados.
datetime_timestamp (NUMBER)	<i>Marca de tiempo</i>	Devuelve el valor de marca de tiempo del número especificado de segundos.
datetime_weekday(DATE)	<i>Entero</i>	Devuelve el día de la semana a partir de la <i>FECHA</i> o marca de tiempo especificada.
datetime_year(DATE)	<i>Entero</i>	Devuelve el año a partir de una <i>FECHA</i> o marca de tiempo. El resultado es un entero como, por ejemplo, 2002.
date_weeks_difference (DATE1, DATE2)	<i>Real</i>	Devuelve el tiempo en semanas desde la fecha o marca de tiempo representada por <i>FECHA1</i> hasta la fecha representada por <i>FECHA2</i> , como un número real. Éste se basa en una semana de 7,0 días. Si <i>FECHA2</i> es anterior a <i>FECHA1</i> , esta función devuelve un número negativo.
date_years_difference (DATE1, DATE2)	<i>Real</i>	Devuelve el tiempo en años desde la fecha o marca de tiempo representada por <i>FECHA1</i> hasta la fecha representada por <i>FECHA2</i> , como un número real. Esta cifra es aproximada; se basa en un año de 365.25 días. Si <i>FECHA2</i> es anterior a <i>FECHA1</i> , esta función devuelve un número negativo.
time_before(TIME1, TIME2)	<i>Booleana</i>	Devuelve un valor verdadero si <i>HORA1</i> representa una hora o marca de tiempo anterior a la representada por <i>HORA2</i> . En el resto de los casos, esta función devuelve 0.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
time_hours_difference (TIME1, TIME2)	<i>Real</i>	Devuelve el número de horas de diferencia que hay entre las horas o marcas de tiempo representadas por <i>HORA1</i> y <i>HORA2</i> , como un número real. Si selecciona Admitir fecha/mín. negativos en el cuadro de diálogo de propiedades de la ruta, se toma un valor superior de <i>HORA1</i> para hacer referencia al día anterior. Si no selecciona la opción de admisión de negativos, un valor superior de <i>HORA1</i> hará que el valor devuelto sea negativo.
time_in_hours(TIME)	<i>Real</i>	Devuelve el número de horas representado por <i>HORA</i> , como un número real. Por ejemplo, con el formato de hora HHMM, la expresión time_in_hours('0130') se evalúa como 1,5. <i>HORA</i> puede representar una hora o una marca de tiempo.
time_in_mins(TIME)	<i>Real</i>	Devuelve el número de minutos representado por <i>HORA</i> , como un número real. <i>HORA</i> puede representar una hora o una marca de tiempo.
time_in_secs(TIME)	<i>Entero</i>	Devuelve el número de segundos representado por <i>HORA</i> , como un entero. <i>HORA</i> puede representar una hora o una marca de tiempo.
time_mins_difference(TIME1, TIME2)	<i>Real</i>	Devuelve el número de minutos de diferencia que hay entre las horas o marcas de tiempo representadas por <i>HORA1</i> y <i>HORA2</i> , como un número real. Si selecciona Admitir fecha/mín. negativos en el cuadro de diálogo de propiedades de la ruta, se toma un valor superior de <i>HORA1</i> para hacer referencia al día anterior (o la hora anterior, si sólo se especifican minutos y segundos en el formato actual). Si no selecciona la opción de admisión de negativos, un valor superior de <i>HORA1</i> hará que el valor devuelto sea negativo.
time_secs_difference(TIME1, TIME2)	<i>Entero</i>	Devuelve el número de segundos de diferencia que hay entre las horas o marcas de tiempo representadas por <i>HORA1</i> y <i>HORA2</i> , como un entero. Si selecciona Admitir fecha/mín. negativos en el cuadro de diálogo de propiedades de la ruta, se toma un valor superior de <i>HORA1</i> para hacer referencia al día anterior (o la hora anterior, si sólo se especifican minutos y segundos en el formato actual). Si no selecciona la opción de admisión de negativos, un valor superior de <i>HORA1</i> hará que el valor devuelto sea negativo.

### **Conversión de valores de fecha y hora**

Tenga en cuenta que las funciones de conversión (y otras muchas funciones que requieren un tipo específico de entrada, como un valor de fecha u hora) dependen de los formatos actuales especificados en el cuadro de diálogo de opciones de ruta. Por ejemplo, si tiene un campo denominado *FECHA* que se guarda como una cadena con los valores *Ene 2003*, *Feb 2003*, etcétera, puede convertirla en campos de almacenamiento de la forma siguiente:

to\_date(FECHA)

Para que esta conversión funcione, seleccione el formato de fecha coincidente MES AAAA como formato de fecha por defecto de la ruta. [Si desea obtener más información, consulte el tema Configuración de opciones generales de las rutas en el capítulo 5 en \*Manual de usuario de IBM SPSS Modeler 15\*.](#)

Para ver un ejemplo que convierte valores de cadena a fecha utilizando un nodo Rellenar, consulte la ruta *broadband\_create\_models.str*, en la carpeta *\Demos* en la subcarpeta *streams*. [Si desea obtener más información, consulte el tema Predicciones con el nodo Serie temporal en el capítulo 14 en \*Guía de aplicaciones de IBM SPSS Modeler 15\*.](#)

**Fechas almacenadas como números.** Tenga en cuenta que *FECHA* en el ejemplo anterior es el nombre de un campo, mientras que *to\_date* es una función de CLEM. Si tiene datos guardados como números, puede convertirlos utilizando la función *datetime\_date*, donde el número se interpreta como un número de segundos desde la fecha de base (o época).

```
datetime_date(FECHA)
```

Al convertir una fecha a un número de segundos (y viceversa), puede ejecutar operaciones como calcular la fecha actual más o menos un número fijo de días, por ejemplo:

```
datetime_date((date_in_days(FECHA)-7)*60*60*24)
```

## ***Funciones de secuencia***

Para algunas operaciones, la secuencia de eventos tiene su importancia. La aplicación permite trabajar con las siguientes secuencias de registros:

- Secuencias y series temporales
- Funciones de secuencia
- Indización de registros
- Media, suma y comparación de valores
- Control de cambio - diferenciación
- @SINCE
- Valores de desplazamiento
- Facilidades de secuencias adicionales

Para muchas aplicaciones, cada registro que pasa por una ruta se puede considerar como un caso individual, independiente de todos los demás. En estas situaciones, el orden de los registros suele no ser importante.

Sin embargo, para algunos tipos de problemas, la secuencia de registros es de gran importancia. Suelen ser situaciones de series temporales, en las que la secuencia de registros representa una secuencia de eventos o instancias ordenada. Cada registro representa una instantánea de un momento concreto; sin embargo, gran parte de la información más importante puede no estar contenida en valores instantáneos, sino en el modo en que tales valores cambian y se comportan durante un período de tiempo.

Por supuesto, el parámetro relevante puede no ser temporal. Por ejemplo, los registros podrían representar análisis realizados a distintas distancias a lo largo de una línea, pero se aplicarían los mismos principios.

Las funciones de secuencia y especiales se reconocen de inmediato por las siguientes características:

- Todas tienen el prefijo @.
- Sus nombres se proporcionan en mayúsculas.

Las funciones de secuencia pueden hacer referencia al registro que está siendo procesado actualmente por un nodo, los registros que ya han pasado por un nodo e, incluso, a los registros que aún tienen que pasar por un nodo. Las funciones de secuencia se pueden mezclar libremente con otros componentes de expresiones CLEM, aunque algunas tienen restricciones en cuanto a qué pueden utilizar como sus argumentos.

### **Ejemplos**

Puede que le resulte útil saber cuánto tiempo ha pasado desde que se produjo un determinado evento o una condición fue verdadera. Para ello, utilice la función @SINCE; por ejemplo:

```
@SINCE(Ingresos > Pagos)
```

Esta función devuelve el desplazamiento del último registro donde esta condición fue verdadera; es decir, el número de registros anteriores a éste en el que la condición fue verdadera. Si la condición nunca ha sido verdadera, @SINCE devuelve @INDEX + 1.

Puede que en algunas ocasiones desee hacer referencia a un valor del registro actual en la expresión utilizada por @SINCE. Puede hacerlo mediante la función @THIS, que especifica que un nombre de campo siempre se aplique al registro actual. Para buscar el desplazamiento del último registro que tuvo un valor de campo Concentration más de dos veces que el registro actual, podría utilizar:

```
@SINCE(Concentración > 2 * @THIS(Concentración))
```

En algunos casos, la condición especificada para @SINCE es verdadera en el registro actual por definición; por ejemplo:

```
@SINCE(ID == @THIS(ID))
```

Por este motivo, @SINCE no evalúa su condición para el registro actual. Utilice una función similar, @SINCE0, si desea evaluar la condición para el registro actual además de los anteriores; si la condición es verdadera en el registro actual, @SINCE0 devuelve 0.

*Nota:* @ Las funciones no se pueden llamar desde procesos. Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.

Función	Resultado	Descripción
MEAN(FIELD)	<i>Real</i>	Devuelve el valor promediado de los valores para el CAMPO o CAMPOS especificados.
@MEAN(FIELD, EXPR)	<i>Real</i>	Devuelve el valor promediado de los valores para CAMPO en los últimos registros de EXPR recibidos por el nodo actual, incluyendo el registro actual. CAMPO debe ser el nombre de un campo numérico. EXPR puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si EXPR se omite, o si supera el número de registros recibidos hasta el momento, se devuelve el promedio de todos los registros recibidos hasta el momento. Observe que esta función no se puede llamar desde un proceso. Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.
@MEAN(FIELD, EXPR, INT)	<i>Real</i>	Devuelve el valor promediado de los valores para CAMPO en los últimos registros de EXPR recibidos por el nodo actual, incluyendo el registro actual. CAMPO debe ser el nombre de un campo numérico. EXPR puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si EXPR se omite, o si supera el número de registros recibidos hasta el momento, se devuelve el promedio de todos los registros recibidos hasta el momento. ENT especifica el número máximo de valores que buscar hacia atrás, lo que es mucho más eficaz que utilizar tan sólo dos argumentos.
@DIFF1(FIELD)	<i>Real</i>	Devuelve el primer diferencial de CAMPO1. El formato de un único argumento devuelve simplemente la diferencia entre el valor actual y el valor anterior del campo. Devuelve 0 si los registros anteriores relevantes no existen.
@DIFF1(FIELD1, FIELD2)	<i>Real</i>	El formato de dos argumentos proporciona el primer diferencial de CAMPO1 respecto a CAMPO2. Devuelve 0 si los registros anteriores relevantes no existen.
@DIFF2(FIELD)	<i>Real</i>	Devuelve el segundo diferencial de CAMPO1. El formato de un único argumento devuelve simplemente la diferencia entre el valor actual y el valor anterior del campo. Devuelve 0 si los registros anteriores relevantes no existen.
@DIFF2(FIELD1, FIELD2)	<i>Real</i>	El formato de dos argumentos proporciona el primer diferencial de CAMPO1 respecto a CAMPO2. Devuelve 0 si los registros anteriores relevantes no existen.
@INDEX	<i>Entero</i>	Devuelve el índice del registro actual. Los índices se asignan a los registros a medida que llegan al nodo actual. Al primer registro se le asigna el índice 1, y el índice se va incrementando en 1 para cada registro posterior.

Función	Resultado	Descripción
@LAST_NON_BLANK(FIELD)	Cualquiera	Devuelve el último valor de <i>CAMPO</i> que no está vacío, tal y como se define en un nodo Tipo o de origen anterior de la ruta. Si no hay ningún valor no vacío para <i>CAMPO</i> en los registros leídos hasta el momento, se devuelve \$null\$. Recuerde que los valores vacíos, también llamados valores perdidos por el usuario, se pueden definir por separado para cada campo.
@MAX(FIELD)	Number	Devuelve el valor máximo del <i>CAMPO</i> especificado.
@MAX(FIELD, EXPR)	Number	Devuelve el valor máximo de <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos hasta el momento, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0.
@MAX(FIELD, EXPR, INT)	Number	Devuelve el valor máximo de <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos hasta el momento, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si <i>EXPR</i> se omite, o si supera el número de registros recibidos hasta el momento, se devuelve el valor máximo de todos los registros recibidos hasta el momento. <i>ENT</i> especifica el número máximo de valores que buscar hacia atrás, lo que es mucho más eficaz que utilizar tan sólo dos argumentos.
@MIN(FIELD)	Number	Devuelve el valor mínimo del <i>CAMPO</i> especificado.
@MIN(FIELD, EXPR)	Number	Devuelve el valor mínimo del <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos hasta el momento, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0.
@MIN(FIELD, EXPR, INT)	Number	Devuelve el valor mínimo del <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos hasta el momento, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si <i>EXPR</i> se omite, o si supera el número de registros recibidos hasta el momento, se devuelve el valor mínimo de todos los registros recibidos hasta el momento. <i>ENT</i> especifica el número máximo de valores que buscar hacia atrás, lo que es mucho más eficaz que utilizar tan sólo dos argumentos.



Función	Resultado	Descripción
@OFFSET(FIELD, EXPR)	Cualquiera	<p>Devuelve el valor de <i>CAMPO</i> en el desplazamiento de registro respecto al registro actual con el valor <i>EXPR</i>. Un positivo hace referencia a un registro que ya ha pasado, mientras que uno negativo especifica una “lectura previa” de un registro que aún tiene que llegar. Por ejemplo, @OFFSET(Status, 1) devuelve el valor del campo Status en el registro anterior, mientras que @OFFSET(Status, -4) hace una “lectura previa” de cuatro registros de la secuencia (es decir, en los registros que aún no han pasado por este nodo) para obtener el valor. <i>Tenga en cuenta que un desplazamiento negativo (lectura previa) se debe especificar como una constante.</i> En desplazamientos positivos únicamente, <i>EXPR</i> también debe ser una expresión CLEM arbitraria, que se evalúa para que el registro actual proporcione el desplazamiento. En este caso, la versión de tres argumentos de esta función debería mejorar el rendimiento (ver función siguiente). Si la expresión devuelve cualquier cosa que no sea un entero no negativo, se producirá un error; es decir, no es legal haber calculado desplazamientos de lectura previa.</p> <p><i>Nota:</i> Una función @OFFSET autorreferida no puede utilizar lectura previa literal. Por ejemplo, en un nodo Rellenar, no puede sustituir el valor de field1 utilizando una expresión como @OFFSET(field1,-2).</p>
@OFFSET(FIELD, EXPR, INT)	Cualquiera	<p>Realiza la misma operación que la función @OFFSET con la adición de un tercer argumento, <i>ENT</i>, que especifica el número máximo de valores que buscar hacia atrás. En casos en los que el desplazamiento se calcula a partir de una expresión, este tercer argumento debería mejorar el rendimiento. Por ejemplo, en una expresión como @OFFSET(Foo, Month, 12), el sistema sabe que debe mantener sólo los últimos doce valores de Foo; de lo contrario, tiene que almacenar todos los valores, por si acaso. Para los desplazamientos fijos, incluyendo los desplazamientos de “lectura previa” negativos, que deben ser fijos, el tercer argumento no es importante, y debe utilizarse la versión de dos argumentos de esta función. Consulte también la nota acerca de las funciones autorreferidas en la versión de dos argumentos descritas anteriormente.</p>
@SDEV(FIELD)	Real	Devuelve la desviación típica de los valores del <i>CAMPO</i> o <i>CAMPOS</i> especificados.
@SDEV(FIELD, EXPR)	Real	Devuelve la desviación típica de los valores de <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos por el nodo actual, incluyendo el registro <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si <i>EXPR</i> se omite, o si supera el número de registros recibidos hasta el momento, se devuelve la desviación típica de todos los registros recibidos hasta el momento.

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
@SDEV(FIELD, EXPR, INT)	<i>Real</i>	Devuelve la desviación típica de los valores de <i>CAMPO</i> en los últimos registros de <i>EXPR</i> recibidos por el nodo actual, incluyendo el registro <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si <i>EXPR</i> se omite, o si supera el número de registros recibidos hasta el momento, se devuelve la desviación típica de todos los registros recibidos hasta el momento. <i>ENT</i> especifica el número máximo de valores que buscar hacia atrás, lo que es mucho más eficaz que utilizar tan sólo dos argumentos.
@SINCE(EXPR)	<i>Cualquiera</i>	Devuelve el número de registros que se han aceptado ya que <i>EXPR</i> , una expresión CLEM arbitraria, era verdadera.
@SINCE(EXPR, INT)	<i>Cualquiera</i>	La adición del segundo argumento, <i>ENT</i> , especifica el número máximo de registros a buscar hacia atrás. Si <i>EXPR</i> nunca ha sido verdadera, <i>ENT</i> es @INDEX+1.
@SINCE0(EXPR)	<i>Cualquiera</i>	Considera el registro actual, mientras que @SINCE no lo hace; @SINCE0 devuelve 0 si <i>EXPR</i> es verdadera para el registro actual.
@SINCE0(EXPR, INT)	<i>Cualquiera</i>	La adición del segundo argumento, <i>ENT</i> , especifica el número máximo de registros a buscar hacia atrás.
@SUM(FIELD)	<i>Number</i>	Devuelve la suma de los valores del <i>CAMPO</i> o <i>CAMPOS</i> especificados.
@SUM(FIELD, EXPR)	<i>Number</i>	Devuelve la suma de los valores de <i>CAMPO</i> en los últimos <i>EXPR</i> registros recibidos por el nodo actual, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si se omite <i>EXPR</i> , o si supera el número de registros recibidos hasta el momento, se devuelve la suma de todos los registros recibidos hasta el momento.
@SUM(FIELD, EXPR, INT)	<i>Number</i>	Devuelve la suma de los valores de <i>CAMPO</i> en los últimos <i>EXPR</i> registros recibidos por el nodo actual, incluyendo el registro actual. <i>CAMPO</i> debe ser el nombre de un campo numérico. <i>EXPR</i> puede ser cualquier expresión que se evalúe como un entero mayor que 0. Si se omite <i>EXPR</i> , o si supera el número de registros recibidos hasta el momento, se devuelve la suma de todos los registros recibidos hasta el momento. <i>ENT</i> especifica el número máximo de valores que buscar hacia atrás, lo que es mucho más eficaz que utilizar tan sólo dos argumentos.
@THIS(FIELD)	<i>Cualquiera</i>	Devuelve el valor del campo con el nombre <i>CAMPO</i> en el registro actual. Sólo se utiliza en las expresiones @SINCE.

### **Funciones globales**

Las funciones @MEAN, @SUM, @MIN, @MAX y @SDEV funcionan, en su mayoría, en todos los registros leídos y que se incluyen en el registro actual. Sin embargo, en algunos casos resulta útil poder saber cómo los valores del registro actual se comparan con los valores considerados en todo

el conjunto de datos. Utilizando un nodo Val. globales para generar valores en todo el conjunto de datos, puede acceder a los mismos en una expresión CLEM mediante las funciones globales.

Por ejemplo:

@GLOBAL\_MAX(Edad)

devuelve al valor más alto de Age en el conjunto de datos, mientras que la expresión

(Valor - @GLOBAL\_MEAN(Valor)) / @GLOBAL\_SDEV(Valor)

expresa la diferencia entre el Value de este registro y la media global como un número de desviaciones típicas. Puede utilizar valores globales sólo una vez que éstos han sido calculados por un nodo Val. globales. Todos los valores globales actuales se pueden cancelar pulsando en el botón Borrar valores globales de la pestaña Valores globales del cuadro de diálogo de propiedades de la ruta.

*Nota:* @ Las funciones no se pueden llamar desde procesos. [Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.](#)

Función	Resultado	Descripción
@GLOBAL_MAX(FIELD)	Number	Devuelve el valor máximo de <i>CAMPO</i> en todo el conjunto de datos, según lo haya generado previamente un nodo Val. globales. <i>CAMPO</i> debe ser el nombre de un campo numérico. Si no se ha establecido el valor global correspondiente, se produce un error. Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.</a>
@GLOBAL_MIN(FIELD)	Number	Devuelve el valor mínimo de <i>CAMPO</i> en todo el conjunto de datos, según lo haya generado previamente un nodo Val. globales. <i>CAMPO</i> debe ser el nombre de un campo numérico. Si no se ha establecido el valor global correspondiente, se produce un error.
@GLOBAL_SDEV(FIELD)	Number	Devuelve la desviación típica de los valores de <i>CAMPO</i> en todo el conjunto de datos, según lo haya generado previamente un nodo Val. globales. <i>CAMPO</i> debe ser el nombre de un campo numérico. Si no se ha establecido el valor global correspondiente, se produce un error.
@GLOBAL_MEAN(FIELD)	Number	Devuelve el valor promediado de los valores de <i>CAMPO</i> en todo el conjunto de datos, según lo haya generado previamente un nodo Val. globales. <i>CAMPO</i> debe ser el nombre de un campo numérico. Si no se ha establecido el valor global correspondiente, se produce un error.
@GLOBAL_SUM(FIELD)	Number	Devuelve la suma de los valores de <i>CAMPO</i> en todo el conjunto de datos, según lo haya generado previamente un nodo Val. globales. <i>CAMPO</i> debe ser el nombre de un campo numérico. Si no se ha establecido el valor global correspondiente, se produce un error.

### **Funciones para gestionar los valores vacíos y nulos**

Con CLEM, puede especificar que ciertos valores de un campo se consideren “vacíos” o perdidos. Las siguientes funciones trabajan con valores o elementos vacíos.

*Nota:* @ Las funciones no se pueden llamar desde procesos. Si desea obtener más información, consulte el tema [Expresiones CLEM en los procesos en el capítulo 3 el p. 28](#).

<b>Función</b>	<b>Resultado</b>	<b>Descripción</b>
@BLANK(FIELD)	<i>Booleana</i>	Devuelve verdadero para todos los registros cuyos valores están vacíos de acuerdo con el conjunto de reglas de tratamiento de los elementos vacíos en un nodo Tipo o de origen (pestaña Tipos) anterior de la ruta. Observe que esta función no se puede llamar desde un proceso. <a href="#">Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28</a> .
@LAST_NON_BLANK(FIELD)	<i>Cualquiera</i>	Devuelve el último valor de CAMPO que no está vacío, tal y como se define en un nodo Tipo o de origen anterior de la ruta. Si no hay ningún valor no vacío para CAMPO en los registros leídos hasta el momento, se devuelve \$null\$. Recuerde que los valores vacíos, también llamados valores perdidos por el usuario, se pueden definir por separado para cada campo.
@NULL(FIELD)	<i>Booleana</i>	Devuelve verdadero si el valor de CAMPO es el valor \$null\$. perdido por el sistema. Devuelve falso para el resto de valores, incluyendo los elementos vacíos definidos por el usuario. Si desea comprobar los dos, utilice @BLANK(FIELD) y @NULL(FIELD).
undef	<i>Cualquiera</i>	Se utiliza generalmente en CLEM para introducir un valor \$null\$; por ejemplo, para rellenar valores vacíos con nulos en el nodo Rellenar.

Los campos vacíos se pueden “rellenar” con el nodo Rellenar. En los nodos Rellenar y Derivar (sólo modo múltiple), la función CLEM especial @FIELD hace referencia al campo o campos actuales que están siendo examinados.

### ***Campos especiales***

Las funciones especiales se utilizan para denotar los campos específicos que están siendo examinados o para generar una lista de campos como entrada. Por ejemplo, cuando se derivan varios campos al mismo tiempo, debe utilizar @FIELD para denotar “realizar esta acción de derivación en los campos seleccionados”. Con la expresión log(@FIELD) se deriva un nuevo campo de log para cada campo seleccionado.

*Nota:* @ Las funciones no se pueden llamar desde procesos. Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.

Función	Resultado	Descripción
@FIELD	Cualquiera	Realiza una acción en todos los campos especificados en el contexto de la expresión. Observe que esta función no se puede llamar desde un proceso. Si desea obtener más información, consulte el tema Expresiones CLEM en los procesos en el capítulo 3 el p. 28.
@TARGET	Cualquiera	Cuando se utiliza una expresión CLEM en una función de análisis definida por el usuario, @TARGET representa el campo objetivo o “valor correcto” de la pareja objetivo/pronosticado que se está analizando. Esta función se suele utilizar en un nodo Análisis.
@PREDICTED	Cualquiera	Cuando se utiliza una expresión CLEM en una función de análisis definida por el usuario, @PREDICTED representa el valor pronosticado de la pareja objetivo/pronosticado que se está analizando. Esta función se suele utilizar en un nodo Análisis.
@PARTITION_FIELD	Cualquiera	Sustituye el nombre del campo de partición actual.
@TRAINING_PARTITION	Cualquiera	Devuelve el valor de la partición de entrenamiento actual. Por ejemplo, para seleccionar registros de entrenamiento mediante el nodo Seleccionar, utilice la expresión CLEM: @PARTITION_FIELD = @TRAINING_PARTITION. De este modo se garantiza que el nodo Seleccionar funcione siempre independientemente de los valores que se utilicen para representar cada partición de los datos.
@TESTING_PARTITION	Cualquiera	Devuelve el valor de la partición de comprobación actual.
@VALIDATION_PARTITION	Cualquiera	Devuelve el valor de la partición de validación actual.
@FIELDS_BETWEEN(start, end)	Cualquiera	Devuelve la lista de nombres de campo entre los campos de inicio y fin especificados (inclusive) según el orden natural (es decir, de inserción) de los campos en los datos. Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.

Función	Resultado	Descripción
@FIELDS_MATCHING(pattern)	Cualquiera	Devuelve una lista de nombres de campo que coinciden un patrón especificado. Se puede incluir una interrogación (?) en el patrón para que coincida exactamente con un carácter, un asterisco (*) coincide con cero o más caracteres. Para que coincida con un asterisco o una interrogación literal (en lugar de utilizarlos como comodines), se puede utilizar una barra invertida (\) como carácter de escape. <a href="#">Si desea obtener más información, consulte el tema Resumen de varios campos en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>
@MULTI_RESPONSE_SET	Cualquiera	Devuelve la lista de campos del conjunto de respuestas múltiples nombrado. <a href="#">Si desea obtener más información, consulte el tema Trabajo con datos de respuestas múltiples en el capítulo 7 en Manual de usuario de IBM SPSS Modeler 15.</a>

***Parte II:***  
***Referencia de propiedades***

# Referencia de propiedades

## Conceptos básicos de referencia de propiedades

Puede especificar varias propiedades diferentes para los nodos, rutas, supernodos y proyectos. Algunas propiedades son comunes a todos los nodos, como el nombre, la anotación y la información sobre herramientas, mientras que otras son específicas para determinados tipos de nodos. Otras propiedades hacen referencia a operaciones de rutas de alto nivel, como el comportamiento del Supernodo o el almacenamiento en caché. Se puede acceder a las propiedades a través de la interfaz de usuario estándar (por ejemplo, al abrir un cuadro de diálogo para editar opciones para un nodo) y se pueden utilizar también de varias otras formas.

- Las propiedades se pueden modificar a través de los procesos, como se describe en esta sección. Si desea obtener más información, consulte Sintaxis de las propiedades a continuación.
- Las propiedades de los nodos se pueden utilizar en los parámetros de Supernodo. [Si desea obtener más información, consulte el tema Uso de parámetros de Supernodo para acceder a las propiedades de los nodos en el capítulo 9 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)
- Asimismo, las propiedades de los nodos se pueden utilizar como parte de una opción de línea de comandos (mediante la marca -P) al iniciar IBM® SPSS® Modeler.

En el contexto de los procesos de SPSS Modeler, las propiedades de nodos y rutas se suelen llamar **parámetros de propiedades**. En este manual, se denominan propiedades de nodos y rutas.

Si desea obtener más información sobre el lenguaje de procesos, consulte el capítulo 3.

## Sintaxis de las propiedades

Las propiedades deben utilizar la siguiente estructura de sintaxis:

NOMBRE:TIPO.PROPIEDAD

donde NAME es el nombre de un nodo, y TYPE es su tipo (por ejemplo, `multiplotnode` o `derivenode`). Puede omitir NAME o bien TYPE, pero debe incluir al menos uno de ellos. PROPERTY es el nombre del parámetro del nodo o de la ruta a la que hace referencia su expresión. Por ejemplo, la siguiente sintaxis se utiliza para filtrar el campo *Edad* de los datos que se encuentran debajo:

```
set minodo:filternode.include.Edad = false
```

Para utilizar un valor personalizado para cualquiera de los parámetros (NAME, TYPE o PROPERTY), establezca en primer lugar el valor en una instrucción, como `set derive.new_name = mynewfield`. A partir de este punto, puede utilizar el valor, `mynewfield`, como el parámetro, precediéndolo del



símbolo `^`. Por ejemplo, puede establecer el tipo para el nodo Derivar, denominado anteriormente, mediante la siguiente sintaxis:

```
set ^minuevocampo.result_type = "Conditional"
```

Todos los nodos utilizados en IBM® SPSS® Modeler se pueden especificar en el parámetro TYPE de la sintaxis NAME:TYPE.PROPERTY.

### **Propiedades estructuradas**

Hay dos formas en las que los procesos utilizan propiedades estructuradas para mejorar la claridad durante el análisis:

- Otorgando estructura a los nombres de las propiedades para los nodos complejos, como Tipo, Filtro o Equilibrar.
- Proporcionando un formato para especificar varias propiedades a la vez.

#### **Estructuración para las interfaces complejas**

Los procesos para los nodos con tablas y otras interfaces complejas, como, por ejemplo, los nodos Tipo, Filtro o Equilibrar, deben seguir una estructura determinada para realizar el análisis correctamente. Estas propiedades estructuradas necesitan un nombre más complejo que el de un solo identificador. Por ejemplo, en un nodo Filtro, cada campo disponible (en la parte superior) se activa o desactiva. Para poder consultar esta información, el nodo Filtro almacena un elemento de información por campo (independientemente de que el campo sea verdadero o falso) y una sola propiedad denominada **campo** accede a estos múltiples elementos y los actualiza. Esta propiedad debe tener (o se le ha dado) el valor `true` o `false`. Supongamos que un nodo Filtro denominado `mynode` tiene (en la parte superior) un campo denominado `Edad`. Para desactivarlo, defina la propiedad `mynode.include.Age` en el valor `false`, como se indica a continuación:

```
set minodo.include.Edad = false
```

#### **Estructuración para definir varias propiedades**

Si hay muchos nodos, puede asignar más de una propiedad de nodo o ruta al mismo tiempo. Esto se denomina **comando de conjunto múltiple** o **bloque de conjuntos**. [Si desea obtener más información, consulte el tema Comando set en el capítulo 4 el p. 33.](#)

En algunos casos, una propiedad estructurada puede ser bastante compleja. El carácter barra invertida (`\`) se puede utilizar como un carácter de continuación de línea que permita alinear los argumentos para la claridad. A continuación se muestra un ejemplo:

```
mynode:sortnode.keys = [{ 'K' Descending} \
                        { 'Edad' Ascending}\
                        { 'Na' Descending }]
```

Otra ventaja de las propiedades estructuradas es la capacidad de definir varias propiedades en un nodo antes de que éste sea estable. Un conjunto múltiple define por defecto, todas las propiedades del bloque antes de realizar una acción basada en una configuración de propiedades individuales. Por ejemplo, al definir un nodo Archivo fijo, el uso de dos pasos para definir las propiedades del

campo daría lugar a errores porque el nodo no será constante hasta que las dos configuraciones sean válidas. La definición de las propiedades como un conjunto múltiple salva este problema al definir ambas propiedades antes de actualizar el modelo de datos.

### Abreviaturas

Las abreviaturas estándar se utilizan en la sintaxis para las propiedades de nodos. El aprendizaje de las abreviaturas le ayudará en la creación de procesos.

Abreviatura	Significado
abs	Valor absoluto
lon	Longitud
min	Mínimo
max	Máximo
correl	Correlation
covar	Covariance
núm	Número o numérico
pct	Porcentaje
transp	Transparencia
xval	Validación cruzada
var	Varianza o variable (en nodos de origen)

### Ejemplos de propiedades de nodos y rutas

Las propiedades de nodos y rutas se pueden utilizar de varias formas con IBM® SPSS® Modeler. Normalmente se utilizan como parte de un proceso, bien un **proceso independiente**, utilizado para automatizar rutas u operaciones o un **proceso de ruta**, utilizado para automatizar procesos en una sola ruta. Los parámetros de nodo se pueden especificar también utilizando las propiedades para los nodos del Supernodo. En el nivel más básico, las propiedades se pueden utilizar también como una opción de línea de comandos para iniciar SPSS Modeler. Si utiliza el argumento -p como parte de la invocación de la línea de comandos, podrá utilizar una propiedad de ruta para cambiar una configuración de la ruta.

s.max_size	Hace referencia a la propiedad max_size del nodo denominado s.
s:samplenode.max_size	Hace referencia a la propiedad max_size del nodo denominado s que debe ser un nodo Muestrear.
:samplenode.max_size	Hace referencia a la propiedad max_size del nodo Muestrear de la ruta actual (debe haber sólo un nodo Muestrear).
s:sample.max_size	Hace referencia a la propiedad max_size del nodo denominado s que debe ser un nodo Muestrear.
t.direction.Age	Hace referencia al papel del campo <i>Edad</i> del nodo Tipo t.
:.max_size	*** NO ES LEGAL *** Debe especificar el nombre o el tipo de nodo.

El ejemplo `s:sample.max_size` muestra que no es necesario deletrear los tipos de nodos al completo.

El ejemplo `t.direction.Age` muestra que algunos nombres de propiedades se pueden estructurar por sí mismos, en aquellos casos en que los atributos de un nodo sean más complejos que las simples propiedades individuales con valores individuales. Dichas propiedades se denominan **estructuradas** o **complejas**.

## Conceptos básicos de las propiedades de nodos

Cada tipo de nodo tiene su propio conjunto de propiedades legales y cada propiedad tiene un tipo. Este tipo puede ser general (número, marca o cadena) en cuyo caso, las configuraciones para la propiedad se forzarán para el tipo correcto. Surgirá un error en caso de que no se puedan forzar. También se puede dar el caso de que la referencia de la propiedad pueda especificar el rango de valores legales como `Discard`, `PairAndDiscard` e `IncludeAsText`, en cuyo caso se producirá un error si se utiliza otro valor. Las propiedades de marcas se deben leer o definir mediante los valores `true` y `false`. (Las variaciones que contengan `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `false`, `False`, `FALSE` o `0` también se reconocen al configurar los valores, pero pueden provocar errores al leer los valores de propiedad en algunos casos. El resto de valores se consideran verdaderos. El uso de `true` y `false` de forma consistente evitará confusiones). En las tablas de referencia de este manual, las propiedades estructuradas se indican como tales en la columna *Descripción de la propiedad* y se proporcionan los formatos de uso.

## Propiedades de nodos comunes

Existen varias propiedades que son comunes a todos los nodos (incluidos los Supernodos) en IBM® SPSS® Modeler.

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
<code>use_custom_name</code>	<i>marca</i>	
<code>name</code>	<i>cadena</i>	Propiedad de sólo lectura que lee el nombre (automático o personalizado) para un nodo del lienzo.
<code>custom_name</code>	<i>cadena</i>	Especifica un nombre personalizado para el nodo.
<code>tooltip</code>	<i>cadena</i>	
<code>annotation</code>	<i>cadena</i>	
<code>keywords</code>	<i>cadena</i>	Propiedad estructurada que especifica una lista de palabras clave asociadas al objeto (por ejemplo, ["Keyword1" "Keyword2"]).
<code>cache_enabled</code>	<i>marca</i>	
<code>node_type</code>	<code>source_supernode</code> <code>process_supernode</code> <code>terminal_supernode</code> todos los nombres de nodos tal como se especifican para los procesos	Propiedad de sólo lectura utilizada para hacer referencia a un nodo por tipo. Por ejemplo, en lugar de hacer referencia al nodo sólo por el nombre, como <code>real_income</code> , puede también especificar el tipo, como <code>userinputnode</code> o <code>filternode</code> .

Las propiedades específicas del Supernodo se tratan aparte como con los demás nodos. [Si desea obtener más información, consulte el tema Propiedades de Supernodos en el capítulo 22 el p. 319.](#)

## *Propiedades de ruta*

Los procesos pueden controlar una serie de propiedades de la ruta. Para hacer referencia a las propiedades de la ruta, debe utilizar una variable de ruta especial, indicada por el símbolo ^ antes de la ruta.

```
set ^stream.execute_method = Script
```

### **Ejemplo**

La propiedad nodes se utiliza para hacer referencia a los nodos de la ruta actual. El siguiente proceso de ruta muestra un ejemplo:

```
var listofnodes
var thenode
set listofnodes = ^stream.nodes

set ^stream.annotation = ^stream.annotation >< "\n\nEsta ruta se llama \"\" >< ^stream.name >
< \"\" y contiene / los siguientes nodos\n"

for thenode in listofnodes
set ^stream.annotation = ^stream.annotation >< "\n" >< ^thenode.node_type
endfor
```

El ejemplo anterior utiliza la propiedad de nodos para crear una lista con todos los nodos de la ruta y escribir dicha lista en las anotaciones de la ruta. La anotación generada tendrá el siguiente aspecto:

Esta ruta se llama "druglearn" y contiene los siguientes nodos

```
derivenode
neuralnetworknode
variablefilenode
typenode
c50node
filternode
```

Las propiedades de la ruta se describen en la tabla siguiente.

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
execute_method	Normal Script	
date_format	"DDMMAA" "MMDDAA" "AAMMDD" "AAAAMMDD" "AAAADD" DÍA MES "DD-MM-AA" "DD-MM-AAAA" "MM-DD-AA" "MM-DD-AAAA" "DD-MES-AA" "DD-MES-AAAA" "AAAA-MM-DD" "DD.MM.AA" "DD.MM.AAAA" "MM.DD.AA" "MM.DD.AAAA" "DD.MES.AA" "DD.MES.AAAA" "DD/MM/AA" "DD/MM/AAAA" "MM/DD/AA" "MM/DD/AAAA" "DD/MES/AA" "DD/MES/AAAA" MES AAAA t T AAAA ss SE AAAA	
date_baseline	número	
date_2digit_baseline	número	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
time_rollover	marca	
import_datetime_as_string	marca	
decimal_places	número	

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
decimal_symbol	Default Period Comma	
angles_in_radians	marca	
use_max_set_size	marca	
max_set_size	número	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	marca	Se utiliza para actualizar los nodos de fuente de forma automática al realizar la ejecución de la ruta.
script	cadena	
annotation	cadena	Ejemplo: set ^stream.annotation = "something interesting"
name	cadena	Ejemplo: set x = ^stream.name Nota: Ésta es una propiedad de solamente lectura. Si desea cambiar el nombre de una ruta, debe guardarla con un nombre diferente.
parameters		Se utiliza para actualizar los parámetros de ruta desde un proceso independiente. Ejemplo: set ^stream.parameters.height = 23
nodes		Consulte la información detallada que se muestra a continuación.
encoding	SystemDefault "UTF-8"	
stream_rewriting	booleano	
stream_rewriting_maximise_sql	booleano	
stream_rewriting_optimise_clem_execution	booleano	
stream_rewriting_optimise_syntax_execution	booleano	
enable_parallelism	booleano	
sql_generation	booleano	
database_caching	booleano	
sql_logging	booleano	
sql_generation_logging	booleano	
sql_log_native	booleano	
sql_log_prettyprint	booleano	
record_count_suppress_input	booleano	
record_count_feedback_interval	entero	

## Propiedades de proyecto

Una serie de propiedades está disponible para procesos con proyectos.

### Ejemplo

```
load project "C:/clemdata/DrugData.cpj"
set ^project.summary="Initial modeling work on the latest drug data."
set ^project.ordering=NameAddedType
execute_project
```

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
summary	<i>cadena</i>	El resumen del proyecto, normalmente una versión abreviada de la anotación.
title	<i>cadena</i>	El título del informe.
author	<i>cadena</i>	El autor del informe.
structure	Phase Class	Determina cómo se organiza el proyecto; por fase de minería de datos o por tipo de objeto (clase).
include_mode	IncludedItems ExcludedItems AllItems	Determina qué elementos incluir en el informe del proyecto.
select_mode	AllItems RecentItems OldItems	Determina (por edad) qué elementos incluir en el informe.
recent_item_limit	<i>entero</i>	Se utiliza cuando <b>select_mode</b> es <b>RecentItems</b> .
old_item_limit	<i>entero</i>	Se utiliza cuando <b>select_mode</b> es <b>OldItems</b> .
ordering	TypeNameAdded TypeAddedName NameAddedType AddedNameType	Determina el orden en que los elementos se enumeran en el informe.



# Propiedades de nodos de fuente

## Propiedades comunes de nodos de fuente

Las propiedades comunes a todos los nodos de fuente se enumeran a continuación, con información sobre nodos específicos en los temas siguientes.

### Ejemplo

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.use_custom_values.Age = True
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
#storage is read only
set :variablefilenode.check.Age = None
set :variablefilenode.values.Age = [1 100]
```

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
direction	Input Target Both None Partition Split Frequency RecordID	Propiedad adaptada para los papeles de los campos. Formato de uso: NODE.direction.FIELDNAME <i>Nota:</i> Se han desaprobado los valores In y Out. Deberán de ser compatibles en versiones posteriores.
type	Range Flag Set Typeless Discrete Ordered Set Default	Tipo de campo. Si se establece esta propiedad como <i>Default</i> , se borrará cualquier configuración de la propiedad values y si value_mode tiene el valor <i>Specify</i> , se restablecerá a <i>Read</i> . Si value_mode está definido como <i>Pass</i> o <i>Read</i> , la configuración de type no le afectará. Formato de uso: NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	Propiedad adaptada de solamente lectura para el tipo de almacenamiento de campos. Formato de uso: NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	Propiedad adaptada para la comprobación del rango y el tipo de campo. Formato de uso: NODE.check.FIELDNAME

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
values	[ <i>valor valor</i> ]	Para un campo continuo (rango), el primer valor es el mínimo y el último valor es el máximo. Para campos nominales (conjunto), especifique todos los valores. Para los campos marca, el primer valor representa <i>falso</i> y el último, <i>verdadero</i> . La configuración de esta propiedad establece de forma automática la propiedad <i>value_mode</i> en <i>Specify</i> . Formato de uso: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Determina la forma en que se han establecido los valores para un campo en el siguiente paso de datos. Formato de uso: NODE.value_mode.FIELDNAME Tenga en cuenta que no puede establecer esta propiedad directamente en <i>Specify</i> . Para utilizar valores específicos, establezca la propiedad <i>values</i> .
default_value_mode	Read Pass	Especifica el método por defecto para configurar los valores de todos los campos. Formato de uso: NODE.default_value_mode Ejemplo: set mynode.default_value_mode = Pass Esta configuración puede anularse para determinados campos mediante la propiedad <i>value_mode</i> .
extend_values	<i>marca</i>	Se aplica cuando <i>value_mode</i> se ha establecido como <i>Read</i> . Establézcala en <i>T</i> para añadir nuevos valores de lectura a los valores existentes del campo. Establézcala en <i>F</i> para descartar los valores existentes y favorecer a los nuevos valores de lectura. Formato de uso: NODE.extend_values.FIELDNAME
value_labels	<i>cadena</i>	Se utiliza para especificar una etiqueta de valor. Ejemplo: set :varfilenode.value_labels.Age = [{3 three}{5 five}] Tenga en cuenta que estos valores se deben especificar primero.
enable_missing	<i>marca</i>	Cuando está definida como <i>T</i> , activa el seguimiento de los valores perdidos para el campo. Formato de uso: NODE.enable_missing.FIELDNAME
missing_values	[ <i>valor valor ...</i> ]	Especifica los valores de datos que denotan los datos perdidos. Formato de uso: NODE.missing_values.FIELDNAME
range_missing	<i>marca</i>	Cuando esta propiedad se establece como <i>T</i> , especifica si se define un rango de valores ausentes (en blanco) para un campo. Formato de uso: NODE.range_missing.FIELDNAME
missing_lower	<i>cadena</i>	Si <i>range_missing</i> es verdadero, especifica el límite inferior del rango de valores perdidos. Formato de uso: NODE.missing_lower.FIELDNAME

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
missing_upper	cadena	Si range_missing es verdadero, especifica el límite superior del rango de valores perdidos. Formato de uso: NODE.missing_upper.FIELDNAME
null_missing	marca	Cuando esta propiedad está definida como <i>T</i> , los valores nulos (valores no definidos que se muestran como \$null\$ en el software) se consideran valores perdidos. Formato de uso: NODE.null_missing.FIELDNAME
whitespace_missing	marca	Cuando esta propiedad está definida como <i>T</i> , los valores que solamente contienen un espacio en blanco (espacios, tabulaciones y líneas nuevas) se consideran valores perdidos. Formato de uso: NODE.whitespace_missing.FIELDNAME
description	cadena	Se utiliza para especificar la descripción o etiqueta de un campo.
default_include	marca	Propiedad adaptada para especificar si el comportamiento por defecto es para pasar o filtrar los campos: NODE.default_include Ejemplo: set mynode:filternode.default_include = false
include	marca	Propiedad adaptada que se utiliza para determinar si los campos individuales se han incluido o se han filtrado: NODE.include.FIELDNAME. Ejemplo: set mynode:filternode.include.Age = true
new_name	cadena	Ejemplo: set mynode:filternode.new_name.'Age' = "years"

## Propiedades de cognosimportnode



El nodo de origen de IBM Cognos BI importa datos desde las bases de datos de Cognos BI. Si desea obtener más información, consulte el tema [Importación de datos de Cognos](#) en el capítulo 2 en *Nodos de origen, proceso y resultado de IBM SPSS Modeler 15*.

### Ejemplo

```
create cognosimportnode
set :cognosimportnode.cognos_connection = {'http://mycogsvr1:9300/p2pd/servlet/dispatch', true, "", ""}
set :cognosimportnode.cognos_package_name = '/Public Folders/GOSALES'
set :cognosimportnode.cognos_items = {"[GreatOutdoors].[BRANCH].[BRANCH_CODE]"}
```

"[GreatOutdoors].[BRANCH].[COUNTRY\_CODE]"]

propiedades de cognosimportnode	Tipo de datos	Descripción de la propiedad
mode	Data Report	Especifica si se importarán los datos de Cognos BI (por defecto) o informes.
cognos_connection	{ "campo", "campo", ... , "campo" }	Una propiedad de la lista que contiene los detalles de conexión para el servidor de Cognos. El formato es: <pre>{ "Cognos_server_URL", login_mode, "espacio de nombre", "nombre de usuario", "contraseña" }</pre> donde: <i>Cognos_server_URL</i> es la URL del servidor de Cognos que contiene la fuente <i>login_mode</i> indica si se utiliza el inicio de sesión anónimo, y si es true o false; si se establece como true, los campos siguientes deberían establecerse como "" <i>espacio de nombre</i> especifica el proveedor de autenticación de seguridad utilizado para registrarse en el servidor <i>nombre de usuario</i> y <i>contraseña</i> son aquellos utilizados para registrarse en el servidor de Cognos
cognos_package_name	cadena	La ruta y el nombre del paquete de Cognos desde el que está importando objetos de datos, por ejemplo: /Public Folders/GOSALES Nota: solo es válida la barra diagonal.
cognos_items	{ "campo", "campo", ... , "campo" }	El nombre de uno o más objetos de datos que van a importarse. El formato de <i>campo</i> es [ <i>espacio de nombre</i> ].[ <i>asunto de consulta</i> ].[ <i>elemento de consulta</i> ] Ejemplo: <pre>set :cognosimport.cognos_items = { "[Inventory (query)].[Inventory].[Opening inventory]", "[Inventory (query)].[Inventory].[Quantity shipped]", "[Inventory (query)].[Inventory].[Additions]", "[Inventory (query)].[Inventory].[Unit cost]", "[Inventory (query)].[Inventory].[Closing inventory]", "[Inventory (query)].[Inventory].[Average unit cost]" }</pre>
cognos_filters	campo	El nombre de uno o más filtros que van a aplicarse antes de importar datos. Ejemplo: <pre>set :cognosimport.cognos_filters = { "[Inventory].[Filter].[MyFilter]" }</pre>
cognos_data_parameters	lista	Valores de parámetros de solicitud de datos. Los pares nombre-valor están entre llaves, los pares múltiples están separados por comas y toda la cadena está entre corchetes. Formato: <pre>[ { "param1", "valor" }, ..., { "paramN", "valor" } ]</pre> Ejemplo: <pre>set :cognosimport.cognos_data_parameters = [ { "SexValue", "F" }, { "a", "1" }, { "b", "1" } ]</pre>

propiedades de cognosimportn-ode	Tipo de datos	Descripción de la propiedad
cognos_report_location	campo	La ruta de Cognos de una carpeta o paquete de la que importar informes, por ejemplo: /Public Folders/GOSALES Nota: solo es válida la barra diagonal.
cognos_report_name	campo	La ruta y nombre dentro de la ubicación del informe de un informe para importar, por ejemplo: set :cognosimport.cognos_report_name = /Jimmy/Package/Drug4nPackage/3columns
cognos_report_parameters	lista	Valores de parámetros de informe. Los pares nombre-valor están entre llaves, los pares múltiples están separados por comas y toda la cadena está entre corchetes. Formato: [{"param1", "valor"}, ..., {"paramN", "valor"}] Ejemplo: set :cognosimport.cognos_report_parameters = [{"SexValue", "F"}, {"a", "1"}, {"b", "1"}]

## Propiedades de databasenode



El nodo Base de datos se puede utilizar para importar datos desde otros paquetes mediante ODBC (del inglés, Open Database Connectivity), incluidos Microsoft SQL Server, DB2, Oracle, etc. [Si desea obtener más información, consulte el tema Nodo de fuente Base de datos en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create databasenode
set :databasenode.mode = Table
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = "Drug4n_db"
set :databasenode.username = "spss"
set :databasenode.password = "spss"
var test_e
set test_e = :databasenode.epassword
set :databasenode.tablename = ".Drug4n"
```

propiedades de databasenode	Tipo de datos	Descripción de la propiedad
mode	Table Query	Especifique <i>Table</i> para conectarse a una tabla de base de datos mediante los controles del cuadro de diálogo, o <i>Query</i> para realizar una consulta a la base de datos seleccionada mediante SQL.
datasource	cadena	Nombre de la base de datos (consulte la siguiente nota).
username	cadena	Detalles de conexión de la base de datos (consulte la siguiente nota).

propiedades de databasenode	Tipo de datos	Descripción de la propiedad
password	<i>cadena</i>	
epassword	<i>cadena</i>	Especifica una contraseña codificada como una alternativa a codificar una contraseña en un proceso. <a href="#">Si desea obtener más información, consulte el tema Generación de una contraseña codificada en el capítulo 5 el p. 61.</a> Esta propiedad es de solamente lectura durante la ejecución.
tablename	<i>cadena</i>	Nombre de la tabla a la que se desea tener acceso.
strip_spaces	None Left Right Both	Opciones para descartar los espacios precedentes y posteriores en las cadenas.
use_quotes	AsNeeded Always Never	Especifica si los nombres de columna y tabla aparecen entre comillas cuando las consultas se envían a la base de datos (en el caso, por ejemplo, de que contengan espacios o signos de puntuación).
query	<i>cadena</i>	Especifica el código SQL para la consulta que desea enviar.

*Nota:* Si el nombre de la base de datos (en la propiedad `datasource`) contiene espacios, entonces en vez de las propiedades individuales para `datasource`, `username` y `password`, utilice una única propiedad de fuente de datos en el siguiente formato:

propiedades de databasenode	Tipo de datos	Descripción de la propiedad
<code>datasource</code>	<i>cadena</i>	Formato: {database_name,username,password[,true   false]} El último parámetro se usa con contraseñas cifradas. Si se define como <code>true</code> , la contraseña se cifrará antes de usarse.

### **Ejemplo**

```
create databasenode
set :databasenode.mode = Table
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = {"ORA 10gR2", user1, mypsw, true}
var test_e
set test_e = :databasenode.epassword
set :databasenode.tablename = "Drug4n"
```

Utilice este formato también si está cambiando la fuente de datos, sin embargo, si tan solamente desea cambiar el nombre de usuario o contraseña, puede usar las propiedades `username` o `password`.

## Propiedades de datacollectionimportnode



El nodo de importación de datos IBM® SPSS® Data Collection importa datos de encuesta basados en el modelo de datos de Data Collection que utilizan los productos de investigación de mercados de IBM Corp.. Se debe instalar la biblioteca de datos de Data Collection para utilizar este nodo. [Si desea obtener más información, consulte el tema Nodo Data Collection en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create datacollectionimportnode
set :datacollectionimportnode.metadata_name="mrQvDsc"
set :datacollectionimportnode.metadata_file="C:/Program Files/IBM/SPSSDataCollection/DDL/Data/
Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.casedata_name="mrQvDsc"
set :datacollectionimportnode.casedata_source_type=File
set :datacollectionimportnode.casedata_file="C:/Program Files/IBM/SPSSDataCollection/DDL/Data/
Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.import_system_variables = Common
set :datacollectionimportnode.import_multi_response = MultipleFlags
```

propiedades de datacollection-importnode	Tipo de datos	Descripción de la propiedad
metadata_name	<i>cadena</i>	El nombre del MDSC. El valor especial DimensionsMDD indica que se debería utilizar el documento de metadatos de Data Collection estándar. Otro posibles valores podrían ser: mrADODsc mrI2dDsc mrLogDsc mrQdiDrDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC El valor especial none indica que no existe ningún MDSC.
metadata_file	<i>cadena</i>	Nombre del archivo en el que se almacenan los metadatos.
casedata_name	<i>cadena</i>	El nombre del CDSC. Entre los posibles valores se encuentran: mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmlDsc El valor especial none indica que no existe ningún CDSC.

propiedades de datacollection-importnode	Tipo de datos	Descripción de la propiedad
casedata_source_type	Unknown File Folder UDL DSN	Indica el tipo de fuente del CDSC.
casedata_file	cadena	Cuando <i>casedata_source_type</i> es <i>File</i> , especifica el archivo que contiene los datos de casos.
casedata_folder	cadena	Cuando <i>casedata_source_type</i> es <i>Folder</i> , especifica la carpeta que contiene los datos de casos.
casedata_udl_string	cadena	Cuando <i>casedata_source_type</i> es <i>UDL</i> , especifica la cadena de conexión OLD-DB de la fuente de datos que contiene los datos de casos.
casedata_dsn_string	cadena	Cuando <i>casedata_source_type</i> es <i>DSN</i> , especifica la cadena de conexión ODBC de la fuente de datos.
casedata_project	cadena	Al leer datos de casos de una base de datos de Data Collection, puede escribir el nombre del proyecto. Para el resto de tipos de datos de casos, esta configuración se deberá dejar en blanco.
version_import_mode	All Latest Specify	Define el modo en que deben tratarse las versiones.
specific_version	cadena	Cuando <i>version_import_mode</i> es <i>Specify</i> , define la versión de los datos de casos que se van a importar.
use_language	cadena	Determina si deben usarse las etiquetas de un idioma concreto.
language	cadena	Si <i>use_language</i> es verdadero, define el código de idioma que se va a usar en la importación. Este código de idioma debe incluirse entre aquellos disponibles en los datos de casos.
use_context	cadena	Determina si se debe importar un contexto específico. Los contextos se utilizan para modificar la descripción asociada con las respuestas.
context	cadena	Cuando <i>use_context</i> es verdadero, define el contexto de la importación. Este contexto debe encontrarse entre aquellos disponibles en los datos de casos.
use_label_type	cadena	Determina si se debe importar un tipo de etiqueta específico.
label_type	cadena	Cuando <i>use_label_type</i> es verdadero, define el tipo de etiqueta de la importación. Este tipo de etiqueta debe encontrarse entre aquellos disponibles en los datos de casos.



propiedades de datacollection-importnode	Tipo de datos	Descripción de la propiedad
user_id	cadena	En el caso de las bases de datos que requieren un inicio de sesión explícito, puede proporcionar un ID de usuario y una contraseña para acceder a la fuente de datos.
password	cadena	
import_system_variables	Common None All	Especifica las variables del sistema que se importan.
import_codes_variables	marca	
import_sourcefile_variables	marca	
import_multi_response	MultipleFlags Single	

## Propiedades de excelimportnode



El nodo de importación Excel importa datos de cualquier versión de Microsoft Excel. No es necesaria una fuente de datos ODBC. [Si desea obtener más información, consulte el tema Nodo de fuente de Excel en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
#Para usar un rango con nombre:
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.use_named_range = True
set :excelimportnode.named_range = "DRUG"
set :excelimportnode.read_field_names = True
```

```
#Para usar un rango explícito:
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.worksheet_mode = Name
set :excelimportnode.worksheet_name = "Drug"
set :excelimportnode.explicit_range_start = A1
set :excelimportnode.explicit_range_end = F300
```

propiedades de excelimportnode	Tipo de datos	Descripción de la propiedad
excel_file_type	Excel2003 Excel2007	
full_filename	cadena	El nombre completo del archivo, incluyendo la ruta.

propiedades de <b>excelimportnode</b>	Tipo de datos	Descripción de la propiedad
use_named_range	<i>Booleana</i>	Si usar un rango con nombre. Si es verdadero, la propiedad <i>named_range</i> se utiliza para especificar el rango de lectura y se ignoran el resto de configuraciones de rango de datos y hojas de trabajo.
named_range	<i>cadena</i>	
worksheet_mode	Index Name	Determina si la hoja de trabajo se define por el índice o por el nombre.
worksheet_index	<i>entero</i>	Índice de la hoja de trabajo que se va a leer, siendo 0 la primera hoja de trabajo, 1 la segunda, etc.
worksheet_name	<i>cadena</i>	Nombre de la hoja de trabajo que se va a leer.
data_range_mode	FirstNonBlank ExplicitRange	Especifica cómo debe establecerse el rango.
blank_rows	StopReading ReturnBlankRows	Cuando <i>data_range_mode</i> es <i>FirstNonBlank</i> , especifica cómo deben tratarse las filas en blanco.
explicit_range_start	<i>cadena</i>	Cuando <i>data_range_mode</i> es <i>ExplicitRange</i> , especifica el punto de partida del rango de lectura.
explicit_range_end	<i>cadena</i>	
read_field_names	<i>Booleana</i>	Determina si la primera fila del rango concreto debería usarse como nombres de campo (columna).

## Propiedades de *evimportnode*



El nodo Enterprise View crea una conexión a IBM SPSS Collaboration and Deployment Services Repository, lo que permite leer los datos de Enterprise View en una ruta y empaquetar un modelo en un escenario al que otros usuarios pueden acceder desde el repositorio. [Si desea obtener más información, consulte el tema \*Nodo Enterprise View\* en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create evimportnode
set :evimportnode.connection = ['Training data', '/Application views/Marketing', 'LATEST', 'Analytic',
'/Data Providers/Marketing']
```

```
set :evimportnode.tablename = "cust1"
```

propiedades de evimportnode	Tipo de datos	Descripción de la propiedad
connection	lista	Propiedad estructura: lista de parámetros que componen una conexión de Enterprise View. Formato de uso: evimportnode.connection = [description,app_view_path, app_view_version_label,environment,DPD_path]
tablename	cadena	Nombre de una tabla de la vista de aplicación.

## Propiedades de fixedfilenode



El nodo Archivo fijo importa datos desde archivos de texto de campo fijo; esto es, archivos cuyos campos no están delimitados pero empiezan en la misma posición y tienen una longitud fija. Los datos heredados o generados por la máquina se suelen almacenar en formato de campo fijo. [Si desea obtener más información, consulte el tema Nodo Archivo fijo en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create fixedfilenode
set :fixedfilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :fixedfilenode.record_len = 32
set :fixedfilenode.skip_header = 1
set :fixedfilenode.fields = [{'Age' 1 3} {'Sex' 5 7} {'BP' 9 10} {'Cholesterol' 12 22} {'Na' 24 25} {'K' 27 27} {'Drug' 29 32}]
set :fixedfilenode.decimal_symbol = Period
set :fixedfilenode.lines_to_scan = 30
```

propiedades de fixedfilenode	Tipo de datos	Descripción de la propiedad
record_len	número	Especifica el número de caracteres de cada registro.
line_oriented	marca	Omite el carácter de nueva línea al final de cada registro.
decimal_symbol	Default Comma Period	Tipo de separador decimal utilizado en la fuente de datos. Ejemplo: set :fixedfilenode.decimal_symbol = Period
skip_header	número	Especifica el número de líneas que se ignorarán al principio del primer registro. Esto resulta útil para ignorar las cabeceras de columna.
auto_recognize_datetime	marca	Especifica si las fechas o las horas se identifican automáticamente en los datos de fuente.
lines_to_scan	número	Ejemplo: set :fixedfilenode.lines_to_scan = 50.

propiedades de fixedfilenode	Tipo de datos	Descripción de la propiedad
fields	lista	Propiedad estructurada. Formato de uso: fixedfilenode.fields = [{field start length} {field start length}]
full_filename	cadena	Nombre completo del archivo que se va a leer, incluido el directorio.
strip_spaces	None Left Right Both	Descarta los espacios precedentes y posteriores en las cadenas de importación.
invalid_char_mode	Discard Replace	Elimina los caracteres no válidos (nulo, 0 o cualquier carácter que no exista en la codificación actual) de la entrada de datos o sustituye los caracteres no válidos con el símbolo especificado de un carácter.
invalid_char_replacement	cadena	
use_custom_values	marca	Propiedad adaptada en la siguiente forma: set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	Propiedad adaptada en la siguiente forma: set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	Aplicable solamente si ha especificado un almacenamiento personalizado. Ejemplo: set:varfilenode.custom Propiedad adaptada en la siguiente forma: set :varfilenode.custom_date_format.'LaunchDate' = "DDMMYY"

propiedades de fixedfilenode	Tipo de datos	Descripción de la propiedad
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Aplicable solamente si ha especificado un almacenamiento personalizado. Propiedad adaptada en la siguiente forma: set :varfilenode.custom_time_format. 'Initialize' = "HHMM"
custom_decimal_symbol	<i>campo</i>	Aplicable solamente si ha especificado un almacenamiento personalizado. Propiedad adaptada en la siguiente forma: set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	Especifica el método de codificación de textos.

## Propiedades de sasimportnode



El nodo importar SAS importa datos SAS a IBM® SPSS® Modeler. [Si desea obtener más información, consulte el tema Nodo de fuente SAS en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create sasimportnode
set :sasimportnode.format = Windows
set :sasimportnode.full_filename = "C:/data/retail.sas7bdat"
set :sasimportnode.member_name = "Test"
set :sasimportnode.read_formats = False
set :sasimportnode.full_format_filename = "Test"
set :sasimportnode.import_names = True
```

propiedades de sasimportnode	Tipo de datos	Descripción de la propiedad
format	Windows UNIX Transport SAS7 SAS8 SAS9	Formato del archivo que se va a importar.
full_filename	<i>cadena</i>	El nombre completo del archivo introducido, incluyendo su ruta.

propiedades de sasimportnode	Tipo de datos	Descripción de la propiedad
member_name	cadena	Especifique el miembro para importar desde el archivo de transporte SAS especificado.
read_formats	marca	Lee formatos de datos (como etiquetas de variables) del archivo de formatos especificado.
full_format_filename	cadena	
import_names	NamesAndLabels LabelsasNames	Especifica el método para la asignación de nombres y etiquetas de variables en la importación.

## Propiedades de statisticsimportnode



El nodo Archivo de IBM® SPSS® Statistics lee los datos desde un formato de archivo *.sav* que utiliza SPSS Statistics y archivos caché guardados en IBM® SPSS® Modeler, que también puede utilizar el mismo formato. [Si desea obtener más información, consulte el tema Nodo de archivos estadísticos en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

Las propiedades de este nodo están descritas en [Propiedades de statisticsimportnode](#) el p. 315.

## Propiedades de userinputnode



El nodo Datos Usuario proporciona una manera fácil de crear datos sintéticos, ya sea partiendo de cero o modificando los datos existentes. Esto resulta útil, por ejemplo, cuando desee crear un conjunto de datos de comprobación para el modelado. [Si desea obtener más información, consulte el tema Nodo Datos Usuario en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create userinputnode
set :userinputnode.data.test1 = "2, 4, 8"
set :userinputnode.names = [pruebat1 pruebat2]
set :userinputnode.custom_storage.test1 = Integer
set :userinputnode.data_mode = "Ordered"
```

propiedades de userinputnode	Tipo de datos	Descripción de la propiedad
data		Propiedad adaptada de la siguiente forma: set :userinputnode.data.Age = "1 2 3 4" La cadena también puede especificar valores de tamaño del paso, altos y bajos, separados por comas. Ejemplo: set :userinputnode.data.Age = "10, 70, 5" Los datos de cada campo pueden tener longitudes diferentes pero deben ser consistentes con el almacenamiento del campo. La configuración de los valores para un campo que no está presente, supone la creación de dicho campo. También se pueden establecer los valores

propiedades de userinputnode	Tipo de datos	Descripción de la propiedad
		para un campo en una cadena vacía (" ") para eliminar el campo especificado.
names		Propiedad estructurada que establece o devuelve una lista de nombres de campos generados por el nodo. Ejemplo: ['Field1' 'Field2']
custom_storage	Unknown String Integer Real Time Date Timestamp	Propiedad adaptada que establece o devuelve el almacenamiento para un campo. Ejemplo: set :userinputnode.custom_storage.'Age' = "Real"
data_mode	Combined Ordered	Si se especifica Combined, los registros se generarán para cada combinación de valores del conjunto y valores mínimos y máximos. El número de registros generados será igual al producto del número de valores de cada campo. Si se especifica Ordered, se tomará un valor de cada columna para cada registro con el fin de generar una fila de datos. El número de registros generados será igual al número más grande de valores asociados a un campo. Los campos que tengan menos valores de datos se rellenarán con valores nulos.
values		<i>Esta propiedad se ha desaprobadado en favor de userinputnode.data y no debe volver a utilizarse.</i>

## Propiedades de variablefilenode



El nodo Archivo variable lee datos desde los archivos de texto de campo libre—campos cuyos registros contienen un número constante de campos, pero un número variado de caracteres. Este nodo resulta también útil para los archivos con texto de cabecera de longitud fija y determinados tipos de anotaciones. [Si desea obtener más información, consulte el tema Nodo Archivo variable en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.read_field_names = True
set :variablefilenode.delimit_other = True
set :variablefilenode.other = ','
set :variablefilenode.quotes_1 = Discard
set :variablefilenode.decimal_symbol = Comma
set :variablefilenode.invalid_char_mode = "Replace"
set :variablefilenode.invalid_char_replacement = "|"
set :variablefilenode.use_custom_values.Age = True
```

```
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
set :variablefilenode.values.Age = [1 100]
```

propiedades de variablefilenode	Tipo de datos	Descripción de la propiedad
skip_header	número	Especifica el número de caracteres que se ignorarán al principio del primer registro. Formato de uso: variablefilenode:skip_header = 3
num_fields_auto	marca	Determina el número de campos de cada registro de forma automática. Los registros deben terminar con un carácter de nueva línea. Formato de uso: variablefilenode:num_fields_auto
num_fields	número	Especifica manualmente el número de campos de cada registro.
delimit_space	marca	Especifica el carácter utilizado para delimitar los límites de los campos del archivo.
delimit_tab	marca	
delimit_new_line	marca	
delimit_non_printing	marca	
delimit_comma	marca	En aquellos casos en los que la coma sea el delimitador del campo y el separador decimal para rutas, establezca delimit_other como true y especifique una coma como delimitador mediante la propiedad other.
delimit_other	marca	Permite especificar un delimitador personalizado mediante la propiedad other.
other	cadena	Especifica el delimitador utilizado cuando delimit_other es true.
decimal_symbol	Default Comma Period	Especifica el separador decimal utilizado en la fuente de datos.
multi_blank	marca	Trata varios caracteres delimitadores vacíos adyacentes como un único delimitador.
read_field_names	marca	Trata la primera fila del archivo de datos como etiquetas para la columna.
strip_spaces	None Left Right Both	Descarta los espacios precedentes y posteriores en las cadenas de importación.
invalid_char_mode	Discard Replace	Elimina los caracteres no válidos (nulo, 0 o cualquier carácter que no exista en la codificación actual) de la entrada de datos o sustituye los caracteres no válidos con el símbolo especificado de un carácter.
invalid_char_replacement	cadena	
lines_to_scan	número	Especifica cuántas líneas se van a explorar para los tipos de datos especificados.



propiedades de variablefilenode	Tipo de datos	Descripción de la propiedad
auto_recognize_datetime	<i>marca</i>	Especifica si las fechas o las horas se identifican automáticamente en los datos de fuente.
quotes_1	Discard PairAndDiscard IncludeAsText	Especifica cómo se tratarán las comillas simples en la importación.
quotes_2	Discard PairAndDiscard IncludeAsText	Especifica cómo se tratarán las comillas dobles en la importación.
full_filename	<i>cadena</i>	Nombre completo del archivo que se va a leer, incluido el directorio.
use_custom_values	<i>marca</i>	Propiedad adaptada en la siguiente forma: set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	Propiedad adaptada en la siguiente forma: set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	Aplicable solamente si ha especificado un almacenamiento personalizado. Ejemplo: set:varfilenode.custom Propiedad adaptada en la siguiente forma: set :varfilenode.custom_date_format. 'LaunchDate' = "DDMMYY"

propiedades de variablefilenode	Tipo de datos	Descripción de la propiedad
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Aplicable solamente si ha especificado un almacenamiento personalizado. Propiedad adaptada en la siguiente forma: set :varfilenode.custom_time_format. 'Initialize' = "HHMM"
custom_decimal_symbol	<i>campo</i>	Aplicable solamente si ha especificado un almacenamiento personalizado. Propiedad adaptada en la siguiente forma: set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	Especifica el método de codificación de textos.

## Propiedades de xmlimportnode



El nodo de origen XML importa datos en formato XML en la ruta. Puede importar un único archivo o todos los archivos en un directorio. Puede especificar opcionalmente un archivo de esquema para leer la estructura XML. [Si desea obtener más información, consulte el tema Nodo Fuente XML en el capítulo 2 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create xmlimportnode
set :xmlimportnode.full_filename = "c:\import\books.xml"
set :xmlimportnode.records = "/author/name"
```

propiedades de xmlimportnode	Tipo de datos	Descripción de la propiedad
read	single directory	Lee un único archivo de datos (predeterminado) o todos los archivos XML de un directorio.
recurse	<i>marca</i>	Especifica si leer además archivos XML de todos los subdirectorios del directorio especificado.
full_filename	<i>cadena</i>	(obligatorio) Ruta completa y nombre de archivo del archivo XML a importar (si read = single).
directory_name	<i>cadena</i>	(obligatorio) Ruta completa y nombre del directorio desde el que importar los archivos XML (si read = directory).

<b>propiedades de xmlimportnode</b>	<b>Tipo de datos</b>	<b>Descripción de la propiedad</b>
full_schema_filename	<i>cadena</i>	Ruta completa y nombre de archivo del archivo XSD o DTD desde el que leer la estructura XML. Si omite este parámetro, se leerá la estructura desde el archivo de fuente XML.
records	<i>cadena</i>	Expresión XPath (p.ej. /author/name) para definir el límite del registro. Cada vez que este elemento se encuentra en el archivo de fuente se crea un nuevo registro.
mode	<b>read specify</b>	Lee todos los datos (predeterminado) o especifica qué elementos leer.
fields		Lista de elementos (elementos y atributos) para importar. Cada elemento de la lista es una expresión XPath.

# Propiedades de nodos de operaciones con registros

## Propiedades de appendnode



El nodo Añadir concatena conjuntos de registros. Es útil para combinar conjuntos de datos con estructuras parecidas, pero con datos diferentes. [Si desea obtener más información, consulte el tema Nodo Añadir en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create appendnode
set :appendnode.match_by = Name
set :appendnode.match_case = True
set :appendnode.include_fields_from = All
set :appendnode.create_tag_field = True
set :appendnode.tag_field_name = "Agregar_marca"
```

propiedades de appendnode	Tipo de datos	Descripción de la propiedad
match_by	Position Name	Se pueden añadir conjuntos de datos basándose en la posición que tienen los campos en el origen de datos principal o el nombre de los campos en los conjuntos de datos de entrada.
match_case	marca	Activa la coincidencia de mayúsculas y minúsculas al hacer coincidir nombres de campos.
include_fields_from	Main All	
create_tag_field	marca	
tag_field_name	cadena	

## Propiedades de aggregatenode



El nodo Agregar reemplaza una secuencia de registros de entrada con registros de salida agregados y resumidos. [Si desea obtener más información, consulte el tema Nodo Agregar en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create aggregatenode
connect :databasenode to :aggregatenode
set :aggregatenode.contiguous = True
```

```

set :aggregatenode.keys = ['Medicamento']
set :aggregatenode.aggregates.Edad = [Suma Media]
set :aggregatenode.inc_record_count = True
set :aggregatenode.count_field = "indice"
set :aggregatenode.extension = "Agregado_"
set :aggregatenode.add_as = Prefix

```

propiedades de <b>aggregatenode</b>	Tipo de datos	Descripción de la propiedad
keys	<i>[campo campo ... campo]</i>	Enumera los campos que se pueden usar como claves en la agregación. Por ejemplo, si <b>Sex</b> y <b>Region</b> son los campos clave, cada combinación única de M y F con las regiones N y S (cuatro combinaciones exclusivas) tendrá un registro agregado.
contiguous	<i>marca</i>	Seleccione esta opción si sabe que todos los registros con los mismos valores clave se agrupan en la entrada (por ejemplo, si la entrada se clasifica en los campos clave). Con ello puede mejorar el rendimiento.
aggregates		Enumera los campos numéricos cuyos valores se añadirán, así como los modos de agregación elegidos. Ejemplo: set :aggregatenode. aggregates.Age = [Sum Mean Min Max SDev Median Count Variance Firstquartile Thirdquartile], donde los métodos de agregación deseados se incluyen en la lista.
extension	<i>cadena</i>	Especifica un prefijo o sufijo para campos agregados duplicados (consulte un ejemplo a continuación).
add_as	Suffix Prefix	
inc_record_count	<i>marca</i>	Crea un campo adicional que indica la cantidad de registros de entrada agregados para conformar cada registro agregado.
count_field	<i>cadena</i>	Especifica el nombre del campo de recuento de registros.

## Propiedades de **balancenode**



El nodo Equilibrar corrige los desequilibrios de un conjunto de datos para que cumpla una condición determinada. La directiva de equilibrado ajusta la proporción de registros si una condición es verdadera por el factor determinado. [Si desea obtener más información, consulte el tema Nodo Equilibrar en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```

create balancenode
set :balancenode.training_data_only = true
set :balancenode.directives = \
  [{1.3 "Edad > 60"}{1.5 "Na > 0.5"}]

```

propiedades de balancenode	Tipo de datos	Descripción de la propiedad
directives		Propiedad estructurada para equilibrar la proporción de los valores de campos basados en un número determinado (consulte el ejemplo a continuación).
training_data_only	marca	Especifica que sólo se deben equilibrar los datos de entrenamiento. Si no se incluye ningún campo de partición en la ruta, se ignorará esta opción.

### Ejemplo

```
create balancenode
set :balancenode.directives = \
  [{1.3 "Edad > 60"}{1.5 "Na > 0.5"}]
```

Esta propiedad de nodo utiliza el siguiente formato:

[{ *número cadena* } \ \ { *número cadena* } \ ... { *número cadena* }].

*Nota:* Si las cadenas (que utilizan comillas dobles) están incrustadas en la expresión, han de estar precedidas del carácter de escape "\". El carácter "\" es también el carácter de continuación de línea, que le permite alinear los argumentos para la claridad.

## Propiedades de distinctnode



El nodo Distinguir se puede usar para eliminar registros duplicados pasando el primero de los registros distintos a la ruta de datos o descartando el primer registro y pasando cualquier duplicado a la ruta de datos en su lugar. [Si desea obtener más información, consulte el tema Nodo Distinguir en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create distinctnode
set :distinctnode.mode = Include
set :distinctnode.fields = ['Edad' 'Sexo']
set :distinctnode.keys_pre_sorted = True
```

propiedades de distinctnode	Tipo de datos	Descripción de la propiedad
mode	Include Discard	Se puede incluir el primer nodo distinto en la ruta o descartar el primer nodo distinto y pasar en su lugar todos los registros duplicados a la ruta de datos.
fields	[ <i>campo campo campo</i> ]	Enumera los campos utilizados para determinar si los registros son idénticos.
low_distinct_key_count	<i>marca</i>	Especifica que sólo tiene un pequeño número de registros y/o un pequeño número de valores únicos del campo(s) clave.
keys_pre_sorted	<i>marca</i>	Especifica que todos los registros con los mismos valores clave se agrupan en la entrada.

## Propiedades de mergenode



El nodo Fundir toma varios registros de entrada y crea un registro de salida único que contiene todos o algunos de los campos de entrada. Es útil para fusionar datos desde diferentes orígenes, como datos de clientes internos y datos demográficos adquiridos. [Si desea obtener más información, consulte el tema Nodo Fundir en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create mergenode
connect customerdata to :mergenode
connect salesdata to :mergenode
set :mergenode.method = Keys
set :mergenode.key_fields = ['id']
set :mergenode.common_keys = true
set :mergenode.join = PartialOuter
set :mergenode.outer_join_tag.2 = true
set :mergenode.outer_join_tag.4 = true
set :mergenode.single_large_input = true
set :mergenode.single_large_input_tag = '2'
set :mergenode.use_existing_sort_keys = true
```

```
set :mergenode.existing_sort_keys = [{'id' Ascending}]
```

propiedades de mergenode	Tipo de datos	Descripción de la propiedad
method	Order Keys Condition	Determina si los registros se fusionan en el orden en el que aparecen enumerados en los archivos de datos, si se usan uno o varios campos clave para fusionar registros con el mismo valor en los campos clave o si los registros se fusionan si se cumple una condición determinada.
condition	<i>cadena</i>	Si method se establece en Condition, especifica la condición para incluir o descartar registros.
key_fields	[ <i>campo campo campo</i> ]	
common_keys	<i>marca</i>	
join	Inner FullOuter PartialOuter Anti	A continuación se muestra un ejemplo: set :merge.join = FullOuter
outer_join_tag.n	<i>marca</i>	En esta propiedad, <i>n</i> es el nombre de etiqueta tal y como recoge el cuadro de diálogo Seleccionar conjunto de datos. Tenga en cuenta que es posible que existan varios nombres de etiquetas especificados, ya que pueden ser varios los conjuntos de datos que aporten registros incompletos.
single_large_input	<i>marca</i>	Determina si se va a usar la optimización para tener una entrada relativamente grande en comparación con el resto de entradas.
single_large_input_tag	<i>cadena</i>	Especifica el nombre de etiqueta tal y como se muestra en el cuadro de diálogo Seleccionar conjunto de datos grande. Tenga en cuenta que el uso de esta propiedad es ligeramente distinto que el de la propiedad <i>outer_join_tag</i> ( <i>marca</i> frente a <i>cadena</i> ), ya que solamente se puede especificar un único conjunto de datos de entrada.
use_existing_sort_keys	<i>marca</i>	Determina si las entradas ya se han ordenado en función de uno o varios campos clave.
existing_sort_keys	[{ <i>cadena</i> Ascending} \ { <i>cadena</i> Descending}]	Especifica los campos que ya están ordenados y la dirección en que dicho orden se ha establecido.

## Propiedades rfmaggatenode



El nodo Adición de RFM (actualidad, frecuencia, monetario) permite tomar datos de transacciones históricas de clientes, deshacerse de los datos no utilizados y combinar todos los datos de transacciones restantes en una única fila que indica cuándo hizo negociaciones con los clientes por última vez, cuántas transacciones hicieron y el valor monetario total de dichas transacciones. [Si desea obtener más información, consulte el tema Nodo Adición de RFM en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)



**Ejemplo**

```

create rfmaggregatenode
connect :fillernode to :rfmaggregatenode
set :rfmaggregatenode.relative_to = Fixed
set :rfmaggregatenode.reference_date = "2007-10-12"
set :rfmaggregatenode.id_field = "CardID"
set :rfmaggregatenode.date_field = "Fecha"
set :rfmaggregatenode.value_field = "Cantidad"
set :rfmaggregatenode.only_recent_transactions = True
set :rfmaggregatenode.transaction_date_after = "2000-10-01"

```

propiedades de rfmaggregatenode	Tipo de datos	Descripción de la propiedad
relative_to	Fixed Today	Especifica la fecha a partir de la que se calculará la actualidad de las transacciones.
reference_date	<i>fecha</i>	Disponible sólo si Fixed se establece como relative_to.
contiguous	<i>marca</i>	Si los datos se han clasificado previamente de forma que todos los registros con el mismo ID aparecen en la ruta de datos, al seleccionar esta opción acelerará el procesamiento.
id_field	<i>campo</i>	Especifica el campo que desea utilizar para identificar el cliente y sus transacciones.
date_field	<i>campo</i>	Especifica el campo de fecha que se utilizará para calcular la actualidad.
value_field	<i>campo</i>	Especifica el campo que se utilizará para calcular el valor monetario.
extension	<i>cadena</i>	Especifica un prefijo o sufijo para campos agregados duplicados.
add_as	Suffix Prefix	Especifica si extension se debe añadir como sufijo o prefijo.
discard_low_value_records	<i>marca</i>	Activa el uso del parámetro discard_records_below.
discard_records_below	<i>número</i>	Especifique un valor mínimo por debajo del cual no se utilice ningún detalle de transacción al calcular los totales de RFM. Las unidades del valor se relacionan con el campo value seleccionado.
only_recent_transactions	<i>marca</i>	Activa el uso de los parámetros specify_transaction_date o transaction_within_last.
specify_transaction_date	<i>marca</i>	
transaction_date_after	<i>fecha</i>	Sólo está disponible si selecciona specify_transaction_date. Especifica la fecha de transacción tras la que se incluirán los registros en su análisis.
transaction_within_last	<i>número</i>	Sólo está disponible si selecciona transaction_within_last. Especifica el número y tipo de períodos (días, semanas, meses o años) desde la fecha Calcular actualidad relativa a tras la cual se incluirán los registros en su análisis.

propiedades de rfmaggregatenode	Tipo de datos	Descripción de la propiedad
transaction_scale	Days Weeks Months Years	Sólo está disponible si selecciona transaction_within_last. Especifica el número y tipo de períodos (días, semanas, meses o años) desde la fecha Calcular actualidad relativa a tras la cual se incluirán los registros en su análisis.
save_r2	marca	Muestra la fecha de la segunda transacción más reciente para cada cliente.
save_r3	marca	Sólo está disponible si selecciona save_r2. Muestra la fecha de la tercera transacción más reciente para cada cliente.

## Propiedades de samplenode



El nodo Muestrear selecciona un subconjunto de registros. Se admite una variedad de tipos de muestras, entre las que se incluyen las muestras estratificadas, conglomeradas y no aleatorias (estructuradas). El muestreo puede ser de gran utilidad para mejorar el rendimiento y para seleccionar grupos de registros o transacciones relacionadas para un análisis. [Si desea obtener más información, consulte el tema Nodo Muestrear en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

/\* Crea dos modos Muestrear para extraer distintas muestras de los mismos datos \*/

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"

set node = create samplenode at 300 100
rename ^node as 'First 500'
connect :variablefilenode to 'First 500'
set 'First 500':samplenode.method = Simple
set 'First 500':samplenode.mode = Include
set 'First 500':samplenode.sample_type = First
set 'First 500':samplenode.first_n = 500

set node = create samplenode at 300 200
rename ^node as 'Custom Strata'
connect :variablefilenode to 'Custom Strata'
set 'Custom Strata':samplenode.method = Complex
set 'Custom Strata':samplenode.stratify_by = ['Sexo' 'Colesterol']
set 'Custom Strata':samplenode.sample_units = Proportions
set 'Custom Strata':samplenode.sample_size_proportions = Custom
set 'Custom Strata':samplenode.sizes_proportions= \
  [{"M" "Alto" "Por defecto"} {"M" "Normal" "Por defecto"} \
```

```
{"F" "Alta" "0.3"}{"F" "Normal" "0.3"}}
```

propiedades de samplenode	Tipo de datos	Descripción de la propiedad
method	Simple Complex	
mode	Include Discard	Incluye o descarta los registros que reúnan la condición especificada.
sample_type	First OneInN RandomPct	Especifica el método de muestreo. A continuación se muestra un ejemplo: set :samplenode.sample_type = First set :samplenode.first_n = 100
first_n	entero	Se incluirán o descartarán los registros hasta el punto de corte especificado.
one_in_n	número	Incluye o descarta cada <i>n</i> registros.
rand_pct	número	Especifica el porcentaje de registros que incluir o descartar.
use_max_size	marca	Activa el uso del parámetro maximum_size.
maximum_size	entero	Especifica la muestra más grande que se va a incluir o descartar de la ruta de datos. Esta opción es redundante, por lo que se desactiva cuando se especifican las opciones First e Include.
set_random_seed	marca	Activa el uso del parámetro de semillas aleatorias.
random_seed	entero	Especifica el valor utilizado como semilla aleatoria.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	número	
fixed_counts	entero	
variable_proportions	campo	
variable_counts	campo	
use_min_stratum_size	marca	
minimum_stratum_size	entero	Esta opción sólo se aplica cuando se toma una muestra compleja con Sample units=Proportions.
use_max_stratum_size	marca	
maximum_stratum_size	entero	Esta opción sólo se aplica cuando se toma una muestra compleja con Sample units=Proportions.
clusters	campo	
stratify_by	[campo1 ... campoN]	
specify_input_weight	marca	

propiedades de <b>sampl</b> node	Tipo de datos	Descripción de la propiedad
input_weight	campo	
new_output_weight	cadena	
sizes_proportions	[{stringvalor de cadena}{stringvalor de cadena}...]	En caso de sample_units=proportions y sample_size_proportions=Custom, se especifica un valor para cada combinación de valores posible de campos de estratificación.
default_proportion	número	
sizes_counts	[{stringvalor de cadena}{stringvalor de cadena}...]	Especifica un valor para cada combinación de valores posible de campos de estratificación. Se utiliza de forma similar a sizes_proportions pero especificando un entero en lugar de una proporción.
default_count	número	

## Propiedades de **select**node



El nodo Seleccionar selecciona o descarta un subconjunto de registros de la ruta de datos en función de una condición específica. Por ejemplo, podría seleccionar los registros que pertenezcan a una región de ventas determinada. [Si desea obtener más información, consulte el tema Nodo Seleccionar en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create selectnode
set :selectnode.mode = Include
set :selectnode.condition = "Edad < 18"
```

propiedades de <b>select</b> node	Tipo de datos	Descripción de la propiedad
mode	Include Discard	Especifica si incluir o descartar los registros seleccionados.
condition	cadena	Condición para incluir o descartar registros.

## Propiedades de **sort**node



Los nodos Ordenar organizan registros en orden ascendente o descendente atendiendo a los valores de uno o varios campos. [Si desea obtener más información, consulte el tema Nodo Ordenar en el capítulo 3 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create sortnode
set :sortnode.keys = [{'Edad' Ascending}{'Sexo' Descending}]
set :sortnode.default_ascending = False
set :sortnode.use_existing_keys = True
```

```
set :sortnode.existing_keys = [{'Edad' Ascending}]
```

propiedades de sortnode	Tipo de datos	Descripción de la propiedad
keys	[{cadena Ascending} \ {cadena Descending}]	Especifica los campos por los que quiere realizar la ordenación (consulte el ejemplo que se muestra a continuación). Si no se especifica ninguna dirección, se utilizará la establecida por defecto.
default_ascending	marca	Especifica el orden de clasificación por defecto.
use_existing_keys	marca	Determina si la clasificación se optimiza usando el orden de clasificación anterior de los campos que ya están ordenados.
existing_keys		Especifica los campos que ya están ordenados y la dirección en que dicho orden se ha establecido. Utiliza el mismo formato que las propiedades keys.

# Propiedades de nodos de operaciones con campos

## Propiedades de anonymizenode



El nodo Anonimizar transforma la manera en que se representan los nombres y los valores de los campos a partir de ese punto de la ruta, lo que permite disfrazar los datos originales. Puede resultar útil si desea permitir que otros usuarios generen modelos utilizando datos confidenciales, como los nombres de los clientes u otros detalles. [Si desea obtener más información, consulte el tema Nodo Anonimizar en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create anonymizenode
set: anonymizenode.enable_anonymize = age
set: anonymizenode.use_prefix = true
set: anonymizenode.prefix = "mi prefijo"
set: anonymizenode.transformation = Random
set: anonymizenode.set_random_seed = true
set: anonymizenode.random_seed = "123"
```

propiedades de anonymizenode	Tipo de datos	Descripción de la propiedad
enable_anonymize	marca	Cuando se establece en T, activa la anonimización de los valores de los campos (equivale a seleccionar Sí para dicho campo en la columna Anonimizar valores).
use_prefix	marca	Cuando se establece en T, se utilizará un prefijo personalizado, si es que se ha establecido uno. Se aplica a los campos que se anonimizarán mediante el método Hash y es equivalente a elegir el botón de radio Personalizado en el cuadro de diálogo Reemplazar valores correspondiente a dicho campo.
prefix	cadena	Equivale a escribir un prefijo en el cuadro de texto del cuadro de diálogo Reemplazar valores. El prefijo por defecto es el valor por defecto, si no se ha especificado otra cosa.
transformation	Random Fixed	Determina si los parámetros de transformación que se aplican a un campo anonimizado mediante el método Transformar serán aleatorios o fijos.
set_random_seed	marca	Cuando se establece en T, se utilizará el valor especificado para la semilla (si también se ha establecido transformation en Random).
random_seed	entero	Si set_random_seed se define a T, es la semilla de los números aleatorios.

propiedades de anonymizenode	Tipo de datos	Descripción de la propiedad
scale	número	Si transformation se establece en Fixed, este valor se utiliza para escalar los valores. El valor máximo de la escala suele ser 10, pero puede reducirse para evitar desbordamientos.
translate	número	Si transformation se establece en Fixed, este valor se utiliza para traducir. El valor máximo de traslación suele ser 1000, pero puede reducirse para evitar desbordamientos.

## Propiedades autodatapreprenode



El nodo de preparación automática de datos (ADP) puede analizar sus datos e identificar los valores fijos, filtrar los campos problemáticos o que no serán útiles y derivar nuevos atributos cuando sea necesario y mejorar el rendimiento mediante técnicas de filtrado y muestreo inteligente. Puede utilizar el nodo de forma totalmente automática, permitiendo que el nodo seleccione y aplique valores fijos, o bien puede tener una vista previa de los cambios antes de que se apliquen y aceptarlos o rechazarlos. [Si desea obtener más información, consulte el tema Preparación automática de datos en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create autodatapreprenode
set: autodatapreprenode.objective = Balanced
set: autodatapreprenode.excluded_fields = Filter
set: autodatapreprenode.prepare_dates_and_times = true
set: autodatapreprenode.compute_time_until_date = true
set: autodatapreprenode.reference_date = Today
set: autodatapreprenode.units_for_date_durations = Automatic
```

propiedades de autodatapreprenode	Tipo de datos	Descripción de la propiedad
objective	Balanced Speed Accuracy Custom	
custom_fields	marca	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
target	campo	Especifica un campo de objetivo único.
inputs	[campo1 ... campoN]	Campos de entrada o predictor utilizados por el modelo.
use_frequency	marca	
frequency_field	campo	
use_weight	marca	
weight_field	campo	

propiedades de autodataprepn-ode	Tipo de datos	Descripción de la propiedad
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	<i>marca</i>	Controla el acceso a todos los campos de fecha y hora
compute_time_until_date	<i>marca</i>	
reference_date	Today Fixed	
fixed_date	<i>fecha</i>	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	<i>marca</i>	
reference_time	CurrentTime Fixed	
fixed_time	<i>hora</i>	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	<i>marca</i>	
extract_month_from_date	<i>marca</i>	
extract_day_from_date	<i>marca</i>	
extract_hour_from_time	<i>marca</i>	
extract_minute_from_time	<i>marca</i>	
extract_second_from_time	<i>marca</i>	
exclude_low_quality_inputs	<i>marca</i>	
exclude_too_many_missing	<i>marca</i>	
maximum_percentage_missing	<i>número</i>	
exclude_too_many_categories	<i>marca</i>	
maximum_number_categories	<i>número</i>	
exclude_if_large_category	<i>marca</i>	
maximum_percentage_category	<i>número</i>	
prepare_inputs_and_target	<i>marca</i>	
adjust_type_inputs	<i>marca</i>	
adjust_type_target	<i>marca</i>	
reorder_nominal_inputs	<i>marca</i>	
reorder_nominal_target	<i>marca</i>	
replace_outliers_inputs	<i>marca</i>	
replace_outliers_target	<i>marca</i>	
replace_missing_continuous_inputs	<i>marca</i>	



propiedades de autodataprepn-ode	Tipo de datos	Descripción de la propiedad
replace_missing_continuous_target	marca	
replace_missing_nominal_inputs	marca	
replace_missing_nominal_target	marca	
replace_missing_ordinal_inputs	marca	
replace_missing_ordinal_target	marca	
maximum_values_for_ordinal	número	
minimum_values_for_continuous	número	
outlier_cutoff_value	número	
outlier_method	Replace Delete	
rescale_continuous_inputs	marca	
rescaling_method	MinMax ZScore	
min_max_minimum	número	
min_max_maximum	número	
z_score_final_mean	número	
z_score_final_sd	número	
rescale_continuous_target	marca	
target_final_mean	número	
target_final_sd	número	
transform_select_input_fields	marca	
maximize_association_with_target	marca	
p_value_for_merging	número	
merge_ordinal_features	marca	
merge_nominal_features	marca	
minimum_cases_in_category	número	
bin_continuous_fields	marca	
p_value_for_binning	número	
perform_feature_selection	marca	
p_value_for_selection	número	
perform_feature_construction	marca	
transformed_target_name_extension	cadena	
transformed_inputs_name_extension	cadena	
constructed_features_root_name	cadena	
years_duration_name_extension	cadena	
months_duration_name_extension	cadena	
days_duration_name_extension	cadena	
hours_duration_name_extension	cadena	
minutes_duration_name_extension	cadena	

propiedades de autodataprepnode	Tipo de datos	Descripción de la propiedad
seconds_duration_name_extension	cadena	
year_cyclical_name_extension	cadena	
month_cyclical_name_extension	cadena	
day_cyclical_name_extension	cadena	
hour_cyclical_name_extension	cadena	
minute_cyclical_name_extension	cadena	
second_cyclical_name_extension	cadena	

## Propiedades de binningnode



El nodo Intervalos crea automáticamente nuevos campos nominales (conjunto) en función de los valores de uno o más campos continuos (rango numérico) existentes. Por ejemplo, puede transformar un campo de ingresos continuo en un campo categórico nuevo que contenga grupos de ingresos como desviaciones desde la media. Una vez creados los intervalos para el campo nuevo, puede generar un nodo Derivar en función de los puntos de corte. [Si desea obtener más información, consulte el tema Nodo Intervalos en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create binningnode
set :binningnode.fields = [Na K]
set :binningnode.method = Rank
set :binningnode.fixed_width_name_extension = "_intervalos"
set :binningnode.fixed_width_add_as = Suffix
set :binningnode.fixed_bin_method = Count
set :binningnode.fixed_bin_count = 10
set :binningnode.fixed_bin_width = 3.5
set :binningnode.tile10 = true
```

propiedades de binningnode	Tipo de datos	Descripción de la propiedad
fields	[campo1 campo2 ... campon]	Los campos continuos (rango numérico) pendientes de transformación. Se pueden crear intervalos de varios campos de forma simultánea.
method	FixedWidth EqualCount Rank SDev Optimal	Método utilizado para determinar los puntos de corte de los intervalos de campo nuevos (categorías).
rcalculate_bins	Always IfNecessary	Especifica si se vuelven a calcular los intervalos y los datos se colocan en el intervalo adecuado cada vez que se ejecuta el nodo o si los datos solamente se añaden a los intervalos existentes y cualquier nuevo intervalo que se haya añadido.
fixed_width_name_extension	cadena	La extensión por defecto es <i>_BIN</i> .

propiedades de binningnode	Tipo de datos	Descripción de la propiedad
fixed_width_add_as	Suffix Prefix	Determina si la extensión se debe añadir al principio (prefijo) o al final (sufijo) del nombre de campo. La extensión por defecto es <i>income_BIN</i> .
fixed_bin_method	Width Count	
fixed_bin_count	entero	Especifica un número entero para determinar el número de intervalos de anchura fija (categorías) para los nuevos campos.
fixed_bin_width	real	Valor (entero o real) para calcular el ancho del intervalo.
equal_count_name_extension	cadena	La extensión por defecto es <i>_TILE</i> .
equal_count_add_as	Suffix Prefix	Especifica una extensión, sufijo o prefijo, utilizada para el nombre de los campos generados con p-tiles estándar. La extensión por defecto es <i>_TILE</i> más <i>N</i> , donde <i>N</i> es el número de cuantil.
tile4	marca	Genera cuatro intervalos de cuantiles, cada uno con el 25% de los casos.
tile5	marca	Genera cinco intervalos de quintiles.
tile10	marca	Genera 10 intervalos de deciles.
tile20	marca	Genera 20 intervalos de veintiles.
tile100	marca	Genera 100 intervalos de percentiles.
use_custom_tile	marca	
custom_tile_name_extension	cadena	La extensión por defecto es <i>_TILEN</i> .
custom_tile_add_as	Suffix Prefix	
custom_tile	entero	
equal_count_method	RecordCount ValueSum	El método <i>RecordCount</i> trata de asignar el mismo número de registros a cada intervalo, mientras que <i>ValueSum</i> asigna registros de manera que la suma de los valores de cada intervalo sea la misma.
tied_values_method	Next Current Random	Especifica en qué intervalo se van a insertar los datos de valor empatado.
rank_order	Ascending Descending	Esta propiedad incluye <i>Ascending</i> (el valor más bajo se marca con 1) o <i>Descending</i> (el valor más alto se marca con 1).
rank_add_as	Suffix Prefix	Esta opción se aplica al rango, rango fraccional y rango como porcentaje.
rank	marca	
rank_name_extension	cadena	La extensión por defecto es <i>_RANK</i> .

propiedades de binningnode	Tipo de datos	Descripción de la propiedad
rank_fractional	marca	Establece rangos de casos en los que el valor del campo nuevo es igual al rango dividido por la suma de las ponderaciones de los casos que no están perdidos. Los rangos fraccionales están dentro del rango de 0 a 1.
rank_fractional_name_extension	cadena	La extensión por defecto es <i>_F_RANK</i> .
rank_pct	marca	Cada rango se divide por el número de registros con valores válidos y se multiplica por 100. Los rangos fraccionales como porcentaje están dentro del rango de 1 a 100.
rank_pct_name_extension	cadena	La extensión por defecto es <i>_P_RANK</i> .
sdev_name_extension	cadena	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	cadena	La extensión por defecto es <i>_OPTIMAL</i> .
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	campo	Campo elegido como campo supervisor, con el que se relacionan los campos seleccionados para los intervalos.
optimal_merge_bins	marca	Especifica que todos los intervalos con un número pequeño de casos se añadirán a un intervalo vecino de mayor tamaño.
optimal_small_bin_threshold	entero	
optimal_pre_bin	marca	Indica si debe agruparse previamente en intervalos el conjunto de datos.
optimal_max_bins	entero	Especifica un límite superior con el fin de evitar que se genere un número desmesurado de intervalos.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded Bounded	

## Propiedades de derivenode



El nodo Derivar modifica los valores de datos o crea campos nuevos desde uno o más campos existentes. Crea campos del tipo fórmula, marca, nominal, estado, recuento y condicional. [Si desea obtener más información, consulte el tema Nodo Derivar en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```
# Crear y configurar un nodo de campo Derivar marca
create derivenode
rename derive:derivenode as "Marca"
set Flag:derivenode.new_name = "MedicamentoX_Marca"
set Flag:derivenode.result_type = Flag
set Flag:derivenode.flag_true = 1
set Flag:derivenode.flag_false = 0
set Flag:derivenode.flag_expr = "Medicamento = X"
# Crear y configurar un nodo de campo Derivar condicional
create derivenode
rename derive:derivenode as "Condicional"
set Conditional:derivenode.result_type = Conditional
set Conditional:derivenode.cond_if_cond = "@OFFSET('\Edad\', 1) = \Edad\\""
set Conditional:derivenode.cond_then_expr = "@OFFSET('\Edad\', 1) = \Edad\' >< @INDEX"
set Conditional:derivenode.cond_else_expr = "\Edad\\""

```

propiedades de derivenode	Tipo de datos	Descripción de la propiedad
new_name	<i>cadena</i>	Nombre del campo nuevo.
mode	Single Multiple	Especifica si los campos son únicos o múltiples.
fields	[ <i>campo campo</i> <i>campo</i> ]	Se utiliza en modo Múltiple solamente para seleccionar varios campos.
name_extension	<i>cadena</i>	Especifica la extensión de los nombres de los nuevos campos.
add_as	Suffix Prefix	Añade la extensión como un prefijo (al principio) o como un sufijo (al final) del nombre de los campos.
result_type	Formula Flag Set State Count Conditional	Los seis tipos de campos nuevos que se pueden crear.
formula_expr	<i>cadena</i>	Expresión para calcular un nuevo valor de campo en el nodo Derivar.
flag_expr	<i>cadena</i>	
flag_true	<i>cadena</i>	
flag_false	<i>cadena</i>	
set_default	<i>cadena</i>	
set_value_cond	<i>cadena</i>	Estructurada para proporcionar la condición asociada a un valor dado. Formato de uso: set :derivenode. set_value_cond. Retired = 'age > 65'
state_on_val	<i>cadena</i>	Especifica el valor del campo nuevo cuando se cumple la condición Activado.
state_off_val	<i>cadena</i>	Especifica el valor del campo nuevo cuando se cumple la condición Desactivado.

propiedades de <b>derivenode</b>	Tipo de datos	Descripción de la propiedad
state_on_expression	cadena	
state_off_expression	cadena	
state_initial	On Off	Asigna a cada registro del nuevo campo un valor inicial de On o Off. Este valor puede cambiar a medida que se cumplan las condiciones.
count_initial_val	cadena	
count_inc_condition	cadena	
count_inc_expression	cadena	
count_reset_condition	cadena	
cond_if_cond	cadena	
cond_then_expr	cadena	
cond_else_expr	cadena	

## Propiedades de **ensemblenode**



El nodo Conjunto combina dos o más nugget de modelo para obtener pronósticos más precisos que pueden conseguirse de cualquier modelo. [Si desea obtener más información, consulte el tema Nodo Conjunto en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
# Crear y configurar un nodo Conjunto
# Utilizar este nodo con los modelos en demos\streams\pm_binaryclassifier.str
create ensemblenode
set :ensemblenode.ensemble_target_field = response
set :ensemblenode.filter_individual_model_output = false
set :ensemblenode.flag_ensemble_method = ConfidenceWeightedVoting
set :ensemblenode.flag_voting_tie_selection = HighestConfidence
```

propiedades de <b>ensemblenode</b>	Tipo de datos	Descripción de la propiedad
ensemble_target_field	campo	Especifica el campo objetivo de todos los modelos utilizados en el conjunto.
filter_individual_model_output	marca	Especifica si los resultados de puntuación de los modelos individuales se deben eliminar.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto solamente se aplica si el objetivo seleccionado es un campo de marca.

propiedades de <b>ensembnode</b>	Tipo de datos	Descripción de la propiedad
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto solamente se aplica si el objetivo seleccionado es un campo nominal.
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto solamente se aplica si el objetivo seleccionado es un campo de marca.
set_voting_tie_selection	Random HighestConfidence	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto solamente se aplica si el objetivo seleccionado es un campo nominal.
calculate_standard_error	marca	Si el campo objetivo es continuo, se ejecuta un error estándar por defecto para calcular la diferencia entre los valores medidos o estimados y los valores true; y para mostrar si las estimaciones coinciden.

## Propiedades de fillernode



El nodo Rellenar sustituye valores de campos y cambia el almacenamiento. Puede sustituir los valores en función de una condición CLEM, como @BLANK(@FIELD). También puede sustituir todos los espacios vacíos o valores nulos por un valor específico. Un nodo Rellenar suelen utilizarse junto con un nodo Tipo para sustituir valores perdidos. [Si desea obtener más información, consulte el tema Nodo Rellenar en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create fillernode
set :fillernode.fields = ['Edad']
set :fillernode.replace_mode = Always
set :fillernode.condition = "(\ 'Edad\ ' > 60) and (\ 'Sexo\ ' = \ 'M\ ')"
set :fillernode.replace_with = "\ hombre mayor\ "
```

propiedades de <b>fillernode</b>	Tipo de datos	Descripción de la propiedad
fields	[campo campo campo]	Campos del conjunto de datos cuyos valores se van a examinar y sustituir.
replace_mode	Always Conditional Blank Null BlankAndNull	Se pueden sustituir todos los valores, valores vacíos o valores nulos, o bien reemplazar aquellos basados en una condición específica.
condition	cadena	
replace_with	cadena	

## Propiedades de filternode



El nodo Filtro filtra (descarta) campos, vuelve a nombrar campos y establece asociaciones de un nodo de origen a otro. [Si desea obtener más información, consulte el tema Cambio de nombre o filtrado de campos en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create filternode
set :filternode.default_include = True
set :filternode.new_name.'Medicamento' = 'Química'
set :filternode.include.'Medicamento' = off
```

**Usando la propiedad default\_include.** Tenga en cuenta que, si establece el valor de la propiedad default\_include, no se incluirán o excluirán automáticamente todos los campos, sino que simplemente establecerá por defecto los seleccionados actualmente. Esto equivale funcionalmente a pulsar en el botón Incluir campos por defecto del cuadro de diálogo del nodo Filtro. Por ejemplo, imagine que ejecuta el siguiente proceso:

```
set Filter.default_include=False
# Incluir solamente los campos en la lista
for f in Age Sex
  set Filter.include.^f=True
endfor
```

Esto hará que el nodo pase los campos *Edad* y *Sexo* y descarte el resto. Ahora, imagine que ejecuta de nuevo el mismo proceso, pero designa dos campos diferentes:

```
set Filter.default_include=False
# Incluir solamente los campos en la lista
for f in PS Na
  set Filter.include.^f=True
endfor
```

De esta forma, se añadirán dos campos más al filtro, de manera que pasan un total de cuatro campos (*Edad*, *Sexo*, *PS* y *Na*). En otras palabras, al volver a definir el valor de default\_include a False no se restablecen automáticamente todos los campos.



Además, si ahora cambia `default_include` a `True` (ya sea mediante un proceso o con el cuadro de diálogo del nodo Filtro), cambiará el comportamiento de forma que los cuatro campos anteriores no se incluirían, sino que se descartarán. Si no está seguro, se recomienda experimentar con los controles del cuadro de diálogo del nodo Filtro para entender esta interacción.

propiedades de <code>filternode</code>	Tipo de datos	Descripción de la propiedad
<code>default_include</code>	<i>marca</i>	Propiedad adaptada para especificar si el comportamiento por defecto es para pasar o filtrar los campos: <code>NODE.include.FIELDNAME</code> A continuación se muestra un ejemplo: <code>set mynode:filternode.default_include = false</code> Tenga en cuenta que, si se establece esta propiedad, no se incluirán o excluirán automáticamente todos los campos, sino que simplemente se determinará si los campos seleccionados se incluyen o excluyen por defecto. Vea el ejemplo que se muestra a continuación para ver más comentarios.
<code>include</code>	<i>marca</i>	Propiedad adaptada para incluir y eliminar el campo. Formato de uso: <code>NODE.include.FIELDNAME</code> A continuación se muestra un ejemplo: <code>set mynode:filternode.include.Age = false</code>
<code>new_name</code>	<i>cadena</i>	A continuación se muestra un ejemplo: <code>set mynode:filternode.new_name. Age = "age"</code>

## Propiedades de `historynode`



El nodo Histórico se utiliza para crear campos nuevos que contienen datos de los campos de registros anteriores. Los nodos Histórico se suelen utilizar para los datos secuenciales, como los datos de series temporales. Antes de utilizar un nodo Histórico, puede desear ordenar los datos utilizando un nodo Ordenar. [Si desea obtener más información, consulte el tema Nodo Histórico en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create historynode
set :historynode.fields = ['Medicamento']
set :historynode.offset = 1
set :historynode.span = 3
set :historynode.unavailable = Discard
```

```
set :historynode.fill_with = "sin definir"
```

propiedades de historynode	Tipo de datos	Descripción de la propiedad
fields	[ <i>campo campo campo</i> ]	Campos para los que desea un histórico.
offset	<i>número</i>	Especifica el último registro anterior al registro actual desde el que desea extraer valores del campo histórico.
span	<i>número</i>	Especifica el número de registros anteriores de los que desea extraer valores.
unavailable	Discard Leave Fill	Para tratar registros que no tienen valores históricos, suele hacer referencia a los primeros registros (en la parte superior del conjunto de datos), de los que no hay registros previos que utilizar como histórico.
fill_with	String Number	Especifica un valor o cadena que utilizar en el caso de los registros en los que no existen valores históricos disponibles.

## Propiedades de partitionnode



El nodo Partición genera un campo de partición, que divide los datos en subconjuntos diferentes para las fases de entrenamiento, comprobación y validación en la generación del modelo. [Si desea obtener más información, consulte el tema Nodo Partición en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create partitionnode
set :partitionnode.create_validation = True
set :partitionnode.training_size = 33
set :partitionnode.testing_size = 33
set :partitionnode.validation_size = 33
set :partitionnode.set_random_seed = True
set :partitionnode.random_seed = "123"
set :partitionnode.value_mode = System
```

propiedades de partitionnode	Tipo de datos	Descripción de la propiedad
new_name	<i>cadena</i>	Nombre del campo de partición generado por el nodo.
create_validation	<i>marca</i>	Especifica si se debe crear una partición de validación.
training_size	<i>entero</i>	Porcentaje de registros (0–100) que se van a asignar a la partición de entrenamiento.
testing_size	<i>entero</i>	Porcentaje de registros (0–100) que se van a asignar a la partición de comprobación.

propiedades de partitionnode	Tipo de datos	Descripción de la propiedad
validation_size	entero	Porcentaje de registros (0–100) que se van a asignar a la partición de validación. Se ignora si no se crea una partición de validación.
training_label	cadena	Etiqueta para la partición de entrenamiento.
testing_label	cadena	Etiqueta para la partición de comprobación.
validation_label	cadena	Etiqueta para la partición de validación. Se ignora si no se crea una partición de validación.
value_mode	System SystemAndLabel Label	Especifica los valores utilizados para representar cada partición en los datos. Por ejemplo, el entero del sistema 1, la etiqueta Training o una combinación de los dos, 1_Training, pueden representar la muestra de entrenamiento.
set_random_seed	Booleana	Especifica si se debe utilizar una semilla aleatoria especificada por el usuario.
random_seed	entero	Valor de semilla aleatoria especificada por el usuario. Para que se pueda utilizar este valor, set_random_seed se debe definir a True.
enable_sql_generation	Booleana	Especifica si se utiliza la retrotracción SQL para asignar registros a particiones.
unique_field		Especifica el campo de entrada que se utiliza para garantizar que los registros se asignan a particiones de una forma aleatoria pero reproducible. Para que se pueda utilizar este valor, enable_sql_generation se debe definir a True.

## Propiedades de reclassifynode



El nodo Reclasificar transforma un conjunto de valores categóricos en otro. La reclasificación es útil para contraer categorías o reagrupar datos para su análisis. Si desea obtener más información, consulte el tema [Nodo Reclasificar en el capítulo 4 en Nodos de origen, proceso y resultado de IBM SPSS Modeler 15](#).

### Ejemplo

```
create reclassifynode
set :reclassifynode.mode = Multiple
set :reclassifynode.replace_field = true
set :reclassifynode.field = "Medicamento"
set :reclassifynode.new_name = "Química"
set :reclassifynode.fields = [Medicamento, BP]
set :reclassifynode.name_extension = "reclasificado"
set :reclassifynode.add_as = Prefix
set :reclassifynode.reclassify.'MedicamentoA' = 'Sí'
set :reclassifynode.use_default = True
set :reclassifynode.default = "MarcaX"
```

```
set :reclassifynode.pick_list = [MarcaX, Placebo, Genérico]
```

propiedades de reclassifynode	Tipo de datos	Descripción de la propiedad
mode	Single Multiple	Single reclasifica las categorías de un campo. Multiple activa las opciones que permiten la transformación de varios campos al mismo tiempo.
replace_field	marca	
field	cadena	Solamente se utiliza en modo Single.
new_name	cadena	Solamente se utiliza en modo Single.
fields	[campo1 campo2 ... campon]	Solamente se utiliza en modo Multiple.
name_extension	cadena	Solamente se utiliza en modo Multiple.
add_as	Suffix Prefix	Solamente se utiliza en modo Multiple.
reclassify	cadena	Propiedad estructurada para valores de campos. Formato de uso: NODE.reclassify. OLD_VALUE A continuación se muestra un ejemplo: set :reclassifynode.reclassify.'drugB' = 'Yes'
use_default	marca	Utiliza el valor por defecto.
default	cadena	Especifica un valor por defecto.
pick_list	[cadena cadena ... cadena]	Permite al usuario importar una lista de valores nuevos conocidos para rellenar la lista desplegable de la tabla. A continuación se muestra un ejemplo: set :reclassify.pick_list = [fruit dairy cereals]

## Propiedades de reordernode



El nodo Reorg. campos define el orden natural utilizado para mostrar los campos en la parte posterior de la ruta. Este orden afecta a la visualización de los campos en diversas ubicaciones, como las tablas, las listas y el selector de campos. Esta operación resulta útil al trabajar con conjuntos de datos amplios que hacen más visibles los campos de interés. [Si desea obtener más información, consulte el tema Nodo Reorg. campos en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create reordernode
set :reordernode.mode = Custom
set :reordernode.sort_by = Storage
set :reordernode.ascending = "falso"
set :reordernode.start_fields = [Edad Colesterol]
```

```
set :reordernode.end_fields = [Medicamento]
```

propiedades de reordernode	Tipo de datos	Descripción de la propiedad
mode	Custom Auto	Se pueden ordenar los valores de forma automática o especificar un orden personalizado.
sort_by	Name Type Storage	
ascending	marca	
start_fields	[campo1 campo2 ... campon]	Los campos nuevos se han insertado después de estos campos.
end_fields	[campo1 campo2 ... campon]	Los campos nuevos se han insertado antes de estos campos.

## Propiedades de restructurenode



El nodo Reestructurar convierte un campo nominal o marca en un grupo de campos que se puede rellenar con los valores todavía de otro campo. Por ejemplo, para un campo determinado llamado *tipo de pago*, con valores de *crédito*, *efectivo*, y *débito*, se crearían tres campos nuevos (*crédito*, *efectivo*, *débito*), que contendría cada uno el valor del pago real realizado. [Si desea obtener más información, consulte el tema Nodo Reestructurar en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create restructurenode
connect :typenode to :restructurenode
set :restructurenode.fields_from.Droga = ["MedicamentoA" "MedicamentoX"]
set :restructurenode.include_field_name = "Verdadero"
set :restructurenode.value_mode = "Otros campos"
set :restructurenode.value_fields = ["Edad" "BP"]
```

propiedades de restructurenode	Tipo de datos	Descripción de la propiedad
fields_from	[ <i>categoría</i> <i>categoría categoría</i> ] all	Por ejemplo: set :restructurenode.fields_from.Drug = [drugA drugB] crea campos denominados Drug_drugA y Drug_drugB. Para utilizar todas las categorías del campo específico: set :restructurenode.fields_from.Drug = all
include_field_name	marca	Indica si se debe usar el nombre del campo en el nombre de campo reestructurado.
value_mode	OtherFields Flags	Indica el modo de definir los valores para los campos reestructurados. Con OtherFields, debe especificar los campos que se van a usar (consulte a continuación). Con Flags, los valores son marcas numéricas.
value_fields	[ <i>campo campo</i> <i>campo</i> ]	Obligatorio si value_mode es OtherFields. Especifica los campos que se van a usar como campos de valores.

## Propiedades de rfmanalysisnode



El nodo Análisis de RFM (actualidad, frecuencia, monetario) permite determinar cuantitativamente qué clientes son los mejores examinando cuándo ha sido la compra más reciente de un cliente (actualidad), cuántas veces suelen comprar (frecuencia) y cuánto gasta el cliente en todas las transacciones (valor monetario). [Si desea obtener más información, consulte el tema Nodo Análisis de RFM en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create rfmanalysisnode
connect :rfmaggregatenode to :rfmanalysisnode
set :rfmanalysisnode.recency = Recency
set :rfmanalysisnode.frequency = Frequency
set :rfmanalysisnode.monetary = Monetary
set :rfmanalysisnode.tied_values_method = Next
set :rfmanalysisnode.recalculate_bins = IfNecessary
set :rfmanalysisnode.recency_thresholds = [1, 500, 800, 1500, 2000, 2500]
```

propiedades de rfmanalysisnode	Tipo de datos	Descripción de la propiedad
recency	<i>campo</i>	Especifica el campo de actualidad. Puede ser una fecha, marca de tiempo o un número simple.
frequency	<i>campo</i>	Especifica el campo de frecuencia.
monetary	<i>campo</i>	Especifica el campo de monetario.
recency_bins	<i>entero</i>	Especifica el número de intervalos de actividades recientes que se van a generar.
recency_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de actividades recientes. El valor predeterminado es 100.
frequency_bins	<i>entero</i>	Especifica el número de intervalos de frecuencia que se van a generar.
frequency_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de frecuencia. El valor predeterminado es 10.
monetary_bins	<i>entero</i>	Especifica el número de intervalos de monetario que se van a generar.
monetary_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de monetario. El valor predeterminado es 1.
tied_values_method	Next Current	Especifica en qué intervalo se van a insertar los datos de valor empatado.
recalculate_bins	Always IfNecessary	
add_outliers	<i>marca</i>	Disponible solamente si recalculate_bins se establece como IfNecessary. Si se selecciona, los registros por debajo del intervalo más inferior se añaden al intervalo inferior y los registros por encima, se añaden al intervalo superior.

propiedades de rfmanalysisnode	Tipo de datos	Descripción de la propiedad
binned_field	Recency Frequency Monetary	
recency_thresholds	valor valor	Disponible solamente si recalculate_bins se establece como Always. Especifica los umbrales superior e inferior de los intervalos de actividades recientes. El umbral superior de un intervalo se utiliza como el umbral inferior del siguiente, por ejemplo, [10 30 60] definiría dos intervalos, el primer intervalo con los umbrales superior e inferior de 10 y 30, con los umbrales del segundo intervalo de 30 y 60.
frequency_thresholds	valor valor	Disponible solamente si recalculate_bins se establece como Always.
monetary_thresholds	valor valor	Disponible solamente si recalculate_bins se establece como Always.

## Propiedades de settoflagnode



El nodo Marcas deriva varios campos de marcas en función de los valores categóricos definidos para uno o más campos nominales. [Si desea obtener más información, consulte el tema Nodo Marcas en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create settoflagnode
connect :typenode to :settoflag
set :settoflagnode.fields_from.Drug = ["MedicamentoA" "MedicamentoX"]
set :settoflagnode.true_value = "1"
set :settoflagnode.false_value = "0"
set :settoflagnode.use_extension = "Verdadero"
set :settoflagnode.extension = "Marca_Medicamento"
set :settoflagnode.add_as = Suffix
set :settoflagnode.aggregate = True
set :settoflagnode.keys = ['Colesterol']
```

propiedades de settoflagnode	Tipo de datos	Descripción de la propiedad
fields_from	[ <i>categoría</i> <i>categoría categoría</i> ] all	Por ejemplo: set :settoflagnode.fields_from.Drug = [drugA drugB] crea campos de marcas denominados Drug_drugA y Drug_drugB. Para utilizar todas las categorías del campo específico: set :settoflagnode.fields_from.Drug = all
true_value	<i>cadena</i>	Especifica el valor para verdadero utilizado por el nodo al configurar una marca. El valor por defecto es T (del inglés 'True').

propiedades de settoflagnode	Tipo de datos	Descripción de la propiedad
false_value	cadena	Especifica el valor para falso utilizado por el nodo al configurar una marca. El valor por defecto es F (del inglés 'True').
use_extension	marca	Utiliza una extensión como sufijo o prefijo para el nuevo campo de marca.
extension	cadena	
add_as	Suffix Prefix	Especifica si la extensión es un sufijo o un prefijo.
aggregate	marca	Agrupar registros en función de campos clave. Si algún registro se establece como verdadero, se activarán todos los campos de marca de un grupo.
keys	[campo campo campo]	Campos clave.

## Propiedades de statisticstransformnode



El nodo Transformación Statistics ejecuta una selección de comandos de sintaxis de IBM® SPSS® Statistics en los orígenes de datos de IBM® SPSS® Modeler. Este nodo requiere una copia de SPSS Statistics con licencia. [Si desea obtener más información, consulte el tema Nodo Transformación de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

Las propiedades de este nodo están descritas en [Propiedades de statisticstransformnode](#) el p. 315.

## Propiedades de timeintervalsnode



El nodo Intervalos de tiempo especifica los intervalos y genera etiquetas (si es necesario) para modelar los datos de series temporales. Si los valores no están espaciados de manera uniforme, el nodo puede rellenar o agregar valores según sea necesario para crear un intervalo uniforme entre registros. [Si desea obtener más información, consulte el tema Nodo de intervalos de tiempo en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create timeintervalsnode
set :timeintervalsnode.interval_type=SecondsPerDay
set :timeintervalsnode.days_per_week=4
set :timeintervalsnode.week_begins_on=Tuesday
set :timeintervalsnode.hours_per_day=10
set :timeintervalsnode.day_begins_hour=7
set :timeintervalsnode.day_begins_minute=5
set :timeintervalsnode.day_begins_second=17
set :timeintervalsnode.mode=Label
set :timeintervalsnode.year_start=2005
set :timeintervalsnode.month_start=January
set :timeintervalsnode.day_start=4
set :timeintervalsnode.pad.AGE=MeanOfRecentPoints
set :timeintervalsnode.agg_mode=Specify
```



set :timeintervalsnode.agg\_set\_default=Last

propiedades de timeintervalsnode	Tipo de datos	Descripción de la propiedad
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	Determina si desea etiquetar los registros de manera consecutiva o crear la serie según una fecha, marca de tiempo o campo de tiempo concretos.
field	<i>campo</i>	Al crear la serie a partir de los datos, especifica el campo que informa de la fecha u hora de cada registro.
period_start	<i>entero</i>	Especifica el intervalo de inicio para períodos o períodos cíclicos.
cycle_start	<i>entero</i>	Ciclo de inicio de los períodos cíclicos.
year_start	<i>entero</i>	En el caso de los tipos de intervalo que procedan, el año en que el primer intervalo tiene lugar.
quarter_start	<i>entero</i>	En el caso de los tipos de intervalo que procedan, el trimestre en que el primer intervalo tiene lugar.
month_start	Enero Febrero Marzo Abril Mayo Junio Julio Agosto Septiembre Octubre Noviembre Diciembre	
day_start	<i>entero</i>	
hour_start	<i>entero</i>	
minute_start	<i>entero</i>	
second_start	<i>entero</i>	
periods_per_cycle	<i>entero</i>	En el caso de los períodos cíclicos, número en cada ciclo.

propiedades de timeintervalsnode	Tipo de datos	Descripción de la propiedad
fiscal_year_begins	Enero Febrero Marzo Abril Mayo Junio Julio Agosto Septiembre Octubre Noviembre Diciembre	En el caso de los intervalos trimestrales, especifica el mes en el que comienza el año fiscal.
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	En el caso de los intervalos periódicos (días a la semana, horas al día, minutos al día y segundos al día), especifica el día en el que comienza la semana.
day_begins_hour	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica la hora a la que comienza el día. Se puede usar junto con <i>day_begins_minute</i> y <i>day_begins_second</i> para determinar una hora exacta, como <i>8:05:01</i> . Vea el ejemplo de uso que se muestra a continuación.
day_begins_minute	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el minuto en el que comienza el día (por ejemplo, <i>5</i> en <i>8:05</i> ).
day_begins_second	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el segundo en el que comienza el día (por ejemplo, <i>17</i> en <i>8:05:17</i> ).
days_per_week	<i>entero</i>	En el caso de los intervalos periódicos (días a la semana, horas al día, minutos al día y segundos al día), especifica el número de días a la semana.
hours_per_day	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el número de horas al día.
interval_increment	1 2 3 4 5 6 10 15 20 30	En el caso de los minutos al día y de los segundos al día, especifica el número de minutos o segundos que se va a aumentar en cada registro.

propiedades de timeintervalsnode	Tipo de datos	Descripción de la propiedad
field_name_extension	cadena	
field_name_extension_as_prefix	marca	
date_format	"DDMMAA" "MMDDAA" "AAMMDD" "AAAAMMDD" "AAAADD" DÍA MES "DD-MM-AA" "DD-MM-AAAA" "MM-DD-AA" "MM-DD-AAAA" "DD-MES-AA" "DD-MES-AAAA" "AAAA-MM-DD" "DD.MM.AA" "DD.MM.AAAA" "MM.DD.AA" "MM.DD.AAAA" "DD.MES.AA" "DD.MES.AAAA" "DD/MM/AA" "DD/MM/AAAA" "MM/DD/AA" "MM/DD/AAAA" "DD/MES/AA" "DD/MES/AAAA" MES AAAA t T AAAA ss SE AAAA	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
aggregate	Mean Sum Mode Min Max First Last TrueIfAnyTrue	Especifica el método de agregación de un campo (por ejemplo, aggregate.AGE=Mean).

propiedades de timeintervalsnode	Tipo de datos	Descripción de la propiedad
pad	Blank MeanOfRecentPoints True False	Especifica el método de relleno de un campo (por ejemplo, pad.AGE=MeanOfRecentPoints).
agg_mode	All Specify	Determina si se agregan o rellenan todos los campos con las funciones por defecto según sea necesario o bien si se especifican los campos y las funciones que deben usarse.
agg_range_default	Mean Sum Mode Min Max	Especifica la función por defecto que se va a usar al agregar campos continuos.
agg_set_default	Mode First Last	Especifica la función por defecto que se va a usar al agregar campos nominales.
agg_flag_default	TruelfAnyTrue Mode First Last	
pad_range_default	Blank MeanOfRecentPoints	Especifica la función por defecto que se va a usar al agregar campos continuos.
pad_set_default	Blank MostRecentValue	
pad_flag_default	Blank True False	
max_records_to_create	<i>entero</i>	Especifica el número máximo de registros que se van a crear al rellenar la serie.
estimation_from_beginning	<i>marca</i>	
estimation_to_end	<i>marca</i>	
estimation_start_offset	<i>entero</i>	
estimation_num_holdouts	<i>entero</i>	
create_future_records	<i>marca</i>	
num_future_records	<i>entero</i>	
create_future_field	<i>marca</i>	
future_field_name	<i>cadena</i>	

## Propiedades de transposenode



El nodo Transponer intercambia los datos en filas y columnas de manera que los registros se conviertan en campos y los campos en registros. [Si desea obtener más información, consulte el tema Nodo Transponer en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```
create transposenode
set :transposenode.transposed_names=Read
set :transposenode.read_from_field="EtiquetaTiempo"
set :transposenode.max_num_fields="1000"
set :transposenode.id_field_name="ID"
```

propiedades de transposenode	Tipo de datos	Descripción de la propiedad
transposed_names	Prefix Read	Se pueden generar nuevos nombres de campo automáticamente a partir de un prefijo concreto o bien se pueden leer desde un campo existente en los datos.
prefix	<i>cadena</i>	
num_new_fields	<i>entero</i>	Al usar un prefijo, especifica el número máximo de campos nuevos que se van a crear.
read_from_field	<i>campo</i>	Campo del que se leen los nombres. Debe tratarse de un campo instanciado o, de lo contrario, se producirá un error al ejecutar el nodo.
max_num_fields	<i>entero</i>	Al leer nombres de un campo, especifica un límite superior con el fin de evitar que se genere un número desmesurado de campos.
transpose_type	Numeric String Custom	Por defecto, solamente los campos continuos (rango numérico) se transponen, si bien se puede elegir un subconjunto personalizado de campos numéricos o, en su lugar, transponer todos los campos de cadena.
transpose_fields	[ <i>campo campo campo</i> ]	Especifica los campos que se van a transponer cuando se usa la opción Custom.
id_field_name	<i>campo</i>	

**Propiedades de typenode**

El nodo Tipo especifica propiedades y metadatos de campo. Por ejemplo, puede especificar un nivel de medición (continuo, nominal, ordinal o marca) para cada campo, establecer las opciones para gestionar valores perdidos y nulos del sistema, establecer el papel de un campo con fines de modelado, especificar las etiquetas de valor y campo y especificar los valores de un campo. [Si desea obtener más información, consulte el tema Nodo Tipo en el capítulo 4 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```
create typenode
connect :variablefilenode to :typenode
set :typenode.check.'Colesterol' = Coerce
set :typenode.direction.'Medicamento' = Input
set :typenode.type.K = Range
set :typenode.values.Drug = [MedicamentoA MedicamentoB MedicamentoC MedicamentoD MedicamentoX MedicamentoY MedicamentoZ]
```

```

set :typenode.null_missing.BP = false
set :typenode.whitespace_missing.BP = "falso"
set :typenode.description.BP = "Presión sanguínea"
set :typenode.value_labels.BP = {{HIGH 'Presión sanguínea alta'}}{NORMAL 'Presión sanguínea normal'}}
set :typenode.display_places.K = 5
set :typenode.export_places.K = 2
set :typenode.grouping_symbol.Drug = None
set :typenode.column_width.Cholesterol = 25
set :typenode.justify.Cholesterol = Right

```

Observe que en algunos casos puede que sea necesario instanciar totalmente el nodo Tipo para que otros nodos funcionen adecuadamente, como, por ejemplo, la propiedad `fields` from del nodo `Marcas`. Simplemente conecte un nodo `Tabla` y ejecútelo para instanciar los campos:

```

create tablenode
connect :typenode to :tablenode
execute :tablenode
delete :tablenode

```

propiedades de typenode	Tipo de datos	Descripción de la propiedad
<code>direction</code>	Input Target Both None Partition Split Frequency RecordID	Propiedad adaptada para los papeles de los campos. Formato de uso: <code>NODE.direction.FIELDNAME</code> <i>Nota:</i> Se han desaprobado los valores <code>In</code> y <code>Out</code> . Deberán de ser compatibles en versiones posteriores.
<code>type</code>	Range Flag Set Typeless Discrete Ordered Set Default	Tipo de campo. La configuración de <code>type</code> a <code>Default</code> borrará cualquier configuración del parámetro <code>values</code> , y si <code>value_mode</code> tiene el valor <code>Specify</code> , se restablecerá a <code>Read</code> . Si <code>value_mode</code> se ha establecido como <code>Pass</code> o <code>Read</code> , la configuración de <code>type</code> no afectará a <code>value_mode</code> . Formato de uso: <code>NODE.type.FIELDNAME</code>
<code>storage</code>	Unknown String Integer Real Time Date Timestamp	Propiedad adaptada de solamente lectura para el tipo de almacenamiento de campos. Formato de uso: <code>NODE.storage.FIELDNAME</code>
<code>check</code>	None Nullify Coerce Discard Warn Abort	Propiedad adaptada para la comprobación del rango y el tipo de campo. Formato de uso: <code>NODE.check.FIELDNAME</code>

propiedades de typenode	Tipo de datos	Descripción de la propiedad
values	[valor valor]	Para un campo continuos, el primer valor es el mínimo y el último es el máximo. Para campos nominales, especifique todos los valores. Para los campos marca, el primer valor representa <i>falso</i> y el último, <i>verdadero</i> . La configuración de esta propiedad establece de forma automática la propiedad value_mode en Specify. Formato de uso: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Determina la forma en la que se establecen los valores. Tenga en cuenta que no puede establecer esta propiedad directamente en Specify. Para utilizar valores específicos, establezca la propiedad values. Formato de uso: NODE.value_mode.FIELDNAME
extend_values	marca	Se aplica cuando value_mode se ha establecido como Read. Establézcala en T para añadir nuevos valores de lectura a los valores existentes del campo. Establézcala en F para descartar los valores existentes y favorecer a los nuevos valores de lectura. Formato de uso: NODE.extend_values.FIELDNAME
enable_missing	marca	Si se define a T, activa el seguimiento de los valores perdidos para el campo. Formato de uso: NODE.enable_missing.FIELDNAME
missing_values	[valor valor ...]	Especifica los valores de datos que denotan los datos perdidos. Formato de uso: NODE.missing_values.FIELDNAME
range_missing	marca	Especifica si se ha definido un rango de valores perdidos (vacíos) para un campo.
missing_lower	cadena	Si range_missing es verdadero, especifica el límite inferior del rango de valores perdidos.
missing_upper	cadena	Si range_missing es verdadero, especifica el límite superior del rango de valores perdidos.
null_missing	marca	Si se define a T, los valores <i> nulos</i> (valores no definidos que se muestran como \$null\$ en el software) se consideran valores perdidos. Formato de uso: NODE.null_missing.FIELDNAME
whitespace_missing	marca	Si se define a T, los valores que solamente contienen un espacio en blanco (espacios, tabulaciones y líneas nuevas) se consideran valores perdidos. Formato de uso: NODE.whitespace_missing.FIELDNAME
description	cadena	Especifica la descripción de un campo.

propiedades de typenode	Tipo de datos	Descripción de la propiedad
value_labels	<i>[[Valor CadenaEtiquetas} { Valor CadenaEtiquetas} ...]</i>	Se utiliza para especificar etiquetas para los pares de valores. A continuación se muestra un ejemplo: set :typenode.value_labels.'Drug'={{drugA label1} {drugB label2}}
display_places	<i>entero</i>	Establece el número de cifras decimales para el campo cuando se muestra (solamente se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.display_places.FIELDNAME
export_places	<i>entero</i>	Establece el número de cifras decimales para el campo cuando se exporta (solamente se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD COMMA	Establece el separador decimal para el campo (solamente se aplica a campos con almacenamiento REAL). Formato de uso: NODE.decimal_separator.FIELDNAME
date_format	"DDMMAA" "MMDDAA" "AAMMDD" "AAAAMMDD" "AAAADDD" DÍA MES "DD-MM-AA" "DD-MM-AAAA" "MM-DD-AA" "MM-DD-AAAA" "DD-MES-AA" "DD-MES-AAAA" "AAAA-MM-DD" "DD.MM.AA" "DD.MM.AAAA" "MM.DD.AA" "MM.DD.AAAA" "DD.MES.AA" "DD.MES.AAAA" "DD/MM/AA" "DD/MM/AAAA" "MM/DD/AA" "MM/DD/AAAA" "DD/MES/AA" "DD/MES/AAAA" MES AAAA t T AAAA ss SE AAAA	Establece el formato de fecha para el campo (solamente se aplica a campos con almacenamiento DATE o TIMESTAMP). Formato de uso: NODE.date_format.FIELDNAME A continuación se muestra un ejemplo: set :tablnode.date_format.'LaunchDate' = "DDMMYY"



propiedades de typenode	Tipo de datos	Descripción de la propiedad
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Establece el formato de hora para el campo (solamente se aplica a campos con almacenamiento TIME o TIMESTAMP). Formato de uso: NODE.time_format.FIELDNAME A continuación se muestra un ejemplo: set :tablnode.time_format.'BOF_enter' = "HHMMSS"
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	Establece el formato de presentación de los números para el campo. Formato de uso: NODE.number_format.FIELDNAME
standard_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato estándar. Un valor de -1 utilizará el valor por defecto de la ruta. Tenga en cuenta que la propiedad display_places existente también cambiará esto, pero ahora se ha desaprobado. Formato de uso: NODE.standard_places.FIELDNAME
scientific_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato científico. Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.scientific_places.FIELDNAME
currency_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato moneda. Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.currency_places.FIELDNAME
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	Establece el símbolo de agrupación para el campo. Formato de uso: NODE.grouping_symbol.FIELDNAME
column_width	entero	Establece el ancho de columna para el campo. Un valor de -1 establecerá el ancho de columna en Auto. Formato de uso: NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	Establece la justificación de columna para el campo. Formato de uso: NODE.justify.FIELDNAME

# Propiedades de nodos de gráficos

## Propiedades comunes de nodos de gráficos

Esta sección describe las propiedades disponibles para los nodos de gráficos, incluidas las comunes y aquellas específicas de cada tipo de nodo.

Propiedades comunes de nodos de gráficos	Tipo de datos	Descripción de la propiedad
title	<i>cadena</i>	Especifica el título. Ejemplo: "Esto es un título".
caption	<i>cadena</i>	Especifica el pie. Ejemplo: "Esto es un pie".
output_mode	Screen File	Determina si el resultado del nodo de gráficos se muestra o si se guarda en un archivo.
output_format	BMP JPEG PNG HTML output (.cou)	Especifica el tipo de resultado. El tipo exacto de resultado permitido para cada nodo varía.
full_filename	<i>cadena</i>	Especifica la ruta de destino y el nombre de archivo del resultado generado por el nodo de gráficos.
use_graph_size	<i>marca</i>	Controla si el tamaño del gráfico se ha establecido de manera explícita mediante las propiedades de ancho y altura a continuación. Afecta solamente a los gráficos que tienen salida a pantalla. No disponible para el nodo Distribución.
graph_width	<i>número</i>	Si use_graph_size es True, establece el ancho del gráfico en píxeles.
graph_height	<i>número</i>	Si use_graph_size es True, establece la altura del gráfico en píxeles.

### Notas

**Desactivación de los campos opcionales.** Los campos opcionales, como un campo de superposición para gráficos, se pueden desactivar estableciendo el valor de la propiedad en "" (cadena vacía), tal y como se muestra en el siguiente ejemplo:

```
set :plotnode.color_field = ""
```

**Especificación de los colores.** Los colores de los títulos, pies, fondos y etiquetas se pueden especificar mediante las cadenas hexadecimales que comiencen con el símbolo almohadilla (#). Por ejemplo, para establecer el fondo del gráfico en cielo azul, debe utilizar la siguiente instrucción:

```
set migráfico.graph_background="#87CEEB"
```

Aquí, los dos primeros dígitos, 87, especifican el contenido rojo, los dos del medio, CE, especifican el contenido verde y los dos últimos, EB, el contenido azul. Cada dígito puede tomar un valor del rango 0–9 o A–F. Juntos, estos valores puede especificar un color rojo-verde-azul o RVA.

*Nota:* al especificar colores en RVA, puede utilizar el selector de campos en la interfaz de usuario para determinar el código de color correcto. Basta con colocarse sobre el color para ver un texto con la información deseada.

## Propiedades de collectionnode



El nodo Colección muestra la distribución de valores de un campo numérico relativo a los valores de otro. (Crea gráficos parecidos a los histogramas.) Es útil para ilustrar una variable o un campo cuyos valores cambian con el tiempo. Con los gráficos 3D también puede incluir un eje simbólico que muestra las distribuciones por categoría. [Si desea obtener más información, consulte el tema Pestaña Gráfico de Colección en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create collectionnode
position :collectionnode at ^posX ^posY
# Pestaña "Gráfico"
set :collectionnode.three_D = true
set :collectionnode.collect_field = 'Medicamento'
set :collectionnode.over_field = 'Edad'
set :collectionnode.by_field = 'PS'
set :collectionnode.operation = Sum
# Sección "Superponer"
set :collectionnode.color_field = 'Medicamento'
set :collectionnode.panel_field = 'Sexo'
set :collectionnode.animation_field = ''
# Pestaña "Opciones"
set :collectionnode.range_mode = Automatic
set :collectionnode.range_min = 1
set :collectionnode.range_max = 100
set :collectionnode.bins = ByNumber
set :collectionnode.num_bins = 10
set :collectionnode.bin_width = 5
```

propiedades de collectionnode	Tipo de datos	Descripción de la propiedad
over_field	campo	
over_label_auto	marca	
over_label	cadena	
collect_field	campo	
collect_label_auto	marca	
collect_label	cadena	
three_D	marca	
by_field	campo	
by_label_auto	marca	
by_label	cadena	

propiedades de collectionnode	Tipo de datos	Descripción de la propiedad
operation	Sum Mean Min Max SDev	
color_field	cadena	
panel_field	cadena	
animation_field	cadena	
range_mode	Automatic UserDefined	
range_min	número	
range_max	número	
bins	ByNumber ByWidth	
num_bins	número	
bin_width	número	
use_grid	marca	
graph_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.

## Propiedades de distributionnode



El nodo Distribución muestra las instancias de valores simbólicos (categóricos), como el tipo de hipoteca o el género. Normalmente, podría usar el nodo Distribución para mostrar los desequilibrios de los datos, que pueden rectificarse mediante el nodo Equilibrar antes de crear un modelo. [Si desea obtener más información, consulte el tema Nodo Distribución en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create distributionnode
# Pestaña "Gráfico"
set :distributionnode.plot = Flags
set :distributionnode.x_field = 'Edad'
set :distributionnode.color_field = 'Medicamento'
set :distributionnode.normalize = True
set :distributionnode.sort_mode = ByOccurrence
set :distributionnode.use_proportional_scale = True
```

propiedades de distributionnode	Tipo de datos	Descripción de la propiedad
plot	SelectedFields Flags	
x_field	campo	
color_field	campo	Campo de superposición.
normalize	marca	

propiedades de distributionnode	Tipo de datos	Descripción de la propiedad
sort_mode	ByOccurrence Alphabetic	
use_proportional_scale	marca	

## Propiedades de evaluationnode



El nodo Evaluación ayuda a evaluar y comparar modelos predictivos. El diagrama de evaluación muestra la calidad con que los modelos pronostican resultados particulares. Ordena registros en función del valor pronosticado y la confianza del pronóstico. Divide el registro en grupos de igual tamaño (**cuantiles**) y, a continuación, representa el valor del criterio de negocios de cada cuantil de mayor a menor. El gráfico muestra múltiples modelos como líneas independientes. [Si desea obtener más información, consulte el tema Nodo Evaluación en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create evaluationnode
position :evaluationnode at ^posX ^posY
# Pestaña "Gráfico"
set :evaluationnode.chart_type = Gains
set :evaluationnode.cumulative = False
set :evaluationnode.field_detection_method = Name
set :evaluationnode.inc_baseline = True
set :evaluationnode.n_tile = Deciles
set :evaluationnode.style = Point
set :evaluationnode.point_type = Dot
set :evaluationnode.use_fixed_cost = True
set :evaluationnode.cost_value = 5.0
set :evaluationnode.cost_field = 'Na'
set :evaluationnode.use_fixed_revenue = True
set :evaluationnode.revenue_value = 30.0
set :evaluationnode.revenue_field = 'Edad'
set :evaluationnode.use_fixed_weight = True
set :evaluationnode.weight_value = 2.0
set :evaluationnode.weight_field = 'K'
```

propiedades de evaluationnode	Tipo de datos	Descripción de la propiedad
chart_type	Gains Response Lift Profit ROI	
inc_baseline	marca	
field_detection_method	Metadata Name	
use_fixed_cost	marca	
cost_value	número	
cost_field	cadena	

propiedades de evaluationnode	Tipo de datos	Descripción de la propiedad
use_fixed_revenue	marca	
revenue_value	número	
revenue_field	cadena	
use_fixed_weight	marca	
weight_value	número	
weight_field	campo	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	marca	
style	Line Point	
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
export_data	marca	
data_filename	cadena	
delimiter	cadena	
new_line	marca	
inc_field_names	marca	
inc_best_line	marca	
inc_business_rule	marca	
business_rule_condition	cadena	
plot_score_fields	marca	
score_fields	[campo1 ... campoN]	
target_field	campo	
use_hit_condition	marca	
hit_condition	cadena	
use_score_expression	marca	

propiedades de evaluationnode	Tipo de datos	Descripción de la propiedad
score_expression	cadena	
caption_auto	marca	

## Propiedades de graphboardnode



El nodo Tablero ofrece muchos tipos diferentes de gráficos en un único nodo. Con este nodo puede seleccionar los campos de datos que desee explorar y seleccionar un gráfico de los disponibles para los datos seleccionados. El nodo filtra automáticamente cualquier tipo de gráfico que no sea compatible con las selecciones de campo. [Si desea obtener más información, consulte el tema Nodo Tablero en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

*Nota:* Si establece una propiedad que no es válida para el tipo de gráfico (por ejemplo, especificar y\_field para un histograma), se ignora dicha propiedad.

### Ejemplo

```
create graphboardnode
connect DRUG4n to :graphboardnode
set :graphboardnode.graph_type="Línea"
set :graphboardnode.x_field = "K"
set :graphboardnode.y_field = "Na"
execute :graphboardnode
```

propiedades de graphboard	Tipo de datos	Descripción de la propiedad
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot	Identifica el tipo de gráfico.

propiedades de graphboard	Tipo de datos	Descripción de la propiedad
	Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap PointsOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	
x_field	<i>campo</i>	Especifica una etiqueta personalizada para el eje x. Disponible solamente para etiquetas.
y_field	<i>campo</i>	Especifica una etiqueta personalizada para el eje y. Disponible solamente para etiquetas.
z_field	<i>campo</i>	Se utiliza en algunos gráficos 3D.
color_field	<i>campo</i>	Se utiliza en mapas de calor.
size_field	<i>campo</i>	Se utiliza en gráficos de burbujas.
categories_field	<i>campo</i>	
values_field	<i>campo</i>	
rows_field	<i>campo</i>	
columns_field	<i>campo</i>	
fields	<i>campo</i>	
start_longitude	<i>campo</i>	Se utiliza con flechas en un mapa de referencia.
end_longitude	<i>campo</i>	
start_latitude	<i>campo</i>	
end_latitude	<i>campo</i>	
data_key_field	<i>campo</i>	Se utiliza en diversos mapas.
panelrow_field	<i>cadena</i>	
panelcol_field	<i>cadena</i>	
animation_field	<i>cadena</i>	
longitude	<i>campo</i>	Se utiliza en mapas de coordenadas.
latitude	<i>campo</i>	
map_color_field	<i>campo</i>	



## Propiedades de histogramnode



El nodo Histograma muestra las instancias de valores de los campos numéricos. Se suele utilizar para explorar los datos antes de las manipulaciones y la generación de modelos. Al igual que con el nodo Distribución, con frecuencia el nodo Histograma detecta desequilibrios en los datos. [Si desea obtener más información, consulte el tema Pestaña Gráfico de Histograma en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create histogramnode
position :histogramnode at ^posX ^posY
# Pestaña "Gráfico"
set :histogramnode.field = 'Medicamento'
set :histogramnode.color_field = 'Medicamento'
set :histogramnode.panel_field = 'Sexo'
set :histogramnode.animation_field = ''
# Pestaña "Opciones"
set :histogramnode.range_mode = Automatic
set :histogramnode.range_min = 1.0
set :histogramnode.range_max = 100.0
set :histogramnode.num_bins = 10
set :histogramnode.bin_width = 10
set :histogramnode.normalize = True
set :histogramnode.separate_bands = False
```

propiedades de histogramnode	Tipo de datos	Descripción de la propiedad
field	campo	
color_field	campo	
panel_field	campo	
animation_field	campo	
range_mode	Automatic UserDefined	
range_min	número	
range_max	número	
bins	ByNumber ByWidth	
num_bins	número	
bin_width	número	
normalize	marca	
separate_bands	marca	
x_label_auto	marca	
x_label	cadena	
y_label_auto	marca	
y_label	cadena	
use_grid	marca	
graph_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.

propiedades de histogramnode	Tipo de datos	Descripción de la propiedad
page_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.
normal_curve	marca	Indica si se debe mostrar la curva de distribución normal en los resultados.

## Propiedades de multiplotnode



El nodo G. múltiple crea un gráfico que muestra varios campos *Y* sobre un campo *X* único. Los campos *Y* están representados como líneas coloreadas; cada uno equivale a un nodo Gráfico con el estilo establecido en Línea y el Modo para *X* establecido en Ordenar. Los gráficos múltiples son útiles cuando quiere explorar la fluctuación de varias variables a través del tiempo. [Si desea obtener más información, consulte el tema Nodo G. múltiple en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create multiplotnode
# Pestaña "Gráfico"
set :multiplotnode.x_field = 'Edad'
set :multiplotnode.y_fields = ['Medicamento' 'PS']
set :multiplotnode.panel_field = 'Sexo'
# Sección "Superponer"
set :multiplotnode.animation_field = ''
set :multiplotnode.tooltip = "prueba"
set :multiplotnode.normalize = True
set :multiplotnode.use_overlay_expr = False
set :multiplotnode.overlay_expression = "prueba"
set :multiplotnode.records_limit = 500
set :multiplotnode.if_over_limit = PlotSample
```

propiedades de multiplotnode	Tipo de datos	Descripción de la propiedad
x_field	campo	
y_fields	[campo campo campo]	
panel_field	campo	
animation_field	campo	
normalize	marca	
use_overlay_expr	marca	
overlay_expression	cadena	
records_limit	número	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	marca	
x_label	cadena	
y_label_auto	marca	
y_label	cadena	

propiedades de multiplotnode	Tipo de datos	Descripción de la propiedad
use_grid	marca	
graph_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.

## Propiedades de plotnode



El nodo Gráfico muestra la relación entre los campos numéricos. Puede crear un gráfico mediante puntos (un diagrama de dispersión) o líneas. [Si desea obtener más información, consulte el tema Nodo Gráfico en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create plotnode
# Pestaña "Gráfico"
set :plotnode.three_D = True
set :plotnode.x_field = 'PS'
set :plotnode.y_field = 'Colesterol'
set :plotnode.z_field = 'Medicamento'
# Sección "Superponer"
set :plotnode.color_field = 'Medicamento'
set :plotnode.size_field = 'Edad'
set :plotnode.shape_field = ''
set :plotnode.panel_field = 'Sexo'
set :plotnode.animation_field = 'PS'
set :plotnode.transp_field = ''
set :plotnode.style = Point
# Pestaña "Resultados"
set :plotnode.output_mode =
set :plotnode.output_format = JPEG
set :plotnode.full_filename = "C:/Temp/Graph_Output/plot_output.jpeg"
```

propiedades de plotnode	Tipo de datos	Descripción de la propiedad
x_field	campo	Especifica una etiqueta personalizada para el eje x. Disponible solamente para etiquetas.
y_field	campo	Especifica una etiqueta personalizada para el eje y. Disponible solamente para etiquetas.
three_D	marca	Especifica una etiqueta personalizada para el eje z. Disponible solamente para etiquetas en gráficos 3D.
z_field	campo	
color_field	campo	Campo de superposición.
size_field	campo	
shape_field	campo	

propiedades de plotnode	Tipo de datos	Descripción de la propiedad
panel_field	<i>campo</i>	Especifica un campo de marcas o nominal para crear un gráfico independiente para cada categoría. Los gráficos aparecerán juntos en una ventana de resultados.
animation_field	<i>campo</i>	Especifica un campo de marcas o nominal para ilustrar las categorías de los valores de datos creando una serie de gráficos secuenciados mediante la animación.
transp_field	<i>campo</i>	Especifica un campo para ilustrar las categorías de los valores de datos utilizando un nivel de transparencia distinto para cada categoría. No disponible para gráficos de líneas.
overlay_type	None Smoother Function	Determina si se muestra una función superpuesta o suavizamiento LOESS.
overlay_expression	<i>cadena</i>	Especifica la expresión utilizada si <i>overlay_type</i> se establece en Function.
style	Point Line	
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	<i>número</i>	
x_range_max	<i>número</i>	
y_range_mode	Automatic UserDefined	
y_range_min	<i>número</i>	
y_range_max	<i>número</i>	
z_range_mode	Automatic UserDefined	

propiedades de plotnode	Tipo de datos	Descripción de la propiedad
z_range_min	número	
z_range_max	número	
jitter	marca	
records_limit	número	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	marca	
x_label	cadena	
y_label_auto	marca	
y_label	cadena	
z_label_auto	marca	
z_label	cadena	
use_grid	marca	
graph_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	colores	Al principio de esta sección se describen los colores de gráficos estándar.
use_overlay_expr	marca	Desaprobado en favor de overlay_type.

## Propiedades de timeplotnode



El nodo Gráfico de tiempo muestra uno o más conjuntos de datos de series temporales. Normalmente, primero se utilizaría un nodo Intervalos de tiempo para crear un campo *EtiquetaTiempo*, que se utilizaría para etiquetar el eje *x*. [Si desea obtener más información, consulte el tema Nodo Gráfico de tiempo en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create timeplotnode
set :timeplotnode.y_fields = ['ventas' 'hombres' 'mujeres']
set :timeplotnode.panel = True
set :timeplotnode.normalize = True
set :timeplotnode.line = True
set :timeplotnode.smoother = True
set :timeplotnode.use_records_limit = True
set :timeplotnode.records_limit = 2000
# Configuración de aspecto
set :timeplotnode.symbol_size = 2.0
```

propiedades de timeplotnode	Tipo de datos	Descripción de la propiedad
plot_series	Series Models	
use_custom_x_field	marca	
x_field	campo	

propiedades de timeplotnode	Tipo de datos	Descripción de la propiedad
y_fields	[ <i>campo campo</i> <i>campo</i> ]	
panel	<i>marca</i>	
normalize	<i>marca</i>	
line	<i>marca</i>	
points	<i>marca</i>	
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
smoother	<i>marca</i>	Puede añadir suavizamientos al gráfico únicamente si establece panel en True.
use_records_limit	<i>marca</i>	
records_limit	<i>entero</i>	
symbol_size	<i>número</i>	Especifica el tamaño del símbolo. Por ejemplo: <code>set :webnode.symbol_size = 5.5</code> crea un tamaño de símbolo mayor que el tamaño por defecto.
panel_layout	Horizontal Vertical	

## Propiedades de webnode



El nodo Malla muestra la fuerza de las relaciones entre los valores de dos o más campos simbólicos (categóricos). El gráfico utiliza líneas de varios anchos para indicar la fuerza de la conexión. Podría utilizar un nodo Malla, por ejemplo, para explorar las relaciones existentes entre la compra de un conjunto de elementos en un sitio de comercio electrónico. [Si desea obtener más información, consulte el tema Nodo Malla en el capítulo 5 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```

create webnode
# Pestaña "Gráfico"
set :webnode.use_directed_web = True
set :webnode.to_field = 'Medicamento'
set :webnode.fields = ['PS' 'Colesterol' 'Sexo' 'Medicamento']
set :webnode.from_fields = ['PS' 'Colesterol' 'Sexo']
set :webnode.true_flags_only = False
set :webnode.line_values = Absolute
set :webnode.strong_links_heavier = True
# Pestaña "Opciones"
set :webnode.max_num_links = 300
set :webnode.links_above = 10
set :webnode.num_links = ShowAll
set :webnode.discard_links_min = True
set :webnode.links_min_records = 5
set :webnode.discard_links_max = True
set :webnode.weak_below = 10
set :webnode.strong_above = 19
set :webnode.link_size_continuous = True
set :webnode.web_display = Circular

```

propiedades de webnode	Tipo de datos	Descripción de la propiedad
use_directed_web	marca	
fields	[campo campo campo]	
to_field	campo	
from_fields	[campo campo campo]	
true_flags_only	marca	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	marca	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	número	
links_above	número	
discard_links_min	marca	
links_min_records	número	
discard_links_max	marca	
links_max_records	número	
weak_below	número	
strong_above	número	
link_size_continuous	marca	

<b>propiedades de webnode</b>	<b>Tipo de datos</b>	<b>Descripción de la propiedad</b>
web_display	Circular Network Directed Grid	
graph_background	<i>colores</i>	Al principio de esta sección se describen los colores de gráficos estándar.
symbol_size	<i>número</i>	Especifica el tamaño del símbolo. Por ejemplo: <code>set :webnode.symbol_size = 5.5</code> crea un tamaño de símbolo mayor que el tamaño por defecto.



# Propiedades de nodos de modelado

## Propiedades comunes de nodos de modelado

Las siguientes propiedades son comunes a algunos o todos los nodos de modelado. Las excepciones se indican en la documentación de los nodos de modelado individuales según sea adecuado.

Propiedad	Valores	Descripción de la propiedad
custom_fields	<i>marca</i>	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
target o targets	<i>campo</i> o [ <i>campo1 ... campoN</i> ]	Especifica un único campo objetivo o varios campos objetivo dependiendo del tipo de modelo.
inputs	[ <i>campo1 ... campoN</i> ]	Campos de entrada o predictor utilizados por el modelo.
partition	<i>campo</i>	
use_partitioned_data	<i>marca</i>	Si se ha definido un campo de partición, esta opción garantiza que solamente se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
use_split_data	<i>marca</i>	
splits	[ <i>campo1 ... campoN</i> ]	Especifica el campo o campos para utilizar en el modelado de divisiones. Eficaz sólo si <i>use_split_data</i> se establece como True.
use_frequency	<i>marca</i>	Los modelos específicos utilizan campos de ponderación y frecuencia como se indica en cada tipo de modelo.
frequency_field	<i>campo</i>	
use_weight	<i>marca</i>	
weight_field	<i>campo</i>	
use_model_name	<i>marca</i>	
model_name	<i>cadena</i>	Nombre personalizado para nuevo modelo.
mode	Simple Expert	

## Propiedades de anomalydetectionnode



El nodo Detección de anomalías identifica casos extraños, o valores atípicos, que no se ajustan a patrones de datos “normales”. Con este nodo, es posible identificar valores atípicos aunque no se ajusten a ningún patrón previamente conocido o no se realice una búsqueda exacta. [Si desea obtener más información, consulte el tema Nodo Detección de anomalías en el capítulo 4 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create anomalydetectionnode
set :anomalydetectionnode.anomaly_method=PerRecords
set :anomalydetectionnode.percent_records=95
set :anomalydetectionnode.mode=Expert
set :anomalydetectionnode.peer_group_num_auto=true
set :anomalydetectionnode.min_num_peer_groups=3
set :anomalydetectionnode.max_num_peer_groups=10
```

Propiedades de anomalydetectionnode	Valores	Descripción de la propiedad
inputs	<i>[campo1 ... campoN]</i>	Los modelos de detección de anomalías criban registros basándose en los campos de entrada especificados. No utilizan un campo objetivo. Los campos de ponderación y frecuencia tampoco se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	Especifica el método utilizado para determinar el valor de corte para marcar los registros como anómalos.
index_level	<i>número</i>	Especifica el valor de corte mínimo para el marcado de anomalías.
percent_records	<i>número</i>	Establece el umbral para marcar registros de acuerdo al porcentaje de registros en los datos de entrenamiento.
num_records	<i>número</i>	Establece el umbral para marcar registros de acuerdo al número de registros en los datos de entrenamiento.
num_fields	<i>entero</i>	El número de campos de los que se va a informar por cada registro anómalo.
impute_missing_values	<i>marca</i>	
adjustment_coeff	<i>número</i>	Valor que se usa para equilibrar la ponderación relativa asignada a los campos categóricos y continuos al calcular la distancia.

Propiedades de anomalydetectionnode	Valores	Descripción de la propiedad
peer_group_num_auto	marca	Calcula automáticamente el número de grupos de homólogos.
min_num_peer_groups	entero	Especifica el número mínimo de grupos de homólogos empleado cuando peer_group_num_auto está definido como True.
max_num_per_groups	entero	Especifica el número máximo de grupos de homólogos.
num_peer_groups	entero	Especifica el número de grupos de homólogos empleado si peer_group_num_auto está definido como False.
noise_level	número	Determina el modo en que se tratan los valores atípicos durante el conglomerado. Especifique un valor entre 0 y 0.5.
noise_ratio	número	Especifica la parte de memoria asignada al componente que debería usarse para el almacenamiento en búfer de ruido. Especifique un valor entre 0 y 0.5.

## Propiedades de apriorinode



El nodo A priori extrae un conjunto de reglas de los datos y destaca aquellas reglas con un mayor contenido de información. A priori ofrece cinco métodos diferentes para la selección de reglas y utiliza un sofisticado esquema de indización para procesar eficientemente grandes conjuntos de datos. En los problemas de mucho volumen, A priori se entrena más rápidamente, no tiene un límite arbitrario para el número de reglas que puede retener y puede gestionar reglas que tengan hasta 32 precondiciones. A priori requiere que todos los campos de entrada y salida sean categóricos, pero ofrece un mejor rendimiento ya que está optimizado para este tipo de datos. [Si desea obtener más información, consulte el tema Nodo A priori en el capítulo 12 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create apriorinode
# Pestaña "Campos"
set :apriorinode.custom_fields = True
set :apriorinode.use_transactional_data = True
set :apriorinode.id_field = 'Age'
set :apriorinode.contiguous = True
set :apriorinode.content_field = 'Drug'
# Parecen haber cambiado, solían ser:
#help set :apriorinode.consequents = ['Age']
#help set :apriorinode.antecedents = ['BP' 'Cholesterol' 'Drug']
# ahora parece que tenemos;
#help set :apriorinode.content = ['Age']
set :apriorinode.partition = Test
# Pestaña "Modelo"
```

```

set :apriorinode.use_model_name = False
set :apriorinode.model_name = "Apriori_bp_choles_drug"
set :apriorinode.min_supp = 7.0
set :apriorinode.min_conf = 30.0
set :apriorinode.max_antecedents = 7
set :apriorinode.true_flags = False
set :apriorinode.optimize = Memory
# Pestaña "Experto"
set :apriorinode.mode = Expert
set :apriorinode.evaluation = ConfidenceRatio
set :apriorinode.lower_bound = 7

```

Propiedades de apriorinode	Valores	Descripción de la propiedad
consequents	<i>campo</i>	Los modelos A priori utilizan Consecuentes y Antecedentes en lugar de los campos estándar objetivo y de entrada. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
antecedents	[ <i>campo1 ... campoN</i> ]	
min_supp	<i>número</i>	
min_conf	<i>número</i>	
max_antecedents	<i>número</i>	
true_flags	<i>marca</i>	
optimize	Speed Memory	
use_transactional_data	<i>marca</i>	
contiguous	<i>marca</i>	
id_field	<i>cadena</i>	
content_field	<i>cadena</i>	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	<i>número</i>	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.

## Propiedades de autotransformación



El nodo Clasificador automático crea y compara varios modelos diferentes para obtener resultados binarios (sí o no, pérdida o no de clientes, etc.), lo que le permite seleccionar el mejor enfoque para un análisis determinado. Son compatibles varios algoritmos de modelado, por lo que es posible seleccionar los métodos que desee utilizar, las opciones específicas para cada uno y los criterios para comparar los resultados. El nodo genera un conjunto de modelos basado en las opciones especificadas y clasifica los mejores candidatos en función de los criterios que especifique. [Si desea obtener más información, consulte el tema Nodo Clasificador automático en el capítulo 5 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create autotransformación
set :autotransformación.ranking_measure=Accuracy
set :autotransformación.ranking_dataset=Training
set :autotransformación.enable_accuracy_limit=true
set :autotransformación.accuracy_limit=0.9
set :autotransformación.calculate_variable_importance=true
set :autotransformación.use_costs=true
set :autotransformación.svm=false
```

Propiedades de autotransformación	Valores	Descripción de la propiedad
target	campo	En objetivos de marca, el nodo Clasificador binario requiere un único campo objetivo y uno o más campos de entrada. También se puede especificar campos de ponderación y frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	entero	Número de modelos que se incluirán en el nugget de modelo. Especifique un entero entre 1 y 100.
calculate_variable_importance	marca	
enable_accuracy_limit	marca	
accuracy_limit	entero	Entero entre 0 y 100.
enable_area_under_curve_limit	marca	
area_under_curve_limit	número	Número real entre 0,0 y 1,0.
enable_profit_limit	marca	
profit_limit	número	Entero mayor que 0.
enable_lift_limit	marca	
lift_limit	número	Número real mayor que 1,0.

Propiedades de autotransformador	Valores	Descripción de la propiedad
enable_number_of_variables_limit	marca	
number_of_variables_limit	número	Entero mayor que 0.
use_fixed_cost	marca	
fixed_cost	número	Número real mayor que 0.0.
variable_cost	campo	
use_fixed_revenue	marca	
fixed_revenue	número	Número real mayor que 0.0.
variable_revenue	campo	
use_fixed_weight	marca	
fixed_weight	número	Número real mayor que 0,0
variable_weight	campo	
lift_percentile	número	Entero entre 0 y 100.
enable_model_build_time_limit	marca	
model_build_time_limit	número	Entero que indica el número máximo de minutos que se puede tardar en generar cada uno de los modelos.
enable_stop_after_time_limit	marca	
stop_after_time_limit	número	Número real que indica el número máximo de horas que puede tardar una ejecución del clasificador automático.
enable_stop_after_valid_model_produced	marca	
use_costs	marca	
<algorithm>	marca	Activa o desactiva el uso de un determinado algoritmo, por ejemplo: set :autotransformador.chaid=true
<algorithm>.<property>	cadena	Define un valor de propiedad para un algoritmo específico. <a href="#">Si desea obtener más información, consulte el tema Definición de propiedades de algoritmo el p. 200.</a>

### Definición de propiedades de algoritmo

En el caso de los nodos Clasificador automático, Autonumérico y Autoconglomeración, las propiedades de determinados algoritmos utilizados por el nodo se pueden establecer utilizando el formato general:

```
set :autotransformador.<algorithm>.<property> = <value>
```

```
set :autonumericonode.<algorithm>.<property> = <value>
```

```
set :autoclusternode.<algorithm>.<propiedad> = <valor>
```

Por ejemplo:

```
set :autotransformador.neuralnetwork.method = MultilayerPerceptron
```

Los nombres de algoritmos del nodo Clasificador automático son `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` y `knn`.

Los nombres de algoritmos del nodo Autonumérico son `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` y `knn`.

Los nombres de algoritmos del nodo Autoconglomerado son `twostep`, `k-means` y `kohonen`.

Los nombres de las propiedades son los nombres estándar, según se han documentado para cada nodo de algoritmo.

Las propiedades de algoritmos que contienen puntos u otros signos de puntuación deben encerrarse entre comillas simples. Por ejemplo:

```
set :autoclassifiernode.logreg.tolerance = '1.0E-5'
```

También es posible asignar varios valores a una propiedad. Por ejemplo:

```
set :autoclassifiernode.decisionlist.search_direction = [Up Down]
```

Para activar o desactivar el uso de un determinado algoritmo:

```
set :autoclassifiernode.chaid=true
```

*Notas:*

- se deben utilizar minúsculas al utilizar los valores `true` y `false` (en vez de `False`).
- En los casos en los que determinadas opciones de algoritmos no están disponibles en el nodo Clasificador automático o cuando sólo se puede especificar un único valor, en lugar de un intervalo de valores, se aplican los mismos límites que tienen los procesos cuando se accede al nodo de la manera estándar.

## ***Propiedades de nodo de autoconglomeración***



El nodo Autoconglomeración calcula y compara los modelos de conglomerado que identifican grupos de registros con características similares. El nodo funciona de la misma manera que otros nodos de modelado de conglomerado, permitiéndole experimentar con múltiples combinaciones de opciones en una única pasada de modelado. Los modelos se pueden comparar utilizando medidas básicas con las que se intenta filtrar y definir la utilidad de los modelos de conglomerado y proporcionar una medida según la importancia de campos concretos. [Si desea obtener más información, consulte el tema Nodo Autoconglomeración en el capítulo 5 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### ***Ejemplo***

```
create autoclusternode
set :autoclusternode.ranking_measure=Silhouette
set :autoclusternode.ranking_dataset=Training
set :autoclusternode.enable_silhouette_limit=true
```

```
set :autoclusternode.silhouette_limit=5
```

Propiedades de autoclusternode	Valores	Descripción de la propiedad
evaluation	<i>campo</i>	<i>Nota:</i> nodo Autoconglomeración únicamente. Identifica el campo cuyo valor de importancia se calculará. Como alternativa, se puede utilizar para identificar la calidad con la que el conglomerado diferencia el valor de este campo y, por lo tanto; la calidad con la que el modelo pronosticará este campo.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	<i>entero</i>	Número de modelos que se incluirán en el informe. Especifique un entero entre 1 y 100.
enable_silhouette_limit	<i>marca</i>	
silhouette_limit	<i>entero</i>	Entero entre 0 y 100.
enable_number_less_limit	<i>marca</i>	
number_less_limit	<i>número</i>	Número real entre 0,0 y 1,0.
enable_number_greater_limit	<i>marca</i>	
number_greater_limit	<i>número</i>	Entero mayor que 0.
enable_smallest_cluster_limit	<i>marca</i>	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	<i>número</i>	
smallest_cluster_limit_count	<i>entero</i>	Entero mayor que 0.
enable_largest_cluster_limit	<i>marca</i>	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percentage	<i>número</i>	
largest_cluster_limit_count	<i>entero</i>	
enable_smallest_largest_limit	<i>marca</i>	
smallest_largest_limit	<i>número</i>	
enable_importance_limit	<i>marca</i>	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	<i>número</i>	Entero entre 0 y 100.
importance_limit_less_than	<i>número</i>	Entero entre 0 y 100.



Propiedades de autoclusternode	Valores	Descripción de la propiedad
<algorithm>	<i>marca</i>	Activa o desactiva el uso de un determinado algoritmo, por ejemplo: set :autoclusternode.kohonen=true
<algorithm>.<property>	<i>cadena</i>	Define un valor de propiedad para un algoritmo específico. <a href="#">Si desea obtener más información, consulte el tema Definición de propiedades de algoritmo el p. 200.</a>

## Propiedades de autonumericnode



El nodo Autonumérico calcula y compara modelos para resultados de rango numérico continuo utilizando cierto número de métodos diferentes. El nodo funciona de la misma manera que el nodo Clasificador automático, lo que le permite seleccionar los algoritmos que desee utilizar y experimentar con varias combinaciones de opciones en una única pasada de modelado. Los algoritmos admitidos incluyen redes neuronales, C&RT, CHAID, regresión lineal, regresión lineal generalizada y máquinas de vectores de soporte (SVM). Los modelos se pueden comparar basándose en la correlación, el error relativo o el número de variables utilizado. [Si desea obtener más información, consulte el tema Nodo Autonumérico en el capítulo 5 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create autonumericnode
set :autonumericnode.ranking_measure=Correlation
set :autonumericnode.ranking_dataset=Training
set :autonumericnode.enable_correlation_limit=true
set :autonumericnode.correlation_limit=0.8
set :autonumericnode.calculate_variable_importance=true
set :autonumericnode.neuralnetwork=true
set :autonumericnode.chaid=false
```

Propiedades de autonumericnode	Valores	Descripción de la propiedad
custom_fields	<i>marca</i>	Si es verdadero, se utilizará la configuración de campos personalizada en lugar de la configuración del nodo Tipo.
target	<i>campo</i>	El nodo Autonumérico requiere un único campo objetivo y uno o más campos de entrada. También se puede especificar campos de ponderación y frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
inputs	<i>[campo1 ... campo2]</i>	
partition	<i>campo</i>	
use_frequency	<i>marca</i>	
frequency_field	<i>campo</i>	

Propiedades de autonumericnode	Valores	Descripción de la propiedad
use_weight	marca	
weight_field	campo	
use_partitioned_data	marca	Si se ha definido un campo de partición, sólo se utilizarán los datos de entrenamiento para la generación del modelo.
ranking_measure	Correlation NumberOfFields	
ranking_dataset	Test Training	
number_of_models	entero	Número de modelos que se incluirán en el nugget de modelo. Especifique un entero entre 1 y 100.
calculate_variable_importance	marca	
enable_correlation_limit	marca	
correlation_limit	entero	
enable_number_of_fields_limit	marca	
number_of_fields_limit	entero	
enable_relative_error_limit	marca	
relative_error_limit	entero	
enable_model_build_time_limit	marca	
model_build_time_limit	entero	
enable_stop_after_time_limit	marca	
stop_after_time_limit	entero	
stop_if_valid_model	marca	
<algorithm>	marca	Activa o desactiva el uso de un determinado algoritmo, por ejemplo: set :autonumericnode.chaid=true
<algorithm>.<property>	cadena	Define un valor de propiedad para un algoritmo específico. <a href="#">Si desea obtener más información, consulte el tema Definición de propiedades de algoritmo el p. 200.</a>

## Propiedades de bayesnetnode



El nodo Red bayesiana le permite crear un modelo de probabilidad combinando pruebas observadas y registradas con conocimiento del mundo real para establecer la probabilidad de instancias. El nodo se centra en las redes Naïve Bayes aumentado a árbol (TAN) y de cadena de Markov que se utilizan principalmente para la clasificación. [Si desea obtener más información, consulte el tema Nodo Red bayesiana en el capítulo 7 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create bayesnetnode
set :bayesnetnode.continue_training_existing_model = True
set :bayesnetnode.structure_type = MarkovBlanket
```

```

set :bayesnetnode.use_feature_selection = True
# Expert tab
set :bayesnetnode.mode = Expert
set :bayesnetnode.all_probabilities = True
set :bayesnetnode.independence = Pearson

```

Propiedades de bayesnetnode	Valores	Descripción de la propiedad
inputs	<i>[campo1 ... campoN]</i>	Los modelos de red bayesiana utilizan un único campo objetivo y uno o más campos de entrada. Los campos continuos se establecen en intervalos automáticamente. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue_training_existing_model	<i>marca</i>	
structure_type	TAN MarkovBlanket	Seleccione la estructura que desea utilizar cuando cree la red bayesiana.
use_feature_selection	<i>marca</i>	
parameter_learning_method	Likelihood Bayes	Especifica el método utilizado para calcular las tablas de probabilidad condicional entre nodos donde se conocen los valores de los elementos principales.
mode	Expert Simple	
missing_values	<i>marca</i>	
all_probabilities	<i>marca</i>	
independence	Likelihood Pearson	Especifica el método utilizado para determinar si las observaciones relacionadas de dos variables son independientes entre sí.
significance_level	<i>número</i>	Especifica el valor de corte para determinar la independencia.
maximal_conditioning_set	<i>número</i>	Establece el número máximo de variables de condición que se utilizarán para la comprobación de la independencia.
inputs_always_selected	<i>[campo1 ... campoN]</i>	Especifica qué campos del conjunto de datos se deben utilizar siempre al generar la red bayesiana. <i>Nota:</i> el campo objetivo siempre está seleccionado.
maximum_number_inputs	<i>número</i>	Especifica el número máximo de campos de entrada que se deben utilizar al generar la red bayesiana.
calculate_variable_importance	<i>marca</i>	
calculate_raw_propensities	<i>marca</i>	
calculate_adjusted_propensities	<i>marca</i>	
adjusted_propensity_partition	Test Validation	

## Propiedades de c50node



El nodo C5.0 genera un árbol de decisión o un conjunto de reglas. El modelo divide la muestra basándose en el campo que ofrece la máxima ganancia de información en cada nivel. El campo objetivo debe ser categórico. Se permiten varias divisiones en más de dos subgrupos. [Si desea obtener más información, consulte el tema Nodo C5.0 en el capítulo 6 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create c50node
# Pestaña "Modelo"
set :c50node.use_model_name = False
set :c50node.model_name = "C5_Drug"
set :c50node.use_partitioned_data = True
set :c50node.output_type = DecisionTree
set :c50node.use_xval = True
set :c50node.xval_num_folds = 3
set :c50node.mode = Expert
set :c50node.favor = Generality
set :c50node.min_child_records = 3
# Pestaña "Costes"
set :c50node.use_costs = True
set :c50node.costs = [{"drugA" "drugX" 2}]
```

Propiedades de c50node	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos C50 utilizan un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de ponderación. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
output_type	DecisionTree RuleSet	
group_symbolics	<i>marca</i>	
use_boost	<i>marca</i>	
boost_num_trials	<i>número</i>	
use_xval	<i>marca</i>	
xval_num_folds	<i>número</i>	
mode	Simple Expert	
favor	Accuracy Generality	Generalización o precisión de favor.
expected_noise	<i>número</i>	
min_child_records	<i>número</i>	
pruning_severity	<i>número</i>	
use_costs	<i>marca</i>	
costs	<i>estructurado</i>	Ésta es una propiedad estructurada.
use_winning	<i>marca</i>	
use_global_pruning	<i>marca</i>	Activado (True) por defecto.

Propiedades de c50node	Valores	Descripción de la propiedad
calculate_variable_importance	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de carmanode



El modelo CARMA extrae un conjunto de reglas de los datos sin necesidad de especificar campos de entrada ni de objetivo. A diferencia de A priori, el nodo CARMA ofrece configuraciones de generación basadas en el soporte de las reglas (soporte tanto para el antecedente como el consecuente) en lugar de hacerlo sólo respecto al soporte del antecedente. Esto significa que las reglas generadas se pueden utilizar en una gama de aplicaciones más amplia, por ejemplo, para buscar una lista de productos o servicios (antecedentes) cuyo consecuente es el elemento que se desea promocionar durante esta temporada de vacaciones. [Si desea obtener más información, consulte el tema Nodo CARMA en el capítulo 12 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create carmanode
# Pestaña "Campos"
set :carmanode.custom_fields = True
set :carmanode.use_transactional_data = True
set :carmanode.inputs = ['BP' 'Cholesterol' 'Drug']
set :carmanode.partition = Test
# Pestaña "Modelo"
set :carmanode.use_model_name = False
set :carmanode.model_name = "age_bp_drug"
set :carmanode.use_partitioned_data = False
set :carmanode.min_supp = 10.0
set :carmanode.min_conf = 30.0
set :carmanode.max_size = 5
# Opciones de Expert
set :carmanode.mode = Expert
#help set :carmanode.exclude_simple = True
set :carmanode.use_pruning = True
set :carmanode.pruning_value = 300
set :carmanode.vary_support = True
set :carmanode.estimated_transactions = 30
```

```
set :carmanode.rules_without_antecedents = True
```

Propiedades de carmanode	Valores	Descripción de la propiedad
inputs	<i>[field1 ... fieldn]</i>	Los modelos CARMA utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
id_field	<i>campo</i>	Campo utilizado como el campo de ID para la generación del modelo.
contiguous	<i>marca</i>	Se utiliza para especificar si los ID del campo de ID son contiguos.
use_transactional_data	<i>marca</i>	
content_field	<i>campo</i>	
min_supp	<i>número (porcentaje)</i>	Está relacionado con el soporte de regla en lugar de con el soporte de antecedentes. El valor por defecto es 20%.
min_conf	<i>número (porcentaje)</i>	El valor por defecto es 20%.
max_size	<i>número</i>	El valor predeterminado es 10.
mode	Simple Expert	El valor por defecto es Simple (del inglés 'True').
exclude_multiple	<i>marca</i>	Excluye las reglas con varios consecuentes. El valor por defecto es False (del inglés 'True').
use_pruning	<i>marca</i>	El valor por defecto es False (del inglés 'True').
pruning_value	<i>número</i>	El valor predeterminado es 500.
vary_support	<i>marca</i>	
estimated_transactions	<i>entero</i>	
rules_without_antecedents	<i>marca</i>	

## Propiedades de cartnode



El nodo de árbol de clasificación y regresión (C&R) genera un árbol de decisión que permite pronosticar o clasificar observaciones futuras. El método utiliza la partición reiterada para dividir los registros de entrenamiento en segmentos minimizando las impurezas en cada paso, donde un nodo se considera "puro" si el 100% de los casos del nodo corresponden a una categoría específica del campo objetivo. Los campos de entrada y objetivo pueden ser continuos (rango numérico) o categóricos (nominal, ordinal o marca). Todas las divisiones son binarias (sólo se crean dos subgrupos). [Si desea obtener más información, consulte el tema Nodo Árbol C&R en el capítulo 6 en Nodos de modelado de IBM SPSS Modeler 15.](#)

### Ejemplo

```
create cartnode
# "Fields" tab
set :cartnode.custom_fields = True
```

```

set :cartnode.target = 'Drug'
set :cartnode.inputs = ['Age' 'BP' 'Cholesterol']
# "Build Options" tab, 'Objective' panel
set :cartnode.model_output_type = InteractiveBuilder
set :cartnode.use_tree_directives = True
set :cartnode.tree_directives = ""Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""
# "Build Options" tab, 'Basics' panel
set :cartnode.prune_tree = False
set :cartnode.use_std_err_rule = True
set :cartnode.std_err_multiplier = 3.0
set :cartnode.max_surrogates = 7
# "Build Options" tab, 'Stopping Rules' panel
set :cartnode.use_percentage = True
set :cartnode.min_parent_records_pc = 5
set :cartnode.min_child_records_pc = 3
# "Build Options" tab, 'Costs & Priors' panel
set :cartnode.use_costs = True
set :cartnode.costs = [{"drugA" "drugX" 2}]
set :cartnode.priors = Custom
# custom priors must add to 1
set :cartnode.custom_priors = [{"drugA" 0.3}{drugX" 0.7}]
set :cartnode.adjust_priors = True
# "Build Options" tab, 'Advanced' panel
set :cartnode.min_impurity = 0.0003
set :cartnode.impurity_measure = Twoing
# "Model Options" tab
set :cartnode.use_model_name = False
set :cartnode.model_name = "Cart_Drug"

```

Propiedades de cartnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos de árbol C&R requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue_training_existing_model	<i>marca</i>	
objective	Standard Boosting Bagging psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>marca</i>	

Propiedades de cartnode	Valores	Descripción de la propiedad
tree_directives	<i>cadena</i>	Especifique directivas para desarrollar el árbol. Las directivas se pueden escribir entre comillas triples para evitar comillas o saltos de línea no deseados. Recuerde que las directivas pueden ser muy sensibles a las pequeñas modificaciones de las opciones de modelado o los datos y es posible que no se puedan generalizar para otros conjuntos de datos. <a href="#">Si desea obtener más información, consulte el tema Directivas de desarrollo de árboles en el capítulo 6 en <i>Nodos de modelado de IBM SPSS Modeler 15</i>.</a>
use_max_depth	Default Custom	
max_depth	<i>entero</i>	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
prune_tree	<i>marca</i>	Poda del árbol para evitar sobreajustes.
use_std_err	<i>marca</i>	Use la diferencia máxima en riesgos (en errores típicos).
std_err_multiplier	<i>número</i>	Diferencia máxima.
max_surrogates	<i>número</i>	Número máximo de sustitutos.
use_percentage	<i>marca</i>	
min_parent_records_pc	<i>número</i>	
min_child_records_pc	<i>número</i>	
min_parent_records_abs	<i>número</i>	
min_child_records_abs	<i>número</i>	
use_costs	<i>marca</i>	
costs	<i>estructurado</i>	Propiedad estructurada que utiliza la siguiente forma: [[drugA drugB 1.5] {drugA drugC 2.1}], donde los argumentos entre llaves ({} ) son los costes pronosticados actuales.
priors	Data Equal Custom	
custom_priors	<i>estructurado</i>	Propiedad estructurada que utiliza la siguiente forma: set :cartnode. custom_priors = [ { drugA 0.3 } { drugB 0.6 } ]
adjust_priors	<i>marca</i>	
trails	<i>número</i>	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Mean Median	Regla de combinación predeterminada para objetivos continuos.



Propiedades de cartnode	Valores	Descripción de la propiedad
large_boost	marca	Aplicar aumento a conjunto de datos muy grandes.
min_impurity	número	
impurity_measure	Gini Twoing Ordered	
train_pct	número	Conjunto de prevención sobreajustado.
set_random_seed	marca	Opción replicar resultados.
seed	número	
calculate_variable_importance	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de chaidnode



El nodo CHAID genera árboles de decisión utilizando estadísticos de chi-cuadrado para identificar las divisiones óptimas. A diferencia de los nodos C&RT y QUEST, CHAID puede generar árboles no binarios, lo que significa que algunas divisiones generarán más de dos ramas. Los campos de entrada y objetivo pueden ser continuos (rango numérico) o categóricos. CHAID exhaustivo es una modificación de CHAID que examina con mayor precisión todas las divisiones posibles, aunque necesita más tiempo para realizar los cálculos. [Si desea obtener más información, consulte el tema Nodo CHAID en el capítulo 6 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create chaidnode
set :chaidnode.custom_fields = True
set :chaidnode.target = Drug
set :chaidnode.inputs = [Edad Na K Colesterol BP]
set :chaidnode.use_model_name = true
set :chaidnode.model_name = "CHAID"
set :chaidnode.method = Chaid
set :chaidnode.model_output_type = InteractiveBuilder
set :chaidnode.use_tree_directives = True
set :chaidnode.tree_directives = "Test"
set :chaidnode.mode = Expert
set :chaidnode.split_alpha = 0.03
set :chaidnode.merge_alpha = 0.04
set :chaidnode.chi_square = Pearson
set :chaidnode.use_percentage = True
set :chaidnode.min_parent_records_abs = 40
set :chaidnode.min_child_records_abs = 30
set :chaidnode.epsilon = 0.003
set :chaidnode.max_iterations = 75
set :chaidnode.split_merged_categories = true
```

```
set :chaidnode.bonferroni_adjustment = true
```

Propiedades de chaidnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos CHAID requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue_training_existing_model	<i>marca</i>	
objective	Standard Boosting Bagging psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>marca</i>	
tree_directives	<i>cadena</i>	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	<i>entero</i>	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
use_percentage	<i>marca</i>	
min_parent_records_pc	<i>número</i>	
min_child_records_pc	<i>número</i>	
min_parent_records_abs	<i>número</i>	
min_child_records_abs	<i>número</i>	
use_costs	<i>marca</i>	
costs	<i>estructurado</i>	Propiedad estructurada que utiliza la siguiente forma: <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , donde los argumentos entre llaves ({} ) son los costes pronosticados actuales.
trails	<i>número</i>	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Mean Median	Regla de combinación predeterminada para objetivos continuos.
large_boost	<i>marca</i>	Aplicar aumento a conjunto de datos muy grandes.
split_alpha	<i>número</i>	Nivel de significancia para división.
merge_alpha	<i>número</i>	Nivel de significancia para fusión.
bonferroni_adjustment	<i>marca</i>	Los valores de significancia de ajuste utilizando el método de Bonferroni.

Propiedades de chaidnode	Valores	Descripción de la propiedad
split_merged_categories	marca	Permitir segunda división de categorías fusionadas.
chi_square	Pearson LR	Método usado para calcular la estadística de chi cuadrado: Pearson o cociente de verosimilitudes (LR)
epsilon	número	Cambio mínimo en frecuencias de casillas esperadas.
max_iterations	número	Número máximo de iteraciones para la convergencia.
set_random_seed	entero	
seed	número	
calculate_variable_importance	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	entero	

## Propiedades de coxregnode



El nodo Regresión de Cox le permite crear un modelo de supervivencia para datos de tiempo hasta el evento en presencia de registros censurados. El modelo produce una función de supervivencia que pronostica la probabilidad de que el evento de interés se haya producido en el momento dado ( $t$ ) para valores determinados de las variables de entrada. [Si desea obtener más información, consulte el tema Nodo Cox en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create coxregnode
set :coxregnode.survival_time = tenure
set :coxregnode.method = BackwardsStepwise
# Pestaña Experto
set :coxregnode.mode = Expert
set :coxregnode.removal_criterion = Conditional
set :coxregnode.survival = True
```

Propiedades de coxregnode	Valores	Descripción de la propiedad
survival_time	campo	Los modelos de regresión de Cox requieren un único campo con los tiempos de supervivencia.
target	campo	Los modelos de regresión de Cox requieren un único campo objetivo y uno o más campos de entrada. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>

Propiedades de coxregnode	Valores	Descripción de la propiedad
method	Enter Stepwise BackwardsStepwise	
groups	<i>campo</i>	
model_type	MainEffects Custom	
custom_terms	[ "BP*Sexo" "BP*Edad" ]	Ejemplo: set :coxregnode. custom_terms = ["BP*Sex" "BP" "Age"]
mode	Expert Simple	
max_iterations	<i>número</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	<i>número</i>	
probability_removal	<i>número</i>	
output_display	EachStep LastStep	
ci_enable	<i>marca</i>	
ci_value	90 95 99	
correlation	<i>marca</i>	
display_baseline	<i>marca</i>	
survival	<i>marca</i>	
hazard	<i>marca</i>	
log_minus_log	<i>marca</i>	
one_minus_survival	<i>marca</i>	

Propiedades de coxregnode	Valores	Descripción de la propiedad
separate_line	<i>campo</i>	
value	<i>número o cadena</i>	Si no se especifica ningún valor para un campo, se utilizará la opción por defecto "Mean" para dicho campo. Uso para un campo numérico: coxnode.value = [{"age" "35.8"}] Uso para un campo categórico: coxnode.value = [{"color" "pink"}]

## Propiedades de decisionlistnode



El nodo Lista de decisiones identifica subgrupos, o segmentos, que muestran una mayor o menor posibilidad de proporcionar un resultado binario relacionado con la población global. Por ejemplo, puede buscar clientes que tengan menos posibilidades de perder clientes o más posibilidades de responder favorablemente a una campaña. Puede incorporar su conocimiento empresarial al modelo añadiendo sus propios segmentos personalizados y previsualizando modelos alternativos uno junto a otro para comparar los resultados. Los modelos de listas de decisiones constan de una lista de reglas en las que cada regla tiene una condición y un resultado. Las reglas se aplican en orden, y la primera regla que coincide determina el resultado. [Si desea obtener más información, consulte el tema Lista de decisiones en el capítulo 9 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create decisionlistnode
set :decisionlistnode.search_direction=Down
set :decisionlistnode.target_value=1
set :decisionlistnode.max_rules=4
set :decisionlistnode.min_group_size_pct = 15
```

Propiedades de decisionlistnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos de listas de decisiones utilizan un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	Hace referencia a la localización de segmentos, donde Up es el equivalente a Alta probabilidad y Down es el equivalente a Baja probabilidad.
target_value	<i>cadena</i>	Si no se especifica, se supondrá el valor true para las marcas.
max_rules	<i>entero</i>	Número máximo de segmentos sin incluir el resto.
min_group_size	<i>entero</i>	Tamaño mínimo del segmento.

Propiedades de decisionlistnode	Valores	Descripción de la propiedad
min_group_size_pct	número	Tamaño mínimo del segmento como porcentaje.
confidence_level	número	Umbral mínimo que un campo de entrada tiene que mejorar la probabilidad de la respuesta (aumentar la elevación) para que merezca la pena añadirlo a la definición de un segmento.
max_segments_per_rule	entero	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	número	
max_models_per_cycle	entero	Amplitud de búsqueda de las listas.
max_rules_per_cycle	entero	Amplitud de búsqueda de las reglas de segmentación.
segment_growth	número	
include_missing	marca	
final_results_only	marca	
reuse_fields	marca	Permite la reutilización de los atributos (los campos de entrada que aparecen en las reglas).
max_alternatives	entero	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de discriminantnode



El análisis discriminante realiza más supuestos rigurosos que regresiones logísticas, pero puede ser una alternativa o un suplemento valioso al análisis de regresión logística si se cumplen dichos supuestos. [Si desea obtener más información, consulte el tema Nodo Discriminante en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create discriminantnode
set :discriminantnode.target = custcat
set :discriminantnode.use_partitioned_data = False
```

set :discriminantnode.method = Stepwise

Propiedades de discriminantnode	Valores	Descripción de la propiedad
target	campo	Los modelos discriminantes requieren un único campo objetivo y uno o más campos de entrada. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	marca	Opciones de estadísticos del cuadro de diálogo Salida avanzada.
univariate_anovas	marca	
box_m	marca	
within_group_covariance	marca	
within_groups_correlation	marca	
separate_groups_covariance	marca	
total_covariance	marca	
fishers	marca	
unstandardized	marca	
casewise_results	marca	Opciones de clasificación del cuadro de diálogo Salida avanzada.
limit_to_first	número	El valor por defecto es 10.
summary_table	marca	
leave_one_classification	marca	
combined_groups	marca	
separate_groups_covariance	marca	Opción de matrices Covarianza de grupos separados.
territorial_map	marca	
combined_groups	marca	Opción de gráfico Grupos combinados.
separate_groups	marca	Opción de gráfico Grupos separados.
summary_of_steps	marca	
F_pairwise	marca	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	número	
criteria	UseValue UseProbability	

Propiedades de discriminantnode	Valores	Descripción de la propiedad
F_value_entry	número	El valor por defecto es 3,84.
F_value_removal	número	El valor por defecto es 2,71.
probability_entry	número	El valor por defecto es 0,05.
probability_removal	número	El valor por defecto es 0,10.
calculate_variable_importance	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de factornode



El nodo PCA/Factorial proporciona técnicas eficaces de reducción de datos para reducir la complejidad de los datos. Análisis de componentes principales (PCA) busca combinaciones lineales de los campos de entrada que realizan el mejor trabajo a la hora de capturar la varianza en todo el conjunto de campos, en el que los componentes son ortogonales (perpendiculares) entre ellos. Análisis factorial intenta identificar factores subyacentes que expliquen el patrón de correlaciones dentro de un conjunto de campos observados. Para los dos métodos, el objetivo es encontrar un número pequeño de campos derivados que resuma de forma eficaz la información del conjunto original de campos. [Si desea obtener más información, consulte el tema Nodo PCA/Factorial en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create factornode
# Pestaña "Campos"
set :factornode.custom_fields = True
set :factornode.inputs = ['BP' 'Na' 'K']
set :factornode.partition = Test
# Pestaña "Modelo"
set :factornode.use_model_name = True
set :factornode.model_name = "Factor_Age"
set :factornode.use_partitioned_data = False
set :factornode.method = GLS
# Opciones de Expert
set :factornode.mode = Expert
set :factornode.complete_records = true
set :factornode.matrix = Covariance
set :factornode.max_iterations = 30
set :factornode.extract_factors = ByFactors
set :factornode.min_eigenvalue = 3.0
set :factornode.max_factor = 7
set :factornode.sort_values = True
set :factornode.hide_values = True
set :factornode.hide_below = 0.7
# Sección "Rotación"
set :factornode.rotation = DirectOblimin
set :factornode.delta = 0.3
```



set :factornode.kappa = 7.0

Propiedades de factornode	Valores	Descripción de la propiedad
inputs	[ <i>campo1 ... campoN</i> ]	Los modelos PCA/Factorial utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	<i>número</i>	
complete_records	<i>marca</i>	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	<i>número</i>	
max_factor	<i>número</i>	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	<i>número</i>	Si selecciona DirectOblimin como el tipo de datos de rotación, podrá especificar un valor para delta. Si no especifica ningún valor, se utilizará el valor por defecto para delta.
kappa	<i>número</i>	Si selecciona Promax como el tipo de datos de rotación, podrá especificar un valor para kappa. Si no especifica ningún valor, se utilizará el valor por defecto para kappa.
sort_values	<i>marca</i>	
hide_values	<i>marca</i>	
hide_below	<i>número</i>	

## Propiedades de featureselectionnode



El nodo Selección de características filtra los campos de entrada para su eliminación en función de un conjunto de criterios (como el porcentaje de valores perdidos); a continuación, clasifica el grado de importancia del resto de entradas de acuerdo con un objetivo específico. Por ejemplo, a partir de un conjunto de datos dado con cientos de entradas potenciales, ¿cuáles tienen mayor probabilidad de ser útiles para el modelado de resultados de pacientes? [Si desea obtener más información, consulte el tema Nodo Selección de características en el capítulo 4 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create featureselectionnode
set :featureselectionnode.screen_single_category=true
set :featureselectionnode.max_single_category=95
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.unimportant_below = 0.8
set :featureselectionnode.important_above = 0.9
set :featureselectionnode.important_label = "Check Me Out!"
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
```

Para obtener un ejemplo más detallado en el que se cree y aplique un modelo de selección de características, consulte [Ejemplo de proceso independiente: Generar un modelo de selección de características en el capítulo 2 el p. 14](#).

Propiedades de featureselectionnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos de selección de características ordenan predictores por rangos con respecto a su objetivo específico. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
screen_single_category	<i>marca</i>	En caso de establecer True, filtra campos que tienen demasiados registros dentro de la misma categoría respecto al número total de registros.
max_single_category	<i>número</i>	Especifica el umbral utilizado si screen_single_category es True.
screen_missing_values	<i>marca</i>	Si está definida como True, filtra campos con demasiados valores perdidos, expresados como un porcentaje del número total de registros.
max_missing_values	<i>número</i>	

Propiedades de featureselectionn-ode	Valores	Descripción de la propiedad
screen_num_categories	marca	Si está definida como True, filtra campos con demasiadas categorías respecto al número total de registros.
max_num_categories	número	
screen_std_dev	marca	Si está definida como True, filtra campos con una desviación típica menor o igual que el mínimo especificado.
min_std_dev	número	
screen_coeff_of_var	marca	Si está definida como True, filtra campos con un coeficiente de varianza menor o igual que el mínimo especificado.
min_coeff_of_var	número	
criteria	Pearson Likelihood CramersV Lambda	Al ordenar por rango los predictores categóricos en función de un objetivo categórico, especifica la medida en la que se basa el valor de importancia.
unimportant_below	número	Especifica los valores $p$ de umbral utilizados para ordenar por rango las variables como importantes, marginales o sin importancia. Acepta valores de 0,0 a 1,0.
important_above	número	Acepta valores de 0,0 a 1,0.
unimportant_label	cadena	Especifica la etiqueta para el rango sin importancia.
marginal_label	cadena	
important_label	cadena	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	marca	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos importantes.
select_marginal	marca	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos marginales.
select_unimportant	marca	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos no importantes.
importance_value	número	Si selection_mode está definido como ImportanceValue, determina el valor de corte que se va a usar. Acepta valores de 0 a 100.
top_n	entero	Si selection_mode está definido como TopN, determina el valor de corte que se va a usar. Acepta valores de 0 a 1000.

## Propiedades de genlinnode



El modelo lineal generalizado amplía el modelo lineal general, de manera que la variable dependiente está relacionada linealmente con los factores y las covariables mediante una determinada función de enlace. Además, el modelo permite que la variable dependiente tenga una distribución que no sea normal. Cubre la funcionalidad de un amplio número de modelos estadísticos, incluyendo regresión lineal, regresión logística, modelos log lineales para recuento de datos y modelos de supervivencia censurados por intervalos. [Si desea obtener más información, consulte el tema Nodo GenLin en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create genlinnode
set :genlinnode.model_type = MainAndAllTwoWayEffects
set :genlinnode.offset_type = Variable
set :genlinnode.offset_field = Claimant
```

Propiedades de genlinnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos lineales generalizados requieren un único campo objetivo, que debe ser un campo nominal o marca, y uno o más campos de entrada. También se puede especificar un campo de ponderación. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
use_weight	<i>marca</i>	
weight_field	<i>campo</i>	El tipo de campo es únicamente continuo.
target_represents_trials	<i>marca</i>	
trials_type	Variable FixedValue	
trials_field	<i>campo</i>	El tipo de campo es continuo, marca u ordinal.
trials_number	<i>número</i>	El valor por defecto es 10.
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	<i>campo</i>	El tipo de campo es únicamente continuo.
offset_value	<i>número</i>	Debe ser un número real.
base_category	Last First	
include_intercept	<i>marca</i>	
mode	Simple Expert	

Propiedades de <b>genlinnode</b>	Valores	Descripción de la propiedad
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: De Gauss inversa. NEGBIN: Binomial negativa.
negbin_para_type	Specify Estimate	
negbin_parameter	<i>número</i>	El valor por defecto es 1. Debe contener un número real no negativo.
tweedie_parameter	<i>número</i>	
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: log-log complementario. LOGC: complemento log. NEGBIN: Binomial negativa. NLOGLOG: log-log negativa. CUMCAUCHIT: Cauchit acumulada. CUMCLOGLOG: Log-log complementario acumulado. CUMLOGIT: Logit acumulado. CUMNLOGLOG: Log-log negativo acumulado. CUMPROBIT: Probit acumulado.
power	<i>número</i>	El valor debe ser real y distinto de cero.
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	<i>número</i>	El valor por defecto es 1; sólo se admiten enteros positivos.
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	<i>número</i>	El valor por defecto es 1; debe ser mayor que 0.
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	<i>número</i>	El valor por defecto es 100; sólo enteros no negativos.
max_step_halving	<i>número</i>	El valor por defecto es 5; sólo enteros positivos.
check_separation	<i>marca</i>	
start_iteration	<i>número</i>	El valor por defecto es 20; sólo se admiten enteros positivos.
estimates_change	<i>marca</i>	
estimates_change_min	<i>número</i>	El valor por defecto es 1E-006; sólo se admiten números positivos.

Propiedades de <code>genlinnode</code>	Valores	Descripción de la propiedad
<code>estimates_change_type</code>	Absolute Relative	
<code>loglikelihood_change</code>	<i>marca</i>	
<code>loglikelihood_change_min</code>	<i>número</i>	Sólo se admiten números positivos.
<code>loglikelihood_change_type</code>	Absolute Relative	
<code>hessian_convergence</code>	<i>marca</i>	
<code>hessian_convergence_min</code>	<i>número</i>	Sólo se admiten números positivos.
<code>hessian_convergence_type</code>	Absolute Relative	
<code>case_summary</code>	<i>marca</i>	
<code>contrast_matrices</code>	<i>marca</i>	
<code>descriptive_statistics</code>	<i>marca</i>	
<code>estimable_functions</code>	<i>marca</i>	
<code>model_info</code>	<i>marca</i>	
<code>iteration_history</code>	<i>marca</i>	
<code>goodness_of_fit</code>	<i>marca</i>	
<code>print_interval</code>	<i>número</i>	El valor por defecto es 1; debe ser un entero positivo.
<code>model_summary</code>	<i>marca</i>	
<code>lagrange_multiplier</code>	<i>marca</i>	
<code>parameter_estimates</code>	<i>marca</i>	
<code>include_exponential</code>	<i>marca</i>	
<code>covariance_estimates</code>	<i>marca</i>	
<code>correlation_estimates</code>	<i>marca</i>	
<code>analysis_type</code>	Typel Typelll TypeIAndTypeIII	
<code>statistics</code>	Wald LR	
<code>citype</code>	Wald Profile	
<code>tolerancelevel</code>	<i>número</i>	El valor por defecto es 0.0001.
<code>confidence_interval</code>	<i>número</i>	El valor por defecto es 95.
<code>loglikelihood_function</code>	Full Kernel	
<code>singularity_tolerance</code>	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
<code>value_order</code>	Ascending Descending DataOrder	
<code>calculate_variable_importance</code>	<i>marca</i>	
<code>calculate_raw_propensities</code>	<i>marca</i>	

Propiedades de <b>genl</b> node	Valores	Descripción de la propiedad
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de **glmm**node



Un modelo lineal mixto generalizado (GLMM) amplía el modelo lineal de modo que el objetivo pueda tener una distribución no normal, esté linealmente relacionado con los factores y covariables mediante una función de enlace especificada y las observaciones se puedan correlacionar. Los modelos lineales mixtos generalizados cubren una amplia variedad de modelos, desde modelos de regresión lineal simple hasta modelos multinivel complejos para datos longitudinales no normales. [Si desea obtener más información, consulte el tema Nodo GLMM en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

Propiedades de <b>glmm</b> node	Valores	Descripción de la propiedad
residual_subject_spec	estructurado	La combinación de valores de los campos categóricos especificados que definen de manera única los sujetos del conjunto de datos.
repeated_measures	estructurado	Campos utilizados para identificar observaciones repetidas.
residual_group_spec	[campo1 ... campoN]	Campos que definen conjuntos independientes de parámetros de covarianza de efectos repetidos.
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Especifica la estructura de covarianza de residuos.
custom_target	marca	Indica si se puede utilizar un objetivo definido en el nodo anterior (false) o un objetivo personalizado especificado por target_field (true).
target_field	campo	Campo que hay que utilizar como objetivo si custom_target es true.
use_trials	marca	Indica si hay que utilizar el campo adicional o el valor que especifica el número de ensayos cuando la respuesta objetivo es un número de eventos que tienen lugar en un conjunto de ensayos. El valor por defecto es false.
use_field_or_value	Field Value	Indica si se utiliza el campo (predeterminado) o valor para especificar el número de ensayos.
trials_field	campo	Campo a utilizar para especificar el número de ensayos.
trials_value	entero	Valor a utilizar para especificar el número de ensayos. Si se especifica, el valor mínimo es 1.

Propiedades de glmmnode	Valores	Descripción de la propiedad
use_custom_target_reference	<i>marca</i>	Indica si hay que utilizar la categoría de referencia personalizada para un objetivo categórico. El valor por defecto es false.
target_reference_value	<i>cadena</i>	Categoría de referencia que hay que utilizar si use_custom_target_reference es true.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	Modelos comunes para la distribución de valores de objetivo. Elija Custom para especificar una distribución de la lista proporcionada por target_distribution.
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	Distribución de valores de objetivo cuando dist_link_combination es Custom.
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	Función de enlace para relacionar valores de objetivo a los predictores.
link_function_param	<i>número</i>	Valor del parámetro de función de enlace que hay que utilizar. Aplicable únicamente si link_function es POWER.
use_predefined_inputs	<i>marca</i>	Indica si los campos de efectos fijos deben ser aquellos definidos anteriormente como campos de entrada (true) o los campos de fixed_effects_list (false). El valor por defecto es false.
fixed_effects_list	<i>estructurado</i>	Si use_predefined_inputs es false, especifica los campos de entrada que hay que utilizar como campos de efectos fijos.
use_intercept	<i>marca</i>	Si es true (predeterminado), incluye la intercepción en el modelo.
random_effects_list	<i>estructurado</i>	Lista de campos para especificar como efectos aleatorios.
regression_weight_field	<i>campo</i>	Campo a utilizar como campo de ponderación de análisis.
use_offset	None offset_value offset_field	Indica cómo se especifica la compensación. El valor None significa que no se ha utilizado compensación.



Propiedades de glmmnode	Valores	Descripción de la propiedad
offset_value	número	Valor a utilizar para la compensación si se ha definido use_offset a offset_value.
offset_field	campo	Campo a utilizar para el valor de compensación si se ha definido use_offset a offset_field.
target_category_order	Ascending Descending Data	Orden de clasificación para objetivos categóricos. El valor Data especifica que se utiliza el orden de clasificación de los datos. El valor por defecto es Ascending.
inputs_category_order	Ascending Descending Data	Orden de clasificación para predictores categóricos. El valor Data especifica que se utiliza el orden de clasificación de los datos. El valor por defecto es Ascending.
max_iterations	entero	Número máximo de iteraciones que ejecutará el algoritmo. Un número entero no negativo; el valor por defecto es 100.
confidence_level	entero	Nivel de confianza utilizado para calcular estimaciones de intervalo de los coeficientes del modelo. Un número entero no negativo; el valor máximo es 100, el valor por defecto es 95.
degrees_of_freedom_method	Fixed Varied	Especifica cómo se calculan los grados de libertad para la prueba de significación.
test_fixed_effects_coefficients	Model Robust	Método para calcular la matriz de covarianza de las estimaciones de los parámetros.
use_model_name	marca	Indica si hay que especificar un nombre personalizado para el modelo (true), o bien utilizar el nombre generado por el sistema (false). El valor por defecto es false.
model_name	cadena	Si use_model_name es true, especifica el nombre de modelo que se va a utilizar.
confidence	highest difference	Base para calcular el valor de confianza de la puntuación: probabilidad más alta pronosticada, o la diferencia entre la probabilidad más alta pronosticada y la segunda probabilidad más alta.
score_category_probabilities	marca	Si es true, genera las probabilidades predichas para objetivos categóricos. El valor por defecto es false.
max_categories	entero	Si score_category_probabilities es true, especifica el número máximo de categorías que hay que guardar.
score_propensity	marca	Si es true, produce puntuaciones de propensión para campos de objetivo de marca que indican la probabilidad del resultado "true" para el campo.

Propiedades de glmmnode	Valores	Descripción de la propiedad
emeans	<i>structure</i>	Para cada campo categórico de la lista de efectos fijos, especifica si hay que producir medias marginales estimadas.
covariance_list	<i>structure</i>	Para cada campo continuo de la lista de efectos fijos, especifica si hay que usar la media o un valor personalizado al calcular medias marginales estimadas.
mean_scale	Original Transformed	Especifica si las medias marginales estimadas se calculan basándose en la escala original del objetivo (por defecto) o en la transformación de la función de enlace.
comparison_adjustment_method	LSD SEQBONFERRONI SESSIDAK	Método de ajuste que hay que utilizar al realizar pruebas de hipótesis con varios contrastes.

## Propiedades de kmeansnode



El nodo K-medias agrupa conjuntos de datos en grupos distintos (o conglomerados). El método define un número fijo de conglomerados, de forma iterativa asigna registros a los conglomerados y ajusta los centros de los conglomerados hasta que no se pueda mejorar el modelo. En lugar de intentar pronosticar un resultado, los modelos de  $k$ -medias utilizan un proceso conocido como aprendizaje no supervisado para revelar los patrones del conjunto de campos de entrada. [Si desea obtener más información, consulte el tema Nodo K-medias en el capítulo 11 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create kmeansnode
# "Fields" tab
set :kmeansnode.custom_fields = True
set :kmeansnode.inputs = ['Colesterol' 'BP' 'Medicamento' 'Na' 'K' 'Edad']
# pestaña "Modelo"
set :kmeansnode.use_model_name = False
set :kmeansnode.model_name = "Kmeans_allinputs"
set :kmeansnode.num_clusters = 9
set :kmeansnode.gen_distance = True
set :kmeansnode.cluster_label = "Number"
set :kmeansnode.label_prefix = "Kmeans_"
set :kmeansnode.optimize = Speed
# pestaña "Experto"
set :kmeansnode.mode = Expert
set :kmeansnode.stop_on = Custom
set :kmeansnode.max_iterations = 10
set :kmeansnode.tolerance = 3.0
```

```
set :kmeansnode.encoding_value = 0.3
```

Propiedades de kmeansnode	Valores	Descripción de la propiedad
inputs	[ <i>campo1 ... campoN</i> ]	Los modelos de K-Medias realizan un análisis de conglomerados en un conjunto de campos de entrada pero no utilizan ningún campo objetivo. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
num_clusters	<i>número</i>	
gen_distance	<i>marca</i>	
cluster_label	String Number	
label_prefix	<i>cadena</i>	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	<i>número</i>	
tolerance	<i>número</i>	
encoding_value	<i>número</i>	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.

## Propiedades de knnnode



El nodo  $k$  de modelado de vecino (KNN) asocia el nuevo caso con la categoría o valor de los objetos  $k$  junto a él en el espacio de predictores, donde  $k$  es un entero. Los casos parecidos están próximos y los que no lo son están alejados entre sí. [Si desea obtener más información, consulte el tema Nodo KNN en el capítulo 16 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create knnnode
# Pestaña Objetivos
set: knnnode.objective = Custom
# Pestaña Configuración, panel Vecinos
set: knnnode.automatic_k_selection = false
set: knnnode.fixed_k = 2
set: knnnode.weight_by_importance = True
# Pestaña Configuración, panel Analizar
```

set: knnnode.save\_distances = True

Propiedades de knnnode	Valores	Descripción de la propiedad
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	marca	
use_case_labels	marca	Seleccione esta casilla de verificación para activar la siguiente opción.
case_labels_field	campo	
identify_focal_cases	marca	Seleccione esta casilla de verificación para activar la siguiente opción.
focal_cases_field	campo	
automatic_k_selection	marca	
fixed_k	entero	Activado sólo si automatic_k_selectio es False.
minimum_k	entero	Activado sólo si automatic_k_selectio es True.
maximum_k	entero	
distance_computation	Euclidean CityBlock	
weight_by_importance	marca	
range_predictions	Mean Median	
perform_feature_selection	marca	
forced_entry_inputs	[campo1 ... campoN]	
stop_on_error_ratio	marca	
number_to_select	entero	
minimum_change	número	
validation_fold_assign_by_field	marca	
number_of_folds	entero	Activado sólo si validation_fold_as-sign_by_field es False.
set_random_seed	marca	
random_seed	número	
folds_field	campo	Activado sólo si validation_fold_as-sign_by_field es True.
all_probabilities	marca	
save_distances	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de kohonennode



El nodo Kohonen genera un tipo de red neuronal que se puede usar para conglomerar un conjunto de datos en grupos distintos. Cuando la red se termina de entrenar, los registros que son similares se deberían cerrar juntos en el mapa de resultados, mientras que los registros que son diferentes aparecerían aparte. Puede observar el número de observaciones capturadas por cada unidad en el nugget de modelo para identificar unidades fuertes. Esto le proporcionará una idea del número apropiado de conglomerados. [Si desea obtener más información, consulte el tema Nodo Kohonen en el capítulo 11 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create kohonennode
# Pestaña "Modelo"
set :kohonennode.use_model_name = False
set :kohonennode.model_name = "Symbolic Cluster"
set :kohonennode.stop_on = Time
set :kohonennode.time = 1
set :kohonennode.set_random_seed = True
set :kohonennode.random_seed = 12345
set :kohonennode.optimize = Speed
# Pestaña "Experto"
set :kohonennode.mode = Expert
set :kohonennode.width = 3
set :kohonennode.length = 3
set :kohonennode.decay_style = Exponential
set :kohonennode.phase1_neighborhood = 3
set :kohonennode.phase1_eta = 0.5
set :kohonennode.phase1_cycles = 10
set :kohonennode.phase2_neighborhood = 1
set :kohonennode.phase2_eta = 0.2
set :kohonennode.phase2_cycles = 75
```

Propiedades de kohonennode	Valores	Descripción de la propiedad
inputs	[ <i>campo1</i> ... <i>campoN</i> ]	Los modelos Kohonen utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de frecuencia y ponderación no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue	<i>marca</i>	
show_feedback	<i>marca</i>	
stop_on	Default Time	
time	<i>número</i>	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.
cluster_label	<i>marca</i>	

Propiedades de kohonennode	Valores	Descripción de la propiedad
mode	Simple Expert	
width	número	
length	número	
decay_style	Linear Exponential	
phase1_neighborhood	número	
phase1_eta	número	
phase1_cycles	número	
phase2_neighborhood	número	
phase2_eta	número	
phase2_cycles	número	

## Propiedades de linearnode



Los modelos de regresión lineal predicen un destino continuo tomando como base las relaciones lineales entre el destino y uno o más predictores. [Si desea obtener más información, consulte el tema Modelos lineales en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create linearnode
# Pestaña Opciones de generación - panel Objetivos
set: linearnode.objective = Estándar
# pestaña "Opciones de generación - panel Selección de modelo
set: linearnode.model_selection = BestSubsets
set: linearnode.criteria_best_subsets = ASE
# pestaña Opciones de generación - panel Conjuntos
set: linearnode.combining_rule_categorical = HighestMeanProbability
```

Propiedades de linearnode	Valores	Descripción de la propiedad
target	campo	Especifica un campo de objetivo único.
inputs	[campo1 ... campoN]	Campos de predictor utilizados por el modelo.
continue_training_existing_model	marca	
objective	Standard Bagging Boosting psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al servidor.
use_auto_data_preparation	marca	
confidence_level	número	
model_selection	ForwardStepwise BestSubsets None	

Propiedades de linearnode	Valores	Descripción de la propiedad
criteria_forward_stepwise	AICC Estadísticas F AdjustedRSquare ASE	
probability_entry	número	
probability_removal	número	
use_max_effects	marca	
max_effects	número	
use_max_steps	marca	
max_steps	número	
criteria_best_subsets	AICC R cuadrado corregido ASE	
combining_rule_continuous	Mean Median	
component_models_n	número	
use_random_seed	marca	
random_seed	number	
use_custom_model_name	marca	
custom_model_name	cadena	
use_custom_name	marca	
custom_name	cadena	
tooltip	cadena	
keywords	cadena	
annotation	cadena	

## Propiedades de logregnode



La regresión logística es una técnica de estadístico para clasificar los registros en función los valores de los campos de entrada. Es análoga a la regresión lineal pero toma un campo objetivo categórico en lugar de uno numérico. [Si desea obtener más información, consulte el tema Nodo Logística en el capítulo 10 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo multinomial

```
create logregnode
# Pestaña "Campos"
set :logregnode.custom_fields = True
set :logregnode.target = 'Drug'
set :logregnode.inputs = ['BP' 'Colesterol' 'Edad']
set :logregnode.partition = Test
#Pestaña "Modelo"
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Drug"
set :logregnode.use_partitioned_data = True
set :logregnode.method = Stepwise
set :logregnode.logistic_procedure = Multinomial
```

```

set :logregnode.multinomial_base_category = BP
set :logregnode.model_type = FullFactorial
set :logregnode.custom_terms = {{BP Sexo}{Edad}{Na K}}
set :logregnode.include_constant = False
# Pestaña "Experto"
set :logregnode.mode = Expert
set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# Sección "Convergencia..."
set :logregnode.max_iterations = 50
set :logregnode.max_steps = 3
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
set :logregnode.delta = 0.03
# Sección "Resultado..."
set :logregnode.summary = True
set :logregnode.likelihood_ratio = True
set :logregnode.asymptotic_correlation = True
set :logregnode.goodness_fit = True
set :logregnode.iteration_history = True
set :logregnode.history_steps = 3
set :logregnode.parameters = True
set :logregnode.confidence_interval = 90
set :logregnode.asymptotic_covariance = True
set :logregnode.classification_table = True
# Opciones "Método por pasos"
set :logregnode.min_terms = 7
set :logregnode.use_max_terms = true
set :logregnode.max_terms = 10
set :logregnode.probability_entry = 3
set :logregnode.probability_removal = 5
set :logregnode.requirements = Containment

```

### ***Ejemplo binomial***

```

create logregnode
# Pestaña "Campos"
set :logregnode.custom_fields = True
set :logregnode.target = 'Cholesterol'
set :logregnode.inputs = ['BP' 'Drug' 'Age']
set :logregnode.partition = Test
# Pestaña "Modelo"
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Cholesterol"
set :logregnode.multinomial_base_category = BP
set :logregnode.use_partitioned_data = True
set :logregnode.binomial_method = Forwards
set :logregnode.logistic_procedure = Binomial
set :logregnode.binomial_categorical_input = Sex
set :logregnode.binomial_input_contrast.Sex = Simple

```



```

set :logregnode.binomial_input_category.Sex = Last
set :logregnode.include_constant = False
# Pestaña "Experto"
set :logregnode.mode = Expert
set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# Sección "Convergencia..."
set :logregnode.max_iterations = 50
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
# Sección "Resultado..."
set :logregnode.binomial_output_display = at_each_step
set :logregnode.binomial_goodness_fit = True
set :logregnode.binomial_iteration_history = True
set :logregnode.binomial_parameters = True
set :logregnode.binomial_ci_enable = True
set :logregnode.binomial_ci = 85
# Opciones de "Método por pasos"
set :logregnode.binomial_removal_criterion = LR
set :logregnode.binomial_probability_removal = 0.2

```

Propiedades de logregnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos de regresión logística requieren un único campo objetivo y uno o más campos de entrada. Los campos de frecuencia y ponderación no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
logistic_procedure	Binomial Multinomial	
include_constant	<i>marca</i>	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	

Propiedades de logregnode	Valores	Descripción de la propiedad
model_type	MainEffects FullFactorial Custom	Si FullFactorial se especifica como el tipo de modelo, no se ejecutarán los métodos por pasos, aunque así se indique. En su lugar, el método utilizado será Enter. Si el tipo de modelo está definido como Custom pero no se ha especificado ningún campo personalizado, se generará un modelo de efectos principales.
custom_terms	[{BP Sexo}{BP}{Edad}]	Ejemplo: set :logregnode. custom_terms = [{Na} {K} {Na K}]
multinomial_base_category	cadena	Especifica cómo se determina la categoría de referencia.
binomial_categorical_input	cadena	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	Propiedad adaptada para la entrada categórica que especifica cómo se determina el contraste. Formato de uso: NODE.binomial_input_contrast.FIELD-NAME
binomial_input_category	First Last	Propiedad adaptada para la entrada categórica que especifica cómo se determina la categoría de referencia. Formato de uso: NODE.binomial_input_category.FIELD-NAME
scale	None UserDefined Pearson Deviance	
scale_value	número	
all_probabilities	marca	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	número	
use_max_terms	marca	
max_terms	número	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	número	
probability_removal	número	
binomial_probability_entry	número	
binomial_probability_removal	número	

Propiedades de logregnode	Valores	Descripción de la propiedad
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	número	
max_steps	número	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	número	
iteration_history	marca	
history_steps	número	
summary	marca	
likelihood_ratio	marca	
asymptotic_correlation	marca	
goodness_fit	marca	
parameters	marca	
confidence_interval	número	
asymptotic_covariance	marca	
classification_table	marca	
stepwise_summary	marca	
info_criteria	marca	
monotonicity_measures	marca	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	marca	
binomial_parameters	marca	
binomial_iteration_history	marca	
binomial_classification_plots	marca	
binomial_ci_enable	marca	
binomial_ci	número	
binomial_residual	outliers all	
binomial_residual_enable	marca	
binomial_outlier_threshold	número	
binomial_classification_cutoff	número	
binomial_removal_criterion	LR Wald Conditional	

Propiedades de logregnode	Valores	Descripción de la propiedad
calculate_variable_importance	marca	
calculate_raw_propensities	marca	

## Propiedades de neuralnetnode

**Precaución:** Una versión más reciente del nodo de modelado Red neural, con características mejoradas, está disponible en esta versión y se describe en la sección siguiente (*red neural*). Aunque aún puede generar y puntuar un modelo con la versión anterior, recomendamos que actualice sus procesos para que se use la nueva versión. Los detalles de la versión anterior se conservan aquí como referencia.

### Ejemplo

```
create neuralnetnode
# Pestaña "Campos"
set :neuralnetnode.custom_fields = True
set :neuralnetnode.targets = ['Drug']
set :neuralnetnode.inputs = ['Age' 'Na' 'K' 'Cholesterol' 'BP']
# Pestaña "Modelo"
set :neuralnetnode.use_partitioned_data = True
set :neuralnetnode.method = Dynamic
set :neuralnetnode.train_pct = 30
set :neuralnetnode.set_random_seed = True
set :neuralnetnode.random_seed = 12345
set :neuralnetnode.stop_on = Time
set :neuralnetnode.accuracy = 95
set :neuralnetnode.cycles = 200
set :neuralnetnode.time = 3
set :neuralnetnode.optimize = Speed
# Sección "Opciones de experto para método múltiple"
set :neuralnetnode.m_topologies = "5 30 5; 2 20 3, 1 10 1"
set :neuralnetnode.m_non_pyramids = False
set :neuralnetnode.m_persistence = 100
```

Propiedades de neuralnetnode	Valores	Descripción de la propiedad
targets	[ <i>campo1 ... campoN</i> ]	El nodo Red neuronal espera uno o varios campos objetivo y uno o más campos de entrada. Los campos de frecuencia y ponderación se omiten. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	marca	

Propiedades de neuralnetnode	Valores	Descripción de la propiedad
train_pct	número	
set_random_seed	marca	
random_seed	número	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	Modo de parada.
accuracy	número	Precisión de parada.
cycles	número	Ciclos para entrenar.
time	número	Tiempo para entrenar (minutos).
continue	marca	
show_feedback	marca	
binary_encode	marca	
use_last_model	marca	
gen_logfile	marca	
logfile_name	cadena	
alpha	número	
initial_eta	número	
high_eta	número	
low_eta	número	
eta_decay_cycles	número	
hid_layers	One Two Three	
hl_units_one	número	
hl_units_two	número	
hl_units_three	número	
persistence	número	
m_topologies	cadena	
m_non_pyramids	marca	
m_persistence	número	
p_hid_layers	One Two Three	
p_hl_units_one	número	
p_hl_units_two	número	
p_hl_units_three	número	
p_persistence	número	
p_hid_rate	número	
p_hid_pers	número	
p_inp_rate	número	
p_inp_pers	número	
p_overall_pers	número	

Propiedades de neuralnetnode	Valores	Descripción de la propiedad
r_persistence	número	
r_num_clusters	número	
r_eta_auto	marca	
r_alpha	número	
r_eta	número	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.
calculate_variable_importance	marca	Nota: la propiedad sensitivity_analysis utilizada en versiones anteriores se ha desaprobado en favor de esta propiedad. La propiedad anterior se sigue admitiendo, pero se recomienda calculate_variable_importance.
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de neuralnetworknode



El nodo Red neuronal utiliza un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades simples de procesamiento interconectadas que parecen versiones abstractas de neuronas. Las redes neuronales son estimadores potentes de funciones generales y requieren un conocimiento matemático o estadístico mínimo para entrenarlas o aplicarlas.

### Ejemplo

```
create neuralnetworknode
# Pestaña Opciones de generación - panel Objetivos
set: neuralnetworknode.objective = Estándar
# Pestaña Opciones de generación - panel Reglas de parada
set: neuralnetworknode.model_selection = BestSubsets
set: neuralnetworknode.criteria_best_subsets = ASE
# Pestaña Opciones de generación - panel Conjuntos
set: neuralnetworknode.combining_rule_categorical = HighestMeanProbability
```

Propiedades de neuralnetworknode	Valores	Descripción de la propiedad
targets	[campo1 ... campoN]	Especifica campos objetivo.
inputs	[campo1 ... campoN]	Campos de predictor utilizados por el modelo.
splits	[campo1 ... campoN]	Especifica el campo o campos para utilizar en el modelado de divisiones.

Propiedades de neuralnetworknode	Valores	Descripción de la propiedad
use_partition	marca	Si se ha definido un campo de partición, esta opción garantiza que sólo se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
continue	marca	Continuar entrenando modelo existente.
objective	Standard Bagging Boosting psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	marca	
first_layer_units	número	
second_layer_units	número	
use_max_time	marca	
max_time	number	
use_max_cycles	marca	
max_cycles	número	
use_min_accuracy	marca	
min_accuracy	número	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	número	
overfit_prevention_pct	número	
use_random_seed	marca	
random_seed	número	
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	marca	
custom_model_name	cadena	
confidence	onProbability onIncrease	
score_category_probabilities	marca	
max_categories	número	
score_propensity	marca	
use_custom_name	marca	
custom_name	cadena	
tooltip	cadena	
keywords	cadena	
annotation	cadena	

## Propiedades de questnode



El nodo QUEST proporciona un método de clasificación binario para generar árboles de decisión; está diseñado para reducir el tiempo de procesamiento necesario para realizar los análisis de C&RT y reducir la tendencia de los métodos de clasificación de árboles para favorecer a las entradas que permitan realizar más divisiones. Los campos de entrada pueden ser continuos (rango numérico), sin embargo el campo objetivo debe ser categórico. Todas las divisiones son binarias. [Si desea obtener más información, consulte el tema Nodo QUEST en el capítulo 6 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create questnode
set :questnode.custom_fields = True
set :questnode.target = Drug
set :questnode.inputs = [Edad Na K Colesterol BP]
set :questnode.model_output_type = InteractiveBuilder
set :questnode.use_tree_directives = True
set :questnode.mode = Expert
set :questnode.max_surrogates = 5
set :questnode.split_alpha = 0.03
set :questnode.use_percentage = False
set :questnode.min_parent_records_abs = 40
set :questnode.min_child_records_abs = 30
set :questnode.prune_tree = True
set :questnode.use_std_err = True
set :questnode.std_err_multiplier = 3
set :questnode.priors = Custom
set :questnode.custom_priors = [{drugA 0.3}{drugB 0.4}]
set :questnode.adjust_priors = true
```

Propiedades de questnode	Valores	Descripción de la propiedad
target	<i>campo</i>	Los modelos QUEST requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue_training_existing_model	<i>marca</i>	
objective	Standard Boosting Bagging psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>marca</i>	
tree_directives	<i>cadena</i>	
use_max_depth	Default Custom	



Propiedades de questnode	Valores	Descripción de la propiedad
max_depth	<i>entero</i>	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
prune_tree	<i>marca</i>	Poda del árbol para evitar sobreajustes.
use_std_err	<i>marca</i>	Use la diferencia máxima en riesgos (en errores típicos).
std_err_multiplier	<i>número</i>	Diferencia máxima.
max_surrogates	<i>número</i>	Número máximo de sustitutos.
use_percentage	<i>marca</i>	
min_parent_records_pc	<i>número</i>	
min_child_records_pc	<i>número</i>	
min_parent_records_abs	<i>número</i>	
min_child_records_abs	<i>número</i>	
use_costs	<i>marca</i>	
costs	<i>estructurado</i>	Propiedad estructurada que utiliza la siguiente forma: [[drugA drugB 1.5] {drugA drugC 2.1}], donde los argumentos entre llaves ({} ) son los costes pronosticados actuales.
priors	Data Equal Custom	
custom_priors	<i>estructurado</i>	Propiedad estructurada que utiliza la siguiente forma: set :cartnode. custom_priors = [ { drugA 0.3 } { drugB 0.6 } ]
adjust_priors	<i>marca</i>	
trails	<i>número</i>	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Mean Median	Regla de combinación predeterminada para objetivos continuos.
large_boost	<i>marca</i>	Aplicar aumento a conjunto de datos muy grandes.
split_alpha	<i>número</i>	Nivel de significancia para división.
train_pct	<i>número</i>	Conjunto de prevención sobreajustado.
set_random_seed	<i>marca</i>	Opción replicar resultados.
seed	<i>número</i>	
calculate_variable_importance	<i>marca</i>	
calculate_raw_propensities	<i>marca</i>	
calculate_adjusted_propensities	<i>marca</i>	
adjusted_propensity_partition	Test Validation	

## Propiedades de regressionnode



La regresión lineal es una técnica de estadístico común utilizada para resumir datos y realizar pronósticos ajustando una superficie o línea recta que minimice las discrepancias existentes entre los valores de salida reales y los pronosticados.

*Nota:* el nodo Lineal reemplazará al nodo Regresión en una versión futura. Recomendamos que a partir de ahora utilice modelos lineales para la regresión lineal.

### Ejemplo

```
create regressionnode
# Pestaña "Campos"
set :regressionnode.custom_fields = True
set :regressionnode.target = 'Age'
set :regressionnode.inputs = ['Na' 'K']
set :regressionnode.partition = Test
set :regressionnode.use_weight = True
set :regressionnode.weight_field = 'Drug'
# Pestaña "Modelo"
set :regressionnode.use_model_name = False
set :regressionnode.model_name = "Regression Age"
set :regressionnode.use_partitioned_data = True
set :regressionnode.method = Stepwise
set :regressionnode.include_constant = False
# Pestaña "Experto"
set :regressionnode.mode = Expert
set :regressionnode.complete_records = False
set :regressionnode.tolerance = "1.0E-3"
# Sección "Método por paso"
set :regressionnode.stepping_method = Probability
set :regressionnode.probability_entry = 0.77
set :regressionnode.probability_removal = 0.88
set :regressionnode.F_value_entry = 7.0
set :regressionnode.F_value_removal = 8.0
# Sección "Resultados"
set :regressionnode.model_fit = True
set :regressionnode.r_squared_change = True
set :regressionnode.selection_criteria = True
set :regressionnode.descriptives = True
set :regressionnode.p_correlations = True
set :regressionnode.collinearity_diagnostics = True
set :regressionnode.confidence_interval = True
set :regressionnode.covariance_matrix = True
```

```
set :regressionnode.durbin_watson = True
```

Propiedades de regressionnode	Valores	Descripción de la propiedad
target	campo	Los modelos de regresión requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de ponderación. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
method	Enter Stepwise Backwards Forwards	
include_constant	marca	
use_weight	marca	
weight_field	campo	
mode	Simple Expert	
complete_records	marca	
tolerance	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	Utilice comillas dobles para los argumentos.
stepping_method	useP useF	useP : utilice probabilidad de F useF: usar valor de F
probability_entry	número	
probability_removal	número	
F_value_entry	número	
F_value_removal	número	
selection_criteria	marca	
confidence_interval	marca	
covariance_matrix	marca	
collinearity_diagnostics	marca	
regression_coefficients	marca	
exclude_fields	marca	
durbin_watson	marca	
model_fit	marca	
r_squared_change	marca	
p_correlations	marca	
descriptives	marca	
calculate_variable_importance	marca	

## Propiedades de sequencenode



El nodo Secuencia encuentra reglas de asociación en datos secuenciales o en datos ordenados en el tiempo. Una secuencia es una lista de conjuntos de elementos que tiende a producirse en un orden previsible. Por ejemplo, es probable que un cliente que compra una cuchilla y una loción para después del afeitado compre crema para afeitar la próxima vez que vaya a comprar. El nodo Secuencia se basa en el algoritmo de reglas de asociación de CARMA, que utiliza un método de dos pasos para encontrar las secuencias. [Si desea obtener más información, consulte el tema \*Nodo Secuencia en el capítulo 12 en \*Nodos de modelado de IBM SPSS Modeler 15\*\*.](#)

### Ejemplo

```
create sequencenode
connect :databasenode to :sequencenode
# Pestaña "Campos"
set :sequencenode.id_field = 'Age'
set :sequencenode.contiguous = True
set :sequencenode.use_time_field = True
set :sequencenode.time_field = 'Date1'
set :sequencenode.content_fields = ['Drug' 'BP']
set :sequencenode.partition = Test
# Pestaña "Modelo"
set :sequencenode.use_model_name = True
set :sequencenode.model_name = "Sequence_test"
set :sequencenode.use_partitioned_data = False
set :sequencenode.min_supp = 15.0
set :sequencenode.min_conf = 14.0
set :sequencenode.max_size = 7
set :sequencenode.max_predictions = 5
# Pestaña "Experto"
set :sequencenode.mode = Expert
set :sequencenode.use_max_duration = True
set :sequencenode.max_duration = 3.0
set :sequencenode.use_pruning = True
set :sequencenode.pruning_value = 4.0
set :sequencenode.set_mem_sequences = True
set :sequencenode.mem_sequences = 5.0
set :sequencenode.use_gaps = True
set :sequencenode.min_item_gap = 20.0
set :sequencenode.max_item_gap = 30.0
```

Propiedades de sequencenode	Valores	Descripción de la propiedad
id_field	<i>campo</i>	Para crear un modelo de secuencias, es necesario especificar un campo de ID, un campo de tiempo opcional y uno o varios campos de contenido. Los campos de ponderación y frecuencia no se usan. <a href="#">Si desea obtener más información, consulte el tema <i>Propiedades comunes de nodos de modelado</i> el p. 195.</a>
time_field	<i>campo</i>	

Propiedades de sequencenode	Valores	Descripción de la propiedad
use_time_field	marca	
content_fields	[campo1 ... campon]	
contiguous	marca	
min_supp	número	
min_conf	número	
max_size	número	
max_predictions	número	
mode	Simple Expert	
use_max_duration	marca	
max_duration	número	
use_gaps	marca	
min_item_gap	número	
max_item_gap	número	
use_pruning	marca	
pruning_value	número	
set_mem_sequences	marca	
mem_sequences	entero	

## Propiedades de slrmnode



El nodo Modelo de respuesta de autoaprendizaje (SLRM) permite crear un modelo en el que un solo caso nuevo o un pequeño número de casos nuevos se pueden utilizar para volver a calcular el modelo sin tener que entrenar de nuevo el modelo utilizando todos los datos. [Si desea obtener más información, consulte el tema Nodo SLRM en el capítulo 14 en Nodos de modelado de IBM SPSS Modeler 15.](#)

### Ejemplo

```
create slrmnode
set :slrmnode.target = Offer
set :slrmnode.target_response = Response
set :slrmnode.inputs = ['Cust_ID' 'Age' 'Ave_Bal']
```

Propiedades de slrmnode	Valores	Descripción de la propiedad
target	campo	El campo objetivo debe ser un campo nominal o marca. También se puede especificar un campo de frecuencia. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
target_response	campo	El tipo debe ser marca.
continue_training_existing_model	marca	
target_field_values	marca	Use all: usar todos los valores del origen. Specify: son necesarios determinados valores.

Propiedades de <code>slrmnode</code>	Valores	Descripción de la propiedad
<code>target_field_values_specify</code>	<i>[campo1 ... campoN]</i>	
<code>include_model_assessment</code>	<i>marca</i>	
<code>model_assessment_random_seed</code>	<i>número</i>	Debe ser un número real.
<code>model_assessment_sample_size</code>	<i>número</i>	Debe ser un número real.
<code>model_assessment_iterations</code>	<i>número</i>	Número de iteraciones.
<code>display_model_evaluation</code>	<i>marca</i>	
<code>max_predictions</code>	<i>número</i>	
<code>randomization</code>	<i>número</i>	
<code>scoring_random_seed</code>	<i>número</i>	
<code>sort</code>	Ascending Descending	Especifica si se mostrarán primero las ofertas con las puntuaciones más altas o más bajas.
<code>model_reliability</code>	<i>marca</i>	
<code>calculate_variable_importance</code>	<i>marca</i>	

## Propiedades de `statisticsmodelnode`



El nodo Modelo Statistics permite analizar y trabajar con sus datos ejecutando los procedimientos de IBM® SPSS® Statistics que producen PMML. Este nodo requiere una copia de SPSS Statistics con licencia. [Si desea obtener más información, consulte el tema Nodo Modelo de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

Las propiedades de este nodo están descritas en [Propiedades de `statisticsmodelnode` el p. 316](#).

## Propiedades de `svmnode`



El nodo Máquina de vectores de soporte (SVM) le permite clasificar datos en uno o dos grupos sin que haya un ajuste por exceso. SVM funciona bien con conjuntos de datos grandes, como aquellos con un gran número de campos de entrada. [Si desea obtener más información, consulte el tema Nodo SVM en el capítulo 15 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create svmnode
# Pestaña Experto
set :svmnode.mode=Expert
set :svmnode.all_probabilities=True
set :svmnode.kernel=Polynomial
```

```
set :svmnode.gamma=1.5
```

Propiedades de svmnode	Valores	Descripción de la propiedad
all_probabilities	marca	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (por defecto) 1.0E-4 1.0E-5 1.0E-6	Determina cuándo detener el algoritmo de optimización.
regularization	número	También se conoce como el parámetro C.
precision	número	Sólo se utiliza si el nivel de medición del campo objetivo es Continuous.
kernel	RBF (por defecto) Polynomial Sigmoid Linear	Tipo de función kernel utilizada para la transformación.
rbf_gamma	número	Utilizado sólo si kernel es RBF.
gamma	número	Utilizado sólo si kernel es Polynomial o Sigmoid.
bias	número	
degree	número	Utilizado sólo si kernel es Polynomial.
calculate_variable_importance	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	
adjusted_propensity_partition	Test Validation	

## Propiedades de timeseriesnode



El nodo Serie temporal estima modelos de suavizado exponencial, modelos autorregresivos integrados de media móvil (ARIMA) univariados y modelos ARIMA (o de función de transferencia) multivariados para series temporales y genera datos de predicciones. Un nodo Serie temporal debe ir siempre precedido por un nodo Intervalos de tiempo. [Si desea obtener más información, consulte el tema Nodo Modelos de series temporales en el capítulo 13 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create timeseriesnode
set :timeseriesnode.method = Exsmooth
set :timeseriesnode.exsmooth_model_type = HoltsLinearTrend
set :timeseriesnode.exsmooth_transformation_type = None
```

Propiedades de timeseriesnode	Valores	Descripción de la propiedad
targets	<i>campo</i>	El nodo Serie temporal pronostica uno o más objetivos, utilizando opcionalmente uno o más campos de entrada como predictores. Los campos de frecuencia y ponderación no se usan. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
continue	<i>marca</i>	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	<i>marca</i>	
consider_seasonal	<i>marca</i>	
detect_outliers	<i>marca</i>	
expert_outlier_additive	<i>marca</i>	
expert_outlier_level_shift	<i>marca</i>	
expert_outlier_innovational	<i>marca</i>	
expert_outlier_level_shift	<i>marca</i>	
expert_outlier_transient	<i>marca</i>	
expert_outlier_seasonal_additive	<i>marca</i>	
expert_outlier_local_trend	<i>marca</i>	
expert_outlier_additive_patch	<i>marca</i>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	<i>entero</i>	
arima_d	<i>entero</i>	
arima_q	<i>entero</i>	
arima_sp	<i>entero</i>	
arima_sd	<i>entero</i>	
arima_sq	<i>entero</i>	



Propiedades de timeseriesnode	Valores	Descripción de la propiedad
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	marca	
tf_arima_p.nombredcampo	entero	Para funciones de transferencia.
tf_arima_d.nombredcampo	entero	Para funciones de transferencia.
tf_arima_q.nombredcampo	entero	Para funciones de transferencia.
tf_arima_sp.nombredcampo	entero	Para funciones de transferencia.
tf_arima_sd.nombredcampo	entero	Para funciones de transferencia.
tf_arima_sq.nombredcampo	entero	Para funciones de transferencia.
tf_arima_delay.nombredcampo	entero	Para funciones de transferencia.
tf_arima_transformation_type.nombredcampo	None SquareRoot NaturalLog	Para funciones de transferencia.
arima_detect_outlier_mode	None Automatic	
arima_outlier_additive	marca	
arima_outlier_level_shift	marca	
arima_outlier_innovational	marca	
arima_outlier_transient	marca	
arima_outlier_seasonal_additive	marca	
arima_outlier_local_trend	marca	
arima_outlier_additive_patch	marca	
conf_limit_pct	real	
max_lags	entero	
events	campos	
scoring_model_only	marca	Se utiliza para modelos con cifras muy grandes (cientos de miles) o series temporales.

## Propiedades de twostepnode



El nodo Bietápico es un método de conglomerado de dos pasos. El primer paso es hacer una única pasada por los datos para comprimir los datos de entrada de la fila en un conjunto de subconglomerados administrable. El segundo paso utiliza un método de conglomerado jerárquico para fundir progresivamente los subconglomerados en conglomerados cada vez más grandes. El bietápico tiene la ventaja de estimar automáticamente el número óptimo de conglomerados para los datos de entrenamiento. Puede gestionar tipos de campos mixtos y grandes conjuntos de datos eficazmente. [Si desea obtener más información, consulte el tema Nodo de conglomerado Bietápico en el capítulo 11 en \*Nodos de modelado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create twostep
set :twostep.custom_fields = True
set :twostep.inputs = ['Age' 'K' 'Na' 'BP']
set :twostep.partition = Test
set :twostep.use_model_name = False
set :twostep.model_name = "TwoStep_Drug"
set :twostep.use_partitioned_data = True
set :twostep.exclude_outliers = True
set :twostep.cluster_label = "String"
set :twostep.label_prefix = "TwoStep_"
set :twostep.cluster_num_auto = False
set :twostep.max_num_clusters = 9
set :twostep.min_num_clusters = 3
set :twostep.num_clusters = 7
```

Propiedades de twostepnode	Valores	Descripción de la propiedad
inputs	[ <i>campo1</i> ... <i>campoN</i> ]	Los modelos bietápicos utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se reconocen. <a href="#">Si desea obtener más información, consulte el tema Propiedades comunes de nodos de modelado el p. 195.</a>
standardize	<i>marca</i>	
exclude_outliers	<i>marca</i>	
percentage	<i>número</i>	
cluster_num_auto	<i>marca</i>	
min_num_clusters	<i>número</i>	
max_num_clusters	<i>número</i>	
num_clusters	<i>número</i>	
cluster_label	String Number	
label_prefix	<i>cadena</i>	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

## Propiedades de nodos de nugget de modelo

Los nodos de nugget de modelo comparten las mismas propiedades comunes que los otros nodos. Si desea obtener más información, consulte el tema [Propiedades de nodos comunes en el capítulo 9 el p. 117](#).

### Propiedades de *applyanomalydetectionnode*

Los nodos de modelado Detección de anomalías pueden utilizarse para generar un nugget de modelo Detección de anomalías. El nombre de proceso de este nugget de modelo es *applyanomalydetectionnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de anomalydetectionnode en el capítulo 16 el p. 196](#).

Propiedades de <i>applyanomalydetectionnode</i>	Valores	Descripción de la propiedad
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Determina que resultados se crean para puntuación.
<code>num_fields</code>	<i>entero</i>	Campos para informar.
<code>discard_records</code>	<i>marca</i>	Indica si los registros se descartan del resultado o no.
<code>discard_anomalous_records</code>	<i>marca</i>	Indicador de cuando descartar los registros anómalos o <i>no</i> anómalos. Por defecto es <i>off</i> , que significa que se descartan los registros <i>no</i> anómalos. En caso contrario, si es <i>on</i> , se descartan los registros anómalos. Esta propiedad se activa sólo si la propiedad <code>discard_records</code> se activa.

### Propiedades de *applyapriorinode*

Los nodos de modelado Apriori pueden utilizarse para generar un nugget de modelo Apriori. El nombre de proceso de este nugget de modelo es *applyapriorinode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de apriorinode en el capítulo 16 el p. 197](#).

Propiedades de <i>applyapriorinode</i>	Valores	Descripción de la propiedad
<code>max_predictions</code>	<i>número (entero)</i>	
<code>ignore_unmatached</code>	<i>marca</i>	
<code>allow_repeats</code>	<i>marca</i>	

Propiedades de <b>applypriorinode</b>	Valores	Descripción de la propiedad
check_basket	NoPredictions Predictions NoCheck	
criterion	Confianza Support RuleSupport Lift Deployability	

## ***Propiedades de applyautoclassifiernode***

Los nodos de modelado de clasificador automático se pueden utilizar para crear un nugget de modelo Clasificador automático. El nombre de proceso de este nugget de modelo es *applyautoclassifiernode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de autoclassifiernode en el capítulo 16 el p. 199](#).

Propiedades de <b>applyautoclassifiernode</b>	Valores	Descripción de la propiedad
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
flag_voting_tie_selection	Random HighestConfidence RawPropensity	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de conjunto.
set_voting_tie_selection	Random HighestConfidence	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo nominal.

## ***Propiedades de applyautoclusternode***

Los nodos de modelado de Conglomerado automático se pueden utilizar para crear un nugget de modelo Conglomerado automático. El nombre de proceso de este nugget de modelo es *applyautoclusternode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de nodo de autoconglomeración en el capítulo 16 el p. 201](#).

## Propiedades de *applyautonumericnode*

Los nodos de modelado autonumérico se pueden utilizar para crear un nugget de modelo Autonumérico. El nombre de proceso de este nugget de modelo es *applyautonumericnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de autonumericnode en el capítulo 16 el p. 203.](#)

Propiedades de <i>applyautonumericnode</i>	Valores	Descripción de la propiedad
calculate_standard_error	marca	

## Propiedades de *applybayesnetnode*

Los nodos de modelado de red bayesiana pueden utilizarse para generar un nugget de modelo de red bayesiana. El nombre de proceso de este nugget de modelo es *applybayesnetnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de bayesnetnode en el capítulo 16 el p. 204.](#)

Propiedades de <i>applybayesnetnode</i>	Valores	Descripción de la propiedad
all_probabilities	marca	
raw_propensity	marca	
adjusted_propensity	marca	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de *applyc50node*

Los nodos de modelado C5.0 pueden utilizarse para generar un nugget de modelo C5.0. El nombre de proceso de este nugget de modelo es *applyc50node*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de c50node en el capítulo 16 el p. 206.](#)

Propiedades de <i>applyc50node</i>	Valores	Descripción de la propiedad
sql_generate	Nunca NoMissingValues	Se utiliza para establecer las opciones de generación de SQL durante la ejecución del conjunto de reglas.
calculate_conf	marca	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de *applycarmanode*

Los nodos de modelado CARMA pueden utilizarse para generar un nugget de modelo CARMA. El nombre de proceso de este nugget de modelo es *applycarmanode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de carmanode en el capítulo 16 el p. 207](#).

## Propiedades de *applycartnode*

Se pueden utilizar los nodos de modelado C&RT para generar un nugget de modelo C&RT. El nombre de proceso de este nugget de modelo es *applycartnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de cartnode en el capítulo 16 el p. 208](#).

Propiedades de <i>applycartnode</i>	Valores	Descripción de la propiedad
sql_generate	Nunca MissingValues NoMissingValues	Se utiliza para establecer las opciones de generación de SQL durante la ejecución del conjunto de reglas.
calculate_conf	<i>marca</i>	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
display_rule_id	<i>marca</i>	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
calculate_raw_propensities	<i>marca</i>	
calculate_adjusted_propensities	<i>marca</i>	

## Propiedades de *applychaidnode*

Los nodos de modelado CHAID pueden utilizarse para generar un nugget de modelo CHAID. El nombre de proceso de este nugget de modelo es *applychaidnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de chaidnode en el capítulo 16 el p. 211](#).

Propiedades de <i>applychaidnode</i>	Valores	Descripción de la propiedad
sql_generate	Never MissingValues	
calculate_conf	<i>marca</i>	
display_rule_id	<i>marca</i>	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
calculate_raw_propensities	<i>marca</i>	
calculate_adjusted_propensities	<i>marca</i>	

## Propiedades de applycoxregnode

Los nodos de modelado Cox pueden utilizarse para generar un nugget de modelo Cox. El nombre de proceso de este nugget de modelo es *applycoxregnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de coxregnode en el capítulo 16 el p. 213](#).

Propiedades de applycoxregnode	Valores	Descripción de la propiedad
future_time_as	Intervals Fields	
time_interval	número	
num_future_times	entero	
time_field	campo	
past_survival_time	campo	
all_probabilities	marca	
cumulative_hazard	marca	

## Propiedades de applydecisionlistnode

Los nodos de modelado Lista de decisiones pueden utilizarse para generar un nugget de modelo Lista de decisiones. El nombre de proceso de este nugget de modelo es *applydecisionlistnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de decisionlistnode en el capítulo 16 el p. 215](#).

Propiedades de applydecisionlistnode	Valores	Descripción de la propiedad
enable_sql_generation	marca	Cuando se establece en true, IBM® SPSS® Modeler intenta enviar el modelo Lista de decisiones a SQL.
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de applydiscriminantnode

Los nodos de modelado Discriminante pueden utilizarse para generar un nugget de modelo Discriminante. El nombre de proceso de este nugget de modelo es *applydiscriminantnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de discriminantnode en el capítulo 16 el p. 216](#).

Propiedades de applydiscriminantnode	Valores	Descripción de la propiedad
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de *applyfactornode*

Los nodos de modelado PCA/Factorial pueden utilizarse para generar un nugget de modelo PCA/Factorial. El nombre de proceso de este nugget de modelo es *applyfactornode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de factornode en el capítulo 16 el p. 218.](#)

## Propiedades de *applyfeatureselectionnode*

Los nodos de modelado Selección de características pueden utilizarse para generar un nugget de modelo Selección de características. El nombre de proceso de este nugget de modelo es *applyfeatureselectionnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de featureselectionnode en el capítulo 16 el p. 220.](#)

Propiedades de <i>applyfeatureselectionnode</i>	Valores	Descripción de la propiedad
selected_ranked_fields		especifica qué campos clasificados se comprueban en el explorador de modelos.
selected_screened_fields		Especifica qué campos filtrados se comprueban en el explorador de modelos.

## Propiedades de *applygeneralizedlinearnode*

Los nodos de modelado lineal generalizado (genlin) pueden utilizarse para generar un nugget de modelo lineal generalizado. El nombre de proceso de este nugget de modelo es *applygeneralizedlinearnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de genlinnode en el capítulo 16 el p. 222.](#)

Propiedades de <i>applygeneralizedlinearnode</i>	Valores	Descripción de la propiedad
calculate_raw_propensities	<i>marca</i>	
calculate_adjusted_propensities	<i>marca</i>	

## Propiedades de *applykmeansnode*

Los nodos de modelado K-medias pueden utilizarse para generar un nugget de modelo K-medias. El nombre de proceso de este nugget de modelo es *applykmeansnode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de kmeansnode en el capítulo 16 el p. 228.](#)



## Propiedades de *applyknnnode*

Los nodos de modelado KNN pueden utilizarse para generar un nugget de modelo KNN. El nombre de proceso de este nugget de modelo es *applyknnnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de \*knnnode\* en el capítulo 16 el p. 229](#).

Propiedades de <i>applyknnnode</i>	Valores	Descripción de la propiedad
<i>all_probabilities</i>	<i>marca</i>	
<i>save_distances</i>	<i>marca</i>	

## Propiedades de *applykohonennode*

Los nodos de modelado Kohonen pueden utilizarse para generar un nugget de modelo Kohonen. El nombre de proceso de este nugget de modelo es *applykohonennode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de \*c50node\* en el capítulo 16 el p. 206](#).

## Propiedades de *applylinearnode*

Los nodos de modelado lineal pueden utilizarse para generar un nugget de modelo lineal. El nombre de proceso de este nugget de modelo es *applylinearnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de \*linearnode\* en el capítulo 16 el p. 232](#).

Propiedades de <i>lineal</i>	Valores	Descripción de la propiedad
<i>use_custom_name</i>	<i>marca</i>	
<i>custom_name</i>	<i>cadena</i>	
<i>enable_sql_generation</i>	<i>marca</i>	

## Propiedades de *applylogregnode*

Los nodos de modelado Regresión logística pueden utilizarse para generar un nugget de modelo Regresión logística. El nombre de proceso de este nugget de modelo es *applylogregnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de \*logregnode\* en el capítulo 16 el p. 233](#).

Propiedades de <i>applylogregnode</i>	Valores	Descripción de la propiedad
<i>calculate_raw_propensities</i>	<i>marca</i>	

## Propiedades de *applyneuralnetnode*

Los nodos de modelado Red neuronal pueden utilizarse para generar un nugget de modelo Red neuronal. El nombre de proceso de este nugget de modelo es *applyneuralnetnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de neuralnetnode en el capítulo 16 el p. 238](#).

**Precaución:** Una versión más reciente del nugget Red neural, con características mejoradas, está disponible en esta versión y se describe en la sección siguiente (*applyneuralnetwork*). Aunque la versión anterior sigue estando disponible, le recomendamos actualizar sus procesos para que se usen la nueva versión. En este documento se incluyen detalles de la versión anterior como referencia, pero en versiones futuras dejará de ser compatible.

Propiedades de <i>applyneuralnetnode</i>	Valores	Descripción de la propiedad
calculate_conf	marca	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
enable_sql_generation	marca	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de *applyneuralnetworknode*

Los nodos de modelado Red neuronal pueden utilizarse para generar un nugget de modelo Red neuronal. El nombre de proceso de este nugget de modelo es *applyneuralnetworknode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de neuralnetworknode en el capítulo 16 el p. 240](#).

Propiedades de <i>applyneuralnetworknode</i>	Valores	Descripción de la propiedad
use_custom_name	marca	
custom_name	cadena	
confidence	onProbability onIncrease	
score_category_probabilities	marca	
max_categories	número	
score_propensity	marca	

## Propiedades de *applyquestnode*

Los nodos de modelado QUEST pueden utilizarse para generar un nugget de modelo QUEST. El nombre de proceso de este nugget de modelo es *applyquestnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de questnode en el capítulo 16 el p. 242](#).

Propiedades de <i>applyquestnode</i>	Valores	Descripción de la propiedad
sql_generate	Nunca MissingValues NoMissingValues	
calculate_conf	marca	
display_rule_id	marca	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
calculate_raw_propensities	marca	
calculate_adjusted_propensities	marca	

## Propiedades de *applyregressionnode*

Los nodos de modelado Regresión lineal pueden utilizarse para generar un nugget de modelo Regresión lineal. El nombre de proceso de este nugget de modelo es *applyregressionnode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de regressionnode en el capítulo 16 el p. 244](#).

## Propiedades de *applyselflearningnode*

Los nodos de modelado de modelo de respuesta de autoaprendizaje (SLRM) pueden utilizarse para generar un nugget de modelo SLRM. El nombre de proceso de este nugget de modelo es *applyselflearningnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, consulte [Propiedades de slrmnode en el capítulo 16 el p. 247](#).

Propiedades de <i>applyselflearningnode</i>	Valores	Descripción de la propiedad
max_predictions	número	
randomization	número	
scoring_random_seed	número	
sort	ascending descending	Especifica si se mostrarán primero las ofertas con las puntuaciones más altas o más bajas.
model_reliability	marca	Tiene en cuenta la opción de fiabilidad del modelo de la pestaña Configuración.

## Propiedades de *applysequencenode*

Los nodos de modelado Secuencia pueden utilizarse para generar un nugget de modelo Secuencia. El nombre de proceso de este nugget de modelo es *applysequencenode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de sequencenode en el capítulo 16 el p. 246](#).

## Propiedades de *applysvmnode*

Los nodos de modelado SVM pueden utilizarse para generar un nugget de modelo SVM. El nombre de proceso de este nugget de modelo es *applysvmnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de svmnode en el capítulo 16 el p. 248](#).

Propiedades de <i>applysvmnode</i>	Valores	Descripción de la propiedad
<i>all_probabilities</i>	<i>marca</i>	
<i>calculate_raw_propensities</i>	<i>marca</i>	
<i>calculate_adjusted_propensities</i>	<i>marca</i>	

## Propiedades de *applytimeseriesnode*

Los nodos de modelado Serie temporal pueden utilizarse para generar un nugget de modelo Serie temporal. El nombre de proceso de este nugget de modelo es *applytimeseriesnode*. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de timeseriesnode en el capítulo 16 el p. 249](#).

Propiedades de <i>applytime-seriesnode</i>	Valores	Descripción de la propiedad
<i>calculate_conf</i>	<i>marca</i>	
<i>calculate_residuals</i>	<i>marca</i>	

## Propiedades de *applytwostepnode*

Los nodos de modelado Bietápico pueden utilizarse para generar un nugget de modelo Bietápico. El nombre de proceso de este nugget de modelo es *applytwostepnode*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre el procesamiento del propio nodo de modelado, [consulte Propiedades de twostepnode en el capítulo 16 el p. 252](#).

## ***Propiedades de nodos de modelado de base de datos***

IBM® SPSS® Modeler admite la integración con herramientas de modelado y minería de datos ofrecidas por proveedores de bases de datos, incluidos Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM® DB2® InfoSphere Warehouse y IBM® Netezza® Analytics. [Si desea obtener más información, consulte el tema Conceptos básicos del modelado de la base de datos en el capítulo 2 en \*Manual de minería interna de la base de datos de IBM SPSS Modeler 15\*.](#)

Podrá crear y almacenar modelos mediante algoritmos nativos de bases de datos, todo desde la aplicación SPSS Modeler. Los modelos de base de datos también se pueden crear y manipular a través de procesos utilizando las propiedades descritas en esta sección.

Por ejemplo, el siguiente fragmento de proceso muestra la creación de un modelo de árboles de decisión de Microsoft mediante la interfaz de procesos de SPSS Modeler:

```
create mstreenode
rename :mstreenode as msbuilder
set msbuilder.analysis_server_name = 'localhost'
set msbuilder.analysis_database_name = 'TESTDB'
set msbuilder.mode = 'Expert'
set msbuilder.datasource = 'LocalServer'
set msbuilder.target = 'Drug'
set msbuilder.inputs = ['Edad' 'Sexo']
set msbuilder.unique_field = 'IDX'
set msbuilder.custom_fields = true
set msbuilder.model_name = 'MSDRUG'

connect :typenode to msbuilder
execute msbuilder

insert model MSDRUG connected between :typenode and :tablenode
set MSDRUG.sql_generate = true
execute :tablenode
```

## Propiedades de nodos de modelado de Microsoft

### Propiedades de nodos de modelado de Microsoft

#### Propiedades comunes

Las siguientes propiedades son comunes a los nodos de modelado de la base de datos de Microsoft.

Propiedades comunes de nodo de Microsoft	Valores	Descripción de la propiedad
analysis_database_name	cadena	Nombre de la base de datos de Analysis Services.
analysis_server_name	cadena	Nombre del host de Analysis Services.
use_transactional_data	marca	Especifica si los datos de entrada están en formato tabular o transaccional.
inputs	[campo campo campo]	Campos de entrada de datos tabulares.
target	campo	Campo pronosticado (no aplicable a nodo Conglomerado de MS o nodos de Conglomeración de secuencias).
unique_field	campo	Campos clave.
msas_parameters	estructurado	Parámetros del algoritmo. <a href="#">Si desea obtener más información, consulte el tema Parámetros del algoritmo el p. 265.</a>
with_drillthrough	marca	Opción Con exploración.

#### Árbol de decisión de MS

No hay propiedades específicas para los nodos del tipo `mstreenode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Conglomerado de MS

No hay propiedades específicas para los nodos del tipo `msclusternode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Reglas de asociación de MS

Las siguientes propiedades específicas están disponibles para los nodos del tipo `msassocnode`:

Propiedades de msassocnode	Valores	Descripción de la propiedad
id_field	campo	Identifica todas las transacciones en los datos.
trans_inputs	[campo campo campo]	Los campos de entrada de datos transaccionales.
transactional_target	campo	Campo pronosticado (datos transaccionales).

**Naive bayesiano de MS**

No hay propiedades específicas para los nodos del tipo `msbayesnode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

**Regresión lineal de MS**

No hay propiedades específicas para los nodos del tipo `msregressionnode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

**Red neuronal de MS**

No hay propiedades específicas para los nodos del tipo `msneuralnetworknode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

**Regresión logística de MS**

No hay propiedades específicas para los nodos del tipo `mslogisticnode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

**Series temporales de MS**

No hay propiedades específicas para los nodos del tipo `mstimeseriesnode`. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

**Conglomeración de secuencias de MS**

Las siguientes propiedades específicas están disponibles para los nodos del tipo `mssequenceclusternode`:

Propiedades de <code>mssequenceclusternode</code>	Valores	Descripción de la propiedad
<code>id_field</code>	<i>campo</i>	Identifica todas las transacciones en los datos.
<code>input_fields</code>	<i>[campo campo campo]</i>	Los campos de entrada de datos transaccionales.
<code>sequence_field</code>	<i>campo</i>	Identificador de secuencia.
<code>target_field</code>	<i>campo</i>	Campo pronosticado (datos tabulares).

**Parámetros del algoritmo**

Cada tipo de modelo de base de datos de Microsoft tiene parámetros concretos que se pueden establecer mediante la propiedad `msas_parameters` ; por ejemplo:

```
set :msregressionnode.msas_parameters =
[{"MAXIMUM_INPUT_ATTRIBUTES" 255}{ "MAXIMUM_OUTPUT_ATTRIBUTES" 255}]
```

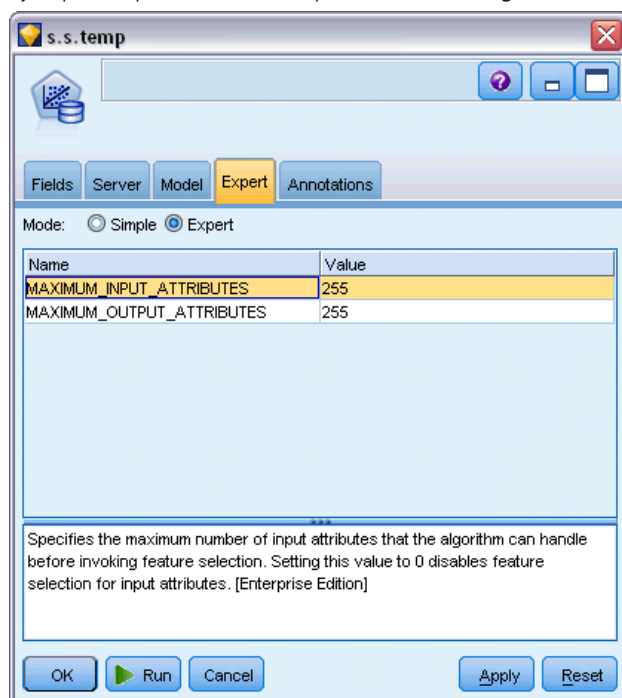
Estos parámetros se derivan de SQL Server. Para ver los parámetros relevantes para cada nodo:

- Coloque un nodo de fuente de base de datos en el lienzo.

- ▶ Abra el nodo de fuente de base de datos.
- ▶ Seleccione una fuente válida en la lista desplegable Fuente de datos.
- ▶ Seleccione una tabla válida en la lista Nombre de tabla.
- ▶ Pulse en Aceptar para cerrar el nodo de fuente de base de datos.
- ▶ Conecte un nodo de modelado de base de datos de Microsoft cuyas propiedades desee conocer.
- ▶ Abra el nodo de modelado de base de datos.
- ▶ Seleccione la pestaña Experto.

Se mostrarán las propiedades `msas_parameters` disponibles para este nodo.

Figura 18-1  
Ejemplo de presentación de parámetros del algoritmo



### **Propiedades de nugget de modelo de Microsoft**

Las siguientes propiedades son para los nugget de modelo creados mediante los nodos de modelado de bases de datos de Microsoft.



**Árbol de decisión de MS**

Propiedades de applymstreenode	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.
datasource	<i>cadena</i>	Nombre de la fuente de datos (DSN) ODBC de SQL Server.
sql_generate	<i>marca</i>	Activa la generación de SQL.

**Regresión lineal de MS**

Propiedades de applymsregressionnode	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.

**Red neuronal de MS**

Propiedades de applymsneural-networknode	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.

**Regresión logística de MS**

Propiedades de applymslogisticnode	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.

**Series temporales de MS**

Propiedades de <code>appliedtime-seriesnode</code>	Valores	Descripción
<code>analysis_database_name</code>	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
<code>analysis_server_name</code>	<i>cadena</i>	Nombre del host del servidor de análisis.
<code>start_from</code>	<code>new_prediction</code> <code>historical_prediction</code>	Especifica si se realizarán pronósticos futuros o históricos.
<code>new_step</code>	<i>número</i>	Define el período de tiempo inicial de pronósticos futuros.
<code>historical_step</code>	<i>número</i>	Define el período de tiempo inicial de pronósticos históricos.
<code>end_step</code>	<i>número</i>	Define el período de tiempo final de los pronósticos.

**Conglomeración de secuencias de MS**

Propiedades de <code>appliedsequenceclusternode</code>	Valores	Descripción
<code>analysis_database_name</code>	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta. Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
<code>analysis_server_name</code>	<i>cadena</i>	Nombre del host del servidor de análisis.

**Propiedades de nodos de modelado de Oracle****Propiedades de nodos de modelado de Oracle**

Las siguientes propiedades son comunes a los nodos de modelado de la base de datos de Oracle.

Propiedades comunes de nodos de Oracle	Valores	Descripción de la propiedad
<code>target</code>	<i>campo</i>	
<code>inputs</code>	<i>Lista de campos</i>	
<code>partition</code>	<i>campo</i>	Campo usado para dividir los datos en muestras independientes para las fases de entrenamiento, comprobación y validación en la generación del modelo.
<code>datasource</code>		
<code>username</code>		
<code>password</code>		
<code>epassword</code>		
<code>use_model_name</code>	<i>marca</i>	
<code>model_name</code>	<i>cadena</i>	Nombre personalizado para nuevo modelo.

Propiedades comunes de nodos de Oracle	Valores	Descripción de la propiedad
use_partitioned_data	marca	Si se ha definido un campo de partición, esta opción garantiza que solamente se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
unique_field	campo	
auto_data_prep	marca	Activa o desactiva la función de preparación de datos automática de Oracle (solamente para bases de datos 11g).
costs	estructurado	Propiedad estructurada de la siguiente forma: {{drugA drugB 1.5} {drugA drugC 2.1}}, donde los argumentos en {} son los costes pronosticados actuales.
mode	Simple Expert	Hace que se ignoren ciertas propiedades si se establece como Simple, como se indica en las propiedades de nodos individuales.
use_prediction_probability	marca	
prediction_probability	cadena	
use_prediction_set	marca	

### Bayesiano Naive de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oranbnode.

Propiedades de oranbnode	Valores	Descripción de la propiedad
singleton_threshold	número	0.0–1.0.*
pairwise_threshold	número	0.0–1.0.*
priors	Data Equal Custom	
custom_priors	estructurado	Propiedad estructurada de la siguiente forma: set :oranbnode.custom_priors = {{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}}

\* Propiedad ignorada si mode se establece como Simple.

### Bayesiano adaptativo de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oraabnnode.

Propiedades de oraabnnode	Valores	Descripción de la propiedad
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	marca	*
execution_time_limit	entero	El valor debe ser mayor que 0.*
max_naive_bayes_predictors	entero	El valor debe ser mayor que 0.*
max_predictors	entero	El valor debe ser mayor que 0.*

Propiedades de oraabnnode	Valores	Descripción de la propiedad
priors	Data Equal Custom	
custom_priors	<i>estructurado</i>	Propiedad estructurada de la siguiente forma: set :oraabnnode.custom_priors = [{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}]

\* Propiedad ignorada si mode se establece como Simple.

### ***Máquinas de vectores de soporte de Oracle***

Las siguientes propiedades están disponibles para los nodos del tipo orasvmnode.

Propiedades de orasvmnode	Valores	Descripción de la propiedad
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	<i>entero</i>	Solamente kernel gaussiano. El valor debe ser mayor que 0.*
convergence_tolerance	<i>número</i>	El valor debe ser mayor que 0.*
use_standard_deviation	<i>marca</i>	Solamente kernel gaussiano.*
standard_deviation	<i>número</i>	El valor debe ser mayor que 0.*
use_epsilon	<i>marca</i>	Solamente modelos de regresión.*
epsilon	<i>número</i>	El valor debe ser mayor que 0.*
use_complexity_factor	<i>marca</i>	*
complexity_factor	<i>número</i>	*
use_outlier_rate	<i>marca</i>	Solamente variantes de una clase.*
outlier_rate	<i>número</i>	Solamente variantes de una clase. 0.0–1.0.*
weights	Data Equal Custom	
custom_weights	<i>estructurado</i>	Propiedad estructurada de la siguiente forma: set :orasvmnode.custom_weights = [{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}]

\* Propiedad ignorada si mode se establece como Simple.

**Modelos lineales generalizados de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo `oraglmnode`.

Propiedades de <code>oraglmnode</code>	Valores	Descripción de la propiedad
<code>normalization_method</code>	<code>zscore</code> <code>minmax</code> <code>none</code>	
<code>missing_value_handling</code>	<code>ReplaceWith-Mean</code> <code>UseCompleteRecords</code>	
<code>use_row_weights</code>	<i>marca</i>	*
<code>row_weights_field</code>	<i>campo</i>	*
<code>save_row_diagnostics</code>	<i>marca</i>	*
<code>row_diagnostics_table</code>	<i>cadena</i>	*
<code>coefficient_confidence</code>	<i>número</i>	*
<code>use_reference_category</code>	<i>marca</i>	*
<code>reference_category</code>	<i>cadena</i>	*
<code>ridge_regression</code>	<code>Auto</code> <code>Off</code> <code>On</code>	*
<code>parameter_value</code>	<i>número</i>	*
<code>vif_for_ridge</code>	<i>marca</i>	*

\* Propiedad ignorada si `mode` se establece como `Simple`.

**Árbol de decisión de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo `oradecisiontreenode`.

Propiedades de <code>oradecisiontreenode</code>	Valores	Descripción de la propiedad
<code>use_costs</code>	<i>marca</i>	
<code>impurity_metric</code>	<code>Entropy</code> <code>Gini</code>	
<code>term_max_depth</code>	<i>entero</i>	2–20.*
<code>term_minpct_node</code>	<i>número</i>	0.0–10.0.*
<code>term_minpct_split</code>	<i>número</i>	0.0–20.0.*
<code>term_minrec_node</code>	<i>entero</i>	El valor debe ser mayor que 0.*
<code>term_minrec_split</code>	<i>entero</i>	El valor debe ser mayor que 0.*
<code>display_rule_ids</code>	<i>marca</i>	*

\* Propiedad ignorada si `mode` se establece como `Simple`.

**O-conglomerado de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo oraoclusternode.

Propiedades de oraoclusternode	Valores	Descripción de la propiedad
max_num_clusters	entero	El valor debe ser mayor que 0.
max_buffer	entero	El valor debe ser mayor que 0.*
sensitivity	número	0.0–1.0.*

\* Propiedad ignorada si mode se establece como Simple.

**K-medias de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo orakmeansnode.

Propiedades de orakmeansnode	Valores	Descripción de la propiedad
num_clusters	entero	El valor debe ser mayor que 0.
normalization_method	zscore minmax none	
distance_function	Euclidean Cosine	
iterations	entero	0–20.*
conv_tolerance	número	0.0–0.5.*
split_criterion	Variance Size	El valor por defecto es Variance.*
num_bins	entero	El valor debe ser mayor que 0.*
block_growth	entero	1–5.*
min_pct_attr_support	número	0.0–1.0.*

\* Propiedad ignorada si mode se establece como Simple.

**NMF de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo oranmfnode.

Propiedades de oranmfnode	Valores	Descripción de la propiedad
normalization_method	minmax none	
use_num_features	marca	*
num_features	entero	0–1. El algoritmo estima el valor por defecto a partir de los datos.*
random_seed	número	*
num_iterations	entero	0–500.*
conv_tolerance	número	0.0–0.5.*
display_all_features	marca	*

\* Propiedad ignorada si mode se establece como Simple.

**Apriori de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo oraapriorinode.

Propiedades de oraapriorinode	Valores	Descripción de la propiedad
content_field	<i>campo</i>	
id_field	<i>campo</i>	
max_rule_length	<i>entero</i>	2–20.
min_confidence	<i>número</i>	0.0–1.0.
min_support	<i>número</i>	0.0–1.0.
use_transactional_data	<i>marca</i>	

**Longitud mínima de la descripción de Oracle (LMD)**

No hay propiedades específicas para los nodos del tipo oramdlnode. Consulte las propiedades comunes de Oracle que se indican al comienzo de esta sección.

**Importancia del atributo de Oracle (AI)**

Las siguientes propiedades están disponibles para los nodos del tipo oraainode.

Propiedades de oraainode	Valores	Descripción de la propiedad
custom_fields	<i>marca</i>	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
selection_mode	ImportanceLevel Importance-Value TopN	
select_important	<i>marca</i>	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos importantes.
important_label	<i>cadena</i>	Especifica la etiqueta para el rango “important”.
select_marginal	<i>marca</i>	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos marginales.
marginal_label	<i>cadena</i>	Especifica la etiqueta para el rango “marginal”.
important_above	<i>número</i>	0.0–1.0.
select_unimportant	<i>marca</i>	Si selection_mode está definido como ImportanceLevel, determina si se seleccionan los campos no importantes.
unimportant_label	<i>cadena</i>	Especifica la etiqueta para el rango “unimportant”.
unimportant_below	<i>número</i>	0.0–1.0.

Propiedades de oraainode	Valores	Descripción de la propiedad
importance_value	número	Si selection_mode está definido como ImportanceValue, determina el valor de corte que se va a usar. Acepta valores de 0 a 100.
top_n	número	Si selection_mode está definido como TopN, determina el valor de corte que se va a usar. Acepta valores de 0 a 1000.

### **Propiedades de nugget de modelo de Oracle**

Las siguientes propiedades son para los nugget de modelo creados mediante los modelos de Oracle.

#### **Bayesiano Naive de Oracle**

No hay propiedades específicas para los nodos del tipo applyoranbnnode.

#### **Bayesiano adaptativo de Oracle**

No hay propiedades específicas para los nodos del tipo applyoraabnnode.

#### **Máquinas de vectores de soporte de Oracle**

No hay propiedades específicas para los nodos del tipo applyorasvmnode.

#### **Árbol de decisión de Oracle**

Las siguientes propiedades están disponibles para los nodos del tipo applyoradecisiontreenode.

Propiedades de applyoradecisiontreenode	Valores	Descripción de la propiedad
use_costs	marca	
display_rule_ids	marca	

#### **O-conglomerado de Oracle**

No hay propiedades específicas para los nodos del tipo applyoraoclusternode.

#### **K-medias de Oracle**

No hay propiedades específicas para los nodos del tipo applyorakmeansnode.

#### **NMF de Oracle**

La siguiente propiedad está disponible para los nodos del tipo applyoranmfnode:

Propiedades de applyoranmfnode	Valores	Descripción de la propiedad
display_all_features	marca	



**Apriori de Oracle**

Este nugget de modelo no se puede aplicar en el procesamiento.

**LMD de Oracle**

Este nugget de modelo no se puede aplicar en el procesamiento.

**Propiedades de nodos para modelado de IBM DB2****Propiedades de nodos de modelado de IBM DB2**

Las siguientes propiedades son comunes a los nodos de modelado de la base de datos de IBM InfoSphere Warehouse (ISW).

Propiedades comunes de nodo de ISW	Valores	Descripción de la propiedad
inputs	<i>Lista de campos</i>	
datasource		
username		
password		
epassword		
enable_power_options	<i>marca</i>	
power_options_max_memory	<i>entero</i>	El valor debe ser mayor que 32.
power_options_cmdline	<i>cadena</i>	
mining_data_custom_sql	<i>cadena</i>	
logical_data_custom_sql	<i>cadena</i>	
mining_settings_custom_sql		

**Árbol de decisión de ISW**

Las siguientes propiedades están disponibles para los nodos del tipo db2imtreeode.

Propiedades de db2imtreeode	Valores	Descripción de la propiedad
target	<i>campo</i>	
perform_test_run	<i>marca</i>	
use_max_tree_depth	<i>marca</i>	
max_tree_depth	<i>entero</i>	Valor mayor que 0.
use_maximum_purity	<i>marca</i>	
maximum_purity	<i>número</i>	Número entre 0 y 100.
use_minimum_internal_cases	<i>marca</i>	
minimum_internal_cases	<i>entero</i>	Valor mayor que 1.
use_costs	<i>marca</i>	
costs	<i>estructurado</i>	Propiedad estructurada de la siguiente forma: <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , donde los argumentos en <code>{ }</code> son los costes pronosticados actuales.

**Asociación de ISW**

Las siguientes propiedades están disponibles para los nodos del tipo db2imassocnode.

Propiedades de db2imassocnode	Valores	Descripción de la propiedad
use_transactional_data	<i>marca</i>	
id_field	<i>campo</i>	
content_field	<i>campo</i>	
data_table_layout	basic limited_length	
max_rule_size	<i>entero</i>	El valor debe ser mayor que 2.
min_rule_support	<i>número</i>	0–100%
min_rule_confidence	<i>número</i>	0–100%
use_item_constraints	<i>marca</i>	
item_constraints_type	Include Exclude	
use_taxonomy	<i>marca</i>	
taxonomy_table_name	<i>cadena</i>	Nombre de la tabla de DB2 para almacenar los detalles de taxonomía.
taxonomy_child_column_name	<i>cadena</i>	Nombre de la columna filial en la tabla de taxonomía. La columna filial contiene los nombres de los elementos o de las categorías.
taxonomy_parent_column_name	<i>cadena</i>	Nombre de la columna parental en la tabla de taxonomía. La columna parental contiene los nombres de las categorías.
load_taxonomy_to_table	<i>marca</i>	Comprueba si la información sobre taxonomía almacenada en IBM® SPSS® Modeler se debe cargar en la tabla de taxonomía en el momento de la generación del modelo. Tenga en cuenta que si la tabla de taxonomía ya existe, se eliminará. La información de taxonomía se almacena con el nodo de generación de modelos y se puede editar mediante los botones Editar categorías y Editar taxonomía.

**Secuencia de ISW**

Las siguientes propiedades están disponibles para los nodos del tipo db2imsequencenode.

Propiedades de db2imsequencenode	Valores	Descripción de la propiedad
id_field	<i>campo</i>	
group_field	<i>campo</i>	
content_field	<i>campo</i>	
max_rule_size	<i>entero</i>	El valor debe ser mayor que 2.
min_rule_support	<i>número</i>	0–100%
min_rule_confidence	<i>número</i>	0–100%
use_item_constraints	<i>marca</i>	

Propiedades de db2imsequencenode	Valores	Descripción de la propiedad
item_constraints_type	Include Exclude	
use_taxonomy	marca	
taxonomy_table_name	cadena	Nombre de la tabla de DB2 para almacenar los detalles de taxonomía.
taxonomy_child_column_name	cadena	Nombre de la columna filial en la tabla de taxonomía. La columna filial contiene los nombres de los elementos o de las categorías.
taxonomy_parent_column_name	cadena	Nombre de la columna parental en la tabla de taxonomía. La columna parental contiene los nombres de las categorías.
load_taxonomy_to_table	marca	Comprueba si la información sobre taxonomía almacenada en SPSS Modeler se debe cargar en la tabla de taxonomía en el momento de la generación del modelo. Tenga en cuenta que si la tabla de taxonomía ya existe, se eliminará. La información de taxonomía se almacena con el nodo de generación de modelos y se puede editar mediante los botones Editar categorías y Editar taxonomía.

### Regresión de ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imregnode.

Propiedades de db2imregnode	Valores	Descripción de la propiedad
target	campo	
regression_method	transform linear polynomial rbf	Consulte las propiedades de la tabla siguiente que solo se aplican si regression_method está definido como rbf.
perform_test_run	campo	
limit_rsquared_value	marca	
max_rsquared_value	número	Valor entre 0,0 y 1,0.
use_execution_time_limit	marca	
execution_time_limit_mins	entero	Valor mayor que 0.
use_max_degree_polynomial	marca	
max_degree_polynomial	entero	
use_intercept	marca	
use_auto_feature_selection_method	marca	
auto_feature_selection_method	normal adjusted	
use_min_significance_level	marca	
min_significance_level	número	
use_min_significance_level	marca	

Las siguientes propiedades solo se aplican si `regression_method` está definido como `rbf`.

Propiedades de <code>db2imregnode</code>	Valores	Descripción de la propiedad
<code>use_output_sample_size</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>output_sample_size</code>	<i>entero</i>	El valor por defecto es 2. El mínimo es 1.
<code>use_input_sample_size</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>input_sample_size</code>	<i>entero</i>	El valor por defecto es 2. El mínimo es 1.
<code>use_max_num_centers</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>max_num_centers</code>	<i>entero</i>	El valor por defecto es 20. El mínimo es 1.
<code>use_min_region_size</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>min_region_size</code>	<i>entero</i>	El valor por defecto es 15. El mínimo es 1.
<code>use_max_data_passes</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>max_data_passes</code>	<i>entero</i>	El valor por defecto es 5. El mínimo es 2.
<code>use_min_data_passes</code>	<i>marca</i>	Si es verdadero, el valor se ajusta automáticamente como el valor por defecto.
<code>min_data_passes</code>	<i>entero</i>	El valor por defecto es 5. El mínimo es 2.

### ***Conglomerados de ISW***

Las siguientes propiedades están disponibles para los nodos del tipo `db2imclusternode`.

Propiedades de <code>db2imclusternode</code>	Valores	Descripción de la propiedad
<code>cluster_method</code>	demographic kohonen birch	
<code>kohonen_num_rows</code>	<i>entero</i>	
<code>kohonen_num_columns</code>	<i>entero</i>	
<code>kohonen_passes</code>	<i>entero</i>	
<code>use_num_passes_limit</code>	<i>marca</i>	
<code>use_num_clusters_limit</code>	<i>marca</i>	
<code>max_num_clusters</code>	<i>entero</i>	Valor mayor que 1.
<code>birch_dist_measure</code>	log_likelihood euclidean	El valor por defecto es <code>log_likelihood</code> .
<code>birch_num_cfleaves</code>	<i>entero</i>	El valor por defecto es 1000.
<code>birch_num_refine_passes</code>	<i>entero</i>	El valor por defecto es 3; el mínimo es 1.
<code>use_execution_time_limit</code>	<i>marca</i>	
<code>execution_time_limit_mins</code>	<i>entero</i>	Valor mayor que 0.
<code>min_data_percentage</code>	<i>número</i>	0–100%

Propiedades de db2imclusternode	Valores	Descripción de la propiedad
use_similarity_threshold	marca	
similarity_threshold	número	Valor entre 0,0 y 1,0.

### **bayesiano Naive de ISW**

Las siguientes propiedades están disponibles para los nodos del tipo db2imnbsnode.

Propiedades de db2imnbsnode	Valores	Descripción de la propiedad
perform_test_run	marca	
probability_threshold	número	El valor por defecto es 0,001. El valor mínimo es 0; el valor máximo es 1.000.
use_costs	marca	
costs	estructurado	Propiedad estructurada de la siguiente forma: {{drugA drugB 1.5} {drugA drugC 2.1}}, donde los argumentos en {} son los costes pronosticados actuales.

### **Regresión logística de ISW**

Las siguientes propiedades están disponibles para los nodos del tipo db2imlognode.

Propiedades de db2imlognode	Valores	Descripción de la propiedad
perform_test_run	marca	
use_costs	marca	
costs	estructurado	Propiedad estructurada de la siguiente forma: {{drugA drugB 1.5} {drugA drugC 2.1}}, donde los argumentos en {} son los costes pronosticados actuales.

### **Serie temporal de ISW**

*Nota:* el parámetro de campos de entrada no se utiliza para este nodo. Si se encuentra el parámetro de campos de entrada en el proceso, se muestra una advertencia para indicar que el nodo tiene *tiempo* y *objetivos* como campos entrantes, pero no de entrada.

Las siguientes propiedades están disponibles para los nodos del tipo db2imtimeseriesnode.

Propiedades de db2imtimeseriesnode	Valores	Descripción de la propiedad
time	campo	Se permite entero, hora o fecha.
targets	lista de campos	
forecasting_algorithm	arima exponen- tial_smoothing sea- sonal_trend_de- composition	

Propiedades de db2imtimeseriesnode	Valores	Descripción de la propiedad
forecasting_end_time	auto integer date time	
use_records_all	booleano	Si es falso, deben establecerse use_records_start y use_records_end.
use_records_start	entero / hora / fecha	Depende del tipo de campo de tiempo.
use_records_end	entero / hora / fecha	Depende del tipo de campo de tiempo.
interpolation_method	none linear exponen- tial_splines cubic_splines	

### **Propiedades de nugget de modelo de IBM DB2**

Las siguientes propiedades son para los nugget de modelo creados mediante los modelos de IBM DB2 ISW.

#### **Árbol de decisión de ISW**

No hay propiedades específicas para los nodos del tipo applydb2imtreenode.

#### **Asociación de ISW**

Este nugget de modelo no se puede aplicar en el procesamiento.

#### **Secuencia de ISW**

Este nugget de modelo no se puede aplicar en el procesamiento.

#### **Regresión de ISW**

No hay propiedades específicas para los nodos del tipo applydb2imregnode.

#### **Conglomerados de ISW**

No hay propiedades específicas para los nodos del tipo applydb2imclusternode.

#### **bayesiano Naive de ISW**

No hay propiedades específicas para los nodos del tipo applydb2imnbnode.

**Regresión logística de ISW**

No hay propiedades específicas para los nodos del tipo applydb2imlognode.

**Serie temporal de ISW**

Este nugget de modelo no se puede aplicar en el procesamiento.

**Propiedades de nodos de modelado de IBM Netezza Analytics****Propiedades de nodos de modelado de Netezza**

Las siguientes propiedades son comunes a los nodos de modelado de la base de datos de IBM Netezza.

Propiedades comunes de nodos de Netezza	Valores	Descripción de la propiedad
custom_fields	<i>marca</i>	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
inputs	<i>[campo1 ... campoN]</i>	Campos de entrada o predictor utilizados por el modelo.
target	<i>campo</i>	Campo de destino (continuo o categórico).
record_id	<i>campo</i>	El campo que se debe utilizar como identificador de registros exclusivo.
use_upstream_connection	<i>marca</i>	Si es verdadero (por defecto), los detalles de conexión especificados en un nodo anterior. No se usa si <i>move_data_to_connection</i> está especificado.
move_data_connection	<i>marca</i>	Si es verdadero, transfiere los datos a la base de datos especificada mediante <i>connection</i> . No se usa si <i>use_upstream_connection</i> está especificado.
connection	<i>estructurado</i>	La cadena de conexión para la base de datos de Netezza donde se almacena el modelo. Propiedad estructurada de la siguiente forma: ['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' {true false}] donde: <dsn> es el nombre de la fuente de datos <username> y <psw> son el nombre de usuario y la contraseña de la base de datos <catname> es el nombre de catálogo <conn_attribs> son los atributos de conexión true   false indica si se necesita la contraseña.
table_name	<i>cadena</i>	Nombre de la base de datos donde se debe almacenar el modelo.
use_model_name	<i>marca</i>	Si es verdadero, utiliza el nombre que especifica <i>model_name</i> como el nombre del modelo, de lo contrario el sistema crea el nombre del modelo.

Propiedades comunes de nodos de Netezza	Valores	Descripción de la propiedad
model_name	<i>cadena</i>	Nombre personalizado para nuevo modelo.
include_input_fields	<i>marca</i>	Si es verdadero, transmite todos los campos de entrada siguientes, de lo contrario solo transmite record_id y los campos generados por el modelo.

### Árbol de decisión de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezzadectreenode.

Propiedades de netezzadectreenode	Valores	Descripción de la propiedad
impurity_measure	Entropy Gini	La medición de impureza, utilizada para valorar la mejor ubicación para dividir el árbol.
max_tree_depth	<i>entero</i>	Número máximo de niveles que puede alcanzar el crecimiento de árbol. El valor predeterminado es 62 (el máximo posible).
min_improvement_splits	<i>número</i>	Mejoras mínimas que se pueden realizar en la impureza de la división. El valor por defecto es 0.01.
min_instances_split	<i>entero</i>	Número mínimo de registros por dividir antes de realizar la división. El valor predeterminado es 2 (el mínimo posible).
weights	<i>estructurado</i>	Ponderaciones relativas para clases. Propiedad estructurada de la siguiente forma: set :netezza_dectree.weights = [{drugA 0.3}{drugB 0.6}] El valor predeterminado es la ponderación de 1 para todas las clases.
pruning_measure	Acc wAcc	El valor predeterminado es Acc (precisión). wAcc alternativo (precisión ponderada) toma en cuenta las ponderaciones de clase mientras se aplica la poda.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	El valor predeterminado es utilizar allTrainingData para calcular la precisión del modelo. Utilice partitionTrainingData para especificar un porcentaje de datos de prueba por utilizar, o useOtherTable para utilizar un conjunto de datos de prueba desde una tabla específica de la base de datos.
perc_training_data	<i>número</i>	Si prune_tree_options está definido como partitionTrainingData, especifica el porcentaje de datos que se utilizará para entrenamiento.
prune_seed	<i>entero</i>	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis si prune_tree_options está definido como partitionTrainingData; el valor predeterminado es 1.
pruning_table	<i>cadena</i>	Nombre de tabla de un conjunto de datos de poda separado para estimar la precisión del modelo.
compute_probabilities	<i>marca</i>	Si es verdadero, produce un campo de nivel de confianza (probabilidad) y el campo de pronóstico.



**K-medias de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzakmeansnode`.

Propiedades de <code>netezzakmeansnode</code>	Valores	Descripción de la propiedad
<code>distance_measure</code>	Euclidean Manhattan Canberra maximum	Método que se debe utilizar para medir la distancia entre puntos de datos.
<code>num_clusters</code>	<i>entero</i>	Número de conglomerados que se deben crear; el valor predeterminado es 3.
<code>max_iterations</code>	<i>entero</i>	Número de iteraciones de algoritmos después de los cuáles se debe detener la prueba del modelo; el valor predeterminado es 5.
<code>rand_seed</code>	<i>entero</i>	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis; el valor predeterminado es 12345.

**Red bayesiana de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezabayesnode`.

Propiedades de <code>netezabayesnode</code>	Valores	Descripción de la propiedad
<code>base_index</code>	<i>entero</i>	Identificador numérico asignado al primer campo de entrada de gestión interna; el valor por defecto es 777.
<code>sample_size</code>	<i>entero</i>	Tamaño de la muestra que se tomará si el número de atributos es muy grande; el valor por defecto es 10.000.
<code>display_additional_information</code>	<i>marca</i>	Si es verdadero, muestra información adicional sobre el progreso en un cuadro de diálogo de mensaje.
<code>type_of_prediction</code>	best neighbors nn-neighbors	Tipo de algoritmo de pronóstico que se utilizará: el mejor (vecino más correlacionado), vecinos (pronóstico ponderado de vecinos), o vecinos NN (vecinos no nulos).

**Bayesiano Naive de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezanaivebayesnode`.

Propiedades de <code>netezanaivebayesnode</code>	Valores	Descripción de la propiedad
<code>compute_probabilities</code>	<i>marca</i>	Si es verdadero, produce un campo de nivel de confianza (probabilidad) y el campo de pronóstico.
<code>use_m_estimation</code>	<i>marca</i>	Si es verdadero, utiliza la técnica m-estimation para evitar probabilidades de cero durante el cálculo.

**KNN de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzaknnnode`.

Propiedades de <code>netezzaknnnode</code>	Valores	Descripción de la propiedad
<code>weights</code>	<i>estructurado</i>	Propiedad estructurada que se utiliza para asignar ponderaciones a clases individuales. Ejemplo: <code>set :netezzaknnnode.weights = [{drugA 0.3}{drugB 0.6}]</code>
<code>distance_measure</code>	Euclidean Manhattan Canberra Maximum	Método que se utiliza para medir la distancia entre puntos de datos.
<code>num_nearest_neighbors</code>	<i>entero</i>	Número de vecinos más próximos de un caso concreto; el valor por defecto es 3.
<code>standardize_measurements</code>	<i>marca</i>	Si es verdadero, estandariza las mediciones de campos de entrada continuos antes de calcular los valores de distancia.
<code>use_coresets</code>	<i>marca</i>	Si es verdadero, utiliza el muestreo del conjunto principal para acelerar el cálculo de conjuntos de datos grandes.

**Conglomerado divisivo de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezadivclusternode`.

Propiedades de <code>netezadivclusternode</code>	Valores	Descripción de la propiedad
<code>distance_measure</code>	Euclidean Manhattan Canberra Maximum	Método que se utiliza para medir la distancia entre puntos de datos.
<code>max_iterations</code>	<i>entero</i>	Número máximo de iteraciones de algoritmo que se ejecutarán antes de detener el entrenamiento del modelo; el valor por defecto es 5.
<code>max_tree_depth</code>	<i>entero</i>	El número máximo de niveles en los que se puede subdividir el conjunto de datos; el valor por defecto es 3.
<code>rand_seed</code>	<i>entero</i>	Semilla aleatoria, se utiliza para replicar los análisis; el valor por defecto es 12345.
<code>min_instances_split</code>	<i>entero</i>	El número mínimo de registros que se pueden dividir, el valor por defecto es 5.
<code>level</code>	<i>entero</i>	El nivel de jerarquía en el que se guardan los registros; el valor por defecto es -1.

**PCA de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzapcanode`.

Propiedades de <code>netezzapcanode</code>	Valores	Descripción de la propiedad
<code>center_data</code>	<i>marca</i>	Si es verdadero (opción por defecto), ejecuta el centrado de datos (también conocido como “sustracción de media”) antes del análisis.
<code>perform_data_scaling</code>	<i>marca</i>	Si es verdadero, ejecuta la adaptación de los datos antes del análisis. De esta forma el análisis será menos arbitrario si las diferentes variables se miden en unidades diferentes.
<code>force_eigensolve</code>	<i>marca</i>	Si es verdadero, utiliza un método menos preciso, pero más rápido para encontrar componentes principales.
<code>pc_number</code>	<i>entero</i>	El número principal de componentes al que se reducirá el conjunto de datos; el valor por defecto es 1.

**Árbol de regresión de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzaregtreenode`.

Propiedades de <code>netezzaregtreenode</code>	Valores	Descripción de la propiedad
<code>max_tree_depth</code>	<i>entero</i>	Número máximo de niveles que puede crecer el árbol por debajo del nodo raíz; el valor por defecto es 10.
<code>split_evaluation_measure</code>	Variance	Medición de clase de impureza, se utiliza para evaluar la mejor ubicación para dividir el árbol; el valor por defecto (y la opción actual) es Variance.
<code>min_improvement_splits</code>	<i>número</i>	Cantidad mínima para reducir la impureza antes de que se cree la nueva división en el árbol.
<code>min_instances_split</code>	<i>entero</i>	El número mínimo de registros que se pueden dividir.
<code>pruning_measure</code>	mse r2 pearson spearman	Método que se utilizará para la poda.
<code>prune_tree_options</code>	allTrainingData partitionTrain- ingData useOtherTable	El valor predeterminado es utilizar <code>allTrainingData</code> para calcular la precisión del modelo. Utilice <code>partitionTrainingData</code> para especificar un porcentaje de datos de prueba por utilizar, o <code>useOtherTable</code> para utilizar un conjunto de datos de prueba desde una tabla específica de la base de datos.
<code>perc_training_data</code>	<i>número</i>	Si <code>prune_tree_options</code> está definido como <code>PercTrainingData</code> , especifica el porcentaje de datos que se utilizará para entrenamiento.
<code>prune_seed</code>	<i>entero</i>	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis si <code>prune_tree_options</code> está definido como <code>PercTrainingData</code> ; el valor predeterminado es 1.

Propiedades de <b>netezzaregtreenode</b>	Valores	Descripción de la propiedad
pruning_table	<i>cadena</i>	Nombre de tabla de un conjunto de datos de poda separado para estimar la precisión del modelo.
compute_probabilities	<i>marca</i>	Si es verdadero, especifica que las varianzas de las clases asignadas se deben incluir en el resultado.

### **Regresión lineal de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzalineressionnode`.

Propiedades de <b>netezzalineressionnode</b>	Valores	Descripción de la propiedad
use_svd	<i>marca</i>	Si es verdadero, utiliza la matriz de descomposición de valores singulares en lugar de la matriz original, para mayor velocidad y precisión numérica.
include_intercept	<i>marca</i>	Si es verdadero (valor por defecto), aumenta la precisión global de la solución.
calculate_model_diagnostics	<i>marca</i>	Si es verdadero, calcula el diagnóstico del modelo.

### **Series temporales de Netezza**

Las siguientes propiedades están disponibles para los nodos del tipo `netezzatimeseriesnode`.

Propiedades de <b>netezzatimeseriesnode</b>	Valores	Descripción de la propiedad
time_points	<i>campo</i>	El campo de entrada que contiene los valores de fecha u hora de la serie temporal.
time_series_ids	<i>campo</i>	El campo de entrada que contiene diversos ID de series temporales; utilice esta opción si la entrada contiene más de una serie temporal.
model_table	<i>campo</i>	Nombre de la tabla de base de datos en la que se guardará el modelo de series temporales de Netezza.
description_table	<i>campo</i>	Nombre de la tabla de entrada que contiene los nombres y las descripciones de las series temporales.
seasonal_adjustment_table	<i>campo</i>	Nombre de la tabla de salida en la que se guardarán los valores ajustados calculados por los algoritmos de suavizado exponencial o de descomposición de tendencia estacional.
algorithm_name	SpectralAnalysis o spectral ExponentialSmoothing o esmoothing ARIMA SeasonalTrend- Decomposition o std	Algoritmo que hay que utilizar para el modelado de series temporales.
trend_name	<i>cadena</i>	Tipo de tendencia del suavizado exponencial.
seasonality_type	<i>cadena</i>	Tipo de estacionalidad del suavizado exponencial.

Propiedades de netezzatimeseriesnode	Valores	Descripción de la propiedad
interpolation_method	linear cubicspline exponential-spline	Método de interpolación que hay que utilizar.
earliest_time	campo	Hora de inicio, si se utiliza una parte de la serie temporal.
latest_time	campo	Hora de finalización, si se utiliza una parte de la serie temporal.
p	entero	ARIMA: grados no estacionales de autocorrelación.
q	entero	ARIMA: valor de derivación no estacional.
d	entero	ARIMA: número no estacional de órdenes de media móvil presentes en el modelo.
sp	entero	ARIMA: grados estacionales de autocorrelación.
sq	entero	ARIMA: valor de derivación estacional.
sd	entero	ARIMA: número estacional de órdenes de media móvil presentes en el modelo.
period	entero	Longitud del ciclo estacional, especificado en conjunción con units_period. No aplicable para análisis espectrales.
units_period	Milliseconds Seconds Minutes Hours Days Weeks Quarters Years	Unidades en que se expresa period. Por ejemplo, para una serie temporal semanal utilice 1 para period y Weeks para units_period.
include_history	marca	Indica si se deben incluir los valores históricos en los resultados.
include_interpolated_values	marca	Indica si se deben incluir los valores interpolados en los resultados. No aplicable si include_history es false.

### Lineal generalizado de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezzaglmnode.

Propiedades de netezzaglmnode	Valores	Descripción de la propiedad
response_variable_distribution	bernoulli gaussian poisson binomial negativebinomial wald gamma	Tipo de distribución; el valor por defecto es bernoulli.
distribution_parameter	número	Valor del parámetro de distribución que hay que utilizar. Aplicable únicamente si distribution es Negativebinomial.

Propiedades de netezzaglmnode	Valores	Descripción de la propiedad
trials	<i>entero</i>	Aplicable únicamente si <i>distribution</i> es Binomial. Cuando la respuesta objetivo es un número de eventos que tienen lugar en un conjunto de ensayos, el campo <i>target</i> contiene el número de eventos, y el campo <i>trials</i> contiene el número de ensayos.
model_table	<i>campo</i>	Nombre de la tabla de base de datos en la que se guardará el modelo lineal generalizado de Netezza.
max_iterations	<i>entero</i>	Número máximo de iteraciones que debe ejecutar el algoritmo; el valor predeterminado es 20.
max_error	<i>número</i>	Valor del error máximo (en notación científica) en el que el algoritmo debería dejar de buscar el modelo de mejor ajuste. El valor predeterminado es -3, lo que significa 1E-3 o 0,001.
error_threshold	<i>número</i>	El valor (en notación científica) por debajo del que los errores se tratan como si su valor fuera cero. El valor predeterminado es -7, lo que significa que los valores de error por debajo de 1E-7 (o 0,0000001) se cuentan como insignificantes.
link_function	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	Función de enlace a utilizar; el valor por defecto es <i>logit</i> .
link_function_parameter	<i>número</i>	Valor del parámetro de función de enlace que hay que utilizar. Aplicable únicamente si <i>link_function</i> es <i>power</i> o <i>oddspower</i> .
intercept	<i>marca</i>	Si es <i>true</i> , incluye la intercepción en el modelo.

### ***Propiedades de nugget de modelo de Netezza***

Las siguientes propiedades son comunes a los nuggets del modelo de la base de datos de Netezza.

Propiedades comunes de nugget de modelo de Netezza	Valores	Descripción de la propiedad
connection	<i>cadena</i>	La cadena de conexión para la base de datos de Netezza donde se almacena el modelo.
table_name	<i>cadena</i>	Nombre de la base de datos donde se almacenará el modelo.

Otras las propiedades del nugget de modelo son las mismas que las del nodo de modelado correspondiente.

Los nombres del proceso de los nuggets de modelo son los siguientes.

<b>Nugget de modelo</b>	<b>Nombre de proceso</b>
Árbol de decisiones	applynetezadectreenode
K-medias	applynetezzakmeansnode
Red bayesiana	applynetezزابayesnode
bayesiano Naive	applynetezzanaivebayesnode
KNN	applynetezzaknnnode
Conglomerado divisivo	applynetezadivclusternode
PCA	applynetezzapcanode
Árbol de regresión	applynetezzaregtreenode
Regresión lineal	applynetezalineregressionnode
Serie temporal	applynetezzatimeseriesnode
Lineal generalizado	applynetezzaglmnode

# Propiedades de los nodos de resultados

Las propiedades de nodos de resultados se diferencian un poco de las de otros tipos de nodos. En lugar de hacer referencia a una opción determinada de nodo, las propiedades de nodos de resultados almacenan una referencia en el objeto de resultado. Esto resulta útil al tomar un valor de una tabla y establecerlo como un parámetro de ruta.

Esta sección describe las propiedades de procesos disponibles para los nodos de resultados.

## Propiedades de analysisnode



El nodo Análisis evalúa la capacidad de los modelos predictivos para generar pronósticos precisos. Los nodos Análisis realizan varias comparaciones entre los valores pronosticados y los valores reales para uno o más nugget de modelo. También pueden comparar modelos predictivos entre sí. [Si desea obtener más información, consulte el tema Nodo Análisis en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create analysisnode
# Pestaña "Análisis"
set :analysisnode.coincidence = True
set :analysisnode.performance = True
set :analysisnode.confidence = True
set :analysisnode.threshold = 75
set :analysisnode.improve_accuracy = 3
set :analysisnode.inc_user_measure = True
#"Definir medida del usuario..."
set :analysisnode.user_if = "@TARGET = @PREDICTED"
set :analysisnode.user_then = "101"
set :analysisnode.user_else = "1"
set :analysisnode.user_compute = [Suma Media]
set :analysisnode.by_fields = ['Medicamento']
#Pestaña "Resultados"
set :analysisnode.output_format = HTML
set :analysisnode.full_filename = "C:/output/analysis_out.html"
```

propiedades de analysisnode	Tipo de datos	Descripción de la propiedad
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.



propiedades de analysisnode	Tipo de datos	Descripción de la propiedad
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_format	Text (.txt) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
by_fields	[campo campo campo]	
full_filename	cadena	Nombre del archivo de resultados, si se trata de resultados HTML, de datos o de disco.
coincidence	marca	
performance	marca	
confidence	marca	
threshold	número	
improve_accuracy	número	
inc_user_measure	marca	
user_if	expresión	
user_then	expresión	
user_else	expresión	
user_compute	[Mean Sum Min Max SDev]	

## Propiedades de dataauditnode



El nodo Auditar datos permite echar un primer vistazo exhaustivo a los datos, incluyendo estadísticos de resumen, histogramas y distribución para cada campo, así como información sobre valores atípicos, valores perdidos y extremos. Los resultados se muestran en una matriz fácil de leer que se puede ordenar y utilizar para generar nodos de preparación de datos y gráficos de tamaño completo. [Si desea obtener más información, consulte el tema Nodo Auditar datos en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create dataauditnode
connect :variablefilenode to :dataauditnode
set :dataauditnode.custom_fields = True
set :dataauditnode.fields = [Edad Na K]
set :dataauditnode.display_graphs = True
set :dataauditnode.basic_stats = True
set :dataauditnode.advanced_stats = True
set :dataauditnode.median_stats = False
set :dataauditnode.calculate = [Desglosar recuento]
set :dataauditnode.outlier_detection_method = std
set :dataauditnode.outlier_detection_std_outlier = 1.0
set :dataauditnode.outlier_detection_std_extreme = 3.0
```

```
set :dataauditnode.output_mode = Screen
```

Propiedades de dataauditnode	Tipo de datos	Descripción de la propiedad
custom_fields	marca	
fields	[campo1 ... campoN]	
overlay	campo	
display_graphs	marca	Se utiliza para activar o desactivar la representación de gráficos en la matriz de resultados.
basic_stats	marca	
advanced_stats	marca	
median_stats	marca	
calculate	Count Breakdown	Se utiliza para calcular valores perdidos. Seleccione uno de los métodos de cálculo, ambos o ninguno.
outlier_detection_method	std iqr	Se utiliza para especificar el método de detección de valores atípicos y extremos.
outlier_detection_std_outlier	número	Si outlier_detection_method es std, especifica el número que se utilizará para definir los valores atípicos.
outlier_detection_std_extreme	número	Si outlier_detection_method es std, especifica el número que se utilizará para definir los valores extremos.
outlier_detection_iqr_outlier	número	Si outlier_detection_method es iqr, especifica el número que se utilizará para definir los valores atípicos.
outlier_detection_iqr_extreme	número	Si outlier_detection_method es iqr, especifica el número que se utilizará para definir los valores extremos.
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
paginate_output	marca	Si output_format es HTML, provocará que los resultados se separen por páginas.

Propiedades de dataauditnode	Tipo de datos	Descripción de la propiedad
lines_per_page	número	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	cadena	

## Propiedades de matrixnode



El nodo Matriz crea una tabla que muestra las relaciones entre campos. Se suele utilizar normalmente para mostrar las relaciones entre dos campos simbólicos, pero también puede mostrar relaciones entre campos de marcas o entre campos numéricos. [Si desea obtener más información, consulte el tema Nodo Matriz en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create matrixnode
# Pestaña "Configuración"
set :matrixnode.fields = Numerics
set :matrixnode.row = 'K'
set :matrixnode.column = 'Na'
set :matrixnode.cell_contents = Function
set :matrixnode.function_field = 'Age'
set :matrixnode.function = Sum
# Pestaña "Aspecto"
set :matrixnode.sort_mode = Ascending
set :matrixnode.highlight_top = 1
set :matrixnode.highlight_bottom = 5
set :matrixnode.display = [Recuentos de residuos esperados]
set :matrixnode.include_totals = True
# Pestaña "Resultados"
set :matrixnode.full_filename = "C:/output/matrix_output.html"
set :matrixnode.output_format = HTML
set :matrixnode.paginate_output = true
set :matrixnode.lines_per_page = 50
```

Propiedades de matrixnode	Tipo de datos	Descripción de la propiedad
fields	Selected Flags Numerics	
row	campo	
column	campo	
include_missing_values	marca	Determina si los valores perdidos por el sistema (nulos) o no especificados por el usuario (vacíos) se incluyen en los resultados de fila y columna.
cell_contents	CrossTabs Function	
function_field	cadena	

Propiedades de matrixnode	Tipo de datos	Descripción de la propiedad
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	<i>número</i>	Si no es cero, es verdadero.
highlight_bottom	<i>número</i>	Si no es cero, es verdadero.
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	<i>marca</i>	
use_output_name	<i>marca</i>	Especifica si se utiliza un nombre de resultado personalizado.
output_name	<i>cadena</i>	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado. Tanto el formato Formatted como el formato Delimited pueden tomar el modificador transposed, mediante el cual se transponen filas y columnas en la tabla. Por ejemplo: NODE.output_format=transposed Delimited
paginate_output	<i>marca</i>	Si output_format es HTML, provocará que los resultados se separen por páginas.
lines_per_page	<i>número</i>	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	<i>cadena</i>	

## Propiedades de meansnode



El nodo Medias compara las medias de grupos independientes o de pares de campos relacionados para probar si existen diferencias significativas. Por ejemplo, puede comparar los ingresos medios antes y después de poner en marcha una promoción o comparar los ingresos de los clientes que no recibieron esa promoción con los que sí lo hicieron. [Si desea obtener más información, consulte el tema Nodo Medias en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create meansnode
set :meansnode.means_mode = BetweenFields
set :meansnode.paired_fields = [{'OPEN_BAL' 'CURR_BAL'}]
set :meansnode.label_correlations = true
set :meansnode.output_view = Advanced
set :meansnode.output_mode = File
set :meansnode.output_format = HTML
set :meansnode.full_filename = "C:/output/means_output.html"
```

Propiedades de meansnode	Tipo de datos	Descripción de la propiedad
means_mode	BetweenGroups BetweenFields	Especifica el tipo de estadístico de las medias que se va a ejecutar en los datos.
test_fields	[field1 ... fieldn]	Especifica el campo de prueba si means_mode se establece en BetweenGroups.
grouping_field	campo	Especifica el campo de agrupación.
paired_fields	{{field1 field2} {field3 field4} ...}	Especifica los pares de campos que se usan si means_mode se establece en BetweenFields.
label_correlations	marca	Determina si las etiquetas de correlación se muestran en el resultado. Esta configuración se aplica únicamente si means_mode se establece en BetweenFields.
correlation_mode	Probability Absolute	Determina si las correlaciones deben etiquetarse según la probabilidad o según el valor absoluto.
weak_label	cadena	
medium_label	cadena	
strong_label	cadena	
weak_below_probability	número	Si correlation_mode se establece en Probability, especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.

Propiedades de meansnode	Tipo de datos	Descripción de la propiedad
strong_above_probability	número	Valor de corte para correlaciones fuertes.
weak_below_absolute	número	Si correlation_mode se establece en Absolute, especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_absolute	número	Valor de corte para correlaciones fuertes.
unimportant_label	cadena	
marginal_label	cadena	
important_label	cadena	
unimportant_below	número	Valor de corte para una importancia del campo baja. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
important_above	número	
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Nombre que se va a usar.
output_mode	Screen File	Especifica la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Especifica el tipo de resultado.
full_filename	cadena	
output_view	Simple Advanced	Determina si el resultado muestra la vista simple o la avanzada.

## Propiedades de reportnode



El nodo Informe crea informes con formato que contienen texto fijo, así como datos y otras expresiones derivadas de los datos. Puede especificar el formato del informe utilizando plantillas de texto para definir el texto fijo y las construcciones de resultados de datos. Puede proporcionar formato de texto personalizado utilizando etiquetas HTML de la plantilla y configurando opciones en la ficha Resultado. Puede incluir valores de datos y otros resultados condicionales mediante el uso de expresiones CLEM en la plantilla. [Si desea obtener más información, consulte el tema Nodo Informe en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```

create reportnode
set :reportnode.output_format = HTML
set :reportnode.full_filename = "C:/report_output.html"
set :reportnode.lines_per_page = 50
set :reportnode.title = "Report node created by a script"
set :reportnode.highlights = False

```

Propiedades de reportnode	Tipo de datos	Descripción de la propiedad
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	HTML (.html) Text (.txt) Output (.cou)	Se utiliza para especificar el tipo de resultado.
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
text	cadena	
full_filename	cadena	
highlights	marca	
title	cadena	
lines_per_page	número	

**Propiedades de setglobalsnode**

El nodo Val. globales explora los datos y calcula los valores de resumen que se pueden utilizar en expresiones CLEM. Por ejemplo, puede utilizar este nodo para calcular estadísticos para un campo denominado *edad* y, a continuación, utilizar la media global de *edad* en expresiones CLEM insertando la función @GLOBAL\_MEAN(*edad*). Si desea obtener más información, consulte el tema [Nodo Val. globales](#) en el capítulo 6 en *Nodos de origen, proceso y resultado de IBM SPSS Modeler 15*.

**Ejemplo**

```

create setglobalsnode
connect :typenode to :setglobalsnode
set :setglobalsnode.globals.Na = [Max Sum Mean]
set :setglobalsnode.globals.K = [Suma media máxima]
set :setglobalsnode.globals.Age = [Suma Media Mín Máx D.T.]
set :setglobalsnode.clear_first = False

```

```
set :setglobalsnode.show_preview = True
```

Propiedades de setglobalsnode	Tipo de datos	Descripción de la propiedad
globals	[Sum Mean Min Max SDev]	Propiedad estructurada en la que los campos que se van a establecer deben hacer referencia a la sintaxis siguiente: set :setglobalsnode.globals.Age = [Sum Mean Min Max SDev]
clear_first	marca	
show_preview	marca	

## Propiedades de statisticsnode



El nodo Estadísticos ofrece información básica de resumen acerca de los campos numéricos. Calcula estadísticos de resumen para campos individuales y correlaciones entre campos. [Si desea obtener más información, consulte el tema Nodo Estadísticos en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create statisticsnode
# Pestaña "Configuración"
set :statisticsnode.examine = ['Edad' 'PS' 'Droga']
set :statisticsnode.statistics = [Suma Media D.T.]
set :statisticsnode.correlate = ['PS' 'Medicamento']
# Sección "Etiquetas de correlación..."
set :statisticsnode.label_correlations = True
set :statisticsnode.weak_below_absolute = 0.25
set :statisticsnode.weak_label = "cuartil inferior"
set :statisticsnode.strong_above_absolute = 0.75
set :statisticsnode.medium_label = "cuartiles intermedios"
set :statisticsnode.strong_label = "cuartil superior"
# Pestaña "Resultado"
set :statisticsnode.full_filename = "c:/output/statistics_output.html"
set :statisticsnode.output_format = HTML
```

Propiedades de statisticsnode	Tipo de datos	Descripción de la propiedad
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.



Propiedades de <b>statisticsnode</b>	Tipo de datos	Descripción de la propiedad
output_format	Text (.txt) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
full_filename	cadena	
examine	[campo campo campo]	
correlate	[campo campo campo]	
statistics	[Count Mean Sum Min Max Range Variance SDev SErr Median Mode]	
correlation_mode	Probability Absolute	Determina si las correlaciones deben etiquetarse según la probabilidad o según el valor absoluto.
label_correlations	marca	
weak_label	cadena	
medium_label	cadena	
strong_label	cadena	
weak_below_probability	número	Si <b>correlation_mode</b> se establece en <b>Probability</b> , especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_probability	número	Valor de corte para correlaciones fuertes.
weak_below_absolute	número	Si <b>correlation_mode</b> se establece en <b>Absolute</b> , especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_absolute	número	Valor de corte para correlaciones fuertes.

## Propiedades de **statisticsoutputnode**



El nodo Resultados de Statistics le permite llamar a un procedimiento de IBM® SPSS® Statistics para analizar los datos de IBM® SPSS® Modeler. A wide variety of SPSS Statistics analytical procedures is available. This node requires a licensed copy of SPSS Statistics. [Si desea obtener más información, consulte el tema Nodo Resultados de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

Las propiedades de este nodo están descritas en [Propiedades de statisticsoutputnode](#) el p. 317.

## Propiedades de tablenode



El nodo Tabla muestra los datos en formato de tabla, que también se puede escribir en un archivo. Esto es útil en cualquier momento en que necesite inspeccionar sus valores de datos o exportarlos en un formato fácilmente legible. [Si desea obtener más información, consulte el tema Nodo Tabla en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create tablenode
set :tablenode.highlight_expr = "Edad > 30"
set :tablenode.output_format = HTML
set :tablenode.transpose_data = true
set :tablenode.full_filename = "C:/output/table_output.htm"
set :tablenode.paginate_output = true
set :tablenode.lines_per_page = 50
```

Propiedades de tablenode	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	Nombre del archivo de resultados, si se trata de resultados HTML, de datos o de disco.
use_output_name	<i>marca</i>	Especifica si se utiliza un nombre de resultado personalizado.
output_name	<i>cadena</i>	Si <i>use_output_name</i> es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
transpose_data	<i>marca</i>	Transpone los datos antes de exportarlos de manera que las filas representan campos y las columnas, registros.
paginate_output	<i>marca</i>	Si <i>output_format</i> es HTML, provocará que los resultados se separen por páginas.
lines_per_page	<i>número</i>	Si se usa con <i>paginate_output</i> , especifica las líneas por página del resultado.
highlight_expr	<i>cadena</i>	
output	<i>cadena</i>	Propiedad de sólo lectura que mantiene una referencia en la última tabla creada por el nodo.
value_labels	[{Valor CadenaEtiquetas} {Valor CadenaEtiquetas} ...]	Se utiliza para especificar etiquetas para los pares de valores. Por ejemplo: set :typenode.value_labels. 'Drug'={{drugA label1} {drugB label2}}

Propiedades de tablenode	Tipo de datos	Descripción de la propiedad
display_places	entero	Establece el número de cifras decimales para el campo cuando se muestra (sólo se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.display_places.FIELDNAME
export_places	entero	Establece el número de cifras decimales para el campo cuando se exporta (sólo se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor por defecto de la ruta. Formato de uso: NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD COMMA	Establece el separador decimal para el campo (sólo se aplica a campos con almacenamiento REAL). Formato de uso: NODE.decimal_separator.FIELDNAME
date_format	"DDMMAA" "MMDDAA" "AAMMDD" "AAAAMMDD" "AAAADDD" DÍA MES "DD-MM-AA" "DD-MM-AAAA" "MM-DD-AA" "MM-DD-AAAA" "DD-MES-AA" "DD-MES-AAAA" "AAAA-MM-DD" "DD.MM.AA" "DD.MM.AAAA" "MM.DD.AA" "MM.DD.AAAA" "DD.MES.AA" "DD.MES.AAAA" "DD/MM/AA" "DD/MM/AAAA" "MM/DD/AA" "MM/DD/AAAA" "DD/MES/AA" "DD/MES/AAAA" MES AAAA t T AAAA ss SE AAAA	Establece el formato de fecha para el campo (sólo se aplica a campos con almacenamiento DATE o TIMESTAMP). Formato de uso: NODE.date_format.FIELDNAME Por ejemplo: set :tablenode.date_format. 'LaunchDate' = "DDMMYY"

Propiedades de <b>tablnode</b>	Tipo de datos	Descripción de la propiedad
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Establece el formato de hora para el campo (sólo se aplica a campos con almacenamiento TIME o TIMESTAMP). Formato de uso: NODE.time_format.FIELDNAME Por ejemplo: set :tablnode.time_format. set 'BOF_enter' = "HHMMSS"
column_width	entero	Establece el ancho de columna para el campo. Un valor de -1 establecerá el ancho de columna en Auto. Formato de uso: NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	Establece la justificación de columna para el campo. Formato de uso: NODE.justify.FIELDNAME

## Propiedades de transformnode



El nodo Transformación permite seleccionar y previsualizar los resultados de las transformaciones antes de aplicarlas a los campos seleccionados. [Si desea obtener más información, consulte el tema Nodo Transformación en el capítulo 6 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create transformnode
set :transformnode.fields = [EDAD INGRESOS]
set :transformnode.formula = Select
set :transformnode.formula_log_n = true
set :transformnode.formula_log_n_offset = 1
```

Propiedades de <b>transformnode</b>	Tipo de datos	Descripción de la propiedad
fields	[ <i>campo1</i> ... <i>campon</i> ]	Campos que se utilizarán en la transformación.
formula	All Select	Indica si se deben calcular todas las transformaciones o sólo las seleccionadas.
formula_inverse	marca	Indica si se debe utilizar la transformación inversa.

Propiedades de transformnode	Tipo de datos	Descripción de la propiedad
formula_inverse_offset	número	Indica el desplazamiento de los datos que se utilizará en la fórmula. Por defecto es 0, a menos que el usuario especifique un valor.
formula_log_n	marca	Indica si debe utilizarse la transformación $\log_n$ .
formula_log_n_offset	número	
formula_log_10	marca	Indica si debe utilizarse la transformación $\log_{10}$ .
formula_log_10_offset	número	
formula_exponential	marca	Indica si se debe utilizar la transformación exponencial ( $e^x$ ).
formula_square_root	marca	Indica si se debe utilizar la transformación de raíz cuadrada.
use_output_name	marca	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
paginate_output	marca	Si output_format es HTML, provocará que los resultados se separen por páginas.
lines_per_page	número	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	cadena	Indica el nombre de archivo que se utilizará para el resultado de archivo.

# Propiedades de nodos Exportar

## Propiedades de nodos Exportar comunes

Las siguientes propiedades son comunes a todos los nodos de exportación:

Propiedad	Valores	Descripción de la propiedad
publish_path	<i>cadena</i>	Introduzca el nombre raíz que se utilizará para la imagen publicada y los archivos de parámetros.
publish_metadata	<i>marca</i>	Especifica si un archivo de metadatos se crea y que describe las entradas y los resultados de la imagen y sus modelos de datos.
publish_use_parameters	<i>marca</i>	Especifica si se incluyen parámetros de ruta en el archivo *.par.
publish_parameters	<i>lista de cadenas</i>	Especifica los parámetros que se van a incluir.
execute_mode	export_data publish	Especifica si el nodo se ejecuta sin publicar la ruta, o si la ruta se publica automáticamente cuando se ejecuta el nodo.

## Propiedades del nodo de exportación Cognos



El nodo Exportar de IBM Cognos BI exporta datos en un formato que pueden leer las bases de datos de Cognos BI. [Si desea obtener más información, consulte el tema Nodo Exportar de IBM Cognos BI en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

*Nota:* Para este nodo, debe definir una conexión de Cognos y una conexión ODBC.

### Conexión de Cognos

Las propiedades de la conexión de Cognos son las siguientes.

propiedades de cognosexportnode	Tipo de datos	Descripción de la propiedad
cognos_connection	{ <i>"campo"</i> , <i>"campo"</i> , ... , <i>"campo"</i> }	Una propiedad de la lista que contiene los detalles de conexión para el servidor de Cognos. El formato es: { <i>"Cognos_server_URL"</i> , <i>login_mode</i> , <i>"espacio de nombre"</i> , <i>"nombre de usuario"</i> , <i>"contraseña"</i> } donde: <i>Cognos_server_URL</i> es la URL del servidor de Cognos al que está realizando la exportación

propiedades de cognosexportnode	Tipo de datos	Descripción de la propiedad
		<i>login_mode</i> indica si se utiliza el inicio de sesión anónimo, y si es true o false; si se establece como true, los campos siguientes deberían establecerse como "" <i>espacio de nombre</i> especifica el proveedor de autenticación de seguridad utilizado para registrarse en el servidor <i>nombre de usuario</i> y <i>contraseña</i> son aquellos utilizados para registrarse en el servidor de Cognos
cognos_package_name	cadena	La ruta y el nombre del paquete de Cognos al que está exportando datos, por ejemplo: /Public Folders/MyPackage
cognos_datasource	cadena	
cognos_export_mode	Publish ExportFile	
cognos_filename	cadena	

### conexión ODBC

Las propiedades de la conexión ODBC son idénticas a las indicadas para databaseexportnode en la sección siguiente, a excepción de la propiedad datasource, que no es válida.

## Propiedades de databaseexportnode



El nodo Opciones de exportación escribe datos en orígenes de datos relacionales compatibles con ODBC. Para escribir en un origen de datos ODBC, el origen de datos debe existir y debe tener permiso para escribir en él. [Si desea obtener más información, consulte el tema Nodo Exportar base de datos en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
/*
Usar este ejemplo con fraud.str de la carpeta demo
Asume que se ha configurado una fuente de datos denominada "Mifuentededatos"
*/
create databaseexport
connect claimvalue:applyneuralnetwork to :databaseexport
# Pestaña Exportar
set :databaseexport.username = "user"
set :databaseexport.datasource = "Mifuentededatos"
set :databaseexport.password = "password"
set :databaseexport.table_name = "predictions"
set :databaseexport.write_mode = Create
set :databaseexport.generate_import = true
set :databaseexport.drop_existing_table = true
set :databaseexport.delete_existing_rows = true
set :databaseexport.default_string_size = 32

# Cuadro de diálogo Esquema
```

```

set :databaseexport.type.region = "VARCHAR(10)"
set :databaseexport.export_db_primarykey.id = true
set :databaseexportnode.use_custom_create_table_command = true
set :databaseexportnode.custom_create_table_command = "My SQL Code"

```

# Indexes dialog

```

set :databaseexport.use_custom_create_index_command = true
set :databaseexport.custom_create_index_command = \
  "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"
set :databaseexport.indexes.MYINDEX.fields = [id region]

```

propiedades de databaseexportnode	Tipo de datos	Descripción de la propiedad
datasource	<i>cadena</i>	
username	<i>cadena</i>	
password	<i>cadena</i>	
epassword	<i>cadena</i>	Esta propiedad es de solamente lectura durante la ejecución. Para generar una contraseña codificada, utilice la herramienta Contraseña disponible del menú Herramientas. <a href="#">Si desea obtener más información, consulte el tema Generación de una contraseña codificada en el capítulo 5 el p. 61.</a>
table_name	<i>cadena</i>	
write_mode	Create Append Merge	
map	<i>cadena</i>	Asigna un nombre de campo de ruta a un nombre de columna de la base de datos (solamente es válido si write_mode es Merge). Ejemplo: set :databaseexportnode.map.streamBP = 'databaseBP' Se permite la correspondencia múltiple de acuerdo con la posición del campo, por ejemplo: set :databaseexportnode.map={streamfield1 field1}{streamfield2 field2}{streamfield3 field3} Para una fusión, todos los campos se deben asignar para que se exporten. Los nombres de campos que no existen en la base de datos se añaden como nuevas columnas.



propiedades de databaseexportnode	Tipo de datos	Descripción de la propiedad
key_fields	[campo campo ... campo]	Especifica el campo de ruta que se utiliza para la clave; la propiedad map muestra los elementos que se corresponden con la base de datos.
join	Database Add	Ejemplo: set : databaseexportnode.join = Database
drop_existing_table	marca	
delete_existing_rows	marca	
default_string_size	entero	
type		Propiedad estructurada que se utiliza para establecer el tipo de esquema. Formato de uso: set :databaseexportnode. type.BP = 'VARCHAR(10)'
generate_import	marca	
use_custom_create_table_command	marca	Utilice la propiedad <i>custom_create_table</i> para modificar el comando de SQL estándar CREATE TABLE.
custom_create_table_command	cadena	Especifica el comando de cadena a utilizar en lugar del comando de SQL estándar CREATE TABLE.
use_batch	marca	Las siguientes propiedades son opciones avanzadas para la carga masiva de la base de datos. Un valor verdadero para Use_batch desactiva fila a fila las confirmaciones en la base de datos.
batch_size	número	Especifica el número de registros para enviar a la base de datos antes de confirmar en la memoria.
bulk_loading	Off ODBC External	Especifica el tipo de carga masiva. A continuación se muestran las opciones adicionales para ODBC y External.
odbc_binding	Row Column	Especifique el enlace a lo largo de las filas o de las columnas para la carga masiva a través de ODBC.

<b>propiedades de databaseexportnode</b>	<b>Tipo de datos</b>	<b>Descripción de la propiedad</b>
loader_delimit_mode	Tab Space Other	Especifique el tipo de delimitador para la carga masiva a través de un programa externo. Seleccione <b>Other</b> junto con la propiedad <b>loader_other_delimiter</b> para especificar los delimitadores, como la coma (,).
loader_other_delimiter	<i>cadena</i>	
specify_data_file	<i>marca</i>	Una marca verdadera activa la siguiente propiedad <b>data_file</b> , en la que puede especificar el nombre de archivo y la ruta de acceso en la que se va a escribir al realizar la carga masiva en la base de datos.
data_file	<i>cadena</i>	
specify_loader_program	<i>marca</i>	Una marca verdadera activa la siguiente propiedad <b>loader_program</b> , en la que puede especificar el nombre y la ubicación de un programa o proceso del cargador externo.
loader_program	<i>cadena</i>	
gen_logfile	<i>marca</i>	Una marca verdadera activa la siguiente propiedad <b>logfile_name</b> , en la que puede especificar el nombre de un archivo en el servidor para generar un registro de errores.
logfile_name	<i>cadena</i>	
check_table_size	<i>marca</i>	Una marca verdadera permite la comprobación de la tabla para garantizar que el aumento del tamaño de la tabla de la base de datos se corresponde con el número de filas exportadas desde IBM® SPSS® Modeler.
loader_options	<i>cadena</i>	Especifique los argumentos adicionales, como <b>-comment</b> y <b>-specialdir</b> , en el programa cargador.
export_db_primarykey	<i>marca</i>	Determina si el campo especificado es una clave primaria.
use_custom_create_index_command	<i>marca</i>	Si se establece <b>true</b> , se activa SQL personalizado para todos los índices.

propiedades de databaseexportnode	Tipo de datos	Descripción de la propiedad
custom_create_index_command	cadena	Especifica el comando de SQL empleado para crear índices cuando SQL personalizado está activado. (Este valor puede anularse para determinados índices, tal y como se indica a continuación.)
indexes.INDEXNAME.fields		Crea el índice especificado si procede y enumera los nombres de campos que se van a incluir en él.
indexes.INDEXNAME.use_custom_create_index_command	marca	Se usa para activar o desactivar el SQL personalizado para un índice específico.
indexes.INDEXNAME.custom_create_command		Especifica el SQL personalizado que se usa para el índice específico.
indexes.INDEXNAME.remove	marca	Si se establece true, se elimina el índice específico del grupo de índices.
table_space	cadena	Especifica el espacio de tabla que se creará.
use_partition	marca	Especifica que se utilizará el campo Distribuir Hash.
partition_field	cadena	Especifica el contenido del campo Distribuir Hash.

*Nota:* Para algunas bases de datos, puede especificar que se crearán tablas de bases de datos para la exportación con compresión (por ejemplo, el equivalente a CREATE TABLE MYTABLE (...) COMPRESS YES; en SQL). Las propiedades use\_compression y compression\_mode se proporcionan para admitir esta función, de la manera siguiente.

propiedades de databaseexportnode	Tipo de datos	Descripción de la propiedad
use_compression	Booleana	Si está establecido como true, crea tablas para la exportación con compresión.
compression_mode	Row Page	Establece el nivel de compresión de las bases de datos de SQL Server.
	Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low	Establece el nivel de compresión de las bases de datos de Oracle. Tenga en cuenta que los valores OLTP, Query_High, Query_Low, Archive_High y Archive_Low requieren un mínimo de Oracle 11gR2.

### **Ejemplo: SQL Server**

```
var DBSource
set DBSource = get node TestCompressionSQL
```

```
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Page
```

```
execute DBSource
```

### **Ejemplo: Oracle 11gR1**

```
var DBSource
set DBSource = get node TestCompressionOracle11gR1
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Direct_Load_Operations
```

```
execute DBSource
```

### **Ejemplo: Oracle 11gR2**

```
var DBSource
set DBSource = get node TestCompressionOracle11gR2
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Basic
```

```
execute DBSource
```

## **Propiedades de datacollectionexportnode**



El nodo de exportación IBM® SPSS® Data Collection abre los datos en el formato utilizado por el software de investigación de mercados Data Collection. Se debe instalar la biblioteca de datos de Data Collection para utilizar este nodo. [Si desea obtener más información, consulte el tema Nodos de exportación IBM SPSS Data Collection en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### **Ejemplo**

```
create datacollectionexportnode
set :datacollectionexportnode.metadata_file = "c:\museums.mdd"
set :datacollectionexportnode.merge_metadata = Overwrite
set :datacollectionexportnode.casedata_file = "c:\museumdata.sav"
set :datacollectionexportnode.generate_import = true
set :datacollectionexportnode.enable_system_variables = true
```

propiedades de datacollectionexportnode	Tipo de datos	Descripción de la propiedad
metadata_file	cadena	Nombre del archivo de metadatos que se va a exportar.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	marca	Especifica si el archivo .mdd exportado debe incluir las variables de sistema de Data Collection.

propiedades de datacollectionexportnode	Tipo de datos	Descripción de la propiedad
casedata_file	cadena	Nombre del archivo .sav donde se exportan los datos de casos.
generate_import	marca	

## Propiedades de excelexportnode



El nodo de exportación Excel ofrece los datos resultantes en formato de Microsoft Excel (.xls). Si lo desea, puede elegir iniciar automáticamente Excel y abrir el archivo exportado cuando se ejecute el nodo. [Si desea obtener más información, consulte el tema Nodo de exportación Excel en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create excelexportnode
set :excelexportnode.full_filename = "C:/output/myexport.xls"
set :excelexportnode.excel_file_type = Excel2007
set :excelexportnode.inc_field_names = True
set :excelexportnode.inc_labels_as_cell_notes = False
set :excelexportnode.launch_application = True
set :excelexportnode.generate_import = True
```

propiedades de excelexportnode	Tipo de datos	Descripción de la propiedad
full_filename	cadena	
excel_file_type	Excel2003 Excel2007	
export_mode	Create Append	
inc_field_names	marca	Determina si los nombres de campos deben incluirse en la primera fila de la hoja de trabajo.
start_cell	cadena	Especifica la casilla de inicio de la exportación.
worksheet_name	cadena	Nombre de la hoja de trabajo que se va a escribir.
launch_application	marca	Determina si Excel debe invocarse para el archivo resultante. Tenga en cuenta que deberá especificar la ruta para iniciar Excel en el cuadro de diálogo Complementos (menú Herramientas, Complementos).
generate_import	marca	Determina si debe crearse un nodo Importar a Excel que leerá el archivo de datos exportado.

## Propiedades de outputfilenode



El nodo Archivo plano produce datos en un archivo de texto delimitado. Esto es útil para exportar datos que se pueden leer con otro software de hoja de cálculo o de análisis. [Si desea obtener más información, consulte el tema nodo de exportación Archivo plano en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create outputfile
set :outputfile.full_filename = "c:/output/flatfile_output.txt"
set :outputfile.write_mode = Append
set :outputfile.inc_field_names = False
set :outputfile.use_newline_after_records = False
set :outputfile.delimit_mode = Tab
set :outputfile.other_delimiter = ","
set :outputfile.quote_mode = Double
set :outputfile.other_quote = "*"
set :outputfile.decimal_symbol = Period
set :outputfile.generate_import = True
```

propiedades de outputfilenode	Tipo de datos	Descripción de la propiedad
full_filename	cadena	Nombre del archivo de resultados.
write_mode	Overwrite Append	
inc_field_names	marca	
use_newline_after_records	marca	
delimit_mode	Comma Tab Space Other	
other_delimiter	carácter	
quote_mode	None Single Double Other	
other_quote	marca	
generate_import	marca	
encoding	StreamDefault SystemDefault "UTF-8"	

## Propiedades de sasexportnode



El nodo Exportar SAS produce datos en formato SAS, para leerlos en SAS o en un paquete de software compatible con SAS. Hay disponibles tres formatos de archivos SAS: SAS para Windows/OS2, SAS para UNIX o SAS versión 7/8. [Si desea obtener más información, consulte el tema Nodo Exportar SAS en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```
create sasexportnode
set :sasexportnode.full_filename = "c:/output/SAS_output.sas7bdat"
set :sasexportnode.format = SAS8
set :sasexportnode.export_names = NamesAndLabels
set :sasexportnode.generate_import = True
```

propiedades de sasexportnode	Tipo de datos	Descripción de la propiedad
format	Windows UNIX SAS7 SAS8	Campos de etiquetas de propiedad de variantes.
full_filename	<i>cadena</i>	
export_names	NamesAndLabels NamesAsLabels	Se utiliza para asignar los nombres de campos de IBM® SPSS® Modeler que se vayan a exportar a nombres de variables de IBM® SPSS® Statistics o SAS.
generate_import	<i>marca</i>	

**Propiedades de statisticsexportnode**

El nodo Exportar Statistics ofrece los resultados en formato IBM® SPSS® Statistics.sav *de*. Los archivos .sav se pueden leer con SPSS Statistics Base y otros productos. Este es también el formato utilizado para los archivos caché de IBM® SPSS® Modeler. [Si desea obtener más información, consulte el tema Nodo Exportación de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

Las propiedades de este nodo están descritas en [Propiedades de statisticsexportnode](#) el p. 317.

**Propiedades de xmlexportnode**

El nodo de exportación XML exporta datos a un archivo en formato XML. También puede crear un nodo de origen XML para leer los datos exportados a la ruta. [Si desea obtener más información, consulte el tema Nodo de exportación XML en el capítulo 7 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Ejemplo**

```
create xmlexportnode
set :xmlexportnode.full_filename = "c:\export\data.xml"
set :xmlexportnode.map = [{"/catalog/book/genre" genre} {"catalog/book/title" title}]
```

propiedades de xmlexportnode	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	(obligatorio) Ruta completa y nombre de archivo del archivo XML para exportar.
use_xml_schema	<i>marca</i>	Especifica si utilizar un esquema XML (archivo XSD o DTD) para controlar la estructura de los datos exportados.

<b>propiedades de xmlexportnode</b>	<b>Tipo de datos</b>	<b>Descripción de la propiedad</b>
full_schema_filename	<i>cadena</i>	Ruta completa y nombre de archivo del archivo XSD o DTD que se quiere utilizar. Es obligatorio si use_xml_schema está establecido como true.
generate_import	<i>marca</i>	Genera un nodo de fuente XML que leerá el archivo de datos exportados de nuevo en la ruta.
records	<i>cadena</i>	Expresión de XPath que denota el límite de registro.
map	<i>cadena</i>	Asigna el nombre de campo a la estructura XML. Ejemplo: set :xmlexportnode.map = [{"/top/node1" field1}{"/top/node2" field2}] Asigna el campo de ruta field1 al elemento XML /top/node1, y así sucesivamente.



# Propiedades de nodos de IBM SPSS Statistics

## Propiedades de statisticsimportnode



El nodo Archivo Statistics lee los datos desde un formato de archivo *.sav* que utiliza IBM® SPSS® Statistics y archivos caché guardados en IBM® SPSS® Modeler, que también puede utilizar el mismo formato. [Si desea obtener más información, consulte el tema Nodo de archivos estadísticos en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create statisticsimportnode
set :statisticsimportnode.full_filename = "C:/data/drug1n.sav"
set :statisticsimportnode.import_names = true
set :statisticsimportnode.import_data = true
```

propiedades de statisticsimportnode	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	El nombre completo del archivo, incluyendo la ruta.
import_names	NamesAndLabels LabelsAsNames	Método para gestionar nombres y etiquetas de variables.
import_data	DataAndLabels LabelsAsData	Método para gestionar valores y etiquetas.
use_field_format_for_storage	<i>Booleana</i>	Especifica si se utiliza la información de formato de campo de SPSS Statistics al importar.

## Propiedades de statisticstransformnode



El nodo Transformación Statistics ejecuta una selección de comandos de sintaxis de IBM® SPSS® Statistics en los orígenes de datos de IBM® SPSS® Modeler. Este nodo requiere una copia de SPSS Statistics con licencia. [Si desea obtener más información, consulte el tema Nodo Transformación de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create statisticstransformnode
set :statisticstransformnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticstransformnode.new_name.NewVar = "Medicamentos mezclados"
```

```
set :statisticstransformnode.check_before_saving = true
```

propiedades de statisticstransformnode	Tipo de datos	Descripción de la propiedad
syntax	cadena	
check_before_saving	marca	Valida la sintaxis introducida antes de guardar las entradas. Muestra un mensaje de error si la sintaxis no es válida.
default_include	marca	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.
include	marca	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.
new_name	cadena	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.

## Propiedades de statisticsmodelnode



El nodo Modelo Statistics permite analizar y trabajar con sus datos ejecutando los procedimientos de IBM® SPSS® Statistics que producen PMML. Este nodo requiere una copia de SPSS Statistics con licencia. Si desea obtener más información, consulte el tema Nodo Modelo de Statistics en el capítulo 8 en *Nodos de origen, proceso y resultado de IBM SPSS Modeler 15*.

### Ejemplo

```
create statisticsmodelnode
set :statisticsmodelnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticsmodelnode.new_name.NewVar = "Medicamentos mezclados"
```

propiedades de statisticsmodelnode	Tipo de datos	Descripción de la propiedad
syntax	cadena	
default_include	marca	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.
include	marca	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.
new_name	cadena	Si desea obtener más información, consulte el tema Propiedades de filternode en el capítulo 14 el p. 162.

## Propiedades de *statisticsoutputnode*



El nodo Resultados de Statistics le permite llamar a un procedimiento de IBM® SPSS® Statistics para analizar los datos de IBM® SPSS® Modeler. A wide variety of SPSS Statistics analytical procedures is available. This node requires a licensed copy of SPSS Statistics. [Si desea obtener más información, consulte el tema Nodo Resultados de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create statisticsoutputnode
set :statisticsoutputnode.syntax = "SORT CASES BY Edad(A) Sexo(A) PS(A) Colesterol(A)"
set :statisticsoutputnode.use_output_name = False
set :statisticsoutputnode.output_mode = File
set :statisticsoutputnode.full_filename = "Casos por edad, sexo e historial médico"
set :statisticsoutputnode.file_type = HTML
```

propiedades de <i>statisticsoutputnode</i>	Tipo de datos	Descripción de la propiedad
mode	Dialog Syntax	Selecciona la opción “cuadro de diálogo de SPSS Statistics” o el editor Sintaxis
syntax	<i>cadena</i>	
use_output_name	<i>marca</i>	
output_name	<i>cadena</i>	
output_mode	Screen File	
full_filename	<i>cadena</i>	
file_type	HTML SPV SPW	

## Propiedades de *statisticsexportnode*



El nodo Exportar Statistics ofrece los resultados en formato IBM® SPSS® Statistics.sav *de*. Los archivos *.sav* se pueden leer con SPSS Statistics Base y otros productos. Este es también el formato utilizado para los archivos caché de IBM® SPSS® Modeler. [Si desea obtener más información, consulte el tema Nodo Exportación de Statistics en el capítulo 8 en \*Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

### Ejemplo

```
create statisticsexportnode
set :statisticsexportnode.full_filename = "c:/output/SPSS_Statistics_out.sav"
set :statisticsexportnode.field_names = Names
set :statisticsexportnode.launch_application = True
```

```
set :statisticsexportnode.generate_import = True
```

<b>propiedades de statisticsexportnode</b>	<b>Tipo de datos</b>	<b>Descripción de la propiedad</b>
full_filename	<i>cadena</i>	
launch_application	<i>marca</i>	
export_names	NamesAndLabels NamesAsLabels	Se utiliza para asignar los nombres de campos de SPSS Modeler que se vayan a exportar a nombres de variables de SPSS Statistics o SAS.
generate_import	<i>marca</i>	

## Propiedades de Supernodos

En las siguientes tablas se describen las propiedades específicas de los Supernodos. Tenga en cuenta que las propiedades de nodos comunes se aplican también a los Supernodos.

Tabla 22-1  
*source\_supernode*

Nombre de la propiedad	Tipo de propiedad / Lista de valores	Descripción de la propiedad
parameters	<i>cualquiera</i>	Utilice esta propiedad para crear y acceder a los parámetros especificados en una tabla de parámetros de Supernodo. Consulte los detalles que se muestran a continuación.

Tabla 22-2  
*process\_supernode*

Nombre de la propiedad	Tipo de propiedad / Lista de valores	Descripción de la propiedad
parameters	<i>cualquiera</i>	Utilice esta propiedad para crear y acceder a los parámetros especificados en una tabla de parámetros de Supernodo. Consulte los detalles que se muestran a continuación.

Tabla 22-3  
*terminal\_supernode*

Nombre de la propiedad	Tipo de propiedad / Lista de valores	Descripción de la propiedad
parameters	<i>cualquiera</i>	Utilice esta propiedad para crear y acceder a los parámetros especificados en una tabla de parámetros de Supernodo. Consulte los detalles que se muestran a continuación.
execute_method	Script Normal	
script	<i>cadena</i>	

### **Parámetros de Supernodos**

Puede utilizar procesos para crear o establecer parámetros de Supernodo utilizando el formato general:

```
set mySuperNode.parameters.minvalue = 30
```

También puede especificar el tipo de Supernodo además del nombre (o en lugar del nombre):

```
set :process_supernode.parameters.minvalue = 30
```

```
set miSupernodo:process_supernode.parameters.minvalue = 30
```

También puede establecer el valor del parámetro utilizando una expresión CLEM:

```
set :process_supernode.parameters.minvalue = "<expresión>"
```

### **Configuración de las propiedades de nodos encapsulados**

Puede establecer las propiedades de determinados nodos encapsulados dentro un Supernodo creando un parámetro de Supernodo que coincida con el nombre literal del nodo y la propiedad que quiera establecer. Por ejemplo, imaginemos que tiene un Supernodo de origen que incluye un nodo de archivo de variables encapsulado para leer los datos. Puede pasar el nombre del archivo que se desea leer (especificado mediante la propiedad `full_filename`) de la siguiente manera:

```
set :source_supernode.parameters.':variablefilenode.full_filename' = "c:/misdatos.txt"
```

Se creará un parámetro de Supernodo llamado `:variablefilenode.full_filename` con el valor `c:/misdatos.txt`. En caso de que exista un nodo del tipo especificado en el Supernodo, se establecerá en consecuencia el valor de la propiedad especificada. Tenga en cuenta que esto se lleva a cabo en el proceso de la ruta (es decir, el proceso de la ruta que *incluye* el Supernodo) en vez de en el proceso del Supernodo. Asegúrese de utilizar comillas simples para especificar el nombre del parámetro.

Este método se puede utilizar con cualquier nodo encapsulado, siempre que sean válidos tanto el nodo como la referencia a la propiedad. Por ejemplo, para establecer la propiedad `rand_pct` de un nodo Muestrear encapsulado, puede utilizarse cualquiera de los siguientes métodos:

```
set mySuperNode.parameters.':samplenode.rand_pct' = 50
```

*o*

```
set mySuperNode.parameters.'Sample.rand_pct'= 50
```

*o*

```
set mySuperNode.parameters.'Sample:samplenode.rand_pct'= 50
```

La primera referencia anterior supone que hay un único nodo Muestrear en la ruta; la segunda, que hay un único nodo con el nombre "Muestrear" independientemente de su tipo. La tercera referencia es la más explícita, ya que especifica tanto el nombre como el tipo del nodo.

[Si desea obtener más información, consulte el tema \*Parámetros de Supernodos en el capítulo 9 en Nodos de origen, proceso y resultado de IBM SPSS Modeler 15\*.](#)

**Limitaciones de los procesos de Supernodos.** Los Supernodos no pueden manipular otras rutas y no pueden cambiar la ruta actual. Por tanto, no se pueden utilizar en procesos de Supernodos los comandos que se aplican a rutas, como `open stream`, `get stream`, `execute_script`, etc.

# Avisos

Esta información se ha desarrollado para los productos y servicios ofrecidos en todo el mundo.

Puede que IBM no ofrezca los productos, los servicios o las características de los que se habla en este documento en otros países. Consulte a su representante local de IBM para obtener información acerca de los productos y servicios que está disponibles actualmente en su zona. Toda referencia que se haga de un producto, programa o servicio de IBM no implica que sólo se deba utilizar ese producto, programa o servicio de IBM. En su lugar, puede utilizarse todo producto, programa o servicio con funcionalidades equivalentes que no infrinjan los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y comprobar el funcionamiento de todo producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o aplicaciones de patentes pendientes que cubren el asunto descrito en este documento. Este documento no le otorga ninguna licencia para estas patentes. Puede enviar preguntas acerca de las licencias, por escrito, a:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, Estados Unidos*

Si tiene alguna pregunta sobre la licencia relacionada con la información del juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe sus preguntas por escrito a:

*Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.*

**El párrafo siguiente no se aplica a los Reino Unido o cualquier otro país donde tales disposiciones son incompatibles con la legislación local:** INTERNATIONAL BUSINESS MACHINES PROPORCIONA ESTA PUBLICACIÓN “TAL CUAL” SIN GARANTÍA DE NINGÚN TIPO, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO, PERO NO LIMITADA A, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO PARTICULAR. Algunos estados no permiten la renuncia a expresar o a garantías implícitas en determinadas transacciones , por lo tanto , esta declaración no se aplique a usted.

Esta información puede incluir imprecisiones técnicas o errores tipográficos. Periódicamente, se efectúan cambios en la información aquí y estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en los productos y/o los programas descritos en esta publicación en cualquier momento sin previo aviso.

Cualquier referencia a sitios Web que no sean de IBM en esta información sólo es ofrecida por comodidad y de ningún modo sirve como aprobación de esos sitios Web. Los materiales en esos sitios Web no forman parte del material de este producto de IBM y el uso de estos sitios Web es bajo su propio riesgo.

IBM puede utilizar cualquier información que le suministre en cualquier forma que considere adecuada, sin incurrir en ninguna obligación para usted.

Los licenciatarios de este programa que deseen tener información sobre el mismo con el objetivo de habilitar: (i) el intercambio de información entre programas creados independientemente y otros programas (incluyendo este) y (ii) el uso común de la información que se ha intercambiado, deben ponerse en contacto con:

*IBM Software Group, a la atención de: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.*

Esta información estará disponible, bajo las condiciones adecuadas, incluyendo en algunos casos el pago de una cuota.

IBM proporciona el programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible para el mismo bajo los términos de IBM Customer Agreement (Acuerdo de cliente de IBM), IBM International Program License Agreement (Acuerdo de licencia de programa internacional de IBM) o cualquier acuerdo equivalente entre las partes.

Cualquier dato de rendimiento mencionado aquí ha sido determinado en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Es posible que algunas medidas se hayan realizado en sistemas en desarrollo y no existe ninguna garantía de que estas medidas sean las mismas en los sistemas comerciales. Además, es posible que algunas medidas hayan sido estimadas a través de extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben consultar los datos que corresponden a su entorno específico.

Se ha obtenido información acerca de productos que no son de IBM de los proveedores de esos productos, de sus publicaciones anunciadas o de otras fuentes disponibles públicamente. IBM no ha probado estos productos y no puede confirmar la precisión de su rendimiento, su compatibilidad o cualquier otra reclamación relacionada con productos que no sean de IBM. Las preguntas acerca de las aptitudes de productos que no sean de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones sobre el futuro del rumbo y la intención de IBM están sujetas a cambio o retirada sin previo aviso y representan únicamente metas y objetivos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos lo máximo posible, los ejemplos incluyen los nombres de las personas, empresas, marcas y productos. Todos esos nombres son ficticios y cualquier parecido con los nombres y direcciones utilizados por una empresa real es pura coincidencia.

Si está viendo esta información en copia electrónica, es posible que las fotografías y las ilustraciones en color no aparezcan.

### **Marcas comerciales**

IBM, el logotipo de IBM, [ibm.com](http://ibm.com) y SPSS son marcas comerciales de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Existe una lista actualizada de marcas comerciales de IBM en Internet en <http://www.ibm.com/legal/copytrade.shtml>.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas comerciales o marcas registradas de Intel Corporation o sus filiales en Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos, otros países o ambos.



---

Microsoft, Windows, Windows NT, y el logotipo de Windows son marcas comerciales de Microsoft Corporation en Estados Unidos, otros países o ambos.

UNIX es una marca registrada de The Open Group en Estados Unidos y otros países.

Java y todas las marcas comerciales y los logotipos basados en Java son marcas comerciales de Sun Microsystems, Inc. en Estados Unidos, otros países o ambos.

Otros productos y nombres de servicio pueden ser marcas comerciales de IBM u otras empresas.



- anotaciones
  - acceso en procesos, 68
- Árbol de decisión de MS
  - propiedades de procesos de nodos, 264, 267
- argumentos
  - archivo de comandos, 71
  - conexión con el servidor, 71
  - IBM SPSS Collaboration and Deployment Services Repository conexión, 73
  - sistema, 73
- avisos legales, 321
  
- bucles
  - uso en procesos, 58, 66–67
- búsqueda de texto, 17
  
- cadena, 76, 78
  - cambio entre minúsculas y mayúsculas, 58
  - procesos, 21
- cadena de texto
  - incrustación en los procesos, 30
- cadena literales
  - incrustación en los procesos, 30
- campos, 76, 78
  - desactivación de los procesos, 180
- campos continuos
  - propiedad de valores, 66
- campos de marcas
  - propiedad de valores, 66
- campos de tiempo
  - conversión , 102
- campos nominales
  - propiedad de valores, 66
- carácter de barra invertida en expresiones CLEM, 78
- caracteres, 76, 78
- CLEM
  - expresiones, 76
  - lengua, 76
  - procesos, 9, 20
  - tipos de datos, 77–78
- CLEM expresiones
  - parámetros, 26
- CLEM funciones
  - a nivel de bit, 91
  - aleatorio, 92
  - cadena, 93
  - comparación, 86
  - fecha y hora, 98
  - funciones especiales, 110
  - globales, 108
  - información, 84
  - lógicas, 88
  - numéricos, 89
  - probabilidad, 91
  - secuencia, 103, 105
  - trigonómicas, 90
  - vacíos y nulos, 109
- comando cargar nodo, 41
- comando clear generated palette, 46, 62
- comando clear stream, 50
- comando close ARCHIVO, 54
- comando close RUTA, 49
- comando connect NODO, 38
- comando create NODO, 37
- comando create stream, 49
- comando de conjunto múltiple, 115
- comando delete model, 46
- comando delete NODO, 38
- comando delete output, 56
- comando disable NODO, 38
- comando disconnect NODO, 39
- comando duplicate NODO, 39
- comando enable NODO, 39
- comando execute NODO, 39
- comando execute\_all, 31
- comando execute\_project, 52
- comando execute\_script, 31
- comando exit, 28, 31
- comando export model, 46
- comando export NODO, 40
- comando export output, 56
- comando flush NODO, 40
- comando for, 24, 28, 58, 64, 67
- comando for...endfor, 32
- comando get, 23
- comando get node, 41
- comando get output, 56
- comando get stream, 50
- comando if, 28, 64
- comando if...then...else, 33
- comando insert model, 47
- comando load model, 48
- comando load output, 57
- comando load project, 53
- comando load state, 53
- comando load stream, 50
- comando open ARCHIVO, 54
- comando open stream, 24, 50
- comando position NODO, 41
- comando rename NODO, 25, 42
- comando retrieve, 59
- comando retrieve model, 48
- comando retrieve node, 42
- comando retrieve output, 57
- comando retrieve project, 53
- comando retrieve stream, 50
- comando save, 23
- comando save model, 48
- comando save node, 42
- comando save output, 57

- comando save project, 53
- comando save RUTA, 51
- comando set, 21, 24–26, 33
- comando store, 59
- comando store model, 48
- comando store node, 43
- comando store output, 57
- comando store project, 53
- comando store stream, 51
- comando value, 54
- comando var, 21, 25, 36
- comando with stream, 24, 52
- comando write ARCHIVO, 55
- comando writeln ARCHIVO, 55, 64, 67
- comentarios
  - procesos, 29
- comprobación de errores
  - procesos, 61
- concatenación de cadenas, 85
- Conglomeración de secuencias de MS
  - propiedades de procesos de nodos, 268
- continuaciones
  - procesos, 29
- contraseñas
  - adición a procesos, 61
  - codificadas, 71
- contraseñas codificadas
  - adición a procesos, 61
- convenciones, 83
- count\_non\_nulls function, 86
  
- directivas de desarrollo de árboles
  - incrustación en los procesos, 30
- distribución chi-cuadrado
  - funciones de probabilidad, 91
- distribución  $f$ 
  - funciones de probabilidad, 91
- distribución normal
  - funciones de probabilidad, 91
- distribución  $t$ 
  - funciones de probabilidad, 91
- documentación, 4
  
- ejecución de procesos, 16
- ejemplos
  - conceptos básicos, 6
  - Manual de aplicaciones, 4
- ejemplos de aplicaciones, 4
- enteros, 76–77
- espacio en blanco
  - eliminación de cadenas, 93
- espacios
  - eliminación de cadenas, 93
- exportación
  - modelos, 46
  - nodos, 40
  - PMML, 40, 46
- SQL, 40, 46
- expresiones, 76
- expresiones CLEM
  - búsqueda y reemplazo de texto, 17
  - procesos, 28, 33
  
- fechas
  - conversión, 102
  - gestión, 102
- formato de texto
  - exportación de modelos, 46
  - exportación de nodos, 40
- formato HTML
  - exportación de modelos, 46
  - exportación de nodos, 40
- formato PMML
  - exportación de modelos, 46
  - exportación de nodos, 40
- formato SQL
  - exportación de nodos, 40, 46
- formatos de fecha, 79–80
- formatos de hora, 79–80
- función @BLANK, 84, 109
- función @DIFF, 103, 105
- función @FIELD, 110
- función @FIELDS\_BETWEEN, 110
- función @FIELDS\_MATCHING, 110
- función @INDEX, 103, 105
- función @LAST\_NON\_BLANK, 103, 105, 109
- función @MAX, 103, 105
- función @MEAN, 103, 105
- función @MIN, 103, 105
- Función @MULTI\_RESPONSE\_SET, 110
- función @NULL, 84, 109
- función @OFFSET, 103, 105
- función @PARTITION\_FIELD, 110
- función @PREDICTED, 110
- función @SDEV, 103, 105
- función @SINCE, 103, 105
- función @TARGET, 110
- función @TESTING\_PARTITION, 110
- función @THIS, 103, 105
- función @TODAY, 98
- función @TRAINING\_PARTITION, 110
- función @VALIDATION\_PARTITION, 110
- función abs, 89
- función allbutfirst, 93
- función allbutlast, 93
- función alphabefore, 93
- función arccos, 90
- función arccosh, 90
- función arcsin, 90
- función arcsinh, 90
- función arctan, 90
- función arctan2, 90
- función arctanh, 90
- función cdf\_chisq, 91

- función cdf\_f, 91
- función cdf\_normal, 91
- función cdf\_t, 91
- función cos, 90
- función cosh, 90
- función count\_equal, 86
- función count\_greater\_than, 86
- función count\_less\_than, 86
- función count\_not\_equal, 86
- función count\_nulls, 86
- función count\_substring, 93
- función date\_before, 86
- Función datetime\_date, 85
- función DIFF, 105
- función div, 89
- función endstring, 93
- función exponential, 89
- Función first\_index, 86
- Función first\_non\_null, 86
- Función first\_non\_null\_index, 86
- función fracof, 89
- función hasendstring, 93
- función hasmidstring, 93
- función hasstartstring, 93
- función hassubstring, 93
- función INDEX, 105
- función integer\_bitcount, 91
- función integer\_leastbit, 91
- función integer\_length, 91
- función intof, 89
- Función is\_date, 84
- Función is\_datetime, 84
- Función is\_integer, 84
- Función is\_number, 84
- Función is\_real, 84
- Función is\_string, 84
- Función is\_time, 84
- Función is\_timestamp, 84
- función isalphacode, 93
- función isendstring, 93
- función islowercode, 93
- función ismidstring, 93
- función isnumbercode, 93
- función isstartstring, 93
- función issubstring, 93
- función issubstring\_count, 93
- función issubstring\_lim, 93
- función isuppercode, 93
- Función last\_index, 86
- función LAST\_NON\_BLANK, 105
- Función last\_non\_null, 86
- Función last\_non\_null\_index, 86
- función length, 93
- función locchar, 93
- función locchar\_back, 93
- función log, 89
- función log10, 89
- función lowertoupper, 58, 93
- función matches, 93
- función max, 86
- función MAX, 105
- Función max\_index, 86
- función max\_n, 86
- función MEAN, 103, 105
- función mean\_n, 89
- función member, 86
- función min, 86
- función MIN, 105
- Función min\_index, 86
- función min\_n, 86
- función mod, 89
- función negate, 89
- función OFFSET, 105
- función oneof, 92
- función pi, 90
- función power (exponential), 89
- función random, 92
- función random0, 92
- función rem, 89
- función replace, 93
- función replicate, 93
- función round, 89
- función SDEV, 105
- función sdev\_n, 89
- función sign, 89
- función sin, 90
- función SINCE, 105
- función sinh, 90
- función skipchar, 93
- función skipchar\_back, 93
- función soundex, 98
- función soundex\_difference, 98
- función sqrt, 89
- función startstring, 93
- función stripchar, 93
- función strmember, 93
- función subscrs, 93
- función substring, 93
- función substring\_between, 93
- función SUM, 105
- función sum\_n, 89
- función tan, 90
- función tanh, 90
- función tesbit, 91
- función THIS, 105
- función time\_before, 86
- función to\_date, 85, 98
- función to\_datetime, 98
- Función to\_datetime, 85
- Función to\_integer, 85
- Función to\_number, 85
- Función to\_real, 85
- Función to\_string, 85
- función to\_time, 85, 98

- función to\_timestamp, 85, 98
  - función trim, 93
  - función trim\_start, 93
  - función trimend, 93
  - función undef, 109
  - función unicode\_char, 93
  - función unicode\_value, 93
  - función uppertolower, 93
  - Función value\_at, 86
  - funciones, 79–80, 83–84, 103
    - @FIELD, 110
    - @GLOBAL\_MAX, 108
    - @GLOBAL\_MEAN, 108
    - @GLOBAL\_MIN, 108
    - @GLOBAL\_SDEV, 108
    - @GLOBAL\_SUM, 108
    - @PARTITION, 110
    - @PREDICTED, 110
    - @TARGET, 110
  - funciones a nivel de bit, 91
  - funciones de cadena, 58, 93
  - Funciones de CLEM
    - de conversión, 85
    - lista de funciones disponibles, 82
  - funciones de comparación, 86
  - funciones de conversión, 85
  - funciones de distribución, 91
  - funciones de fecha, 79–80
    - date\_before, 86, 98
    - date\_days\_difference, 98
    - date\_in\_days, 98
    - date\_in\_months, 98
    - date\_in\_weeks, 98
    - date\_in\_years, 98
    - date\_months\_difference, 98
    - date\_weeks\_difference, 98
    - date\_years\_difference, 98
    - función @TODAY, 98
  - funciones de fecha y hora
    - datetime\_date, 98
    - datetime\_day, 98
    - datetime\_day\_name, 98
    - datetime\_day\_short\_name, 98
    - datetime\_hour, 98
    - datetime\_in\_seconds, 98
    - datetime\_minute, 98
    - datetime\_month, 98
    - datetime\_month\_name, 98
    - datetime\_month\_short\_name, 98
    - datetime\_now datetime\_second, 98
    - datetime\_time, 98
    - datetime\_timestamp, 98
    - datetime\_weekday, 98
    - datetime\_year, 98
  - funciones de hora, 79–80
    - time\_before, 86, 98
    - time\_hours\_difference, 98
    - time\_in\_hours, 98
    - time\_in\_mins, 98
    - time\_in\_secs, 98
    - time\_mins\_difference, 98
    - time\_secs\_difference, 98
  - funciones de hora y fecha, 79–80
  - funciones de información, 84
  - funciones de probabilidad, 91
  - funciones de secuencia, 103, 105
  - funciones especiales, 110
  - funciones globales, 108
  - funciones if, then, else, 88
  - funciones lógicas, 88
  - funciones numéricas, 89
  - funciones trigonométricas, 90
- Generador de expresiones
- búsqueda y reemplazo de texto, 17
- IBM SPSS Collaboration and Deployment Services Repository
- argumentos de la línea de comandos, 73
  - procesos, 59
- IBM SPSS Modeler, 1
- documentación, 4
  - ejecución desde la línea de comandos, 70
- Identificadores de nodos
- referencias en los procesos, 21
- informes
- creación y uso de procesos, 64, 67
- interrupción de procesos, 16
- introducción, 76
- línea de comandos
- ejecución de IBM SPSS Modeler, 70
  - inicio de IBM SPSS Modeler, 70
  - lista de argumentos, 71, 73
  - parámetros, 75
  - procesos, 62
  - varios argumentos, 71
- listas, 76, 78
- literales
- procesos, 20, 30
- marcas
- argumentos de la línea de comandos, 70
  - combinación de varias marcas, 71
- marcas comerciales, 322
- modelado de base de datos, 263
- modelos
- exportación, 46
  - nombres de procesos, 43, 45
  - procesos, 46

- modelos a priori
  - propiedades de procesos de nodos, 197, 253
- Modelos Apriori de Oracle
  - propiedades de procesos de nodos, 273, 275
- modelos autonuméricos
  - propiedades de procesos de nodos, 203
- Modelos autonuméricos
  - propiedades de procesos de nodos, 255
- Modelos bayesianos adaptativos de Oracle
  - propiedades de procesos de nodos, 269, 274
- Modelos bayesianos Naive de IBM ISW
  - propiedades de procesos de nodos, 279–280
- Modelos bayesianos Naive de Netezza
  - propiedades de procesos de nodos, 289
- Modelos bayesianos Naive de Oracle
  - propiedades de procesos de nodos, 269, 274
- modelos bietápicos
  - propiedades de procesos de nodos, 252, 262
- modelos C5.0
  - propiedades de procesos de nodos, 206, 255
- modelos CARMA
  - propiedades de procesos de nodos, 207, 256
- modelos CHAID
  - propiedades de procesos de nodos, 211, 256
- modelos de árbol C&R
  - propiedades de procesos de nodos, 208, 256
- Modelos de árbol de decisión de IBM ISW
  - propiedades de procesos de nodos, 275, 280
- Modelos de árbol de decisión de Oracle
  - propiedades de procesos de nodos, 271, 274
- modelos de árboles de decisión de Netezza
  - propiedades de procesos de nodos, 282, 289
- Modelos de árboles de regresión de Netezza
  - propiedades de procesos de nodos, 285, 289
- Modelos de asociación de IBM ISW
  - propiedades de procesos de nodos, 276, 280
- modelos de Autoconglomerado
  - propiedades de procesos de nodos, 254
- Modelos de bayesiano Naive de Netezza
  - propiedades de procesos de nodos, 283
- Modelos de clasificador automático
  - propiedades de procesos de nodos, 254
- Modelos de conglomerado de IBM ISW
  - propiedades de procesos de nodos, 278, 280
- Modelos de conglomerado divisivo de Netezza
  - propiedades de procesos de nodos, 284, 289
- modelos de detección de anomalías
  - propiedades de procesos de nodos, 196, 253
- modelos de IBM DB2
  - propiedades de procesos de nodos, 275
- Modelos de IBM SPSS Statistics
  - propiedades de procesos de nodos, 316
- modelos de K-medias
  - propiedades de procesos de nodos, 228, 258
- modelos de K-medias de Netezza
  - propiedades de procesos de nodos, 283, 289
- Modelos de K-medias de Oracle
  - propiedades de procesos de nodos, 272, 274
- modelos de la máquina de vectores de soporte
  - propiedades de procesos de nodos, 248, 262
- modelos de listas de decisiones
  - propiedades de procesos de nodos, 215, 257
- Modelos de máquinas de vectores de soporte de Oracle
  - propiedades de procesos de nodos, 270, 274
- modelos de Microsoft
  - propiedades de procesos de nodos, 264, 266
- Modelos de Netezza
  - propiedades de procesos de nodos, 281
- Modelos de NMF de Oracle
  - propiedades de procesos de nodos, 272, 274
- modelos de Oracle
  - propiedades de procesos de nodos, 268
- modelos de red bayesiana
  - propiedades de procesos de nodos, 204
- Modelos de red bayesiana de Netezza
  - propiedades de procesos de nodos, 283, 289
- modelos de red neuronal
  - propiedades de procesos de nodos, 238, 260
- Modelos de redes bayesianas
  - propiedades de procesos de nodos, 255
- Modelos de regresión de Cox
  - propiedades de procesos de nodos, 213, 257
- Modelos de regresión de IBM ISW
  - propiedades de procesos de nodos, 277, 280
- modelos de regresión lineal
  - propiedades de procesos de nodos, 244, 261
- Modelos de regresión lineal de Netezza
  - propiedades de procesos de nodos, 286, 289
- modelos de regresión logística
  - propiedades de procesos de nodos, 233, 259
- Modelos de regresión logística de IBM ISW
  - propiedades de procesos de nodos, 279, 281
- modelos de respuesta de autoaprendizaje
  - propiedades de procesos de nodos, 247, 261
- Modelos de secuencia de IBM ISW
  - propiedades de procesos de nodos, 276, 280
- modelos de secuencias
  - propiedades de procesos de nodos, 246, 262
- modelos de selección de características
  - aplicación, 14
  - procesos, 14
  - propiedades de procesos de nodos, 220, 258
- modelos de series temporales
  - propiedades de procesos de nodos, 249, 262
- Modelos de series temporales de IBM ISW
  - propiedades de procesos de nodos, 279
- Modelos de series temporales de Netezza
  - propiedades de procesos de nodos, 286
- modelos del vecino más próximo
  - propiedades de procesos de nodos, 229
- modelos discriminantes
  - propiedades de procesos de nodos, 216, 257

- modelos generados
  - nombres de procesos, 43, 45
- Modelos GLMM
  - propiedades de procesos de nodos, 225
- modelos KNN
  - propiedades de procesos de nodos, 259
- Modelos KNN de Netezza
  - propiedades de procesos de nodos, 284, 289
- modelos kohonen
  - propiedades de procesos de nodos, 231
- modelos Kohonen
  - propiedades de procesos de nodos, 259
- modelos lineales
  - propiedades de procesos de nodos, 232, 259
- modelos lineales generalizados
  - propiedades de procesos de nodos, 222, 258
- Modelos lineales generalizados de Netezza
  - propiedades de procesos de nodos, 287
- Modelos lineales generalizados de Oracle
  - propiedades de procesos de nodos, 271
- Modelos Oracle AI
  - propiedades de procesos de nodos, 273
- Modelos para LMD de Oracle
  - propiedades de procesos de nodos, 273, 275
- modelos PCA
  - propiedades de procesos de nodos, 218, 258
- Modelos PCA de Netezza
  - propiedades de procesos de nodos, 285, 289
- modelos PCA/Factorial
  - propiedades de procesos de nodos, 218, 258
- modelos QUEST
  - propiedades de procesos de nodos, 242, 261
- modelos SLRM
  - propiedades de procesos de nodos, 247, 261
- modelos SVM
  - propiedades de procesos de nodos, 248
  
- Nodo Adición de RFM
  - propiedades, 146
- nodo Agregar
  - propiedades, 142
- nodo Añadir
  - propiedades, 142
- Nodo Análisis
  - propiedades, 290
- Nodo Análisis de RFM
  - propiedades, 168
- nodo Anonimizar
  - propiedades, 152
- nodo Archivo fijo
  - propiedades, 133
- nodo Archivo plano
  - propiedades, 312
- nodo Archivo var.
  - propiedades, 137
- Nodo Auditar datos
  - propiedades, 291
  
- Nodo Autoconglomeración
  - propiedades de procesos de nodos, 201
- nodo Base de datos
  - propiedades, 127
- Nodo Clasificador automático
  - propiedades de procesos de nodos, 199
- nodo Colección
  - propiedades, 181
- Nodo Conjunto
  - propiedades, 160
- nodo de datos de usuario
  - propiedades, 136
- Nodo de exportación Excel
  - propiedades, 311
- nodo de exportación IBM SPSS Data Collection
  - propiedades, 310
- nodo de exportación IBM SPSS Statistics
  - propiedades, 317
- nodo de exportación XML
  - propiedades, 313
- Nodo de fuente de Excel
  - propiedades, 131
- Nodo de fuente de IBM Cognos BI
  - propiedades, 125
- nodo de fuente IBM SPSS Data Collection
  - propiedades, 129
- Nodo de fuente SAS
  - propiedades, 135
- Nodo de fuente XML
  - propiedades, 140
- nodo de origen IBM SPSS Statistics
  - propiedades, 315
- nodo de transformación
  - propiedades, 302
- Nodo de transformación de IBM SPSS Statistics
  - propiedades, 315
- nodo Derivar
  - propiedades, 158
- nodo Distinguir
  - propiedades, 144
- nodo Distribución
  - propiedades, 182
- nodo Enterprise View
  - propiedades, 132
- nodo Equilibrar
  - propiedades, 143
- nodo Estadísticos
  - propiedades, 298
- nodo Evaluación
  - propiedades, 183
- nodo Exportar base de datos
  - propiedades, 305
- Nodo Exportar SAS
  - propiedades, 312
- nodo Filtro
  - propiedades, 162

- nodo Fundir
  - propiedades, 145
- nodo G. múltiple
  - propiedades, 188
- nodo Gráfico
  - propiedades, 189
- nodo Gráfico de tiempo
  - propiedades, 191
- nodo Histograma
  - propiedades, 187
- nodo Histórico
  - propiedades, 163
- nodo Informe, 64, 67
  - propiedades, 296
- nodo Intervalos
  - propiedades, 156
- nodo Intervalos de tiempo
  - propiedades, 170
- nodo Malla
  - propiedades, 192
- nodo Malla direccional
  - propiedades, 192
- Nodo Marcas
  - propiedades, 169
- nodo Matriz
  - propiedades, 293
- nodo Medias
  - propiedades, 295
- nodo Muestrear
  - propiedades, 148
- nodo Ordenar
  - propiedades, 150
- nodo Partición
  - propiedades, 164
- nodo Reclasificar
  - propiedades, 165
- nodo Reestructurar
  - propiedades, 167
- nodo Rellenar
  - propiedades, 161
- nodo Reordenar
  - propiedades, 166
- nodo Reorg. campos
  - propiedades, 166
- Nodo Resultados de IBM SPSS Statistics
  - propiedades, 317
- nodo Seleccionar
  - propiedades, 150
- nodo Tabla
  - propiedades, 300
- Nodo Tablero
  - propiedades, 185
- nodo Tipo
  - propiedades, 175
- nodo Transponer
  - propiedades, 174
- nodo Val. globales
  - propiedades, 297
- nodos
  - recorrido en bucle en procesos, 58
- nodos de exportación
  - propiedades de procesos de nodos, 304
- nodos de fuente
  - propiedades, 123
- nodos de gráficos
  - propiedades de los procesos, 180
- nodos de modelado
  - propiedades de procesos de nodos, 195
- nodos de resultados
  - propiedades de los procesos, 290
- nombres de campos
  - cambio entre minúsculas y mayúsculas, 58
- nombres de rutas
  - acceso en procesos, 68
- nugget
  - propiedades de procesos de nodos, 253
- nuggets de modelo
  - nombres de procesos, 43, 45
  - propiedades de procesos de nodos, 253
- números, 77
- O-conglomerado de Oracle
  - propiedades de procesos de nodos, 272, 274
- objeto actual
  - referencias en los procesos, 23
- objetos de archivo
  - comandos de procesos, 54
- objetos de nodo
  - comandos de procesos, 37
  - procesos, 21
- objetos de resultado
  - comandos de procesos, 54
- objetos de resultados
  - comandos de procesos, 55
  - nombres de procesos, 55
- objetos de ruta
  - apertura, 24
  - comandos de procesos, 49
  - referencia, 24
- objetos del modelo
  - comandos de procesos, 43
  - nombres de procesos, 43, 45
- objetos state
  - comandos de procesos, 53
- operador and, 88
- operador equals, 86
- operador greater than, 86
- operador less than, 86
- operador not, 88
- operador not equal, 86
- operador or, 88
- operadores
  - procesos, 28



- unión de cadenas, 68, 85
- orden de ejecución
  - modificación con procesos, 58
- orden de ejecución de rutas
  - modificación con procesos, 58
- palabra clave generada, 62
- parámetros, 15, 33, 114–116, 119
  - procesos, 20, 28
  - ruta, 26
  - sesión, 26
  - Supernodos, 319
- parámetros de lista
  - modificación en los procesos, 28
- parámetros de propiedades, 15, 33, 114, 117
- parámetros de ruta, 26, 33
- parámetros de sesión, 26, 33
- preparación automática de datos
  - propiedades, 153
- prioridad, 80
- prioridad del operador, 80
- procesos
  - abreviaturas utilizadas, 116
  - almacenamiento, 10
  - búsqueda y reemplazo de texto, 17
  - comentarios, 29
  - compatibilidad con versiones anteriores, 62
  - comprobación de errores, 61
  - conceptos básicos, 9, 20
  - continuaciones, 29
  - desde la línea de comandos, 62
  - ejecución, 16
  - ejecución de procesos, 28
  - ejemplos, 64, 67
  - en Supernodos, 15
  - expresiones CLEM, 28
  - importación desde archivos de texto, 10
  - interfaz de usuario, 10, 12, 15
  - interrupción, 16
  - modelos de selección de características, 14
  - nodos, 21
  - nodos de gráficos, 180
  - nodos de resultados, 290
  - objeto actual, 23
  - operadores, 28
  - orden de ejecución de rutas, 58
  - Procesos de Supernodo, 9
  - procesos independientes, 9
  - propiedades comunes, 117
  - rutas, 9
  - syntax, 20
- procesos independientes, 9, 12
- propiedad `column_count`, 54
- propiedad de valores, 66
- propiedad `row_count`, 54
- propiedad `stream.nodes`, 58
- propiedades, 33
  - nodos de modelado de base de datos, 263
  - nodos filtro, 115
  - procesos, 114–117, 195, 253, 304
  - procesos comunes, 117
  - proyectos, 122
  - ruta, 119
  - Supernodos, 319
- propiedades `autodatapreprenode`, 153
- propiedades de `aggreatenode`, 142
- propiedades de `analysisnode`, 290
- propiedades de `anomalydetectionnode`, 196
- propiedades de `anonymizenode`, 152
- propiedades de `appendnode`, 142
- propiedades de `applyanomalydetectionnode`, 253
- propiedades de `applyapriorinode`, 253
- propiedades de `applyautoclassifiernode`, 254
- propiedades de `applyautoclusternode`, 254
- propiedades de `applyautonumericnode`, 255
- propiedades de `applybayesnetnode`, 255
- propiedades de `applyc50node`, 255
- propiedades de `applycarmanode`, 256
- propiedades de `applycartnode`, 256
- propiedades de `applychaidnode`, 256
- propiedades de `applycoxregnode`, 257
- propiedades de `applydb2imclusternode`, 280
- propiedades de `applydb2imlognode`, 281
- propiedades de `applydb2imnbnode`, 280
- propiedades de `applydb2imregnode`, 280
- propiedades de `applydb2imtreenode`, 280
- propiedades de `applydecisionlistnode`, 257
- propiedades de `applydiscriminantnode`, 257
- propiedades de `applyfactornode`, 258
- propiedades de `applyfeatureselectionnode`, 258
- propiedades de `applygeneralizedlinearnode`, 258
- propiedades de `applykmeansnode`, 258
- propiedades de `applyknnnode`, 259
- propiedades de `applykohonenode`, 259
- Propiedades de `applylinearnode`, 259
- propiedades de `applylogregnode`, 259
- propiedades de `applymslogisticnode`, 267
- propiedades de `applymsneuralnetworknode`, 267
- propiedades de `applymsregressionnode`, 267
- propiedades de `applymssequenceclusternode`, 268
- propiedades de `applymstimeseriesnode`, 268
- propiedades de `applymstreenode`, 267
- propiedades de `applynetezzbayesnode`, 289
- propiedades de `applynetez zadectreenode`, 289
- propiedades de `applynetez zadivclusternode`, 289
- propiedades de `applynetez zakmeansnode`, 289
- propiedades de `applynetez zaknnnode`, 289
- propiedades de `applynetez zalineregressionnode`, 289
- propiedades de `applynetez zanaivebayesnode`, 289
- propiedades de `applynetez zapcanode`, 289
- propiedades de `applynetez zaregtreenode`, 289
- propiedades de `applyneuralnetnode`, 260
- propiedades de `applyneuralnetworknode`, 260

- propiedades de applyoraabnnode, 274
- propiedades de applyoradecisiontreenode, 274
- propiedades de applyorakmeansnode, 274
- propiedades de applyoranbnode, 274
- propiedades de applyoranmfnode, 274
- propiedades de applyoraoclusternode, 274
- propiedades de applyorasvmnode, 274
- propiedades de applyquestnode, 261
- propiedades de applyregressionnode, 261
- propiedades de applyselflearningnode, 261
- propiedades de applysequencenode, 262
- propiedades de applysvmnode, 262
- propiedades de applytimeseriesnode, 262
- propiedades de applytwestepnode, 262
- propiedades de apriorinode, 197
- propiedades de autotransformnode, 199
- propiedades de autonumericnode, 203
- propiedades de balancenode, 143
- propiedades de bayesnet, 204
- propiedades de binningnode, 156
- propiedades de c50node, 206
- propiedades de carmanode, 207
- propiedades de cartnode, 208
- propiedades de chaidnode, 211
- propiedades de cognosimportnode, 125
- propiedades de collectionnode, 181
- propiedades de coxregnode, 213
- propiedades de dataauditnode, 291
- propiedades de databaseexportnode, 305
- propiedades de databasenode, 127
- propiedades de datacollectionexportnode, 310
- propiedades de datacollectionimportnode, 129
- propiedades de db2imassocnode, 276
- propiedades de db2imclusternode, 278
- propiedades de db2imlognode, 279
- propiedades de db2imnbnode, 279
- propiedades de db2imregnode, 277
- propiedades de db2imsequencenode, 276
- propiedades de db2imtimeseriesnode, 279
- propiedades de db2imtreenode, 275
- propiedades de decisionlist, 215
- propiedades de derivenode, 158
- propiedades de directedwebnode, 192
- propiedades de discriminantnode, 216
- propiedades de distinctnode, 144
- propiedades de distributionnode, 182
- propiedades de ensemblenode, 160
- propiedades de evaluationnode, 183
- propiedades de evimportnode, 132
- Propiedades de excelexportnode, 311
- propiedades de excelimportnode, 131
- propiedades de factornode, 218
- propiedades de featureselectionnode, 14, 220
- propiedades de fillernode, 161
- propiedades de filternode, 162
- propiedades de fixedfilenode, 133
- propiedades de flatfilenode, 312
- propiedades de genlinnode, 222
- Propiedades de glmnode, 225
- propiedades de graphboardnode, 185
- propiedades de histogramnode, 187
- propiedades de historynode, 163
- propiedades de kmeansnode, 228
- propiedades de knnnode, 229
- propiedades de kohonenode, 231
- propiedades de logregnode, 233
- propiedades de los nodos
  - acceso en procesos, 68
- propiedades de matrixnode, 293
- propiedades de meansnode, 295
- propiedades de mergenode, 145
- propiedades de msassocnode, 264
- propiedades de msbayesnode, 264
- propiedades de msclusternode, 264
- propiedades de mslogisticnode, 264
- propiedades de msneuralnetworknode, 264
- propiedades de msregressionnode, 264
- propiedades de mssequenceclusternode, 264
- propiedades de mstimeseriesnode, 264
- propiedades de mstreenode, 264
- propiedades de multiplotnode, 188
- propiedades de netezabayesnode, 283
- propiedades de netezadectreenode, 282
- propiedades de netezadivclusternode, 284
- Propiedades de netezzaglmnode, 287
- propiedades de netezzakmeansnode, 283
- propiedades de netezzaknnode, 284
- propiedades de netezalineregressionnode, 286
- propiedades de netezanaivebayesnode, 283
- propiedades de netezapcanode, 285
- propiedades de netezaregtreenode, 285
- Propiedades de netezatimeseriesnode, 286
- propiedades de neuralnetnode, 238
- Propiedades de neuralnetwork, 240
- propiedades de nodo de autoconglomeración, 201
- propiedades de numericpredictornode, 203
- propiedades de oraabnnode, 269
- propiedades de oraainode, 273
- propiedades de oraapriorinode, 273
- propiedades de oradecisiontreenode, 271
- propiedades de oraglmnode, 271
- propiedades de orakmeansnode, 272
- propiedades de oramdlnode, 273
- propiedades de oranbnode, 269
- propiedades de oranmfnode, 272
- propiedades de oraoclusternode, 272
- propiedades de orasvmnode, 270
- propiedades de outputfilenode, 312
- propiedades de partitionnode, 164
- propiedades de plotnode, 189
- propiedades de procesos de nodos, 263
  - nodos de exportación, 304
  - nodos de modelado, 195
  - nuggets de modelo, 253

- propiedades de questnode, 242
- propiedades de reclassifynode, 165
- propiedades de regressionnode, 244
- propiedades de reordernode, 166
- propiedades de reportnode, 296
- propiedades de restructurenode, 167
- propiedades de rfmanalysisnode, 168
- propiedades de ruta, 68
- propiedades de samplenode, 148
- propiedades de sasexportnode, 312
- propiedades de sasimportnode, 135
- propiedades de selectnode, 150
- propiedades de sequencenode, 246
- propiedades de setglobalsnode, 297
- propiedades de settoflagnode, 169
- propiedades de slrmnode, 247
- propiedades de sortnode, 150
- propiedades de statisticsexportnode, 317
- propiedades de statisticsimportnode, 14, 315
- propiedades de statisticsmodelnode, 316
- propiedades de statisticsnode, 298
- propiedades de statisticsoutputnode, 317
- propiedades de statisticstransformnode, 315
- propiedades de svmnode, 248
- propiedades de tablenode, 300
- propiedades de timeintervalnode, 170
- propiedades de timeplotnode, 191
- propiedades de timeseriesnode, 249
- propiedades de transformnode, 302
- propiedades de transposenode, 174
- propiedades de twostepnode, 252
- propiedades de typenode, 14, 65, 175
- propiedades de userinputnode, 136
- propiedades de variablefilenode, 137
- propiedades de webnode, 192
- propiedades de xmlexportnode, 313
- propiedades de xmlimportnode, 140
- propiedades estructuradas, 115
- propiedades lineales, 232
- propiedades rfmaggregatenode, 146
- proyectos
  - propiedades, 122
- reales, 76–77
- Red neuronal de MS
  - propiedades de procesos de nodos, 264, 267
- redes neuronales
  - propiedades de procesos de nodos, 240, 260
- reemplazo de texto, 17
- Regresión lineal de MS
  - propiedades de procesos de nodos, 264, 267
- Regresión logística de MS
  - propiedades de procesos de nodos, 264, 267
- resultado en formato HTML
  - creación y uso de procesos, 64, 67
- rutas
  - comando de conjunto múltiple, 114
  - procesos, 9–10
  - propiedades, 119
- seguridad
  - contraseñas codificadas, 61, 71
- Serie temporales de MS
  - propiedades de procesos de nodos, 268
- servidor
  - argumentos de la línea de comandos, 71
- sintaxis de intercalación
  - referencias de variables, 21, 25
- sistema
  - argumentos de la línea de comandos, 73
- SPSS Modeler Server, 2
- supernodo, 114
- Supernodo
  - parámetros, 26, 33
- Supernodos
  - configuración de propiedades, 319
  - parámetros, 319
  - procesos, 9, 15–16, 319
  - propiedades, 319
- tratamiento de los elementos vacíos
  - Funciones de CLEM, 109
- variables, 25, 33
  - procesos, 20, 23
  - referencias de nodos, 21
- variables especiales, 23
- variables locales, 25, 33