

*IBM SPSS Modeler 18.2.1 Solution  
Publisher*

**IBM**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos" en la página 37.

**Información sobre el producto**

Esta edición se aplica a la versión 18, release 2, modificación 0 de IBM SPSS Modeler y a todos los releases y las modificaciones posteriores, hasta que se indique lo contrario en nuevas ediciones.

---

# Contenido

|  |           |   |           |
|--|-----------|---|-----------|
| <b>Prefacio</b> . . . . .  | <b>v</b>  | <b>Capítulo 4. Instalación de IBM SPSS Modeler Solution Publisher Runtime</b> . . . . .                     | <b>17</b> |
| <b>Capítulo 1. Acerca de IBM SPSS Modeler</b> <b>1</b>   |           | Instalación de Runtime . . . . .  | 17        |
| Productos IBM SPSS Modeler . . . . .   | 1         | Instalación en Windows . . . . .  | 17        |
| IBM SPSS Modeler . . . . .   | 1         | Instalación en UNIX . . . . .   | 17        |
| IBM SPSS Modeler Server . . . . .  | 1         | Resolución de problemas de instalación . . . . .  | 19        |
| Consola de administración de IBM SPSS Modeler . . . . .  | 2         | Inicio de Runtime . . . . .   | 19        |
| IBM SPSS Modeler Batch . . . . .   | 2         | Modificación del directorio temporal . . . . .  | 20        |
| IBM SPSS Modeler Solution Publisher . . . . .  | 2         | Desinstalación de IBM SPSS Modeler Solution Publisher Runtime . . . . .                                     | 20        |
| Adaptadores de IBM SPSS Modeler Server para IBM SPSS Collaboration and Deployment Services . . . . . | 2         | Procedimiento de desinstalación para Windows . . . . .  | 20        |
| Ediciones de IBM SPSS Modeler . . . . .  | 2         | Procedimiento de desinstalación para UNIX . . . . .   | 20        |
| Documentación . . . . .  | 3         | <b>Capítulo 5. API de IBM SPSS Embedded Predictive Modeling</b> . . . . .                                   | <b>21</b> |
| Documentación de SPSS Modeler Professional . . . . .   | 3         | <b>Capítulo 6. Referencia API de la biblioteca de IBM SPSS Modeler Solution Publisher Runtime</b> . . . . . | <b>23</b> |
| Documentación de SPSS Modeler Premium . . . . .  | 4         | Introducción . . . . .  | 23        |
| Ejemplos de aplicación . . . . .   | 4         | Conceptos básicos del proceso de API . . . . .  | 23        |
| Carpeta Demos . . . . .  | 4         | Funciones API . . . . .   | 24        |
| Rastreo de licencias . . . . .   | 5         | <b>Avisos</b> . . . . .   | <b>37</b> |
| <b>Capítulo 2. IBM SPSS Modeler Solution Publisher</b> . . . . .                                     | <b>7</b>  | Marcas comerciales . . . . .  | 38        |
| Modo de funcionamiento de IBM SPSS Modeler Solution Publisher . . . . .                              | 7         | Términos y condiciones para la documentación del producto . . . . .   | 39        |
| Publicación de rutas . . . . .   | 8         | <b>Índice</b> . . . . .   | <b>41</b> |
| <b>Capítulo 3. Ejecución de rutas publicadas</b> . . . . .   | <b>11</b> |   |           |
| Uso del archivo de parámetros para personalizar la ejecución de una ruta . . . . .                   | 12        |   |           |
| Incrustación de IBM SPSS Modeler Solution Publisher en las aplicaciones . . . . .                    | 14        |   |           |
| Aplicación de muestra de IBM SPSS Modeler Solution Publisher . . . . .                               | 15        |   |           |



---

## Prefacio

IBM® SPSS Modeler es el entorno de trabajo de minería de datos orientado a las empresas de IBM Corp.. 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 los usuarios a aplicar su experiencia empresarial específica, 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 predicciones, 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 confían los encargados de la toma de decisiones para mejorar el rendimiento comercial. Un conjunto integral de inteligencia empresarial, análisis predictivo, rendimiento financiero y gestión de estrategias y aplicaciones de análisis que ofrece una perspectiva clara, inmediata e interactiva del rendimiento actual y la capacidad de 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, automatizar las decisiones de forma fiable 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 IBM SPSS como ventaja ante la competencia para atraer, retener y hacer crecer a los clientes, reduciendo al mismo tiempo el fraude y 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 <https://www.ibm.com/mysupport/s/>.

### Asistencia técnica

Hay asistencia técnica disponible para los clientes de mantenimiento. 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 técnica, consulte el sitio web de IBM Corp. en <https://www.ibm.com/mysupport/s/>. Tenga a mano su acuerdo de asistencia y esté preparado para identificarse a sí mismo y a su organización al solicitar ayuda.



---

## Capítulo 1. 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 desplegarlos en operaciones empresariales para mejorar la toma de decisiones. Con un diseño que sigue el modelo CRISP-DM, estándar del sector, IBM SPSS Modeler admite el proceso completo de minería de datos, desde los propios datos hasta obtener los mejores resultados empresariales.

IBM 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. Sin desea más información, consulte <https://www.ibm.com/analytics/us/en/technology/spss/>.

---

### 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
- Consola de administración de IBM SPSS Modeler (incluido con el Gestor de despliegue de IBM SPSS)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- Adaptadores de IBM SPSS Modeler Server 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. Consulte el tema “Ediciones de IBM SPSS Modeler” en la página 2 para obtener más información.

### 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 esta forma, SPSS Modeler Server ofrece un rendimiento superior en conjuntos de datos grandes, ya que las operaciones que requieren mucha memoria se pueden realizar en el servidor sin tener que descargar datos en el equipo cliente. IBM SPSS Modeler Server también proporciona soporte para las funciones de optimización de SQL y modelado en la base de datos, lo que ofrece ventajas adicionales de rendimiento y automatización.

## **Consola de administración de IBM SPSS Modeler**

El Consola de administración de Modeler región propietaria del archivos una interfaz gráfica de usuario para gestionar muchas de las opciones de configuración de SPSS Modeler Server, que también se pueden configurar a través de un archivo de opciones. La consola se incluye en el Gestor de despliegue de IBM SPSS, se puede utilizar para supervisar y configurar las instalaciones de SPSS Modeler Server y está disponible de forma 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 del usuario. SPSS Modeler Batch es una versión especial del producto que proporciona soporte para todas las prestaciones de análisis de SPSS Modeler sin acceso a la interfaz de usuario habitual. SPSS Modeler Server debe 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 desplegar 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 - Puntuación, para el que se necesita una licencia independiente. Con esta licencia, recibirá SPSS Modeler Solution Publisher Runtime, que le permite ejecutar las rutas publicadas.

Para obtener más información sobre SPSS Modeler Solution Publisher, consulte la documentación de IBM SPSS Collaboration and Deployment Services. El Knowledge Center de IBM SPSS Collaboration and Deployment Services contiene secciones denominadas "IBM SPSS Modeler Solution Publisher" e "IBM SPSS Analytics Toolkit."

## **Adaptadores de IBM SPSS Modeler Server 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 desplegada 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 trabajar con datos especializados y con datos de texto no estructurado. SPSS Modeler Premium incluye IBM SPSS Modeler Text Analytics:

**IBM SPSS Modeler Text Analytics** 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 IBM SPSS Modeler para tomar decisiones mejores y más certeras.

## IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription proporciona todas las mismas prestaciones de análisis predictivo que el cliente tradicional de IBM SPSS Modeler. Con la edición Subscription, puede descargar actualizaciones de producto con regularidad.

---

## Documentación

La documentación está disponible desde el menú Ayuda en SPSS Modeler. Así se abre el Knowledge Center, que está disponible de forma pública fuera del producto.

También está disponible documentación completa para cada producto (incluyendo instrucciones de instalación) en formato PDF, en una carpeta comprimida separada, como parte de la descarga del producto. O bien, los documentos PDF se pueden descargar de la web en <http://www.ibm.com/support/docview.wss?uid=swg27046871>.

## Documentación de SPSS Modeler Professional

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

- **Guía del usuario de IBM SPSS Modeler.** Introducción general para utilizar SPSS Modeler, incluyendo cómo crear corrientes de datos, manejar valores perdidos, crear expresiones de CLEM, trabajar con proyectos e informes y empaquetar corrientes para el despliegue en IBM SPSS Collaboration and Deployment Services o IBM SPSS Modeler Advantage.
- **Nodos Origen, 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 gran variedad de métodos de modelado procedentes del aprendizaje automático, la inteligencia artificial y el estadístico.
- **Guía de aplicaciones de IBM SPSS Modeler.** Los ejemplos de esta guía ofrecen introducciones breves y concisas a métodos y técnicas de modelado específicos. También está disponible una versión en línea de esta guía desde el menú Ayuda. Consulte el tema “Ejemplos de aplicación” en la página 4 si desea más información.
- **Automatización y scripts Python IBM SPSS Modeler.** Información sobre la automatización del sistema mediante scripts de Python, incluidas las propiedades que se pueden utilizar para manipular nodos y rutas.
- **Guía de despliegue de IBM SPSS Modeler.** La información sobre cómo ejecutar rutas de IBM SPSS Modeler como pasos en el proceso de trabajos en el Gestor de despliegue de IBM SPSS.

- **Guía del desarrollador de IBM SPSS Modeler CLEF.** CLEF permite integrar programas de terceros, tales como rutinas de proceso de datos o algoritmos de modelado, como nodos en IBM SPSS Modeler.
- **Guía de la minería interna de base de datos 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.
- **Guía del usuario del Gestor de despliegue de IBM SPSS.** Información sobre cómo utilizar la interfaz de usuario de la consola de administración incluida en la aplicación Gestor de despliegue para supervisar y configurar IBM SPSS Modeler Server.
- **Guía 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.
- **Guía 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.

- **Guía del usuario de SPSS Modeler Text Analytics .** 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.

---

## Ejemplos de aplicación

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 analistas de datos, pero los conceptos y métodos implicados son escalables a aplicaciones del mundo real.

Para acceder a los ejemplos, pulse **Ejemplos de aplicación** en el menú Ayuda en SPSS Modeler.

Los archivos de datos y rutas de ejemplo se instalan en la carpeta Demos en el directorio de instalación del producto. Si desea obtener más información, consulte "Carpeta Demos".

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

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

---

## Carpeta Demos

Los archivos de datos y las corrientes de muestras que se utilizan con los ejemplos de aplicación se instalan en la carpeta Demos en el directorio de instalación del producto (por ejemplo: C:\Archivos de programa\IBM\SPSS\Modeler\<<versión>\Demos). También se puede acceder a esta carpeta desde el grupo de programas de IBM SPSS Modeler en el menú Inicio de Windows, o pulsando Demos en la lista de directorios recientes en el recuadro de diálogo **Archivo > Abrir ruta**.

---

## Rastreo de licencias

Cuando se utiliza SPSS Modeler, el uso de las licencias se rastrea y se registra a intervalos regulares. Las métricas de licencia que se registran son *AUTHORIZED\_USER* y *CONCURRENT\_USER*, y el tipo de métrica que se registra depende del tipo de licencia que tiene para SPSS Modeler.

IBM License Metric Tool puede procesar los archivos de registro que se generan, a partir de los cuales puede crear informes de uso de licencia.

Los archivos de registro de licencia se crean en el mismo directorio donde se registran los archivos de registro del cliente SPSS Modeler (de forma predeterminada, %ALLUSERSPROFILE%/IBM/SPSS/Modeler/<versión>/log).



---

## Capítulo 2. IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher es una potente herramienta de distribución para la integración de resultados de minería de datos en el proceso empresarial para resolver problemas reales. Con IBM SPSS Modeler Solution Publisher, puede crear una versión "empaquetada" de una ruta que se puede ejecutar en un motor de tiempo de ejecución externo o incrustado en una aplicación externa. Esto le permite desplegar las rutas de modelado de datos en un entorno de producción para admitir los procesos empresariales de cada día y proporcionar a las personas encargadas de la toma de decisiones de la organización el conocimiento obtenido a partir de la minería de sus datos.

Utilizar IBM SPSS Modeler Solution Publisher proporciona mayor potencia que simplemente exportar el modelo (como PMML), ya que le permite publicar y desplegar *rutas completas de IBM SPSS Modeler*. Esto significa que puede llevar a cabo preparaciones de datos, así como operaciones con registros y campos, como agregar datos, seleccionar registros o derivar nuevos campos antes de crear predicciones basadas en un modelo. De este modo, puede procesar con mayor detalle los resultados de los modelos antes de guardar los datos, y todo simplemente ejecutando la ruta publicada.

**Nota:** La publicación se consigue utilizando la pestaña Publicar del nodo de exportación apropiado. Consulte el tema "Publicación de rutas" en la página 8 para obtener más información. Las rutas guardadas en versiones anteriores a la 11.0 del producto empleaban un nodo Publicador separado. El nodo aún se utiliza para la compatibilidad con versiones anteriores, pero ya no está disponible en la paleta del nodo.

### Licencias de IBM SPSS Modeler Solution Publisher

**Nota:** IBM SPSS Modeler Solution Publisher se distribuye como parte del servicio de puntuación IBM SPSS Collaboration and Deployment Services, para el cual es necesaria una licencia separada.  
<http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Cuando adquiere una licencia, se proporciona el complemento independiente IBM SPSS Modeler Solution Publisher Runtime Runtime que permite ejecutar las rutas publicadas. Consulte el tema "Instalación de Runtime" en la página 17 para obtener más información.

### Registro

El archivo de registro está controlador por el archivo de configuración `log4cxx.properties`, que se encuentra en la carpeta `config` del directorio de instalación de IBM SPSS Modeler Solution Publisher. Puede modificar el archivo de configuración `log4cxx.properties` para personalizar la salida del registro.

Tenga en cuenta que si utiliza Linux/UNIX, para que la aplicación pueda encontrar el archivo de configuración, debe definir una variable de entorno antes de ejecutar la aplicación (por ejemplo, `export MODELERRUNTIME=DIRECTORIO_INSTALACIÓN_MSP`).

---

### Modo de funcionamiento de IBM SPSS Modeler Solution Publisher

El despliegue de una solución utilizando IBM SPSS Modeler Solution Publisher consta de dos fases: *publicar* una ruta y *ejecutar* una serie.

**Publicación.** Durante el proceso de minería de datos, al final se llega a un modelo que proporciona una buena solución para su problema empresarial. Llegado a ese punto, podrá tomar dicho modelo y aplicarlo a su proceso empresarial. Al publicar una ruta, se guarda en el disco una descripción detallada de la ruta (como un archivo de *imagen* y un archivo de *parámetros*). Consulte el tema "Publicación de rutas" en la página 8 para obtener más información.

**Ejecución.** Una vez publicada la ruta, puede volver a crear el proceso implementado en la ruta simplemente ejecutando la ruta publicada. Esto se puede hacer bien usando el ejecutable independiente IBM SPSS Modeler Runtime (*modelerrun.exe*) o bien desarrollando una aplicación que utilice la biblioteca de IBM SPSS Modeler Runtime para ejecutar la ruta. Para ejecutar rutas fuera de IBM SPSS Modeler (usando Runtime o una aplicación personalizada), primero debe instalar IBM SPSS Modeler Solution Publisher Runtime. Consulte el tema “Instalación de Runtime” en la página 17 para obtener más información.

---

## Publicación de rutas

La publicación de rutas se realiza directamente desde IBM SPSS Modeler utilizando cualquiera de los nodos de exportación estándares: nodos de base de datos, archivo sin formato, exportación de Estadísticas, exportación de extensión, exportación de recopilación de datos, exportación de SAS, Excel y exportación a XML. El tipo de nodo de exportación determina el formato de los resultados que se deben escribir cada vez que se ejecuta la ruta publicada utilizando IBM SPSS Modeler Solution Publisher Runtime o una aplicación externa. Por ejemplo, si desea escribir los resultados en una base de datos cada vez que se ejecute una ruta publicada, utilice el nodo de exportación de Base de datos.

### Para publicar una ruta

1. Abra o cree una ruta de la forma habitual y conecte un nodo de exportación al final.
2. En la pestaña Publicar del nodo de exportación, especifique un nombre de raíz para los archivos publicados (es decir, el nombre de archivo al que se añadirán las extensiones *.pim*, *.par* y *.xml*).
3. Pulse **Publicar** para publicar la ruta o seleccione **Publicar la ruta** para publicar automáticamente la ruta cada vez que se ejecuta el nodo.

**Nombre publicado.** Especifique el nombre de raíz para la imagen publicada y los archivos de parámetros.

- El **archivo de imagen** (*\*.pim*) proporciona toda la información necesaria para que Runtime ejecute la ruta publicada exactamente tal y como era en el momento de la exportación. Si está seguro de que no necesitará modificar ninguno de los valores de la ruta (como el origen de datos de entrada o el archivo de datos de salida), puede desplegar sólo el archivo de imagen.
- El **archivo de parámetros** (*\*.par*) contiene información configurable sobre orígenes de datos, archivos de resultados y opciones de ejecución. Si desea poder controlar la entrada o salida de la ruta sin volver a publicarla, necesitará el archivo de parámetros además del archivo de imagen.
- El **archivo de metadatos** (*\*.xml*) describe las entradas y los resultados de la imagen y sus modelos de datos. Está diseñado para que lo utilicen aplicaciones que incorporan la biblioteca de tiempo de ejecución y que necesitan conocer la estructura de los datos de entrada y salida.

**Nota:** Este archivo sólo se genera si selecciona la opción **Publicar metadatos**.

**Publicar parámetros.** Si es necesario, puede incluir parámetros de ruta en el archivo *\*.par*. Puede cambiar estos valores de parámetros de ruta cuando ejecute la imagen editando el archivo *\*.par* o a través de la API de tiempo de ejecución.

Esta opción habilita el botón **Parámetros**. Al pulsar el botón, aparece el cuadro de diálogo Publicar parámetros.

Elija los parámetros que desea incluir en la imagen publicada seleccionando la opción pertinente en la columna **Publicar**.

**Al ejecutar la ruta.** Especifica si la ruta se publica automáticamente cuando se ejecuta el nodo.

- **Exportar datos.** Ejecuta el nodo de exportación de la forma habitual, sin publicar la ruta. (Básicamente el nodo se ejecuta en IBM SPSS Modeler de la misma manera que lo haría si IBM SPSS Modeler Solution Publisher no estuviera disponible.) Si selecciona esta opción, la ruta no se publicará a menor

que lo haga explícitamente pulsando **Publicar** en el cuadro de diálogo de nodo de exportación. De forma alternativa, puede publicar la ruta actual utilizando la herramienta Publicar en la barra de herramientas o utilizando un script.

- **Publicar la ruta.** Publica la ruta para su despliegue utilizando IBM SPSS Modeler Solution Publisher. Seleccione esta opción si desea publicar automáticamente la ruta cada vez que se ejecuta.

**Nota:**

- Si planea ejecutar la ruta publicada con datos nuevos o actualizados, es importante tener en cuenta que el orden de los campos del archivo de entrada debe ser el mismo que el orden de los campos del archivo de entrada del nodo de origen especificado en la ruta publicada.
- Cuando se publican datos en aplicaciones externas, es recomendable filtrar los campos no pertinentes o cambiar el nombre de los campos para que cumplan los requisitos de entrada. Ambas acciones se pueden realizar utilizando un nodo Filtrar antes del nodo de exportación.





---

## Capítulo 3. Ejecución de rutas publicadas

El principal efecto de la ejecución de una ruta publicada es siempre la generación de un conjunto de datos. Los datos se pueden guardar en un archivo de disco o en una base de datos. Las rutas publicadas no pueden generar gráficos, tablas, modelos generados u otro resultado que no sea datos.

Las rutas publicadas con IBM SPSS Modeler Solution Publisher se pueden ejecutar con el programa IBM SPSS Modeler Solution Publisher Runtime. El programa Runtime, *modelerrun.exe*, se inicia desde la línea de comando, con opciones que indican la ejecución del archivo de imagen y (opcionalmente) el archivo de parámetros. El comando de Runtime es el siguiente:

```
modelerrun -nobanner -p <archivo de parámetros> -o <opciones> <archivo de imagen>
```

donde las entradas de comando se muestran en la siguiente tabla.

Tabla 1. Entradas de comando de Runtime

| Entrada del comando     | Descripción  |
|-------------------------|--|
| -nobanner               | suprime el mensaje de texto de inicio de Runtime (opcional).   |
| <archivo de parámetros> | es el nombre del archivo de parámetros (opcional).   |
| <options>               | es una lista de opciones separadas por comas que constituyen pares opción=valor y especifican las opciones de ejecución de Runtime. Más adelante se detallan las opciones (opcional).<br><b>Nota:</b> No debe haber espacios vacíos después de una coma. |
| <archivo de imagen>     | es el nombre del archivo de imagen que ejecutar (obligatorio).   |

Las opciones de ejecución disponibles se muestran en la siguiente tabla.

Tabla 2. Opciones válidas de Runtime

| Nombre                | Valor predeterminado | Descripción   |
|-----------------------|----------------------|---|
| max_file_size         | -1                   | Tamaño máximo de los archivos guardados. Un valor de -1 indica que no hay límite.   |
| max_sql_string_length | 2.048                | Longitud máxima de una cadena importada desde la base de datos con SQL. Los valores de cadena con más caracteres se truncan por la derecha sin advertencias. El rango válido es de 1 a 65.535 caracteres. |
| memory_usage          | 100                  | Multiplicador de asignación de la memoria dinámica. Ajuste este valor para regular el uso total de memoria del servidor.  |
| temp_directory        | "                    | Especifica el directorio que utilizar para los archivos temporales.   |
| request_passwords     | Y                    | Controla si se debe pedir una contraseña de base de datos cuando se ejecute una ruta publicada que requiera acceso a una base de datos (Y/N).   |

**Nota:** Al publicar una ruta que contiene un nodo de entrada de usuario, tenga en cuenta que la Aplicación de publicación de la solución SPSS Modeler requiere un archivo de entrada para el nodo (mientras que en SPSS Modeler, defina el campo y sus datos directamente en el nodo). El nombre de archivo predeterminado es *userInput.dat*, y puede cambiarlo en el archivo de parámetros.

**Nota:** Al ejecutar una corriente que contiene un nodo Optimización CPLEX en **SPSS Modeler Solution Publisher**, de forma predeterminada, se utiliza la biblioteca CPLEX incorporada de la edición Community. Tiene una limitación de 1000 variables y 1000 restricciones. Si instala la edición completa de IBM ILOG CPLEX y desea utilizar en su lugar el motor CPLEX de la edición completa, que no tiene dichas limitaciones, complete el paso siguiente para la plataforma.

- En Windows, añada la vía de acceso a biblioteca OPL como un argumento de línea de mandatos para modelerrun.exe. Por ejemplo:

```
-o cplex_opl_lib_path="<CPLEX_path>\opl\bin\<Platform_dir>"
```

Donde <CPLEX\_path> es el directorio de instalación de CPLEX como C:\Archivos de programa\IBM\ILOG\CPLEX\_Studio127, y <Platform\_dir> es el directorio específico de plataforma como x64\_win64.

- En Linux, edite modelerrun y añada la vía de acceso a biblioteca OPL. Por ejemplo:

```
CPLEX_OPL_LIB_PATH=<CPLEX_path>/opl/bin/<Platform_dir>
```

Donde <CPLEX\_path> es el directorio de instalación de CPLEX como /root/Libs\_127\_FullEdition/Linux\_x86\_64, y <Platform\_dir> es el directorio específico de plataforma como x86-64\_linux.

## Ejecución de una ruta de Apache Spark local en IBM SPSS Modeler Solution Publisher

Cuando ejecute una ruta de Spark local en IBM SPSS Modeler Solution Publisher, debe especificar lo siguiente en el comando modelerrun (donde C:/Anaconda2/python.exe es la vía de acceso al ejecutable de Python):

```
-o eas_pyspark_python_path=C:/Anaconda2/python.exe
```

---

## Uso del archivo de parámetros para personalizar la ejecución de una ruta

Cuando se publica una ruta, el archivo de imagen contiene información sobre el origen de datos y el destino de los datos de salida seleccionados en la ruta original. No obstante, a veces resulta conveniente ejecutar una ruta con datos de distinta procedencia o redirigir los datos de salida a un destino diferente. Esto se puede hacer editando el archivo de parámetros de la ruta.

Se trata de una tarea de especial importancia cuando se ejecuta la ruta publicada en un ordenador diferente al que se utilizó para crearla. Deberá actualizar las ubicaciones de los archivos de entrada y salida para reflejar la estructura del ordenador de destino. Tenga en cuenta que las rutas publicadas no admiten rutas relativas, lo que da una relevancia especial a la verificación de las ubicaciones de los datos en el archivo de parámetros. (Si especifica rutas relativas en las opciones del nodo de exportación al publicar la ruta, IBM SPSS Modeler las convertirá en rutas absolutas antes de crear los archivos de imagen y de parámetros de la ruta publicada.)

También puede utilizar el archivo de parámetros para definir las opciones de ejecución y cambiar los valores de los parámetros de ruta.

Los valores de los parámetros se especifican en el archivo de parámetros como pares <nombre. atributo>=<valor>. En el caso de algunos parámetros, el nombre del parámetro contiene un dígito identificativo para distinguir los nodos del mismo tipo en la ruta. Por ejemplo, en una ruta donde la entrada y la salida hagan referencia a una conexión con una base de datos, los parámetros que describen la conexión de entrada podrían ser dbconn0.datasource, dbconn0.user, etc., y los asociados con la conexión de salida serían dbconn1.datasource, dbconn1.user, etc. Los valores siempre se acotan con comillas dobles. El archivo de parámetros contiene los siguientes parámetros, que se pueden modificar para personalizar la ejecución de la ruta.

Tabla 3. Parámetros de ejecución

| Parámetro                    | Propósito   |
|------------------------------|---|
| dbconnx.datasource           | Nombre de origen de datos (DSN).  |
| dbconnx.user                 | Nombre del usuario para bases de datos con acceso restringido.  |
| dbconnx.password             | Contraseña para bases de datos con acceso restringido.  |
| dbconnx.epassword            | Contraseña codificada para bases de datos con acceso restringido. Para generar una contraseña codificada, seleccione <b>Codificar contraseña</b> en el menú Herramientas de la interfaz de usuario de IBM SPSS Modeler. Copie y pegue la contraseña codificada si es necesario. |
| dbconnx.password_required    | Marca que indica si se debe solicitar una contraseña para la conexión con esta base de datos. (Los valores son "Y" y "N".) Se establece automáticamente en "N" cuando no se especifica ninguna contraseña en el momento de la publicación.                                      |
| dbconnx.catalog              | Catálogo asociado con la conexión a la base de datos.   |
| dbtablex.name                | Nombre de tabla de base. <i>Nota:</i> el orden de los campos de una nueva tabla debe coincidir con el de los campos de la ruta originalmente publicada.   |
| dbtablex.schema              | Nombre del esquema.   |
| dbtablex.catalog             | Catálogo asociado con la base de datos actual.  |
| filex.name                   | Nombre del archivo de datos. <i>Nota:</i> el orden de los campos de un nuevo archivo de entrada debe ser el mismo que el orden de los campos de la ruta originalmente publicada.  |
| filex.path                   | Ubicación del archivo (nombre del directorio)   |
| filex.field_names_included   | En el caso de archivos con formato de texto, es la marca que indica si los nombres de los campos se incluyen como primera línea en el archivo. (Los valores son "Y" y "N".)   |
| filex.field_separator        | En el caso de archivos con formato de texto, son los caracteres utilizados para separar los valores de los campos.  |
| filex.decimal_separator      | En el caso de archivos con formato de texto, es el carácter utilizado para indicar el separador decimal.  |
| options.angle_in_radians     | Marca que indica si se deben usar radianes como unidad de medida en las expresiones trigonométricas de CLEM. (Los valores son "Y" y "N".)   |
| options.date_2digit_baseline | Define el siglo en las fechas especificadas con años de dos dígitos.  |
| options.date_baseline        | El año de línea base (siempre 1 de enero) que utilizarán las funciones de fecha de CLEM que trabajan con una sola fecha.  |
| options.time_rollover        | Marca que indica si las diferencias de tiempo negativas hacen referencia al pasado.   |
| options.decimal_separator    | Carácter predeterminado para el separador decimal utilizado en los archivos de texto.   |
| options.time_format          | Formato de hora que se debe usar cuando las funciones de hora de CLEM interpretan las cadenas como horas.   |
| options.date_format          | Formato de fecha que se debe usar cuando las funciones de fecha de CLEM interpretan las cadenas como fechas.  |
| options.timestamp_format     | Formato utilizado para leer los campos de marca de tiempo como cadenas desde los orígenes de datos ODBC.  |
| paramx.value                 | Valor del parámetro de ruta.<br><i>Nota:</i> El nombre del parámetro, descripción y tipo de almacenamiento se incluyen como referencia pero no se pueden cambiar. El valor tiene que ser válido para el tipo.   |

---

## Incrustación de IBM SPSS Modeler Solution Publisher en las aplicaciones

Además del motor independiente de Runtime, IBM SPSS Modeler Solution Publisher ofrece una biblioteca de programación de tiempo de ejecución (CLEMRTL) que permite a otros programas controlar la ejecución de rutas de IBM SPSS Modeler Solution Publisher. Puede llamar a procedimientos de CLEMRTL desde programas cliente desarrollados en C, C++. Para usar CLEMRTL, deberá incluir el archivo de cabecera *clemrtl.h* (disponible en la carpeta *..\installation\clemrtl\include*) y asegurarse de vincular el archivo de biblioteca adecuado para su plataforma de desarrollo en la aplicación cuando lo genere. El archivo *clemrtl.dll* está disponible en la carpeta *..\installation\bin*.

Cualquier archivo de origen que haga referencia a procedimientos de biblioteca debe incluir el archivo de cabecera *clemrtl.h*. Este archivo de cabecera proporciona prototipos ANSI C para los procedimientos de biblioteca y define macros de gran utilidad. No requiere incluir otras cabeceras aparte de las que requiera el programa. Para evitar los conflictos de nombres, todos los nombres de funciones y tipos de biblioteca comienzan por *clemrtl\_* y todos los nombres de macro contienen el prefijo *CLEMRTL\_*.

CLEMRTL produce informes con información útil que se debe comunicar de nuevo a la aplicación de algún modo. El CLEMRTL proporciona mecanismos para tratar con dichos mensajes:

- La aplicación puede recuperar los detalles del último error usando la función *clemrtl\_getErrorDetail()*.
- La aplicación puede ofrecer su propio procedimiento de gestión de informes usando la función *clemrtl\_setReportHandler()*.

El sistema de Runtime contiene un catálogo de mensajes localizados para informes. Cualquier informe que se pase a la aplicación debería incluir la cadena del mensaje localizado. Los informes también incluyen el código del informe para que la aplicación pueda elegir la interpretación y presentación del mensaje de maneras diferentes. Con el fin de obtener mensajes detallados (localizados), la aplicación debe albergar el archivo apropiado *messages.cfg* en un directorio *config*. Podrá encontrar distintas versiones de *messages.cfg* para las diferentes configuraciones regionales en el directorio *<DirInstalación>/config/<configuración regional>*, donde *<DirInstalación>* es el directorio en el que ha instalado IBM SPSS Modeler Solution Publisher y *<configuración regional>* es la configuración regional deseada. Seleccione la versión apropiada de *messages.cfg* y añádala al subdirectorio *config* del directorio del proyecto.

El sistema de Runtime no necesita ninguna configuración especial de entorno ni registro para funcionar correctamente. Se deben distribuir bibliotecas independientes con una aplicación vinculada a CLEMRTL. Los archivos de ejemplo se instalan en el directorio *<DirInstalación>/bin*.

### Requisitos

La siguiente tabla muestra las plataformas y compiladores probados con la biblioteca de programación de Runtime.

Tabla 4. Plataformas y compiladores

| Plataforma       | Compiladores recomendados                     |
|------------------|---|
| AIX              | IBM XL C/C++ Enterprise Edition V8.0 para AIX |
| Linux de 64 bits | gcc 3.2.3, Intel C++ Compiler 9.1.043         |
| Windows          | Visual Studio 2008 R2, Visual Studio 2015     |

## Aplicación de muestra de IBM SPSS Modeler Solution Publisher

Se incluye una aplicación de ejemplo, *dlltest.c*, en la carpeta `\clemtrl\demo` del directorio de instalación de IBM SPSS Modeler Solution Publisher. Use la aplicación de ejemplo para iniciarse en la incrustación de la biblioteca de Runtime en sus propias aplicaciones.

La información utilizada por la aplicación de ejemplo se incluye en la carpeta `\clemtrl\demo`; consulte el archivo *LÉAME.txt* para obtener instrucciones. La compilación de una aplicación difiere de una plataforma a otra, por lo que también se incluyen archivos de creación de ejemplo.



---

## Capítulo 4. Instalación de IBM SPSS Modeler Solution Publisher Runtime

---

### Instalación de Runtime

La instalación de IBM SPSS Modeler Solution Publisher Runtime le permite ejecutar las rutas publicadas. Las rutas publicadas son independientes del sistema operativo, así que se puede ejecutar cualquier ruta publicada en cualquier plataforma que Runtime admita.

La funcionalidad que le permite publicar rutas desde IBM SPSS Modeler está instalada con IBM SPSS Modeler. Tenga en cuenta que los archivos publicados deben volver a exportarse con cada versión de IBM SPSS Modeler. Por ejemplo, un archivo exportado con una versión anterior no funcionará con la versión actual de IBM SPSS Modeler Solution Publisher Runtime.

### Instalación en Windows

La siguiente sección describe el procedimiento de instalación de SPSS Modeler Solution Publisher Runtime en el sistema operativo Windows.

1. Inicie la instalación de IBM SPSS Modeler Solution Publisher.
2. En el menú de instalación, seleccione **Instalar IBM SPSS Modeler Solution Publisher Runtime** o **Instalar IBM SPSS Modeler Solution Publisher Runtime for Win64**, dependiendo del sistema en el que está instalando.
3. En el asistente de instalación, pulse en **Siguiente** para empezar.
4. Siga las instrucciones que aparecen en la pantalla. Para continuar, pulse en **Siguiente**.
5. Una vez que haya especificado todas las opciones, podrá instalarlo. Pulse en **Instalar** para comenzar la transferencia de archivos.
6. Después de instalar todos los archivos, pulse en **Finalizar**.

### Instalación en UNIX

Las secciones siguientes describen los requisitos adicionales del sistema y el procedimiento de instalación de IBM SPSS Modeler Solution Publisher Runtime en el sistema operativo UNIX.

#### Requisitos adicionales para UNIX

Debe asegurarse de que los límites de kernel del sistema son suficientes para el funcionamiento de IBM SPSS Modeler Server. Los ulimits de datos, memoria, archivo y procesos son especialmente importantes y se deben establecer como ilimitados en el entorno de IBM SPSS Modeler Server. Para ello:

1. Añada los siguientes comandos a `modelersrv.sh`:

```
ulimit -d unlimited
ulimit -m unlimited
ulimit -f unlimited
ulimit -u unlimited
```

Además, establezca el límite de pila en el máximo permitido por el sistema (`ulimit -s XXXX`), por ejemplo:

```
ulimit -s 64000
```

2. Reinicie IBM SPSS Modeler Server.

También necesita que el programa de utilidad de compresión de archivos *gzip* y *GNU cpio* estén instalados y en la PATH para que el instalador pueda descomprimir los archivos de instalación. Además, en la máquina en la que se ejecuta SPSS Modeler Server, debe establecer el entorno local en EN\_US.UTF-8.

## Procedimiento de instalación de UNIX

Estas instrucciones de instalación se aplican a IBM SPSS Modeler Solution Publisher Runtime para UNIX. Necesita permisos de escritura y lectura para el directorio de instalación de destino, por lo que debe iniciar la sesión con una cuenta que tenga permisos suficientes.

**Nota:** debe haber iniciado sesión como *root* para realizar la instalación.

1. Desde el paquete de instalación, abra el directorio *modelrun*.
2. Cambie al directorio de la plataforma pertinente.
3. Ejecute el script de instalación de *.bin*. Por ejemplo:  

```
./modelersolutionpublisherhpia64.bin -i console
```
4. Se muestran los detalles de introducción. Pulse Intro para continuar.
5. Se mostrará la información de licencias. Lea la licencia, escriba 1 para aceptarla y pulse Intro para continuar.
6. Se muestra una lista de los idiomas disponibles. Introduzca el número de la versión del idioma que desea instalar y pulse Intro para continuar.
7. Se le solicitará que escriba la ubicación de instalación. Para usar el directorio predeterminado, */usr/IBM/SPSS/ModelerSolutionPublisher<nn>* (donde *<nn>* es el número de versión), pulse Entrar. Se instalará Runtime en el directorio especificado.
8. Se le solicitará que confirme la ubicación de instalación. Cuando sea correcta, escriba y pulse Intro.
9. Se mostrará un resumen de instalación previa para confirmar sus entradas hasta el momento. Pulse Intro para continuar.
10. Se mostrará un mensaje que le comunicará que la rutina de instalación está lista para ejecutarse. Pulse Intro para continuar.
11. Se mostrará una barra de progreso durante la ejecución de la rutina de instalación. Cuando finalice la instalación, pulse Intro para salir del instalador.

## Configuración de ODBC en UNIX

### Configurar que ODBC se inicie con IBM SPSS Modeler Solution Publisher Runtime

Cuando conecte satisfactoriamente con la base de datos desde IBM SPSS Modeler Server, puede configurar una instalación de IBM SPSS Modeler Solution Publisher Runtime en el mismo servidor haciendo referencia al mismo script *odbc.sh* desde el script de inicio de IBM SPSS Modeler Solution Publisher Runtime.

1. Edite el archivo de script *modelerrun* de IBM SPSS Modeler Solution Publisher Runtime y añada la línea siguiente inmediatamente encima de la última línea del script:  

```
. <odbc.sh_path>
```

donde *odbc.sh\_path* es la ruta de acceso completa al archivo *odbc.sh*. Por ejemplo:  

```
. /usr/spss/odbc/odbc.sh
```

**Nota:** La sintaxis es importante aquí. Asegúrese de dejar un espacio entre el primer punto y la ruta de acceso al archivo.

2. Guarde el archivo de script *modelerrun*.
3. De forma predeterminada, Driver Manager de DataDirect no está configurado para IBM SPSS Modeler Solution Publisher Runtime para utilizar ODBC en sistemas UNIX. Para configurar UNIX para que cargue Driver Manager de DataDirect, introduzca los siguientes comandos (donde *sp\_install\_dir* es el directorio de instalación de Solution Publisher Runtime):



```
cd sp_install_dir
rm -f libspssodbc.so
ln -s libspssodbc_datadirect.so libspssodbc.so
```

## Resolución de problemas de instalación

### Firma digital no válida en la instalación

Los productos IBM SPSS Modeler utilizan la certificación expedida por IBM para la firma digital. En determinadas circunstancias, puede que vea el siguiente error al intentar instalar productos SPSS Modeler:

Error 1330. No se puede instalar un archivo obligatorio porque el nombre del archivo cabinet tiene una firma digital no válida...

### Todos los usuarios de Windows

Verá este mensaje si intenta instalar productos SPSS Modeler en una máquina que no disponga de conexión a Internet y que no tenga instalado el certificado correcto. Utilice el procedimiento siguiente para corregir este problema.

1. Pulse en **Aceptar** para ver el mensaje.
2. Pulse en **Cancelar** para salir del instalador.
3. Si el equipo en el que quiere realizar la instalación no tiene conexión a Internet, realice el paso siguiente en un equipo conectado a Internet y copie el archivo .cer al equipo donde desee realizar la instalación.
4. Vaya a <https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&id=SO5624> y siga las instrucciones para descargar el certificado raíz **VeriSign Class 3 Primary Certification Authority - G5**. Guárdelo como archivo .cer.
5. Pulse dos veces en el archivo .cer.
6. En la pestaña General, pulse en **Instalar certificado**.
7. Siga las instrucciones del asistente de importación de certificados utilizando las opciones predeterminadas y pulsando en **Finalizar** al final.
8. Vuelva a intentar realizar la instalación.

---

## Inicio de Runtime

Una vez que se ha instalado Runtime, se puede utilizar para ejecutar rutas que se hayan publicado desde IBM SPSS Modeler utilizando cualquiera de los nodos de exportación. Consulte el tema “Publicación de rutas” en la página 8 para obtener más información. Inicie Runtime desde la línea de comandos, con opciones que indiquen el nombre del archivo de imagen publicado y el archivo de parámetro opcional que se va a ejecutar.

Para ejecutar Runtime, escriba la siguiente línea en el indicador de comandos:

```
modelerrun -p <archivo de parámetros> -o <opciones> <archivo de imagen>
```

donde las entradas de comando se muestran en la siguiente tabla.

Tabla 5. Entradas de comando de Runtime

| Entrada del comando     | Descripción   |
|-------------------------|---|
| <archivo de parámetros> | es el nombre del archivo del archivo de parámetros publicado (opcional).  |
| <options>               | es una lista de opciones separadas por comas que constituyen pares opción=valor y especifican las opciones de ejecución de Runtime. <b>Nota:</b> No debe haber espacios vacíos después de una coma. |
| <archivo de imagen>     | es el nombre del archivo de imagen publicado (*.pim) que se va a ejecutar.  |

*Nota:* si utiliza lenguajes multibyte como chino o japonés en un entorno UNIX, debe especificar los códigos de lenguaje que se van a usar. Los siguientes ejemplos muestran los códigos de lenguaje de chino simplificado:

```
./modelerrun -o locale="zh_CN.GB18030",encoding="GB18030" -p
```

donde -p es la ubicación de los archivos pim y par.

Consulte Capítulo 3, "Ejecución de rutas publicadas", en la página 11 si desea obtener más información.

## Modificación del directorio temporal

IBM SPSS Modeler Solution Publisher Runtime realiza algunas operaciones que requieren la creación de archivos temporales. De forma predeterminada, Runtime utiliza el directorio temporal del sistema para crear archivos temporales. Se puede modificar la ubicación del directorio temporal con los pasos siguientes.

*Nota:* Es muy importante utilizar la sintaxis correcta, siendo especialmente cuidadoso con los espacios y la ruta del directorio.

Suponga que está utilizando IBM SPSS Modeler Solution Publisher Runtime con los siguientes archivos publicados: *webtest.pim* y *webtest.par*. En esta situación, desea modificar la ubicación de los archivos temporales para utilizar el directorio *C:\published\temp*. Para alterar la ubicación del directorio temporal utilizado por el ejecutable:

1. Desde el directorio donde están guardados los archivos *.pim* y *.par*, ejecute el comando:

```
modelerrun -o temp_directory="C:\\published\\temp" -p webtest.par webtest.pim
```

De esta forma se obliga a Runtime a usar "*C:\published\temp*" como directorio temporal sólo para ese proceso específico.

---

## Desinstalación de IBM SPSS Modeler Solution Publisher Runtime

La sección siguiente describe los procedimientos para la desinstalación de IBM SPSS Modeler Solution Publisher Runtime.

### Procedimiento de desinstalación para Windows

1. En el Panel de control de Windows, abra **Agregar o quitar programas**.
2. En la lista, seleccione **IBM SPSS Modeler Solution Publisher <nn>**.
3. Pulse en **Eliminar**.
4. Siga las instrucciones que aparecerán en la pantalla para desinstalar Runtime.

*Nota:* para quitar programas, debe haber iniciado sesión en el ordenador con privilegios de administrador.

### Procedimiento de desinstalación para UNIX

Quite el directorio de instalación, incluidos todos los archivos instalados.

---

## Capítulo 5. API de IBM SPSS Embedded Predictive Modeling

La API de IBM SPSS Embedded Predictive Modeling define una interfaz de Java simple que permite integrar los artefactos diseñados de IBM SPSS Modeler con las aplicaciones de un modo incorporado. Se incluye con IBM SPSS Modeler Solution Publisher la documentación completa en formato Javadoc para los codificadores que utilizan la biblioteca.

Tenga en cuenta las restricciones siguientes al utilizar la API de EPM (Embedded Predictive Modeling)

- **Se requieren etiquetas de nodo terminal y de origen exclusivas.** La API de EPM utiliza etiquetas de nodo. Aunque las etiquetas están siempre en un contexto de nodo de origen o terminal, pueden ser ambiguas.
- **No se admiten supernodos como nodos terminales o de origen.** Para proporcionar un origen común para los supernodos en todas las operaciones, no se admiten como nodos de origen ni nodos terminales
- **Restricciones de nombre de campo de entrada.** Algunos caracteres no se pueden utilizar en los nombres de campo.

Consulte la información siguiente para obtener una visión general de la funcionalidad de la API. Para obtener más detalles, consulte el Javadoc incluido con el producto.

### Puntuación de Predictive Model Embedded

**Uso:** Lo utiliza la aplicación para generar un análisis predictivo a petición a medida que entran nuevos datos. La aplicación debe gestionar los modelos actualizados. Todos los nodos de origen así como los nodos terminales de la rama de puntuación se sustituyen por las entradas y salidas de aplicación de este modo de puntuación.

**Acción:** Prepara la rama de puntuación diseñada en la ruta de IBM SPSS Modeler una vez para las llamadas repetidas que generan rápida y eficazmente los análisis predictivos a partir de los datos de entrada gestionados por la aplicación.

### Actualización de modelos predictivos

**Uso:** Realice el entrenamiento inicial de los algoritmos de modelo en el diseño de ruta utilizando datos históricos. Además, "actualice" periódicamente los algoritmos del modelo predictivo configurados con los datos de historial más actuales.

**Acción:** Ejecuta todos los nodos generadores de modelos predictivos en la ruta de IBM SPSS Modeler, que entrena el algoritmo de modelo predictivo como está configurado utilizando técnicas de minería de datos. Actualiza los nugget de aplicador de modelos enlazados con cada nodo generador de modelos. Los parámetros de ruta y las propiedades de nodo de origen/terminal se pueden utilizar para implementar algún grado de dinámica.

### Evaluación de modelos predictivos

**Uso:** Las aplicaciones lo utilizan para comunicar la precisión, la confianza u otra evaluación cualitativa de un modelo predictivo entrenado como está diseñado en la ruta de IBM SPSS Modeler. Esta información se suele utilizar para controlar la promoción de un modelo predictivo recién renovado en la aplicación. La principal razón por la que la Actualización del modelo devuelve un nuevo objeto `PredictiveModel` es habilitar la comparación de los datos de evaluación del nuevo modelo recién actualizado con el original.

**Acción:** Ejecuta todos los nodos generadores de documentos (principalmente los nodos de resultados no interactivos en las paletas de gráfico y resultado de IBM SPSS Modeler) en la ruta de IBM SPSS Modeler que devuelve los documentos generados. Los parámetros de ruta y las propiedades de nodo de origen/terminal se pueden utilizar para implementar algún grado de dinámica.

## **Ejecución de ruta de modelos predictivos**

**Uso:** Todos los casos de uso que no se ajustan a "actualización", "evaluación" ni a los dos modos de puntuación, se deben gestionar mediante la ejecución de ruta.

**Acción:** La ejecución como está controlada mediante las opciones de ruta de IBM SPSS Modeler. Ya sea en el modo "todos los nodos de terminal" similar a pulsar el botón grande de flecha verde en el cliente de IBM SPSS Modeler o como está codificado en el script de nivel de ruta. Los parámetros de ruta y las propiedades de nodo de origen/terminal se pueden utilizar para implementar algún grado de dinámica.

## **Datos externos de puntuación de modelos predictivos**

**Uso:** Normalmente se hace referencia a ellos como procesos "por lotes" debido a la manera en que se procesa un lote completo de datos de entrada.

**Acción:** Ejecuta la rama de puntuación como está diseñada. Las entradas de los nodos de origen de datos configurados y de los análisis predictivos generados persisten como haya sido controlado en el nodo terminal. Los parámetros de ruta y las propiedades de nodo de origen/terminal se pueden utilizar para implementar algún grado de dinámica.

## **Parámetros de ruta de modelo predictivo y especificación de propiedades de nodo de origen y de nodo terminal**

**Uso:** Una forma común es para definir los parámetros referenciados en las expresiones en ramas de la ruta. Otro punto de control es para modificar determinadas propiedades clave de un nodo de origen o terminal para modificar la vía de acceso de archivo utilizada, etc.

**Acción:** Una aplicación puede modificar dinámicamente determinados aspectos de ejecución de ruta de IBM SPSS Modeler.

**Nota:** los cambios en los parámetros se deben realizar antes de crear el PredictiveScorer (consulte la descripción de puntuación incluida anterior) y las propiedades de nodo de origen y terminal no tienen ningún significado para este objeto.

---

## Capítulo 6. Referencia API de la biblioteca de IBM SPSS Modeler Solution Publisher Runtime

---

### Introducción

La API de CLEMRTL permite controlar la ejecución de las rutas publicadas desde su aplicación. Esta sección describe las funciones API disponibles.

A continuación se muestran algunos problemas generales al utilizar API:

- La API tiene vinculación C para una máxima compatibilidad, pero la biblioteca tiene dependencias C++. En algunas plataformas, esto podría significar que sólo se puede utilizar con un vinculador de C++.
- Se debe llamar `clemrtl_initialise_ext()` antes de utilizar cualquier otra función de la biblioteca.
- El tipo `clemrtl_image_handle` se utiliza como un identificador de imagen.
- Cada función devuelve un código de indicador de estado. Los valores del indicador de estado se muestran en la siguiente tabla.

Tabla 6. Valores de indicador de estado

| Resultado     | Descripción  |
|---------------|--|
| CLEMRTL_OK    | Éxito  |
| CLEMRTL_FAIL  | Error sin más detalles disponibles   |
| CLEMRTL_ERROR | Error con información adicional sobre el error disponible a través de la función <code>clemrtl_getErrorDetail()</code> |

### Conceptos básicos del proceso de API

El esquema general de una aplicación que utiliza API es el siguiente:

1. Inicializar la biblioteca mediante `clemrtl_initialise_ext()`.
2. Comprobar y cambiar las opciones de ejecución mediante `clemrtl_getOption()` y `clemrtl_setOption()`.
3. Abrir una imagen mediante `clemrtl_openImage()` y recibir un controlador de imagen.
4. Comprobar y cambiar los parámetros de la imagen mediante `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` y `clemrtl_setParameter()`.
5. Ejecutar la imagen utilizando `clemrtl_execute()`. Si se va a ejecutar varias veces la misma imagen sin cambiar los parámetros, se utiliza primero `clemrtl_prepare()`.
6. Cerrar la imagen mediante `clemrtl_closeImage()`.
7. Para cancelar la ejecución en curso, se utiliza `clemrtl_interrupt()`.
8. Para recuperar información sobre el último error, se utiliza `clemrtl_getErrorDetail()`. Para recibir mensajes de error y de diagnóstico según lleguen, se utiliza `clemrtl_setReportHandler()`.

La biblioteca está diseñada para utilizarla en subprocesamiento múltiple bajo las condiciones siguientes:

- No se puede llamar de manera simultánea a funciones globales, que no toman un controlador de imagen como un argumento.
- No se puede llamar de manera simultánea a funciones locales de imagen, que toman un controlador de imagen como un argumento, cuando se aplican al mismo controlador de imagen, a excepción de `clemrtl_interrupt()`.

- Se puede llamar de manera simultánea a funciones locales de imagen cuando se aplican a diferentes controladores de imagen.

Concretamente, esto significa que varios subprocesos distintos pueden preparar y ejecutar imágenes de manera simultánea, siempre que utilicen diferentes controladores de imagen. Es posible que varios controladores distintos hagan referencia a la misma imagen (`clemrtl_openImage()` crea un nuevo controlador cada vez que se le llama, incluso para la misma imagen). No obstante, cuando ejecute varias instancias de la misma imagen, asegúrese de cambiar los parámetros de la imagen para volver a direccionar los resultados según sea necesario. Si dos imágenes que se están ejecutando de manera simultánea intentan escribir en el mismo archivo de resultados o tabla de base de datos, los resultados serán impredecibles.

## Funciones API

A continuación se muestra una lista completa de las funciones expuestas en la API.

### initialise

Esta función está desaprobadada y es equivalente a `initialise_ext(flags, 0, 0);`

Los nuevos programas deben llamar *initialise\_ext*.

### initialise\_ext

```
int clemrtl_initialise_ext(
    unsigned flags,
    int arg_count,
    const clemrtl_init_arg* args);
```

Inicializa IBM SPSS Modeler Solution Publisher Runtime. *Debe llamarse a esta función antes que a las otras funciones API.*

Tabla 7. Parámetros de *initialise\_ext*

| Parámetro | Descripción  |
|-----------|--|
| marcas    | Controla algunos aspectos del proceso de inicialización. El valor está constituido como OR a nivel de bit de las marcas descritas a continuación.  |
| arg_count | El número de argumentos de inicialización adicionales. Debe ser igual o inferior a la longitud de la matriz args.  |
| args      | Argumentos de inicialización adicionales. Los argumentos se describen según el tipo:<br><pre>typedef struct _clemrtl_init_arg {     const char* nombre;     const char* valor; } clemrtl_init_arg;</pre> |

El campo nombre es el nombre de una propiedad de configuración y el campo valor es su valor. Se reconocen las propiedades de configuración mostradas en la tabla siguiente.

Tabla 8. Propiedades de configuración

| Nombre                 | Valor  |
|------------------------|--|
| directorio_instalación | La ruta completa a la carpeta de instalación de IBM SPSS Modeler Solution Publisher. La carpeta de instalación es la que contiene las carpetas bin y config. |

Devuelve uno de los códigos de error mostrados en la tabla siguiente.

Tabla 9. Códigos de error

| Resultado    | Descripción   |
|--------------|---|
| CLEMRTL_OK   | Completado.   |
| CLEMRTL_FAIL | Error al inicializar, sin más detalles disponibles. |

El parámetro `marks` proporciona control sobre ciertos aspectos del proceso de inicialización. Un valor de 0 (o `CLEMRTL_INIT_DEFAULTS`) especifica el comportamiento predeterminado como se indica a continuación:

- Establezca la zona horaria predeterminada, llamando a `tzset()`.
- Establezca el entorno local `LC_NUMERIC` en "C".
- Establezca el nuevo controlador global para iniciar una excepción cuando la memoria esté vacía.

Este comportamiento se puede modificar especificando combinaciones de los distintivos que se muestran en la tabla siguiente.

Tabla 10. Distintivos

| Distintivo                                  | Descripción   |
|---|---|
| <code>CLEMRTL_INIT_NO_TZ</code>             | No establecer el huso horario.                      |
| <code>CLEMRTL_INIT_NO_LOCALE</code>         | No cambiar el entorno local.                        |
| <code>CLEMRTL_INIT_NO_NEW_HANDLER</code>    | No establecer el controlador nuevo.                 |
| <code>CLEMRTL_INIT_LOCAL_NEW_HANDLER</code> | Localizar el nuevo controlador en cada llamada API. |

La biblioteca requiere que el entorno local `LC_NUMERIC` se establezca en "C" para convertir correctamente entre números y cadenas (el formato numérico de IBM SPSS Modeler utiliza las reglas del entorno local de "C"). Esto incluye la conversión de números leídos y escritos en archivos de texto.

- Especifique `NO_LOCALE` si su aplicación depende de la configuración regional `LC_NUMERIC` y no requiere conversión entre números y cadenas.
- `NO_NEW_HANDLER` y `LOCAL_NEW_HANDLER` se excluyen entre sí. La biblioteca requiere que cuando la memoria está vacía, `new` debe iniciar una excepción.
- Especifique `NO_NEW_HANDLER` si la aplicación establece su propio controlador nuevo o si está seguro de que el compilador/sistema de Runtime inicia una excepción de forma predeterminada.

Especifique `LOCAL_NEW_HANDLER` sólo si su aplicación depende de que `new` devuelva un puntero nulo cuando la memoria está vacía. *Nota:* Esta operación no es segura en un entorno de subprocesamiento múltiple.

## getOption

```
int clemrtl_getOption(
    const char* name,
    char* value,
    int value_size);
```

Recupera el valor de una opción de ejecución.

Tabla 11. Parámetros de `getOption`

| Parámetro                 | Descripción                                  |
|---------------------------|--|
| <code>name</code>         | El nombre de la opción.                      |
| <code>value</code>        | Un búfer para recibir el valor de la opción. |
| <code>tamaño_valor</code> | El tamaño del búfer del valor.               |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 12. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si el nombre de la opción no es ninguno de los nombres de opción de ejecución de la lista anterior (por ejemplo, `memory_usage`) o si el búfer del valor no es lo suficientemente grande como para recibir el valor que incluye un carácter nulo en finalización. Los valores de opción son siempre cadenas, incluso cuando la interpretación es numérica. Consulte Capítulo 3, “Ejecución de rutas publicadas”, en la página 11 si desea obtener más información.

### setOption

```
int clemrtl_setOption(
    const char* name,
    const char* value);
```

Define el valor de una opción de ejecución.

Tabla 13. Parámetros de setOption

| Parámetro | Descripción             |
|-----------|-------------------------|
| name      | El nombre de la opción. |
| value     | El valor de la opción.  |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 14. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si el nombre de la opción no es ninguno de los nombres de opción de ejecución de la lista anterior (por ejemplo, `memory_usage`). Los valores de opción son siempre cadenas, incluso cuando la interpretación es numérica.

### openImage

```
int clemrtl_openImage(
    const char* image_file,
    const char* param_file,
    clemrtl_image_handle* handle);
```

Abre una imagen de una ruta publicada.

Tabla 15. Parámetros de openImage

| Parámetro         | Descripción                         |
|-------------------|-------------------------------------|
| archivo_imagen    | El nombre de archivo de la imagen.  |
| archivo_parámetro | El nombre de archivo del parámetro. |



Tabla 15. Parámetros de openImage (continuación)

| Parámetro   | Descripción                         |
|-------------|-------------------------------------|
| controlador | Recibe el controlador de la imagen. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 16. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

El archivo del parámetro puede ser NULL si no se requiere un archivo de parámetro distinto. La operación da error si el archivo de imagen no se especifica, el nombre de archivo no es válido o el contenido no se puede leer. Si se completa la operación, el controlador de imagen devuelto se puede utilizar para identificar la instancia de la imagen en las llamadas API posteriores.

### closeImage

```
int clemrtl_closeImage(
    clemrtl_image_handle handle);
```

Cierra un controlador de imagen y libera los recursos.

Tabla 17. Parámetros de closeImage

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 18. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si hay otra llamada API en curso en la imagen, especialmente si la imagen todavía se está ejecutando. Si se completa la operación, se cierra la imagen y no se puede utilizar el controlador en ninguna llamada API posterior, excepto en otra llamada de `clemrtl_closeImage()`, lo que no tendría efecto alguno.

### enumerateParameters

```
int clemrtl_enumerateParameters(
    clemrtl_image_handle handle,
    clemrtl_parameter_proc proc,
    void* data);
```

Aplica un procedimiento de devolución de llamada a cada valor y nombre del parámetro de la imagen.

Tabla 19. Parámetros de `enumerateParameters`

| Parámetro     | Descripción   |
|---------------|---|
| controlador   | El controlador de imagen.   |
| procedimiento | El procedimiento de la devolución de llamada del parámetro.       |
| data          | Datos especificados por el usuario para la devolución de llamada. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 20. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

El procedimiento de devolución de llamada tiene el siguiente tipo:

```
typedef void (*clemrtl_parameter_proc)(
    void* data,
    const char* name,
    const char* value);
```

Tabla 21. Parámetros de procedimiento de devolución de llamada

| Parámetro | Descripción   |
|-----------|---|
| data      | Datos especificados por el usuario pasados a <code>clemrtl_enumerateParameters()</code> . |
| name      | El nombre del parámetro.  |
| value     | El valor del parámetro.   |

El procedimiento se aplica exactamente una vez a cada parámetro de imagen en un orden arbitrario.

### **getParameter**

```
int clemrtl_getParameter(
    clemrtl_image_handle handle,
    const char* name,
    char* value,
    int value_size);
```

Recupera el valor de un parámetro de imagen.

Tabla 22. Parámetros de `getParameter`

| Parámetro    | Descripción                                   |
|--------------|---|
| controlador  | El controlador de imagen.                     |
| name         | El nombre del parámetro.                      |
| value        | Un búfer para recibir el valor del parámetro. |
| tamaño_valor | El tamaño del búfer del valor.                |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 23. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si el nombre del parámetro no coincide con el nombre de ningún parámetro de la imagen o si el búfer del valor no es lo suficientemente grande como para recibir el valor del parámetro que incluye un carácter nulo en finalización. Los nombres de los parámetros utilizan el formato `name.attribute` descrito anteriormente (por ejemplo, `file0.name`) y los valores de los parámetros son siempre cadenas, incluso cuando la interpretación es numérica.

### setParameter

```
int clemrtl_setParameter(
    clemrtl_image_handle handle,
    const char* name,
    const char* value);
```

Establece el valor de un parámetro de imagen.

Tabla 24. Parámetros de `setParameter`

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |
| name        | El nombre del parámetro.  |
| value       | El valor del parámetro.   |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 25. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si el controlador de imagen o si el nombre del parámetro no coincide con el nombre de ningún parámetro en la imagen. Los nombres de los parámetros utilizan el formato `name.attribute` descrito anteriormente (por ejemplo, `file0.name`) y los valores de los parámetros son siempre cadenas, incluso cuando la interpretación es numérica.

### getFieldCount

```
int clemrtl_getFieldCount(
    clemrtl_image_handle handle,
    const char* key,
    size_t* field_count);
```

Devuelve el número de campos de un origen de entrada o de salida.

Tabla 26. Parámetros de `getFieldCount`

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Tabla 26. Parámetros de getFieldCount (continuación)

| Parámetro       | Descripción  |
|-----------------|--|
| clave           | El nombre de la entrada o salida que se va a comprobar, tal y como se ha utilizado en el archivo de parámetros. La clave puede hacer referencia o a una base de datos. |
| recuento_campos | Recibe el número de campos.  |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 27. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

## getFieldTypes

```
int clemrtl_getFieldTypes(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int* field_types);
```

Devuelve los tipos de campos de un origen de entrada o de salida.

Tabla 28. Parámetros de getFieldTypes

| Parámetro       | Descripción  |
|-----------------|--|
| controlador     | El controlador de imagen.  |
| clave           | El nombre de la entrada o salida que se va a comprobar, tal y como se ha utilizado en el archivo de parámetros. La clave puede hacer referencia o a una base de datos.   |
| recuento_campos | El número de campos que se van a examinar. Debe ser igual o inferior a la longitud de la matriz tipo_datos.  |
| tipos_campos    | Una matriz de una longitud de al menos recuento_campos que recibe los tipos de campos. Los tipos del primer campo recuento_campos en la entrada o salida se copian en la matriz. Si recuento_campos es mayor que el número real de campos, los elementos adicionales de la matriz se dejan sin definir. Los valores de los tipos de campo se incluyen en la tabla de los tipos de datos siguiente. |

## Tipos de datos

El valor de tipos\_campos debe ser uno de los tipos de datos que aparecen en la tabla siguiente.

Tabla 29. Tipos de datos

| Tipo    | Interpretación   | Declarador 'C' normal |
|---------|--|-----------------------|
| STRING  | Cadena de caracteres UTF con 8 caracteres nulos en finalización. | const char*           |
| INTEGER | Entero con signo de 32 bits.                                     | int                   |
| LONG    | Entero firmado de 64 bits.                                       | long long             |

Tabla 29. Tipos de datos (continuación)

| Tipo          | Interpretación  | Declarador 'C' normal |
|---------------|---|-----------------------|
| REAL          | Con punto flotante de 64 bits.  | double (doble)        |
| TIEMPO        | Entero con signo de 64 bits (segundos desde la medianoche).               | long long             |
| DATE          | Entero con signo de 64 bits (segundos desde la medianoche de 01/01/1970). | long long             |
| MARCADETIEMPO | Entero con signo de 64 bits (segundos desde la medianoche de 01/01/1970). | long long             |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 30. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

## setAlternativeInput

```
int clemrtl_setAlternativeInput(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int field_types,
    void** (*iterator)(void* arg),
    void* arg);
```

Sustituye un origen de entrada de archivo con un origen de entrada alternativo.

Tabla 31. Parámetros setAlternativeInput

| Parámetro       | Descripción   |
|-----------------|---|
| controlador     | El controlador de imagen.   |
| clave           | El nombre del origen de entrada original que se utiliza en el archivo de parámetros. El origen de entrada debe ser un archivo para que la clave siempre tenga el formato "fileN" para algún entero $N \geq 0$ .   |
| recuento_campos | El número de campos en la entrada. El valor debe coincidir exactamente con el número de campos de la entrada del archivo original o la llamada no será válida.  |
| tipos_campos    | Una matriz de tipos de una longitud de al menos recuento_campos. El valor tipos_campos[i] es el tipo del campo i' y debe ser uno de los valores de la tabla de tipos de datos siguientes. Los tipos deben ser compatibles con los de la entrada del archivo original o la llamada no será válida. |

Tabla 31. Parámetros *setAlternativeInput* (continuación)

| Parámetro | Descripción  |
|-----------|--|
| iterador  | Una función que produce los datos de entrada alternativos. La función se aplica a su argumento de la siguiente forma:<br><pre>void** row = iterator(arg);</pre> <p>La función se llama durante su ejecución (en una llamada de <code>clemrtl_execute</code>) y se llama una vez para cada registro de entrada. Un valor NULL indica el final de la entrada, en cuyo caso, la función no se vuelve a llamar y la ejecución finalizará. En caso contrario, el resultado es una matriz de datos de una longitud de al menos <code>recuento_campos</code> donde <code>row[i]</code> proporciona el valor del campo <code>i</code>. Un valor puede ser NULL, de lo contrario debe ser un puntero a un dato cuyo tipo está determinado por el <code>tipo_campos[i]</code> correspondiente. Los punteros deben seguir siendo válidos hasta la siguiente llamada del iterador o hasta el final de la ejecución si ésta termina de forma prematura.</p> |
| arg       | Un argumento opaco que se pasa al iterador en cada llamada.  |

### Tipos de datos

`tipo_campos` debe ser uno de una lista de tipos restringida. Consulte el tema “`getFieldTypes`” en la página 30 para obtener más información.

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 32. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación falla si los tipos de campo no son compatibles con los de la entrada del archivo original o si el recuento de campos no coincide exactamente con el número de campos de la entrada del archivo original.

### setAlternativeOutput

```
int clemrtl_setAlternativeOutput(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int field_types,
    void (*iterator)(void* arg, void** row),
    void* arg);
```

Sustituye un objetivo de salida de archivo con un objetivo de salida alternativo.

Tabla 33. Parámetros de *setAlternativeOutput*

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Tabla 33. Parámetros de `setAlternativeOutput` (continuación)

| Parámetro       | Descripción  |
|-----------------|--|
| clave           | El nombre del objetivo de salida original que se utiliza en el archivo de parámetros. El destino de salida debe ser un archivo para que la clave siempre tenga el formato "fileN" para algún entero $N \geq 0$ .   |
| recuento_campos | El número de campos en la salida. El valor debe coincidir exactamente con el número de campos de la salida del archivo original o la llamada no será válida.   |
| tipos_campos    | Una matriz de tipos de una longitud de al menos <code>recuento_campos</code> . El valor <code>tipos_campos[i]</code> es el tipo del campo <code>i</code> y debe ser uno de los valores de la tabla de tipos de datos siguientes. Los tipos deben ser compatibles con los de la salida del archivo original o la llamada no será válida.  |
| iterador        | Una función que consume la salida de la imagen. La función se aplica a su argumento y a una fila de datos de la siguiente forma:<br><code>iterador(arg, row);</code><br><br>La función se llama durante su ejecución (en una llamada de <code>clemrtl_execute</code> ) y se llama una vez para cada fila de resultados que produce la imagen. Un valor de <code>fila</code> NULL indica el final de la salida después de la cual la función no se vuelve a llamar; una aplicación no se debe basar en esta llamada final y debe vaciar y cerrar cualquier recurso externo, etc., cuando la ejecución ha terminado. En caso contrario, la <code>fila</code> es una matriz de datos de una longitud de al menos <code>recuento_campos</code> donde <code>row[i]</code> proporciona el valor del resultado <code>i</code> . Un valor puede ser NULL, de lo contrario debe ser un puntero a un dato cuyo tipo está determinado por el <code>tipo_campos[i]</code> correspondiente. La función debe copiar cualquier valor de datos necesario porque es posible que no se pueda conservar la memoria una vez se ha devuelto la llamada. |
| arg             | Un argumento opaco que se pasa al iterador en cada llamada.  |

## Tipos de datos

`tipo_campos` debe ser uno de una lista de tipos restringida. Consulte el tema "getFieldTypes" en la página 30 para obtener más información.

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 34. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_FAIL  | Error sin más detalles disponibles. |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación falla si los tipos de campo no son compatibles con los de la salida del archivo original o si el recuento de campos no coincide exactamente con el número de campos de la salida del archivo original.

## execute

```
int clemrtl_execute(clemrtl_image_handle controlador);
```

Ejecuta una imagen.

Tabla 35. Parámetros de execute

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 36. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

Si no se ha preparado la imagen, se prepara primero utilizando los valores del parámetro actual. La operación da error si el controlador de imagen no es válido o si se produce un error durante la preparación o la ejecución. La llamada no se devuelve hasta que la ejecución se completa.

## prepare

```
int clemrtl_prepare(clemrtl_image_handle controlador);
```

Prepara una imagen para que se ejecute.

Tabla 37. Parámetros de prepare

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 38. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación da error si el controlador de imagen no es válido o si se produce un error durante la preparación.

Una imagen se debe preparar para poder ejecutarla. La preparación de una imagen congela los valores del parámetro en la imagen. `clemrtl_execute()` prepara una imagen automáticamente en cada llamada si aún no se ha preparado; resulta útil utilizar `clemrtl_prepare()` para preparar una imagen si se va a ejecutar la imagen varias veces con los mismos valores de parámetro, y permite mejorar el rendimiento considerablemente. Una vez que se ha preparado una imagen, se ignoran los cambios posteriores de los valores del parámetro; vuelva a utilizar `clemrtl_prepare()` para actualizar la imagen con los valores nuevos del parámetro.

## interrupt

```
int clemrtl_interrupt(clemrtl_image_handle controlador);
```

Finaliza la ejecución en curso en una imagen.



Tabla 39. Parámetros de interrupt

| Parámetro   | Descripción               |
|-------------|---------------------------|
| controlador | El controlador de imagen. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 40. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

La operación no tiene efecto alguno si el controlador de imagen no es válido o si la imagen no se está ejecutando.

Es seguro llamar a esta función simultáneamente con otra llamada API en el mismo controlador de imagen.

### getErrorDetail

```
int clemrtl_getErrorDetail(
    clemrtl_image_handle handle,
    char* severity,
    int* code,
    char* text,
    int text_size);
```

Recupera información detallada sobre el último error que se produjo en una imagen.

Tabla 41. Parámetros de getErrorDetail

| Parámetro    | Descripción  |
|--------------|--|
| controlador  | El controlador de imagen.  |
| gravedad     | Recibe el código de gravedad como un carácter único:<br><br>I—información<br>W—aviso<br>E—error<br>X—error del sistema |
| código       | Recibe el número de error.   |
| text         | Un búfer para recibir el texto del mensaje.  |
| tamaño_texto | Tamaño del búfer del texto.  |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 42. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

Los resultados serán inestables si el controlador de imagen no es válido. El texto del mensaje se divide si es necesario para ajustarlo en el búfer del texto.

Si una llamada API da el error CLEMRTL\_ERROR y éste no está relacionado con ningún controlador de imagen en particular, al pasar el valor 0 como el controlador de imagen se recuperarán los detalles del último error no específico de imagen, pero es inestable en un entorno de subprocesamiento múltiple.

## setReportHandler

```
int clemrtl_setReportHandler(
    clemrtl_image_handle handle,
    clemrtl_report_proc proc,
    void* data);
```

Instala un procedimiento de devolución de llamada de informe para una imagen.

Tabla 43. Parámetros de setReportHandler

| Parámetro     | Descripción   |
|---------------|---|
| controlador   | El controlador de imagen.   |
| procedimiento | El procedimiento de devolución de llamada de informe.             |
| data          | Datos especificados por el usuario para la devolución de llamada. |

Devuelve uno de los códigos de estado que se muestran en la tabla siguiente.

Tabla 44. Códigos de estado

| Resultado     | Descripción                         |
|---------------|-------------------------------------|
| CLEMRTL_OK    | Completado.                         |
| CLEMRTL_ERROR | Error con más detalles disponibles. |

El procedimiento de devolución de llamada tiene el siguiente tipo:

```
typedef void (*clemrtl_report_proc)(
    void* data,
    char severity,
    int code,
    const char* text);
```

Tabla 45. Parámetros de procedimiento de devolución de llamada

| Parámetro | Descripción   |
|-----------|---|
| data      | Datos especificados por el usuario pasados a clemrtl_setReportHandler().  |
| gravedad  | El código de gravedad como un carácter único:<br><br>I—información<br>W—aviso<br>E—error<br>X—error del sistema |
| código    | El número del mensaje.  |
| text      | El texto del mensaje.   |

Este procedimiento se aplica a cada mensaje según llega. Al establecer el procedimiento en NULL se elimina cualquier controlador instalado en la imagen.

---

## Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los EE.UU. Este material puede estar disponible en IBM en otros idiomas. Sin embargo, es posible que deba ser propietario de una copia del producto o de la versión del producto en dicho idioma para acceder a él.

Es posible que IBM no ofrezca los productos, servicios o características que se tratan en este documento en otros países. El representante local de IBM le puede informar sobre los productos y servicios que están actualmente disponibles en su localidad. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que solamente se pueda utilizar ese producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja 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 solicitudes de patente en tramitación que cubran la materia descrita 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, MD-NC119  
Armonk, NY 10504-1785  
EE.UU.*

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe sus consultas, por escrito, a:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokio 103-8510, Japón*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍAS DE NINGUNA CLASE, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN, COMERCIALIZACIÓN O ADECUACIÓN A UN PROPÓSITO DETERMINADO. Algunas jurisdicciones no permiten la renuncia a las garantías explícitas o implícitas en determinadas transacciones; por lo tanto, es posible que esta declaración no sea aplicable a su caso.

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 en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen de modo alguno un aval de esos sitios web. La información de esos sitios web no forma parte de la información de este producto de IBM y la utilización de esos sitios web se realiza bajo la responsabilidad del usuario.

IBM puede utilizar o distribuir la información que se le proporcione del modo que considere adecuado sin incurrir por ello en ninguna obligación con el remitente.

Los titulares de licencias de este programa que deseen tener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido este) y (ii) el uso mutuo de la información que se ha intercambiado, deberán ponerse en contacto con:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
EE.UU.*

Esta información estará disponible, bajo las condiciones adecuadas, incluyendo en algunos casos el pago de una cuota.

El programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible los proporciona IBM bajo los términos de las Condiciones Generales de IBM, Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre las partes.

Los ejemplos de datos de rendimiento y de clientes citados se presentan solamente a efectos ilustrativos. Los resultados reales de rendimiento pueden variar en función de las configuraciones específicas y condiciones de operación.

La información relacionada con productos no IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con productos no IBM. Las preguntas sobre las posibilidades de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Las declaraciones sobre el futuro rumbo o 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 estos nombres son ficticios y cualquier parecido con personas o empresas comerciales reales es pura coincidencia.

---

## **Marcas comerciales**

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas o marcas comerciales de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios podrían ser marcas registradas de IBM u otras compañías. En Internet hay disponible una lista actualizada de las marcas registradas de IBM, en "Copyright and trademark information", en [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, el logotipo Adobe, PostScript y el logotipo PostScript son marcas registradas o marcas comerciales de Adobe Systems Incorporated en Estados Unidos y/o otros países.

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 o registradas de Oracle y/o sus afiliados.

---

## **Términos y condiciones para la documentación del producto**

Los permisos para utilizar estas publicaciones se otorgan de acuerdo con los términos y condiciones siguientes.

### **Aplicabilidad**

Estos términos y condiciones son adicionales a los términos de uso del sitio web de IBM.

### **Uso personal**

Estas publicaciones se pueden reproducir para uso personal no comercial siempre que se conserven todos los avisos de propiedad. No puede distribuir, visualizar ni realizar trabajos derivados de estas publicaciones, ni de partes de las mismas, sin el consentimiento expreso de IBM.

### **Uso comercial**

Puede reproducir, distribuir y visualizar estas publicaciones únicamente dentro de la empresa a condición de que se conserven todos los avisos de propiedad. No puede realizar trabajos derivados de estas publicaciones, ni de partes de las mismas, ni reproducirlas, distribuirlas o visualizarlas fuera de su empresa sin el consentimiento expreso de IBM.

### **Derechos**

Excepto de la forma explícitamente otorgada en este permiso, no se otorga ningún permiso, licencia ni derecho, ni explícito ni implícito, sobre las publicaciones ni a ninguna otra información, datos, software u otra propiedad intelectual contenida en ellas.

IBM se reserva el derecho de retirar los permisos aquí otorgados siempre que, a su discreción, el uso de las publicaciones sea perjudicial para su interés o cuando, según determine IBM, las instrucciones anteriores no se sigan correctamente.

No puede descargar, exportar ni volver a exportar esta información si no es cumpliendo totalmente todas las leyes y regulaciones aplicables, incluyendo las leyes y regulaciones de exportación de los Estados Unidos.

IBM NO GARANTIZA EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL" Y SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUYENDO PERO SIN LIMITARSE A LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, NO VULNERACIÓN E IDONEIDAD PARA UN FIN DETERMINADO.



# Índice

## A

apertura  
  rutas 26  
API  
  códigos de error 23  
  códigos de estado 23  
  conceptos básicos 14, 23  
  entrada alternativa 31  
  opciones de ejecución 25, 26  
  recuento de campos 29  
  referencia 23  
  salida alternativa 32  
  setReportHandler 36  
  tipos de campos 30  
archivo .par 8  
archivo .pim 8

## B

biblioteca de programación de tiempo de ejecución (CLEMRTL) 14

## C

cierre  
  rutas 27  
closeImage  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 27  
códigos de error 23  
códigos de estado 23  
configuración regional  
  definir mediante la API 24

## D

desinstalación  
  IBM SPSS Modeler Solution Publisher Runtime 20  
despliegue 7, 21  
directorio temporal  
  IBM SPSS Modeler Solution Publisher Runtime 20  
documentación 3

## E

Editor, nodo 21  
ejecución de rutas  
  utilizando IBM SPSS Modeler Solution Publisher 11, 12  
ejemplos  
  conceptos básicos 4  
  Guía de aplicaciones 3  
ejemplos de aplicaciones 3  
enumerateParameters  
  conceptos básicos del proceso de API 23

enumerateParameters (*continuación*)  
  referencia de funciones API 27  
EPM  
  API 21  
execute  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 34

## F

funciones API  
  closeImage 23, 27  
  enumerateParameters 23, 27  
  execute 23, 34  
  getErrorDetail 23, 35  
  getFieldCount 23, 29  
  getFieldTypes 23, 30  
  getOption 23, 25  
  getParameter 23, 28  
  inicialización 24  
  initializing 23  
  interrupt 23, 34  
  openImage 23, 26  
  prepare 23, 34  
  setAlternativeInput 23, 31  
  setAlternativeOutput 23, 32  
  setLogFile 23  
  setOption 23, 26  
  setParameter 23, 29  
  setReportHandler 23

## G

getErrorDetail  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 35  
getFieldCount  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 29  
getFieldTypes  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 30  
getOption  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 25  
getParameter  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 28

## I

IBM SPSS Embeddable Predictive Analytics  
  API 21

IBM SPSS Modeler 1  
  documentación 3  
IBM SPSS Modeler Server 1  
IBM SPSS Modeler Solution Publisher 7, 8, 21  
  conceptos básicos 7  
  incrustación en aplicaciones 14  
IBM SPSS Modeler Solution Publisher, nodo 21  
IBM SPSS Modeler Solution Publisher Runtime  
  desinstalación 20  
  directorio temporal 20  
  inicio 19  
  instalación 17  
  requisitos del sistema 17  
initialise  
  marcas 24  
  referencia de funciones API 24  
initialise\_ext  
  marcas 24  
  referencia de funciones API 24  
initialize  
  conceptos básicos del proceso de API 23  
interrupt  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 34

## N

nodos de resultados 7, 21  
notificación de errores  
  referencia de funciones API 35, 36

## O

openImage  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 26

## P

parámetros  
  archivos de parámetros de API 26, 27  
  parámetros de ejecución de ruta 12  
  parámetros de imagen de API 27, 28, 29  
  procedimientos de devolución de llamada API 27  
prepare  
  conceptos básicos del proceso de API 23  
  referencia de funciones API 34  
publicación de rutas  
  IBM SPSS Modeler Solution Publisher 7, 8

## R

### Runtime

- desinstalación 20
- directorio temporal 20
- inicio 19
- instalación de Windows 17
- instalación en UNIX 17
- opciones 11

## S

### setAlternativeInput

- conceptos básicos del proceso de API 23
- referencia de funciones API 31

### setAlternativeOutput

- conceptos básicos del proceso de API 23
- referencia de funciones API 32

### setLogFile

- conceptos básicos del proceso de API 23

### setOption

- conceptos básicos del proceso de API 23
- referencia de funciones API 26

### setParameter

- conceptos básicos del proceso de API 23
- referencia de funciones API 29

### setReportHandler

- conceptos básicos del proceso de API 23
- referencia de funciones API 36

## U

### UNIX

- instalación en IBM SPSS Modeler Solution Publisher Runtime 17

## Z

### zona horaria

- definir mediante la API 24







Impreso en España