

IBM DB2 Alphablox



# Introdução

*Versão 8.3*



IBM DB2 Alphablox



# Introdução

*Versão 8.3*

**Nota:**

Antes de utilizar estas informações e o produto suportado por elas, leia as informações nos “Avisos” na página 31.

**Primeira Edição (Novembro de 2005)**

Esta edição aplica-se à versão 8, release 3, do IBM DB2 Alphablox para Linux, UNIX e Windows (produto número 5724-L14) e a todos os releases e modificações subseqüentes, até que seja indicado de outra forma em novas edições.

Quando o Cliente envia seus comentários, concede direitos não-exclusivos à IBM para usá-los ou distribuí-los da maneira que achar conveniente, sem que isso implique em qualquer compromisso ou obrigação para com o Cliente.

**Direitos Autorais © 1996 - 2005 Alphablox Corporation. Todos os direitos reservados.**

**© Direitos Autorais International Business Machines Corporation 1996, 2005. Todos os direitos reservados.**

---

# Índice

## Capítulo 1. Tutorial: Construindo Seu primeiro Aplicativo . . . . . 1

Definindo seu Aplicativo . . . . .	1
Acessando Dados. . . . .	2
Criando a Home Page do Aplicativo . . . . .	2
Configurando uma Home Page Padrão . . . . .	3
Criando a Primeira Visualização Analítica . . . . .	3
Estrutura da Página de sua Primeira Visualização Analítica. . . . .	4
Criando a Segunda Visualização Analítica . . . . .	5
Resumo . . . . .	7

## Capítulo 2. Tutorial: Construindo seu Primeiro Portlet com Componentes Blox 9

Instalando Portlets de Amostra . . . . .	9
Executando os Portlets de Amostra . . . . .	10
Examinando a Estrutura de uma Página JSP do Portlet com um Componente Blox . . . . .	10
Criando sua Própria Página JSP do Portlet com Componentes Blox . . . . .	12
Criando um Projeto de Portlet para Utilizar Componentes Blox . . . . .	14
Configurando um Projeto de Portlet Utilizando o Rational Application Developer. . . . .	15
Próximas Etapas. . . . .	16
Dicas de Desenvolvimento de Portlets . . . . .	16

## Capítulo 3. Tutorial: Construindo Aplicativos Utilizando Ferramentas Rational Developer . . . . . 19

Preparando seu Ambiente de Desenvolvimento . . . . .	19
--	----

Instalando o DB2 Alphablox Toolkit . . . . .	19
Instalando o DB2 Alphablox nos Ambientes de Teste Integrados do WebSphere. . . . .	20
Criando uma Instância do Servidor WebSphere . . . . .	21
Criando Variáveis de Substituição do Servidor WebSphere 5.1 . . . . .	21
Configurando as Instâncias do Servidor WebSphere 5.1 . . . . .	22
Incluindo o Usuário Guest no Grupo de Administradores do DB2 Alphablox . . . . .	23
Criando Aplicativos do DB2 Alphablox . . . . .	23
Criando Arquivos JSP com o Conteúdo do DB2 Alphablox . . . . .	23
Acessando Origens de Dados do DB2 OLAP Server e do Essbase . . . . .	24

## Capítulo 4. Tutorial: Construindo um Cubo do DB2 Alphablox, Utilizando o DB2 Cube Views . . . . . 27

Definindo uma Origem de Dados Relacionais do DB2 . . . . .	27
Definindo uma Origem de Dados do Alphablox Cube Server Adaptor . . . . .	28
Definindo um Cubo do DB2 Alphablox . . . . .	29
Iniciando o Cubo do seu DB2 Alphablox . . . . .	30

## Avisos . . . . . 31

Marcas Registradas . . . . .	33
------------------------------	----

## Índice Remissivo . . . . . 35



---

## Capítulo 1. Tutorial: Construindo Seu primeiro Aplicativo

Este tutorial ensina princípios básicos importantes sobre o que são os componentes do Blox e fornece etapas que você pode seguir para rapidamente construir um aplicativo DB2 Alphablox.

Para construir aplicativos DB2 Alphablox, você deve utilizar a tecnologia JSP (JavaServer Pages), mas o conhecimento de JSP não é obrigatório para este tutorial. Este tutorial explica os aspectos do JSP requeridos para construir aplicativos analíticos de um modo relativamente fácil, que se soma aos seus conhecimentos anteriores de tecnologia da web.

As etapas neste tutorial assumem que você está utilizando o DB2 Alphablox em execução no WebSphere Application Server. Se estiver utilizando o BEA WebLogic ou Apache Tomcat como servidor de aplicativos, você pode precisar modificar algumas das etapas a seguir para ajustar diferenças específicas no servidor.

---

### Definindo seu Aplicativo

Para criar um aplicativo utilizando uma abordagem de desenvolvimento J2EE, você precisa criar uma estrutura de diretórios com um diretório WEB-INF que contenha um arquivo do descritor de aplicativos (web.xml). O modo mais simples de criar esta estrutura no DB2 Alphablox é criar um novo aplicativo utilizando a página Aplicativo no DB2 Alphablox Admin Pages.

Para criar o aplicativo e a pasta::

1. Crie um novo aplicativo chamado MyApp no DB2 Alphablox, seguindo as etapas descritas na seção Application Definitions do *Administrator's Guide*.
  - Digite MyApp no campo **Nome**.
  - Digite My App (com um espaço) no campo **Nome da Exibição**, que define a etiqueta para o aplicativo que aparece na lista na página Aplicativos.
2. Clique na guia superior **Aplicativos** no início da página à esquerda da página. Uma lista de aplicativos disponíveis é aberta.
3. Clique no nome de seu aplicativo criado (My App) na lista de aplicativos. Como você ainda não criou arquivos, o diretório de arquivos está vazio.
4. Navegue para a pasta do novo aplicativo no servidor de aplicativos (para WebSphere, a pasta está no diretório installedApps do WebSphere). Observe que o aplicativo inclui um diretório WEB-INF com um arquivo web.xml e o diretório tlds. O arquivo web.xml define as informações sobre o aplicativo. O diretório tlds inclui um arquivo do descritor de biblioteca de tags Blox (blox.tld), que define as tags Blox que você utilizará para criar visualizações analíticas neste tutorial. O descritor contém outros arquivos do descritor de biblioteca de tags (TLD) para outras bibliotecas de tags Blox (não utiliza neste tutorial), incluindo bloxform.tld, bloxlogic.tld, bloxreport.tld e bloxui.tld.

Seu primeiro aplicativo DB2 Alphablox está agora definido.

---

## Acessando Dados

Para este tutorial, você utilizará uma origem de dados pré-definida com o DB2 Alphablox.

Para abrir a página Origem de Dados para assegurar que esta origem de dados está disponível:

1. Abra o navegador na home page do DB2 Alphablox.
2. Clique na guia **Administração**.
3. Clique nas **Origens de Dados**.
4. Na página Origens de Dados, procure por Canned na lista de seleções no lado esquerdo da página. A origem de dados Canned é uma origem de dados pré-definida que utiliza o Canned Data Adapter interno. Esta origem de dados é útil para teste e aprendizado de conceitos básicos.

---

## Criando a Home Page do Aplicativo

Para esta tarefa, você pode criar uma home page simples utilizando conhecimentos básicos sobre a web.

Para criar uma home page:

1. Crie o arquivo da home page, nomeando-o como `myapp.html`.
2. Edite o arquivo para especificar um título do aplicativo (por exemplo, “My DB2 Alphablox Application”) e dois links para as duas visualizações analíticas serão criadas.

Você pode copiar e colar o seguinte código no arquivo `myapp.html`.

```
<html>
<head>
<title>My DB2 Alphablox Application</title>
</head>
<body>
<h2>My DB2 Alphablox Application</h2>
<p>
<a href="GridandChartViews.jsp">GridBlox and ChartBlox Views</a>
</p>
<p>
<a href="PresentBloxView.jsp">PresentBlox View</a>
</p>
</body>
</html>
```

3. Salve este arquivo no diretório MyApp criado ao definir o aplicativo.
4. Abra o navegador web na home page do DB2 Alphablox. Por padrão, o navegador exibe a guia **Aplicativos**.
5. Clique no link **My App** para verificar que o arquivo esteja incluído no diretório. Clique no link **myapp.html** para visualizar sua home page. Se você estiver utilizando o BEA WebLogic e você ver o erro 403 do HTTP (listagens de diretórios não são permitidas por padrão), você precisará incluir o nome do arquivo `myapp.html` no final da URL.

Agora você tem uma home page com links para as duas visualizações analíticas que você criará em uma tarefa posterior no tutorial.



---

## Configurando uma Home Page Padrão

Nas etapas anteriores, você acessou a home page diretamente clicando no link do nome do arquivo no diretório de aplicativos MyApp ou especificando o arquivo da home page, myapp.html, na URL. Nesta etapa, você especificará a home page padrão para o aplicativo.

Para configurar a home page padrão do aplicativo:

1. Abra o navegador na home page do DB2 Alphablox.
2. Clique na guia **Administração**.
3. Clique no link **Aplicativos**.
4. Selecione **MyApp** na lista de aplicativos e clique no botão **Editar** abaixo da lista.
5. Na página Editar Aplicativo para MyApp, digite myapp.html no campo **URL Home**.
6. Clique no botão **Salvar** para salvar as alterações feitas.
7. Clique na guia principal Aplicativos, na parte superior esquerda da guia da pasta, para retornar à página Aplicativos.
8. Clique no aplicativo **My App**. O aplicativo agora abre diretamente na home page definida.

Agora que a home page está definida, o servidor saberá aonde começar, e direcionará o aplicativo para abrir nesta página, mesmo que o usuário falhe em especificar a home page em sua URL. Desta maneira, se um usuário digitar `http://yourServerName/MyApp/` na barra de endereços do navegador, o aplicativo abrirá automaticamente na página definida, myapp.html.

Nas etapas que virão, você começará a criar visualizações analíticas do DB2 Alphablox utilizando tags Blox simples.

---

## Criando a Primeira Visualização Analítica

Nesta tarefa, você criará uma visualização analítica do DB2 Alphablox exibindo uma grade e um gráfico.

Para esta primeira visualização, você estará utilizando tags customizadas JSP definidas pelo DB2 Alphablox para criar rapidamente uma página JSP com dois componentes Blox de apresentação. Os componentes GridBlox e ChartBlox serão utilizados para criar duas visualizações diferentes da mesma amostra de dados, uma visualização de tabela em grade e uma visualização de gráfico.

**Pré-requisito:** Se você ler a seção de introdução sobre o DB2 Alphablox e seus componentes, você vai entender o que é um Blox e como JSP e tags customizadas funcionam. Se você pulou esta seção, especialmente se você não estiver familiarizado com a tecnologia J2EE e JSP (JavaServer Pages), é importante lê-la antes de continuar. A tecnologia JSP e tags JSP customizadas permitem que páginas HTML incluam conteúdo dinâmico que é processado no servidor, e não nos navegadores do usuário.

Para criar sua primeira página de análise do DB2 Alphablox, inclua duas das três apresentações Blox comuns para uma página JSP:

1. Abra um novo arquivo no editor.
2. Crie uma página de visualização analítica básica com um título, um GridBlox e um ChartBlox, digitando o código a seguir:

```

<%@ taglib uri="bloxtld" prefix="blox" %>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>GridBlox and ChartBlox Views</h2>
<p>
<blox:grid id="MyGridBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
</blox:grid>
</p>
<p>
<blox:chart id="MyChartBlox"
  height="50%"
  width="90%"
  chartType="Bar">
  <blox:data
    dataSourceName="Canned"/>
</blox:chart>
</p>
</body>
</html>

```

3. Nomeie o arquivo GridandChartViews.jsp e salve no diretório MyApp.
4. Verifique os resultados.
  - a. Abra a home page do DB2 Alphablox em seu navegador. Se você ainda não estiver na página Aplicativos, clique na guia **Aplicativos**.
  - b. Clique no nome do aplicativo **My App** para abrir a home page.
  - c. Clique no link **Visualizações GridBlox e ChartBlox** que chamam a página GridAndChartViews.jsp para ver sua primeira visualização do DB2 Alphablox. Se aparecer uma página de erro, verifique se existe algum erro em seu código e tente novamente.

---

## Estrutura da Página de sua Primeira Visualização Analítica

Este tópico sumariza a estrutura da primeira visualização analítica criada na etapa anterior.

A primeira linha da página é:

```
<%@ taglib uri="bloxtld" prefix="blox" %>
```

Esta linha é uma diretiva de taglib JSP que informa ao servidor que você pretende utilizar uma biblioteca de tags Blox. O uri é um ponteiro para o arquivo do descritor da biblioteca de tags. O valor prefix, definido como blox, informa ao servidor para procurar todas as tags nesta página que iniciam com blox e, em seguida, processar o conteúdo, utilizando a biblioteca de tags Blox conforme definido no arquivo do descritor da biblioteca de tags.

A seção <head> da página contém uma tag Blox especial para inclusão de código importante para a página, antes que ela seja renderizada:

```
<blox:header/>
```

Essa tag é utilizada pelo DB2 Alphablox para incluir automaticamente HTML, JavaScript e código CSS requeridos na seção principal da página. Quando uma página é renderizada pelo servidor, esta tag inclui links CSS para temas HTML definidos e uma tag meta para evitar armazenamento em cache. Lembre-se de

digitar esta tag em cada página JSP que utilize componentes Blox. Se esta tag não for incluída, a página não será renderizada corretamente.

No corpo da página, a primeira apresentação do Blox, um GridBlox, é definido pelas seguintes linhas:

```
<blox:grid id="MyGridBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
</blox:grid>
```

A tag `<blox:grid>` especifica que você deseja que um GridBlox apareça aqui com um valor de atributo `id` do `MyGridBlox`. O atributo `id` permite a você identificar este Blox particular para propósitos de script. Existem muitos atributos disponíveis para cada Blox definido utilizando tags, mas a menos que você precise especificar valores de atributos diferentes dos valores padrão, você não precisa incluí-los na tag. No exemplo de grade, você também especificou um valor de altura de 50% e um valor de largura de 90%. A altura e a largura podem ser definidas em pixels ou em porcentagens.

Um DataBlox aninhado é incluído, que define o atributo `dataSourceName` como `Canned`. Se nenhuma origem de dados for especificada, você verá a mensagem "Nenhum dado disponível" na grade. A origem de dados `Canned` é pré-definida durante a instalação. Ela inclui uma pequena quantidade de dados de amostra, e não requer a instalação e configuração de um banco de dados externo real e pode ser utilizada para resolução de problemas e aprendizado. Como não é necessário especificar uma consulta, o atributo `query` não existe.

Nas linhas a seguir, um ChartBlox definido que utiliza a tag `<blox:chart>`:

```
<blox:chart id="MyChartBlox"
  height="50%"
  width="90%"
  chartType="Bar">
  <blox:data
    dataSourceName="Canned"/>
</blox:chart>
```

Como mostrado antes no GridBlox, os atributos especificados incluem `id`, altura e largura. Além desses atributos comuns, um atributo `chartType` com um valor de `Bar` (um gráfico de barras padrão) foi definido para invalidar o valor padrão de `3D Bar` (um gráfico de barras tri-dimensional). Como no GridBlox, um DataBlox aninhado é especificado pelo uso da tag `<blox:data>`. O componente Blox precisa de uma origem de dados para ter um conjunto de dados disponível.

Para resumir, três tags Blox foram utilizadas nessa página: `<blox:header/>`, `<blox:grid>`, e `<blox:chart>`. Estas três tags especificam uma visualização analítica sem requererem qualquer código Java na página. A complexidade da lógica de apresentação está sendo gerenciada por essas tags Blox simples. Incluindo Blox aninhado e modificando os valores de atributos, você pode customizar visualizações com base nos requisitos de negócios.

---

## Criando a Segunda Visualização Analítica

Nesta tarefa, você criará a visualização analítica `PresentBlox`, combinando uma grade e um gráfico em um componente Blox único.

Até o momento, você viu os componentes GridBlox e ChartBlox gerando visualizações analíticas não sincronizadas para exibir os mesmos dados. Muitos aplicativos requerem uma visualização de dados utilizando uma grade ou um gráfico. Em outros momentos, você pode desejar mostrar dados em uma visualização analítica, que permita aos usuários comutarem entre as visualizações de grade e de gráfico. Você pode até mesmo permitir que ambas sejam visualizadas simultaneamente. O componente PresentBlox combina o GridBlox e o ChartBlox como componentes aninhados que são exibidos juntos no componente PresentBlox. Nas etapas a seguir, você definirá um PresentBlox em uma página JSP e utilizará as propriedades do PresentBlox para criar uma visualização que mostra inicialmente uma grade e um gráfico lado a lado. Utilizando o controle de interface com o usuário, o usuário pode escolher visualizar apenas uma grade ou um gráfico. Ao combinar as duas visualizações em um Blox, você não apenas obtém visualizações sincronizadas dos dados, mas também utiliza com eficiência o espaço limitado em uma página.

Para criar a segunda visualização analítica:

1. Abra um novo arquivo no editor HTML.
2. Crie uma página de visualização analítica básica com um título e um PresentBlox único, digitando o código a seguir:

```
<%@ taglib uri="bloxtld" prefix="blox" %>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>PresentBlox View</h2>
<p>
<blox:present id="MyPresentBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
  <blox:grid
    bandingEnabled="true"/>
  <blox:chart
    chartType="Bar"/>
</blox:present>
</p>
</body>
</html>
```

3. Nomeie o arquivo PresentBloxView.jsp e salve-o no diretório MyApp.
4. Verifique os resultados:
  - a. Abra a home page do DB2 Alphablox em seu navegador. Se você ainda não estiver na página Aplicativos, clique na guia **Aplicativos**.
  - b. Clique no nome do aplicativo **My App** para abrir a home page.
  - c. Clique no link **Visualização PresentBlox** que chama o arquivo PresentBloxView.jsp para ver sua nova visualização analítica. Se aparecer uma página de erro, verifique se existe algum erro em seu código e tente novamente.

---

## Resumo

Se você concluiu todas as tarefas do tutorial, você aprendeu como construir um aplicativo básico DB2 Alphablox utilizando o DB2 Alphablox Admin Pages e tags Blox. Se você já leu com atenção a *Referência do Desenvolvedor*, você está ciente que existem muitas propriedades e métodos disponíveis para definir e manipular Blox em páginas JSP. Desta forma qual é o próximo passo?

Se você deseja ver seus dados corporativos em uma visualização do DB2 Alphablox, você sabe agora como criar um aplicativo e incluir visualizações analíticas no mesmo. Se você deseja ver algo imediatamente, você deve fazer algumas modificações simples no aplicativo MyApp. Para exibir dados de seu banco de dados corporativo ao invés de utilizar a origem de dados Canned, você deve criar uma nova origem de dados que aponte para uma origem de dados corporativa, modificar os atributos dataSourceName para apontar para a nova origem de dados definida e, em seguida, incluir um atributo query apropriado. Para obter detalhes sobre como criar origens de dados, consulte o *Administrator's Guide*. Para aprender como criar consultas apropriadas, consulte Retrieving Data no *Developer's Guide* e a seção DataBlox do *Developer's Reference*.



---

## Capítulo 2. Tutorial: Construindo seu Primeiro Portlet com Componentes Blox

Neste tutorial, você aprenderá como incluir componentes Blox em seus portlets. Você:

1. Instalará um portlet de amostra pré-construído. Com esta etapa, você pode rapidamente ver como um componente Blox é incluído a uma página JSP do portlet e como ele é exibido em uma página do portal.
2. Escreverá sua própria página JSP com um GridBlox.

As tarefas neste tutorial não fornecem detalhes sobre o desenvolvimento geral de portlets. Este tutorial foca nas tarefas específicas do DB2 Alphablox e assume que exista alguma familiaridade com os conceitos gerais de ambientes de portais e desenvolvimento de portlets.

### Pré-requisitos

- O DB2 Alphablox deve estar instalado no servidor WebSphere Portal Versão 5.1. Consulte o *Guia de Instalação* para maiores detalhes sobre a instalação.
- O servidor WebSphere Portal deve estar iniciado.
- Você deve ter acesso administrativo para o servidor WebSphere Portal.
- Você deve ter familiaridade com as funções administrativas e interface com o usuário no WebSphere Portal.
- Você deve ter conhecimento básico de Java e JSP.
- Você deve ter um editor JSP instalado.

Embora você possa utilizar qualquer editor JSP ou um editor de texto para este tutorial, quando desenvolver seus próprios portlets, utilize uma ferramenta de desenvolvimento recomendada pelo WebSphere Portal, como Rational Application Developer.

Este tutorial utiliza uma origem de dados pré-definida instalada com o DB2 Alphablox. Você pode utilizar esta origem de dados para rapidamente desenvolver um aplicativo básico. Não se preocupe em configurar uma origem de dados personalizada para este tutorial.

---

## Instalando Portlets de Amostra

A melhor forma de aprender como incluir um Blox em seu portlet, é instalar os portlets de amostra fornecidos com o DB2 Alphablox e carregá-los em uma página do portal. Isso permite examinar a estrutura básica do código JSP e corresponder essa estrutura à saída do portal.

Instale os portlets de amostra fornecidos com o DB2 Alphablox:

1. Abra o navegador e efetue logon no portal como um usuário administrativo (a URL está no formulário do `http://<seu_servidor_de_portal>:<porta>/wps/portal`).
2. Clique no botão **Administração**.
3. Na seção Gerenciamento de Portlet, clique em **Módulos da Web**. A página Gerenciar Módulos da Web aparece à direita.

4. Clique em **Instalar**. Você recebe um aviso para instalar o Módulo da Web.
5. Clique no botão **Navegar** e navegue para o diretório `installableApps` no diretório de instalação do DB2 Alphablox.
6. Escolha o `AlphabloxSamplePortlets.war` e clique em **Avançar**. O aplicativo DB2 Alphablox Sample Portlets com um portlet chamado DB2 Alphablox JSP Page Sample Portlet é exibido na tabela de aplicativos Portlet.
7. Clique em **Concluir**.

O aplicativo DB2 Alphablox Sample Portlets e os portlets incluídos estão agora instalados. Selecione o diretório `installedApps` na instalação do WebSphere Portal. O nome do diretório recém-criado inicia com DB2 Alpha e termina com um ID do portlet gerado dinamicamente no padrão de `_PA_x_x_xx.ear`.

## Executando os Portlets de Amostra

Para executar os portlets de amostra na página do portal:

1. Vá para a página do portal.
2. Crie ou edite uma página existente. Você pode criar uma nova página para testar este portlet de amostra ou pode clicar em uma página existente para editar a página.
3. Clique em um dos botões **Incluir Portlets** na página de layout do portal.
4. Digite DB2 no campo Procurar e pressione **Procurar**. A caixa de opções **Portlet de Amostra da Página JSP do DB2 Alphablox** aparece.
5. Selecione a caixa de opções e clique em **OK**.
6. Clique em **Concluído**.

Quando a página do portal é atualizada, um `PresentBlox` irá aparecer. Esse `PresentBlox` possui:

- Uma barra de menu no início da página
- Dois painéis da barra de ferramentas debaixo da barra de menus
- Um painel de layout de dados na parte esquerda, que lhe permite mover as dimensões pelos diferentes eixos
- Uma grade que mostra os dados em um formato tabular
- Um gráfico de barras tridimensional à direita

Este arquivo `present.jsp` é a página padrão a ser carregada, como especificado no servlet `BloxJSPPagePortlet` incluído na amostra. O código fonte para esse servlet de amostra está disponível no diretório `WEB-INF/src/`.

Você instalou com sucesso os portlets de amostra e incluiu um portlet com `Blox` na página do portal. A próxima tarefa é examinar a estrutura de código nesta página JSP.

## Examinando a Estrutura de uma Página JSP do Portlet com um Componente Blox

Nesta tarefa, você revê a estrutura de código de uma página JSP que contém um `Blox`. Todas as páginas JSP devem ter os mesmos elementos chave.

Para abrir um arquivo JSP:



1. Navegue para o diretório installedApps/ na instalação do WebSphere Portal, e localize a pasta de aplicativos criada mais recentemente que inicia com DB2 Alpha.
2. Navegue para o diretório PA\_x\_x\_xx.war/jsp/html/.
3. Abra o present.jsp no JSP ou no editor Java.
4. Examine o seguinte código e note os elementos chave:

```
<%@ page contentType="text/html"%>

<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>

<portletAPI:init/>

<%
String bloxName = portletResponse.encodeNamespace("presentBlox");
%>

<head>
  <blox:header />
</head>

<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
  <blox:data dataSourceName="canned" />
</blox:present>
```

Este bloco de código contém seis elementos chaves:

1. A primeira linha informa ao navegador que a saída é HTML:

```
<%@ page contentType="text/html"%>
```

2. O próximo conjunto de código especifica as duas bibliotecas de tags JSP utilizadas nesta página:

```
<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
```

O uri é um ponteiro para o local do diretório onde o arquivo do descritor da biblioteca de tags está localizado. Os valores prefix, definidos como blox e portletAPI, informam ao servidor para:

- Procure por tags nesta página que comecem com blox e, em seguida, processe o conteúdo utilizando a Biblioteca de Tags do Blox como definido no arquivo do descritor da biblioteca de tags.
- Procure por tags nesta página que comecem com portletAPI e, em seguida, processe o conteúdo utilizando a Biblioteca de Tags do Portlet como definido no arquivo do descritor da biblioteca de tags.

3. Em seguida, uma tag de inicialização do portlet é incluída:

```
<API_do_potlet:init/>
```

Esta tag fornece acesso aos objetos PortletRequest, PortletResponse e PortletConfig. Com o PortletResponse, você pode chamar o método encodeNamespace() para assegurar que o nome do Blox não conflite com outros objetos em outros portlets executados na mesma página.

4. A próxima tag codifica o espaço de nomes para o Blox a incluir na página:

```
<%
String bloxName = portletResponse.encodeNamespace("presentBlox");
%>
```

Isto permite a você criar um Blox e designar esse nome exclusivo a ele.

5. O próximo bloco de código inclui uma tag de cabeçalho do Blox que é requerida para renderização do Blox e comunicação entre o servidor e o cliente:

```
<head>
  <blox:header />
</head>
```

Esta tag é utilizada pelo DB2 Alphablox para incluir automaticamente o HTML requerido, JavaScript, e código CSS na seção de cabeçalho da página. Quando uma página é renderizada pelo servidor, esta tag resulta na inclusão de links CSS aos temas HTML e uma tag meta para evitar armazenamento em cache. Esta tag deve ser incluída a cada página JSP que contém componentes Blox ou os componentes não serão renderizados corretamente.

6. Inclua um PresentBlox, utilizando tags fornecidas na Biblioteca de Tags Blox:

```
<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
  <blox:data dataSourceName="canned" />
</blox:present>
```

- Este código inclui um PresentBlox com um id do presentBlox e um bloxName do xx\_x\_x\_xxx\_presentBlox, que é o resultado da codificação do espaço de nomes.
- Esse Blox possui uma largura de 800 pixels e a altura de 400 pixels (o padrão).
- Um DataBlox aninhado é incluído, com o atributo tag dataSourceName configurado para canned.

O atributo da tag id do PresentBlox é requerido. Ele especifica o nome do script Java a ser utilizado na página JSP. O atributo bloxName especifica o nome do objeto no servidor. Um bloxName codificado assegura que a instância é exclusiva no servidor.

**Nota:** Se nenhuma origem de dados for especificada, você verá a mensagem **Nenhum dado disponível** na grade. A origem de dados canned é pré-definida durante a instalação e inclui uma pequena quantidade de dados de amostra. Ela não requer a instalação e configuração de um banco de dados externo real e pode ser utilizada para resolução de problemas e aprendizado. Como não é necessário especificar uma consulta, o atributo query não existe no código.

Esta página não tem uma tag outmost <html> pois esta página JSP é exibida em uma página do portal com outros portlets. Nenhuma tag <html> ou <body> adicional é necessária.

Agora que você examinou a estrutura do JSP e aprendeu o código essencial para incluir as páginas JSP do portlet, na próxima tarefa você criará uma nova página JSP com um Blox diferente e especificará alguns atributos Blox comuns para que a página se ajuste melhor na página do portal.

---

## Criando sua Própria Página JSP do Portlet com Componentes Blox

Nesta tarefa, você criará uma nova página JSP com um GridBlox e configurará algumas de suas propriedades. O objetivo é que você fique familiarizado com o constructo da tag geral do Blox durante a criação de um GridBlox que seja bem ajustado em uma página típica de portal.

Um GridBlox possui um tamanho padrão de 400x400 pixels. Ele vem também com uma barra de menus e uma barra de ferramentas. Você configurará algumas da

propriedades comuns do GridBlox utilizadas, portanto, a grade utiliza apenas até 100 pixels de altura, com a barra de menus e a barra de ferramentas desativadas. Isso é feito, configurando as seguintes propriedades do GridBlox:

- height : configurado como 100 pixels
- menubarVisible: configurado como falso
- toolbarVisible: configurado como falso

Verifique se você atendeu aos requisitos especificados no Capítulo 2, “Tutorial: Construindo seu Primeiro Portlet com Componentes Blox”, na página 9 e instalou o aplicativo de portlet de amostra, conforme descrito na seção no “Instalando Portlets de Amostra” na página 9.

Siga as etapas a seguir:

1. Na janela do navegador onde o PresentBlox é exibido, clique no botão Editar do portlet (o botão com um ícone de lápis). Uma lista drop-down de seleção é apresentada.
2. Na lista drop-down, selecione “Grid Blox” e clique em **OK**.

Conforme a página é atualizada, você deve ver um GridBlox de 400x400. Esse GridBlox, por padrão, possui a barra de menus e a barra de ferramentas ativas. Você alterará seu tamanho para 400x100, com a barra de menus e a barra de ferramentas desativadas.

3. Navegue para o diretório PA\_x\_x\_xx.war/jsp/html/ no aplicativo DB2 Alphablox Sample Portlets instalado anteriormente.

4. Abra grid.jsp no editor JSP. Esta página é quase idêntica ao present.jsp, exceto que:

- a tag <blox:present> agora indica que o <blox:grid> e o valor do id são diferentes:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400">
  <blox:data dataSourceName="canned" />
</blox:grid>
```

- o valor de bloxName é diferente:

```
<%
  String bloxName = portletResponse.encodeNamespace("gridBlox");
%>
```

5. Configure a altura desse GridBlox como 100 pixels, incluindo o atributo height e configurando seu valor como 100:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100">
  <blox:data dataSourceName="canned" />
</blox:grid>
```

6. Desative a barra de menus e a barra de ferramentas na parte superior, configurando os atributos menubarVisible e toolbarVisible como false:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100"
  menubarVisible="false" toolbarVisible="false" >
  <blox:data dataSourceName="canned" />
</blox:grid>
```

Verifique se você digitou os nomes dos atributos corretamente, incluindo o uso de maiúsculas e minúsculas (“V” maiúsculo nos dois nomes de atributos). Verifique também se os atributos estão incluídos antes dos colchetes angulares de finalização (“>”).

7. Altere o espaço de nomes para myFirstGrid:

```
<%
  String bloxName = portletResponse.encodeNamespace("myFirstGrid");
%>
```

A alteração do espaço de nomes para esse GridBlox é para garantir que as alterações feitas sejam refletidas conforme você carrega esse JSP no portal. Como você carregou essa página anteriormente, já existe uma instância desse GridBlox executando no servidor para essa sessão. A menos que você altere o espaço de nomes, as alterações feitas não serão refletidas mesmo que você atualize a página. A alteração do espaço de nomes é uma maneira rápida de testar suas alterações em um ambiente de desenvolvimento. De maneira alternativa, também é possível abrir uma nova janela do navegador para que um novo objeto seja criado no servidor para a nova sessão.

8. Salve o arquivo.

Agora você está pronto para testar esse arquivo no WebSphere Portal.

Para testar as alterações feitas:

1. Volte para a página do portal.
2. Clique no botão Atualizar do Navegador para recarregar a página.

Você deve ver um GridBlox de 400x100 sem nenhuma barra de menus ou barra de ferramentas.

**Nota:** Essa lista drop-down de seleção do JSP não é uma funcionalidade interna do WebSphere Portal, mas criada por esse portlet de amostra. Selecione o arquivo `edit.jsp` e os arquivos de origem Java no diretório `WEB-INF/src/`.

---

## Criando um Projeto de Portlet para Utilizar Componentes Blox

Ao criar o projeto de portlet em sua ferramenta de desenvolvimento, verifique se o mapeamento do servlet `AlphabloxServer` e as referências da biblioteca de tags necessárias para os componentes do Blox estão incluídas no arquivo `web.xml` do projeto e se os arquivos `.tld` das Bibliotecas de Tags do DB2 Alphablox estão copiados sobre seu projeto.

1. Modifique o arquivo `web.xml` do projeto para incluir as seguintes linhas:

- Para obter a definição e o mapeamento do servlet:

```
<servlet>
  <servlet-name>AlphabloxServer</servlet-name>
  <servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
  </servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>AlphabloxServer</servlet-name>
  <url-pattern>/abx/*</url-pattern>
</servlet-mapping>
```

- Para referências à biblioteca de tags:

```
<taglib>
  <taglib-uri>bloxtld</taglib-uri>
  <taglib-location>/WEB-INF/tlds/blox.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>bloxformtld</taglib-uri>
  <taglib-location>/WEB-INF/tlds/bloxform.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>bloxlogictld</taglib-uri>
  <taglib-location>/WEB-INF/tlds/bloxlogic.tld</taglib-location>
</taglib>
<taglib>
```

```

    <taglib-uri>bloxreporttld</taglib-uri>
    <taglib-location>/WEB-INF/tlds/bloxreport.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>bloxportlettld</taglib-uri>
    <taglib-location>/WEB-INF/tlds/bloxportlet.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>bloxuitld</taglib-uri>
    <taglib-location>/WEB-INF/tlds/bloxui.tld</taglib-location>
</taglib>

```

2. Copie os arquivos .tld nas Bibliotecas de Tags do DB2 Alphablox no diretório WEB-INF/tlds/ do projeto. Esses arquivos estão localizados no <dir\_do\_db2alphablox>/bin/

em que <dir\_do\_db2alphablox> é o diretório de instalação do DB2 Alphablox.

## Configurando um Projeto de Portlet Utilizando o Rational Application Developer

O WebSphere Portal recomenda que você utilize o Rational Application Developer para desenvolvimento do portlet. Em particular, o Rational Application Developer Versão 6.0 é requerido para o WebSphere Portal Versão 5.1. O Rational Application Developer fornece um assistente que fornece instruções para a configuração e a criação de um projeto de portlet. Com base em sua seleção, ele configura a estrutura e o arquivo do descritor de implementação apropriados e cria automaticamente as classes Java necessárias para o controlador e páginas JSP para a visualização do portlet. Conforme você acompanha o assistente Novo Projeto de Portlet, verifique se o seguinte é configurado corretamente:

- Escolha o "stub do WebSphere Portal v5.1" para o servidor de destino. Isso é especificado na primeira tela do assistente, clicando no botão **Mostrar Avançado** >> .
- Inclua o mapeamento do servlet no DB2 Alphablox no arquivo web.xml criado no WebContent/WEB-INF/. Isso é feito digitando a seguinte definição de servlet e o código do mapeamento no web.xml do projeto:

```

<servlet>
  <servlet-name>AlphabloxServer</servlet-name>
  <servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>AlphabloxServer</servlet-name>
  <url-pattern>/abx/*</url-pattern>
</servlet-mapping>

```

- Inclua as Bibliotecas de Tags do Alphablox no web.xml do projeto. Siga as seguintes etapas:
  1. Com o web.xml aberto, clique na guia **Variáveis**.
  2. Role para a seção Referências a Bibliotecas de Tags na parte inferior.
  3. Clique em **Incluir**.
  4. Para **URL** e **Local**, digite os valores com base na seguinte tabela:

URL	Local
bloxtld	/WEB-INF/tlds/blox.tld
bloxformtld	/WEB-INF/tlds/bloxform.tld

URL	Local
bloxlogictld	/WEB-INF/tlds/bloxlogic.tld
bloxreporttld	/WEB-INF/tlds/bloxreport.tld
bloxportlettld	/WEB-INF/tlds/bloxportlet.tld
bloxuitld	/WEB-INF/tlds/bloxui.tld

Para cada par de URL e Local, é necessário clicar em **Concluir** para incluir a referência à biblioteca de tags e, em seguida, clicar em **Incluir** para incluir o próximo par.

**Nota:** Para obter detalhes sobre as diferentes bibliotecas de tags e seus usos, consulte Utilizando JavaServer Pages e Blox.

Finalmente, lembre-se de copiar os arquivos .tld das Bibliotecas de Tags do DB2 Alphablox no diretório WEB-INF/tlds/ do projeto. Esses arquivos estão localizados no

```
<dir_do_db2alphablox>/bin/
```

em que <dir\_do\_db2alphablox> é o diretório de instalação do DB2 Alphablox.

---

## Próximas Etapas

Você aprendeu como criar um portlet com componentes do Blox e a estrutura de código essencial. Para exibir dados de seus bancos de dados, é necessário:

1. Definir uma nova origem de dados para o DB2 Alphablox que aponte para a sua origem de dados.
2. Modificar o atributo dataSourceName do DataBlox para apontar para a origem de dados definida.
3. Incluir um atributo query com a cadeia de consulta no DataBlox.

Para obter detalhes sobre a criação de origens de dados, consulte o *Administrator's Guide*. Para aprender como criar consultas apropriadas, consulte Retrieving Data no *Developer's Guide* e a seção DataBlox do *Developer's Reference*.

Quando seus próprios dados aparecerem nas visualizações Blox, você também pode começar a explorar as muitas propriedades que podem ser configuradas, utilizando as tags do Blox.

---

## Dicas de Desenvolvimento de Portlets

Conforme você utiliza as informações fornecidas na documentação do DB2 Alphablox e continua o desenvolvimento do portlet, existem alguns tópicos específicos ao portlet e diretrizes gerais das quais você precisa estar ciente. O lista a seguir reúne essas diretrizes de desenvolvimento geral assim como ponteiros para seções específicas na documentação que são de seu interesse.

- Sempre codifique o nome do Blox que utiliza o espaço de nomes do portlet. Um espaço de nomes assegura um nome Blox exclusivo para a sessão J2EE atual.
- Sempre configure a largura e a altura de um Blox utilizando pixels. Fora do ambiente de portal, é possível configurar a largura e a altura para um valor de porcentagem como "50%" ou "100%". Os valores de porcentagem não funcionam no ambiente do portal, pois vários portlets coexistem na página.

- Não utilize URLs relativas para chamar os recursos em um portlet. As URLs devem ser codificadas utilizando o método `encodeURL()` da classe `PortletResponse`. Nos lugares aonde a documentação mostra o uso da URL relativa, você deve sempre codificar a URL.
- Para assegurar que o portlet do Blox utiliza um tema similar ao tema do portal, use o Portal Theme Utility. Este utilitário está disponível a partir de duas origens:
  - A home page do DB2 Alphablox.  
Na guia **Administração**, clique no link **Geral**. O Utility é listado sob a seção Portal. Consulte o *Administrator's Guide* e a ajuda on-line para obter detalhes.
  - O arquivo `AlphabloxAdminPortlets.war`.  
Essa é uma versão de portlet do Portal Theme Utility. Você pode instalar este portlet para executá-lo através do portal sem ter que efetuar login na home page do DB2 Alphablox.

Este utilitário combinará os estilos do ambiente do portal e os estilos no DB2 Alphablox, de modo que o Blox será exibido com cores e fontes semelhantes aos outros portlets na página.

- Selecione o tópico Blox Portlet Tag Library no *Developer's Guide* para obter os problemas que deseja levar em consideração durante a fase de planejamento.
- O DB2 Alphablox possui um Modelo eficiente de UI do Blox. O Modelo UI inclui um objeto `ClientLink` que lhe permite carregar uma URL especificada quando um componente Blox na página é clicado. Para o desenvolvimento do portlet, utilize a Biblioteca de Tags do Portlet Blox para criar o `ClientLink`. As tags manipularão a formação e a manipulação da URL de maneira dinâmica para que os links não se tornem stale depois que a página for atualizada. Consulte o tópico Planning for portlet development no *Developer's Guide* para obter detalhes.
- Para o portlet e ações de URIs, utilize a Biblioteca de Tags do Portlet Blox para criar o link do portlet ou da ação. Em seguida, é possível utilizar a API do Portlet do portal para processar a ação ou o link do portlet. Consulte o tópico Blox Portlet Tag Library no *Developer's Guide* para obter detalhes.





---

## Capítulo 3. Tutorial: Construindo Aplicativos Utilizando Ferramentas Rational Developer

Você pode utilizar ferramentas Rational Developer com o DB2 Alphablox Toolkit para desenvolver aplicativos mais rapidamente utilizando componentes DB2 Alphablox e APIs Java do DB2 Alphabloxs.

Ferramentas Rational Developer (Rational Application Developer ou Rational Web Developer) podem ser utilizadas com o DB2 Alphablox Toolkit, um conjunto de plug-ins baseados no Eclipse, para desenvolver e testar aplicativos construídos utilizando a tecnologia do DB2 Alphablox. Este tutorial o orientará em como configurar o ambiente de desenvolvimento integrado do Rational para permitir o uso do método e conclusão da tag, assim como aprimoramentos personalizados incluídos com o DB2 Alphablox Toolkit.

Para iniciar a configuração da ferramenta Rational Developer com o DB2 Alphablox Toolkit para desenvolver e testar os aplicativos do DB2 Alphablox.

---

### Preparando seu Ambiente de Desenvolvimento

Antes que você possa instalar e iniciar a utilização do DB2 Alphablox Toolkit, prepare seu ambiente de desenvolvimento certificando-se de que você tenha todos os softwares requeridos.

Para preparar o ambiente de desenvolvimento:

1. Instale uma ferramenta Rational Developer (Rational Application Developer ou Rational Web Developer) em sua estação de trabalho.
2. Instale quaisquer atualizações necessárias para assegurar que você tenha ao menos o Rational Application Developer ou o Rational Web Developer versão 6.0.0.1.
3. Se você planeja utilizar ambientes de teste integrados do WebSphere 5.1, certifique-se de que você atualizou a versão do Rational Developer para incluir a instalação deste opcional.

**Nota:** As instalações padrões das ferramentas Rational Developer não incluem os ambientes de teste integrados do WebSphere 5.1.

Você agora está pronto para executar as tarefas neste tutorial.

---

### Instalando o DB2 Alphablox Toolkit

A utilização do DB2 Alphablox Toolkit com a ferramenta Rational Developer facilitará o início do desenvolvimento de aplicativos baseado na web com o conteúdo do DB2 Alphablox.

O DB2 Alphablox Toolkit requer a utilização do Rational Application Developer ou do Rational Web Developer versão 6.0.0.1.

A instalação do DB2 Alphablox Toolkit em uma ferramenta Rational Developer economiza a execução de muitas etapas manuais para permitir a utilização da conclusão do conteúdo para métodos Java e tags customizadas JSP. Assistentes

customizados e folhas de dicas também podem ser utilizados para o orientar na criação de aplicativos com o conteúdo do DB2 Alphablox e na definição de instâncias do servidor WebSphere.

Para instalar o DB2 Alphablox Toolkit no Rational Application Developer ou no Rational Web Developer:

1. Insira o Disco de Instalação do DB2 Alphablox na unidade de CD de sua estação de trabalho.
2. No diretório plugin, localize o subdiretório UpdateSite.
3. Copie o diretório UpdateSite em um local conveniente no disco rígido. Por exemplo, copie o diretório no C:\DB2Alphablox\UpdateSite.
4. Inicie a ferramenta Rational Developer.
5. Na barra de menus, selecione **Ajuda > Atualizações de Software > Localizar e Instalar**.
6. Na janela de Instalação que se abre, selecione a opção **Procurar por novos recursos a serem instalados** e, em seguida, clique em **Avançar**.
7. Na janela Sites de Atualização a Visitar, clique no botão **Novo Site Local** e navegue para o local do diretório UpdateSite.
8. Clique em **Avançar**, selecione o recurso **DB2 Alphablox Toolkit** e clique em **Avançar** novamente.
9. Na janela seguinte, aceite o contrato de licença e selecione o local aonde o recurso DB2 Alphablox Toolkit será instalado.
10. Depois da instalação ser concluída, reinicie a ferramenta Rational Developer

Depois de reiniciar a ferramenta Rational Developer, os recursos DB2 Alphablox Toolkit estarão disponíveis.

---

## Instalando o DB2 Alphablox nos Ambientes de Teste Integrados do WebSphere

Nesta tarefa, você instala o DB2 Alphablox nos ambientes de teste integrados do WebSphere em uma ferramenta Rational Developer.

**Pré-requisitos:** O ambiente de teste integrado do WebSphere suportado deve ser instalado na ferramenta Rational Developer. Você precisa acessar o Disco de Instalação do DB2 Alphablox. Reveja e execute quaisquer etapas requeridas de pré-instalação, descritas na seção Tarefas de Pré-instalação do *Guia de Instalação do DB2 Alphablox*.

O DB2 Alphablox pode ser instalado em servidores de aplicativos WebSphere independentes ou em ambientes de teste integrados do WebSphere disponíveis na ferramenta Rational Developer. O resumo das etapas descritas abaixo explicam as diferenças na instalação ao instalar o DB2 Alphablox em tempos de execução do WebSphere na ferramenta Rational Developer.

Para instalar o DB2 Alphablox em ambientes de teste integrados do WebSphere

Instale o DB2 Alphablox seguindo as etapas na seção de Instalação do *Guia de Instalação do DB2 Alphablox*, com as seguintes exceções:

1. Na janela Configurar WebSphere do instalador, no campo **Diretório Raiz do WebSphere**, especifique o local do tempo de execução do WebSphere que você deseja utilizar como servidor de ambiente de teste integrado.

Por exemplo, para instalar o DB2 Alphablox no ambiente de teste integrado do WebSphere 6 na instalação padrão, selecione o caminho para o diretório base\_v6. Para uma instalação típica do Rational Application Developer, o caminho deve ser semelhante ao seguinte:

C:\Arquivos de programas\IBM\Rational\SDP\6.0\runtimes\base\_v6

2. Na janela Configurações do WebSphere, especifique um nome e senha para o administrador do WebSphere. Estes valores não são utilizados no ambiente de teste integrado do WebSphere, mas são requeridos pelo instalador do DB2 Alphablox.

**Importante:** Não execute as etapas de pós-instalação descritas no *Guia de Instalação do DB2 Alphablox*.

Depois de instalar o DB2 Alphablox no tempo de execução do WebSphere, você pode criar instâncias do servidor WebSphere para testar aplicativos DB2 Alphablox na ferramenta Rational Developer.

---

## Criando uma Instância do Servidor WebSphere

Para executar aplicativos ou arquivos JSP que contenham conteúdo do DB2 Alphablox, você precisa criar uma instância do servidor WebSphere que possa acessar serviços requeridos do DB2 Alphablox e classes Java.

**Pré-requisitos:** Configure a ferramenta Rational Developer. Instale o DB2 Alphablox Toolkit. Instale o DB2 Alphablox nos ambientes de teste integrados do WebSphere.

Para criar uma instância do servidor WebSphere para aplicativos de teste e arquivos JSP contendo conteúdo do DB2 Alphablox:

1. Abra a ferramenta Rational Developer.
2. Clique na guia **Servidores** (se a guia não estiver visível, selecione **Janela > Mostrar Visualização > Servidores**).
3. Clique com o botão direito do mouse dentro da janela de visualização e selecione **Novo > Servidor**. A janela Novo Servidor é aberta.
4. Digite localhost como o nome do host e selecione o tipo do servidor. Clique em **Avançar**.
5. Digite o número da porta do servidor e clique em **Avançar**.
6. Inclua os projetos disponíveis que deseja executar nesta instância do servidor na lista de projetos configurados. Clique em **Concluir**. A nova instância do servidor é aberta na visualização **Servidores**.

Se você criou uma instância do servidor WebSphere 5.1, você precisa executar a próxima etapa, "Criando Variáveis de Substituição do Servidor WebSphere 5.1." Para instâncias do servidor WebSphere 6, prossiga para "Incluindo Guests no Grupo de Administradores do DB2 Alphablox."

---

## Criando Variáveis de Substituição do Servidor WebSphere 5.1

Nesta tarefa, você modificará variáveis de substituição do servidor para instâncias do servidor WebSphere 5.1, para executar apropriadamente o DB2 Alphablox em ambientes de teste integrados do WebSphere.

1. Abra a visualização **Servidores** e clique duas vezes na instância do servidor do WebSphere 5.1 que deseja modificar.

2. Clique na guia **Configuração** da instância do servidor.
3. Selecione **Ativar Console de Administração** e desmarque **Ativar Cliente de Teste Universal**.
4. Clique na guia **Variáveis** da instância do servidor. A janela **Variáveis de Substituição** é aberta.
5. Inclua duas novas variáveis utilizando o botão **Incluir** (localizado próximo à lista **Configurações do Nó**).
  - a. Inclua uma variável denominada `WS_EAR_AlphabloxPlatform` e configure o valor como `$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear`.
  - b. Inclua uma segunda variável denominada `WS_EAR_AlphabloxStudio` e configure o valor como `$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear`.
6. Salve as alterações.

Para instâncias do servidor do WebSphere 5.1, é necessário seguir as etapas em "Configurando as Instâncias do Servidor do WebSphere 5.1" para concluir a modificação da instância do servidor.

---

## Configurando as Instâncias do Servidor WebSphere 5.1

Para instâncias do servidor WebSphere 5.1, você precisa configurar as instâncias do servidor para executar aplicativos e arquivos do DB2 Alphablox.

**Pré-requisitos:** Crie as variáveis de substituição do servidor WebSphere 5.1 requeridas.

Para configurar as instâncias do servidor WebSphere 5.1:

1. Abra a visualização **Servidores** na ferramenta Rational Developer e inicie a instância do servidor que deseja configurar.
2. Clique com o botão direito na instância do servidor e selecione **Executar Console Administrativo**.
3. Não digite um valor no campo **ID**. Clique em **OK**.
4. Clique no botão **OK** abaixo da tabela de configuração.
5. Abra a visualização de navegação **Aplicativos Corporativos** e clique no nome do aplicativo **ApplicationStudio**.
6. Configure **Binários do Aplicativo** como `$(WS_EAR_ApplicationStudio)`.
7. Selecione **Utilizar Metadados de Binários**.
8. Clique no botão **OK** abaixo da tabela de configuração.
9. Clique no botão **Salvar** no WebSphere Administrative Console.
10. Na janela Salvar para Configuração Principal, clique em **Salvar**.
11. Inicie os aplicativos `AlphabloxPlatform` e `ApplicationStudio` no WebSphere Administrative Console em **Aplicativos > Aplicativos Corporativos**.

A instância do servidor está configurada. A seguir você precisa executar a etapa "Incluir Guest no Grupo de Administradores do DB2 Alphablox".

---

## Incluindo o Usuário Guest no Grupo de Administradores do DB2 Alphablox

Incluir o usuário guest ao grupo de Administradores do DB2 Alphablox permite que você acesse as Páginas de Administração do DB2 Alphablox ao utilizar a ferramenta Rational Developer.

**Pré-requisitos:** Crie a instância do servidor WebSphere. Configure a instância do servidor WebSphere.

Para incluir o usuário guest ao grupo de Administradores do DB2 Alphablox.:

**Importante:** Não conceda direitos de administrador ao usuário guest em seus servidores WebSphere de produção.

1. Efetue login no console do DB2 Alphablox utilizando o seguinte comando telnet: `telnet localhost portNumber` . em que *portNumber* é a porta especificada durante a instalação do DB2 Alphablox.
2. No prompt do console do telnet, digite o seguinte comando de console do DB2 Alphablox: `set Administrators guest` e pressione Enter.
3. No console do telnet, digite `Salvar` e pressione Enter.
4. Feche a sessão de telnet.

A instância do servidor WebSphere está agora pronta para utilização.

---

## Criando Aplicativos do DB2 Alphablox

Quando você cria novos aplicativos no Rational, é necessário incluir conteúdo requerido do DB2 Alphablox para que os aplicativos sejam executados de maneira apropriada.

**Pré-requisitos:** Instale o DB2 Alphablox Toolkit na ferramenta Rational Developer.

1. Na barra de menus da ferramenta Rational Developer, selecione **Arquivo > Novo Projeto**. O assistente **Novo Projeto** é aberto.
2. Expanda a opção **Web**, e selecione **Projeto da Web Dinâmico** e, em seguida, clique em **Avançar**.
3. Digite um nome para o projeto e clique no botão **Mostrar Avançado**. Opções adicionais aparecem.
4. Selecione a versão do servlet apropriado e o servidor de destino.
5. Clique em **Avançar**. A janela **Opções de Recursos** é aberta.
6. Selecione a opção **Conteúdo do DB2 Alphablox** e clique em **Concluir**.

Seu novo projeto de aplicativo da Web agora é ativado pelo DB2 Alphablox. As bibliotecas de tags do DB2 Alphablox e da API Java Blox estão disponíveis e o arquivo descritor do aplicativo (`web.xml`) foi modificado para incluir informações requeridas do DB2 Alphablox. Agora você pode começar a incluir arquivos JSP contendo o conteúdo do DB2 Alphablox para o projeto.

---

## Criando Arquivos JSP com o Conteúdo do DB2 Alphablox

Nesta tarefa, você cria novos arquivos JSP com acesso para as bibliotecas de tags do Blox utilizando a ferramenta Rational Developer com o DB2 Alphablox Toolkit.

**Pré-requisitos:** Instale o DB2 Alphablox Toolkit na ferramenta Rational Developer.

1. Na visualização **Explorador de Projetos** do Rational Developer, selecione **Arquivo > Novo > Arquivo JSP**.
2. Digite o nome do arquivo no campo **Nome do Arquivo**.
3. Clique em **Configurar Opções Avançadas** e pressione **Avançar**.
4. Inclua as bibliotecas de tags do DB2 Alphablox que você utilizará no arquivo JSP.
  - a. Clique no botão **Incluir** para abrir a janela Incluir Bibliotecas de Tags.
  - b. Selecione as bibliotecas de tags do DB2 Alphablox que planeja utilizar no arquivo JSP.
  - c. Clique em **Avançar**.
5. Selecione **ISO 10646/Unicode(UTF-8)** da lista **Codificação**. A codificação UTF-8 é requerida para que os aplicativos DB2 Alphablox sejam executados corretamente.
6. Clique em **Concluir**. O novo arquivo JSP aparece na listagem do projeto.
7. Dê um clique duplo no nome do arquivo para abrir o arquivo na janela do editor JSP.
8. Posicione o cursor na tag `<head>` do HTML, mas depois das diretivas de taglib JSP do DB2 Alphablox incluídas acima.
9. Digite a guia do cabeçalho Blox digitando o seguinte na nova linha:  
`<blox:header/>` A tag de cabeçalho do Blox é necessária para incluir o JavaScript do DB2 Alphablox e arquivos CSS ao executar o arquivo JSP.

O novo arquivo JSP está agora ativado para acessar as bibliotecas de tags do DB2 Alphablox que você selecionou. Com o recurso Assistente de Conteúdo da ferramenta Rational Developer, você pode inserir tags Blox e atributos de tags para as bibliotecas de tags selecionadas.

---

## Acessando Origens de Dados do DB2 OLAP Server e do Essbase

Para acessar as origens de dados do DB2 OLAP Server ou do Essbase na ferramenta Rational Developer ao utilizar instâncias do servidor WebSphere 5.1, você precisa criar um arquivo em batch de inicialização que carregue as bibliotecas do cliente requeridas.

Pré-requisitos: Instale o DB2 Alphablox Toolkit. Instale o DB2 Alphablox nos ambientes de teste integrados do WebSphere 5.1. Configure as instâncias do servidor WebSphere 5.1. Instale o IBM DB2 OLAP Server ou Hyperion Essbase na máquina de desenvolvimento.

Devido às limitações nos ambientes de teste integrados do WebSphere 5.1 na ferramenta Rational Developer, você precisa criar um arquivo em batch de inicialização para a ferramenta Rational Developer para acessar origens de dados do DB2 OLAP Server ou Hyperion Essbase. Ao iniciar a ferramenta Rational Developer utilizando o arquivo em batch, as bibliotecas do cliente do Essbase requeridas são incluídas no caminho da biblioteca Java.

Para criar um arquivo em batch para ativar o acesso ao DB2 OLAP Server ou Essbase com instâncias do servidor WebSphere 5.1:

1. Utilizando um editor de texto, crie um novo documento de texto.
2. Inclua uma linha de código para chamar o arquivo `aasetup.bat` do DB2 Alphablox, localizado no diretório de instalação do DB2 Alphablox. Por exemplo, o seguinte código executará o arquivo `aasetup.bat` localizado na instalação do DB2 Alphablox especificada:

```
call C:\alphablox\analytics\bin\assetup.bat
```

3. Inclua uma segunda linha para chamar o arquivo `rationalsdp.exe` do RAD, que iniciará o RAD. Por exemplo, o seguinte código executará a ferramenta Rational Developer:

```
call C:\Arquivos de programas\IBM\Rational\SDP\6.0\rationalsdp.exe
```

4. Salve este arquivo como `startRAD.bat` no desktop da estação de trabalho (ou em outro local conveniente).

Ao dar um clique duplo no arquivo `startRAD.bat`, o arquivo `assetup.bat` do DB2 Alphablox é executado e configura os caminhos requeridos e as variáveis de ambiente, e posteriormente o Rational Application Developer é iniciado. As instalações do DB2 Alphablox incluem várias versões das bibliotecas do cliente Essbase que podem ser utilizadas com as origens de dados do DB2 OLAP Server. A versão das bibliotecas do cliente Essbase utilizadas com o DB2 Alphablox precisam corresponder com a versão do DB2 OLAP Server (ou Hyperion Essbase) que você está utilizando como uma origem de dados. Para modificar as bibliotecas do cliente Essbase utilizadas com o DB2 Alphablox, execute o DB2 OLAP Server / Essbase Client Library Utility (`ChangeEssbase.bat`), que pode ser localizado no diretório `db2_alphablox\analytics\bin`, em que `db2_alphablox` é o diretório raiz da instalação do DB2 Alphablox. Este utilitário modifica um dos arquivos em batch executados quando você inicia o RAD utilizando o arquivo em batch `startRAD` que você criou.

Segue abaixo um exemplo completo do arquivo `startRAD.bat` descrito acima:

```
call C:\alphablox\analytics\bin\assetup.bat  
call C:\Arquivos de programas\IBM\Rational\SDP\6.0\rationalsdp.exe
```

Quando você pretende acessar as origens de dados do DB2 OLAP Server ou Essbase, o arquivo `startRAD.bat` deve ser utilizado para iniciar a ferramenta Rational Developer.





---

## Capítulo 4. Tutorial: Construindo um Cubo do DB2 Alphablox, Utilizando o DB2 Cube Views

O tutorial o orienta através da criação de um cubo DB2 Alphablox construído utilizando um banco de dados de amostra do DB2 Cube Views.

As tarefas no tutorial não fornecem detalhes sobre a construção de cubos DB2 Alphablox customizados. O objetivo é mostrar a você como criar rapidamente um cubo DB2 Alphablox que pode ser utilizado na exploração de recursos do DB2 Alphablox Cube Server. A origem de dados resultante pode também ser utilizada para testar e construir aplicativos DB2 Alphablox.

### Pré-requisitos:

- Instalar o DB2 Alphablox. Consulte o *Guia de Instalação* para maiores detalhes sobre a instalação.
- Ter direitos de acesso para uma implementação do DB2 Cubes Views suportada que tenha o banco de dados de amostra do DB2 Cube Views CVSAMPLE instalado. Consulte o *Guia de Instalação* para maiores detalhes sobre as versões suportadas do DB2 Cube Views.

No tutorial, você aprenderá como construir um cubo DB2 Alphablox baseado no banco de dados de amostra do DB2 Cube Views CVSAMPLE. Durante o tutorial, você aprenderá as seguintes tarefas:

---

### Definindo uma Origem de Dados Relacionais do DB2

Nesta tarefa, você definirá uma definição da origem de dados do DB2 Alphablox para um banco de dados DB2.

Pré-requisitos: Os drivers JDBC requeridos do DB2 devem estar acessíveis para o DB2 Alphablox.

Um cubo do DB2 Alphablox requer que a origem de dados relacionais básica seja predefinida como uma origem de dados do DB2 Alphablox. Os cubos do DB2 Alphablox são gerados utilizando os metadados e dados disponíveis nos bancos de dados relacionais.

Para definir uma origem de dados do DB2 Alphablox para um banco de dados DB2:

1. Efetue logon nas Páginas de Administração do DB2 Alphablox como o usuário admin (ou como um usuário com direitos de administrador).
2. Clique na guia **Administração**.
3. Clique no link **Origem de Dados**.
4. Clique no botão **Criar**.
5. No menu **Adaptador**, selecione o driver JDBC apropriado do IBM DB2 para seu servidor de banco de dados.  
Selecione **Driver JDBC Tipo 4 do IBM DB2** ou **Driver IBM DB2 UDB no iSeries**.
6. No campo **Nome da Origem de Dados**, digite CVSAMPLE como o nome a ser utilizado para a origem de dados.

7. Digite os valores apropriados para o **Nome do Servidor**, **Número da Porta** e o **Nome do Banco de Dados** (o nome deve ser CVSAMPLE).

**Nota:** Se você precisar de ajuda para determinar os valores corretos para estes campos, entre em contato com o administrador do banco de dados.

8. Digite um **Nome de Usuário Padrão** e uma **Senha Padrão**.

O nome do usuário e a senha devem ser válidos no banco de dados relacional. O nome do usuário e a senha padrão são sempre utilizados quando um cubo do DB2 Alphablox acessa o banco de dados relacional. O usuário do banco de dados que você utiliza deve ter direitos de acesso de leitura no banco de dados.

**Nota:** O valor da opção **Utilizar Nome do Usuário e Senha do DB2 Alphablox** é ignorado quando a origem do banco de dados relacional está sendo utilizada para preencher um cubo do DB2 Alphablox. As ACLs (Listas de Controle de Acesso) podem ser utilizadas para restringir o acesso aos cubos do DB2 Alphablox. Para obter informações sobre ACLs, consulte o *Administrator's Guide*.

9. Os valores de **Número Máximo de Linhas** e **Número Máximo de Colunas** são ignorados quando a origem de dados está sendo utilizada para preencher um cubo do DB2 Alphablox. Você ainda podem digitar valores e eles serão utilizados quando outros aplicativos utilizarem a origem de dados.
10. Configure o valor de **Rastreo de JDBC Ativado** como **Não**, a não ser que você queira gravar informações de registro do JDBC no arquivo de registro do DB2 Alphablox. Ative o rastreo do JDBC apenas se você tiver encontrando problemas e precisar depurar suas causas.
11. Clique no botão **Salvar** para salvar a definição da origem de dados.

Você definiu agora a origem de dados do DB2 Alphablox para CVSAMPLE. Agora é possível criar uma definição de cubo do DB2 Alphablox para acessar metadados de cubos de Cube Views nesta origem de dados do DB2.

---

## Definindo uma Origem de Dados do Alphablox Cube Server Adaptor

Nesta tarefa, você definirá uma origem de dados do DB2 Alphablox que utiliza o Alphablox Cube Server Adapter.

Pré-requisitos: Crie a definição da origem de dados do DB2 Alphablox para o banco de dados CVSAMPLE do DB2.

Para definir uma origem de dados do DB2 Alphablox Cube Server Adapter:

1. Efetue logon nas Páginas de Administração do DB2 Alphablox como o usuário admin (ou como um usuário com direitos de administrador).
2. Clique na guia **Administração**.
3. Clique no link **Origem de Dados**.
4. Clique no botão **Criar**.
5. No menu **Adaptador**, selecione a opção **Alphablox Cube Server Adapter**.
6. No campo **Nome da Origem de Dados**, digite DB2AlphabloxCubes como o nome a ser utilizado para sua origem de dados.
7. Clique no botão **Salvar** para salvar a definição da origem de dados.

Você definiu uma origem de dados do DB2 Alphablox que pode ser utilizada para acessar cubos do DB2 Alphablox. A seguir você precisa definir um cubo do DB2 Alphablox que você possa acessar.

---

## Definindo um Cubo do DB2 Alphablox

Nesta tarefa, você definirá um cubo do DB2 Alphablox baseado nos metadados de um cubo CVSAMPLE do DB2 Cube Views.

**Pré-requisitos:** Defina a origem de dados relacional do DB2. Defina uma origem de dados do DB2 Alphablox Cube Server Adapter.

Para definir as propriedades gerais de um cubo DB2 Alphablox:

1. Efetue logon nas Páginas de Administração do DB2 Alphablox como o usuário admin (ou como um usuário com direitos de administrador).
2. Clique na guia **Administração**.
3. Clique no link **Cubos**.
4. Clique no botão **Criar**. A janela Administração do Cubo é aberta.
5. Defina o novo cubo:
  - a. No campo **Nome do Cubo do DB2 Alphablox**, digite **CVSales**.
  - b. Selecione a opção **Ativado**, localizada próxima do campo **Nome do Cubo do DB2 Alphablox**. A seleção desta opção resulta na inicialização automática do cubo quando o servidor é reiniciado.
  - c. No menu **Origem de Dados Relacionais**, selecione **MyDB2**, a origem de dados relacionais que você criou para este tutorial.
  - d. Deixe a opção **Função de Segurança** desmarcada. Esta opção pode ser utilizada para limitar os usuários que podem acessar um cubo específico.
6. Ativar as configurações do DB2 Cube Views e especificar os metadados a serem utilizados:
  - a. Selecione a opção **Ativar Configurações do DB2 Cube Views**.
  - b. No menu **Modelo de Cubo**, selecione **CVSAMPLE.Sales**.
  - c. No menu **Cubo**, selecione o cubo **Vendas Gerais**.
  - d. Selecione o botão de rádio **Utilizar Nomes de Negócios** para especificar os nomes. A seleção desta opção utiliza os nomes de membros que normalmente são mais significativos e legíveis.
  - e. Clique no botão **Importar Definição de Cubo**. O uso desta opção permite importar uma definição de cubo e pré-preencher medidas e dimensões em seu cubo do DB2 Alphablox. A definição do cubo que é importada reflete o cubo do DB2 Cube Views da maneira mais aproximada possível que o DB2 Alphablox pode corresponder, com base no suporte do metadados das DB2 Cube Views. Quando você estiver mais experiente no trabalho com cubos, poderá modificar medidas e dimensões para atender às suas necessidades.
  - f. Clique no botão **Mostrar Registro de Importação** para ver um registro que especifica informações e mensagens de depuração relacionadas à operação de importação. Para este tutorial, esta etapa é incluída apenas para ajudá-lo a familiarizar-se com este recurso.
  - g. Selecione a opção **Importar Definição de Cubo na Inicialização, Reconstrução e Edição**. Esta opção fará com que o seu cubo do DB2 Alphablox utilize a última definição do DB2 Cube Views a cada vez que o cubo do DB2 Alphablox for iniciado, reconstruído ou aberto para edição. Com mais familiaridade com o DB2 Alphablox e o DB2 Cube Views, você pode importar a definição de cubo e customizá-la.

7. Clique no botão **Salvar** para salvar a definição do cubo do DB2 Alphablox.

Você criou uma definição de cubo do DB2 Alphablox. Agora você pode iniciar o novo cubo CVSales.

---

## Iniciando o Cubo do seu DB2 Alphablox

Nesta tarefa, você inicia o cubo CVSales utilizando as Páginas de Administração do DB2 Alphablox.

**Pré-requisitos:** Defina a origem de dados relacional do DB2. Defina uma origem de dados do DB2 Alphablox Cube Server Adapter. Defina o cubo do DB2 Alphablox.

Para iniciar o cubo do CVSales:

1. Efetue logon nas Páginas de Administração do DB2 Alphablox como o usuário admin (ou como um usuário com direitos de administrador).
2. Clique na guia **Administração**.
3. Na seção **Gerenciamento do Tempo de Execução**, clique no link **Cubos**.
4. Na lista **Cubos do DB2 Alphablox**, selecione o cubo DB2 Alphablox que deseja iniciar.
5. Clique no botão **Iniciar**. Quando o cubo tiver sido iniciado, o campo de status exibirá **Executando**.

Você tem agora um cubo de amostra do DB2 Alphablox em execução. Você pode começar a construção de aplicativos utilizando o cubo do DB2 Alphablox que você acabou de criar. Você pode executar uma verificação rápida no novo cubo utilizando o aplicativo Query Builder, selecionando o cubo como a origem de dados e executando consultas MDX contra o cubo.

---

## Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

*Gerência de Relações Comerciais e Industriais da IBM Brasil, Av. Pasteur, 138-146, Botafogo, Rio de Janeiro, RJ, CEP 22290-240.*

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

*IBM World Trade Asia Corporation, Licensing, 2-31 Roppongi 3-chome, Minato-ku, Tokyo 106-0032, Japan*

*O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE NÃO-INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.*

Esta publicação pode conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aprimorar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais desse produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

*Gerência de Relações Comerciais e Industriais da IBM Brasil. Av. Pasteur, 138-146, Botafogo, Rio de Janeiro, RJ, CEP 22290-240.*

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença de Programas Internacional IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas de nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não-IBM. Dúvidas sobre os recursos de produtos não-IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

Estas informações contêm programas de aplicativos de exemplo na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de exemplo sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. O Cliente pode copiar, modificar e distribuir estes programas de exemplo de qualquer maneira

sem pagamento à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

---

## Marcas Registradas

Os termos a seguir são marcas ou marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países:

DB2	DB2 OLAP Server	DB2 Universal Database
IBM	WebSphere	

Alphablox e Blox são marcas ou marcas registradas da Alphablox Corporation nos Estados Unidos e/ou em outros países.

Java e todas as marcas registradas baseadas em Java são marcas registradas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

Linux é marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Outros nomes de empresas, produtos ou serviços podem ser marcas registradas ou marcas de serviço de terceiros.





---

## Índice Remissivo

### P

portlet

amostra 9

estrutura do JSP 10

### T

tutoriais

desenvolvimento de aplicativo 1







Número do Programa: 5724-L14

Impresso em Brazil

G517-8371-00

