



# **Red Hat JBoss Enterprise Application Platform 7.1**

## **Guía de inicio**

Para usar con Red Hat JBoss Enterprise Application Platform 7.1



# Red Hat JBoss Enterprise Application Platform 7.1 Guía de inicio

---

Para usar con Red Hat JBoss Enterprise Application Platform 7.1

## Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Resumen

En esta guía se proporciona la información básica para ayudar a los usuarios a comenzar a utilizar Red Hat JBoss Enterprise Application Platform 7.1.

## Table of Contents

<b>CAPÍTULO 1. INTRODUCCIÓN</b>	<b>4</b>
1.1. ACERCA DE RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7	4
1.2. ACERCA DE LA GUÍA DE INICIO	4
<b>CAPÍTULO 2. ADMINISTRACIÓN DE JBOSS EAP</b>	<b>5</b>
2.1. DESCARGAR E INSTALAR JBOSS EAP	5
2.1.1. Prerrequisitos de instalación	5
2.1.2. Descargar JBoss EAP	5
2.1.3. Instale JBoss EAP	5
2.2. INICIAR Y DETENER JBOSS EAP	6
2.2.1. Iniciar JBoss EAP	6
Iniciar JBoss EAP como un servidor autónomo	6
Inicie JBoss EAP en un dominio administrado	7
2.2.2. Parada de JBoss EAP	7
Pare una instancia interactiva de JBoss EAP	7
Pare una instancia de segundo plano de JBoss EAP	7
2.3. JBOSS EAP MANAGEMENT	8
2.3.1. Usuarios administrativos	8
2.3.1.1. Cómo adicionar un usuario administrativo	8
2.3.1.2. Cómo ejecutar la herramienta Agregar usuario de forma no interactiva	9
Cree un usuario que pertenezca a varios grupos	10
Especifique un archivo de propiedades alternativas	10
2.3.2. Interfaces de administración	10
2.3.2.1. CLI administrativa	10
Lance la CLI administrativa	11
Conéctese a un servidor en ejecución	11
Mostrar ayuda	11
Salga de la CLI administrativa	11
Ver parámetros del sistema	11
Actualice los parámetros del sistema	12
Inicie los servidores	12
2.3.2.2. Consola administrativa	12
2.3.3. Archivos de configuración	13
2.3.3.1. Archivos de configuración de servidor autónomo	13
2.3.3.2. Archivos de configuración de dominio administrado	14
2.3.3.3. Hacer copia de seguridad de los datos de configuración	14
2.3.3.4. Instantáneas de archivos de configuración	15
Tomar una instantánea	15
Listar las instantáneas	15
Borre la instantánea	16
Inicie el servidor con una instantánea	16
2.3.3.5. Reemplazo de propiedad	16
Expresiones anidadas	17
Reemplazo de las propiedades basadas en el descriptor	17
2.4. CONFIGURACIÓN DE RED Y PUERTO	18
2.4.1. Interfaces	18
2.4.1.1. Configuración de interfaz predeterminada	19
2.4.1.2. Configuración de interfaces	19
Agregue una interfaz con un valor NIC	20
Agregue la interfaz con varios valores condicionales	20
Actualice el atributo de interfaz	20

Agregue una interfaz a un servidor en un dominio administrado	20
2.4.2. Enlaces de socket	21
2.4.2.1. Puertos de administración	21
2.4.2.2. Enlaces de sockets predeterminados	21
Servidor autónomo	22
Dominio administrado	22
2.4.2.3. Configuración de enlaces de socket	23
2.4.2.4. Port Offsets	24
2.4.3. Direcciones IPv6	24
Configurar la pila JVM para direcciones IPv6	25
Actualice las declaraciones de interfaz para direcciones IPv6	25
<b>CAPÍTULO 3. DESARROLLO DE APLICACIONES MEDIANTE JBOSS EAP .....</b>	<b>26</b>
3.1. VISIÓN GENERAL	26
3.2. CONFIGURACIÓN DEL ENTORNO DE DESARROLLO	26
3.3. CÓMO USAR EJEMPLOS DE INICIOS RÁPIDOS	26
3.3.1. Acerca de Maven	26
3.3.2. Uso de Maven con Quickstarts	27
3.3.3. Descargar y ejecutar los Quickstarts	27
3.3.3.1. Descargar los Quickstarts	27
3.3.3.2. Ejecutar los Quickstarts en JBoss Developer Studio	27
3.3.3.3. Ejecute los Inicios rápidos mediante la línea de comandos	35
3.4. REVISAR LOS EJEMPLOS DE INICIOS RÁPIDOS	35
3.4.1. Exploración del inicio rápido helloworld	35
Prerrequisitos	36
Examine la estructura del directorio	36
Examine el código	36
3.4.2. Explore el inicio rápido de numberguess	37
Prerrequisitos	38
Examine los archivos de configuración	38
3.4.2.1. Examine el código JSF	39
3.4.2.2. Examine los archivos de clases	40
<b>APÉNDICE A. MATERIAL DE REFERENCIA .....</b>	<b>45</b>
A.1. ARGUMENTOS DE TIEMPO DE EJECUCIÓN DEL SERVIDOR	45
A.2. ARGUMENTOS DE LA HERRAMIENTA ADD-USER	48
A.3. ATRIBUTOS DE INTERFAZ	50
A.4. ATRIBUTOS DE ENLACES DE SOCKETS	51
A.5. ENLACES DE SOCKETS PREDETERMINADOS	52



# CAPÍTULO 1. INTRODUCCIÓN

## 1.1. ACERCA DE RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7

Red Hat JBoss Enterprise Application Platform 7 (JBoss EAP) es una plataforma de middleware desarrollada con estándares abiertos y conforme a la especificación de Java Enterprise Edition 7.

JBoss EAP incluye una nueva estructura modular, la cual permite la habilitación de servicios solo cuando se requieran, mejorando así, la velocidad de arranque.

La consola administrativa y la interfaz de línea de comandos administrativa (CLI) hacen innecesaria la modificación de archivos de configuración XML y agregan la habilidad para utilizar scripts y automatizar tareas.

JBoss EAP proporciona dos modos operativos para instancias JBoss EAP: el servidor autónomo o el dominio administrado. El servidor autónomo representa la ejecución de JBoss EAP como una instancia de servidor sencilla. El modo operativo de dominio administrado permite la administración de múltiples instancias JBoss EAP desde un punto de control único.

Además, JBoss EAP 7 incluye APIs y marcos de trabajo de desarrollo para desarrollar rápidamente aplicaciones Java EE seguras y escalables.

## 1.2. ACERCA DE LA GUÍA DE INICIO

Esta guía tiene como objeto hacer que usted pueda implementar JBoss EAP tan rápido como sea posible. Cubre tareas [administrativas](#), como instalación, administración y configuración básicas para JBoss EAP. Esta guía también ayuda a los [desarrolladores](#) a comenzar a escribir aplicaciones Java EE 7 mediante los inicios rápidos de JBoss EAP.

Para obtener más información, consulte el [paquete de documentación de JBoss EAP](#).



## CAPÍTULO 2. ADMINISTRACIÓN DE JBOSS EAP

### 2.1. DESCARGAR E INSTALAR JBOSS EAP

En esta guía se proporcionan instrucciones básicas para descargar e instalar JBoss EAP mediante la instalación ZIP, que es independiente de la plataforma.

Para conocer detalles adicionales, consulte la [Guía de instalación](#), que incluye instrucciones para instalar JBoss EAP mediante el instalador gráfico o los métodos de instalación de paquetes RPM.

#### 2.1.1. Prerrequisitos de instalación

Verificar que los siguientes prerrequisitos se hayan cumplido antes de instalar JBoss EAP:

##### Prerrequisitos comunes

- Su sistema recibe soporte según [Las configuraciones que reciben soporte de JBoss EAP 7](#).
- Su sistema esté actualizado con las actualizaciones y erratas emitidas de Red Hat.

##### Prerrequisitos de instalación ZIP

- El usuario que ejecutará JBoss EAP debe tener acceso de lectura y escritura para el directorio de instalación.
- Debe instalarse el kit de desarrollo de Java deseado.
- Para Hewlett-Packard HP-UX, se debe instalar una herramienta **unzip**.
- Para Windows Server, se deben establecer las variables de entorno **JAVA\_HOME** y **PATH**.

#### 2.1.2. Descargar JBoss EAP

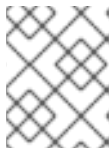
El archivo ZIP de JBoss EAP está disponible en el Portal del cliente de Red Hat. La instalación del archivo ZIP es independiente de la plataforma.

1. Ingrese al [Portal del cliente de Red Hat](#)
2. Haga clic en **Downloads**.
3. Haga clic en **Red Hat JBoss Enterprise Application Platform** en la lista **Descargas de productos**
4. En el menú desplegable **Versión**, seleccione **7.1**.
5. Busque **Red Hat JBoss Enterprise Application Platform 7.1.0** en la lista y haga clic en el enlace **Descargar**.

#### 2.1.3. Instale JBoss EAP

Una vez haya sido descargado el archivo de instalación EAP ZIP, puede ser instalado al extraer el contenido del paquete.

1. Si es necesario, desplace el archivo ZIP al servidor y el sitio donde JBoss EAP debe ser instalado.

**NOTA**

El usuario que ejecutará JBoss EAP debe tener acceso de lectura y escritura para este directorio.

2. Extraer el archivador.

```
$ unzip jboss-eap-7.1.0.zip
```

**NOTA**

Para Windows Server, haga clic derecho en el archivo ZIP y seleccione **Extract All**.

El directorio creado al extraer el archivo ZIP es el directorio del nivel superior para la instalación de JBoss EAP. Este directorio se conoce como **EAP\_HOME**.

## 2.2. INICIAR Y DETENER JBOSS EAP

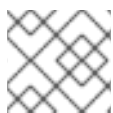
### 2.2.1. Iniciar JBoss EAP

JBoss EAP es compatible con Red Hat Enterprise Linux, Windows Server, Oracle Solaris y Hewlett-Packard HP-UX, y se ejecuta en un modo operativo de servidor autónomo o dominio administrado. El comando específico para iniciar JBoss EAP depende de la plataforma subyacente y del modo operativo deseado.

En un principio, los servidores se inician en modo suspendido y no aceptan solicitudes hasta que todos los servicios requeridos se hayan iniciado, en cuyo momento, los servidores se colocarán en estado de ejecución normal y podrán comenzar a aceptar solicitudes.

#### Iniciar JBoss EAP como un servidor autónomo

```
$ EAP_HOME/bin/standalone.sh
```

**NOTA**

Para Windows Server, use el script **EAP\_HOME\bin\standalone.bat**.

Este script de arranque usa el archivo **EAP\_HOME/bin/standalone.conf** (o **standalone.conf.bat** para Windows Server) para establecer algunas preferencias predeterminadas, como opciones JVM. Puede personalizar los parámetros en este archivo.

JBoss EAP usa el archivo de configuración **standalone.xml** de forma predeterminada, pero puede iniciarse mediante uno diferente. Para obtener más información sobre los archivos de configuración autónomos y la forma de usarlos, consulte la sección [Configuración de servidor autónoma](#).

Para obtener un listado de todos los argumentos de scripts de arranque y sus propósitos, use el argumento **--help** o consulte la sección de argumentos o vea la sección [Argumentos de tiempo de ejecución de servidor](#).

### Inicie JBoss EAP en un dominio administrado

El controlador de dominio se debe iniciar antes que los servidores en los grupos de servidores del dominio. Utilice este script para iniciar primero el controlador de dominio y, luego, para cada controlador de host asociado.

```
$ EAP_HOME/bin/domain.sh
```



#### NOTA

Para Windows Server, use el script **EAP\_HOME\bin\domain.bat**.

Este script de arranque usa el archivo **EAP\_HOME/bin/domain.conf** (o **domain.conf.bat** para Windows Server) para establecer algunas preferencias predeterminadas, como opciones JVM. Puede personalizar los parámetros en este archivo.

JBoss EAP usa el archivo de configuración **standalone.xml** de forma predeterminada, pero puede iniciarse mediante uno diferente. Para obtener más información sobre los archivos de configuración autónomos y la forma de usarlos, consulte la sección [Archivos de configuración de dominio administrado](#).

Al configurar un dominio administrado, se deberán pasar argumentos adicionales en el script de arranque. Para obtener un listado de todos los argumentos de scripts de arranque disponibles y sus propósitos, use el argumento **--help** o consulte la sección [Argumentos de tiempo de ejecución del servidor](#).

## 2.2.2. Parada de JBoss EAP

La forma de parar a JBoss EAP depende de la forma como se ha iniciado.

### Pare una instancia interactiva de JBoss EAP

Presione **Ctrl+C** en el terminal donde se inició JBoss EAP.

### Pare una instancia de segundo plano de JBoss EAP

Use la CLI administrativa para conectarse a la instancia que está en ejecución y apague el servidor.

1. Lance la CLI administrativa.

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

2. Ejecute el comando **shutdown**.

```
shutdown
```



#### NOTA

Cuando se ejecuta en un dominio administrado, debe especificar el nombre de host para el apagado mediante el argumento **--host** con el comando **shutdown**.

## 2.3. JBOSS EAP MANAGEMENT

JBoss EAP usa una configuración simplificada con un archivo de configuración por servidor autónomo o dominio administrado. La configuración predeterminada para un servidor autónomo se almacena en el archivo

**`EAP_HOME/standalone/configuration/standalone.xml`** y la configuración

predeterminada para un dominio administrado se almacena en el archivo

**`EAP_HOME/domain/configuration/domain.xml`**. Además, la configuración predeterminada para un controlador de host se almacena en el archivo

**`EAP_HOME/domain/configuration/host.xml`**.

JBoss EAP se puede configurar mediante la CLI de administración de la línea de comando, la consola de administración basada en la Web, Java API o HTTP API. Los cambios realizados a través de estas interfaces de administración persisten automáticamente y la API de administración sobrescribe los archivos de configuración XML. Los métodos preferidos son la CLI de administración y la consola de administración. No se recomienda modificar de forma manual los archivos de configuración XML.

### 2.3.1. Usuarios administrativos

La configuración predeterminada de JBoss EAP proporciona autenticación local, para que un usuario pueda tener acceso a la CLI de administración en el host local sin requerir autenticación.

Sin embargo, debe agregar un usuario de administración si desea acceder a la CLI de administración de manera remota o usar la consola de administración, que se considera como acceso remoto, incluso si el tráfico se origina en el host local. Si intenta acceder a la consola de administración antes de añadir un usuario de administración, recibirá un mensaje de error.

Si JBoss EAP está instalado usando el instalador gráfico, se crea un usuario de administración durante el proceso de instalación.

Esta guía abarca la administración de usuario simple para JBoss EAP mediante el script **`add-user`**, que es una herramienta que permite añadir nuevos usuarios a los archivos de propiedades para la autenticación externa.

Para obtener opciones de autenticación y autorización más avanzadas, como LDAP o Control de acceso basado en roles (RBAC), consulte la sección [Autenticación de administración principal](#) de la guía *Arquitectura de seguridad* de JBoss EAP.

#### 2.3.1.1. Cómo adicionar un usuario administrativo

1. Ejecute el script de la herramienta **`add-user`** y siga las indicaciones.

```
$ EAP_HOME/bin/add-user.sh
```



#### NOTA

Para Windows Server, use el script **`EAP_HOME\bin\add-user.bat`**.

2. Presione **ENTER** para seleccionar la opción predeterminada para agregar un usuario de administrativo.

Este usuario se agregará al *ManagementRealm* y se autorizará para realizar operaciones administrativas mediante la consola de administración o la CLI de

administración. La otra selección **(b)** agrega un usuario a *ApplicationRealm*, que se utiliza para aplicaciones y no proporciona permisos particulares.

3. Ingrese el nombre de usuario y la contraseña deseados. Se le solicitará confirmar la contraseña.



#### NOTA

Los nombres de usuario solo pueden contener los siguientes caracteres, en cualquier cantidad y orden:

- Caracteres alfanuméricos (a-z, A-Z, 0-9)
- Guiones (-), puntos (.), comas (,), arroba (@)
- Barra diagonal inversa (\)
- Signo igual (=)

De manera predeterminada, JBoss EAP admite contraseñas débiles, pero emite una advertencia.

Consulte la sección [Restricciones para establecer la contraseña de la herramienta Añadir usuario](#) de la *Guía de configuración* de JBoss EAP para obtener detalles sobre cómo cambiar este comportamiento predeterminado.

4. Ingrese una lista separada por comas de los grupos a los cuales pertenece el usuario. Si usted no desea que el usuario pertenezca a ninguno de ellos presione **ENTER** para dejarlo en blanco.
5. Revise la información e ingrese **yes** para confirmar.
6. Determine si este usuario representa una instancia remota de servidor JBoss EAP. Para un usuario administrativo básico, ingrese **no**.  
Un tipo de usuario que necesite ser agregado al *ManagementRealm* es un usuario que representa otra instancia de JBoss EAP, la cual debe ser capaz de autenticarse para unirse como un miembro de un clúster. Si este es el caso, entonces la respuesta es **yes** para indicar que se le otorgará un valor secreto en hash que representa la contraseña, la cual deberá ser agregada a un archivo de configuración diferente.

También puede crear usuarios de modo no interactivo pasando parámetros al script **add-user**. Este método no se recomienda en sistemas compartidos, ya que las contraseñas estarán visibles en archivos de registro y de historiales. Para obtener más información, consulte [Cómo ejecutar la herramienta Agregar usuario de forma no interactiva](#)

#### 2.3.1.2. Cómo ejecutar la herramienta Agregar usuario de forma no interactiva

Ejecute el script **add-user** de forma no interactiva pasando argumentos en la línea de comandos. Como mínimo, se debe proporcionar el nombre de usuario y la contraseña.



## AVISO

Este enfoque no se recomienda en sistemas compartidos, ya que las contraseñas estarán visibles en archivos de registro y de historiales.

### Cree un usuario que pertenezca a varios grupos

El siguiente comando agrega un usuario de administración (**mgmtuser1**) con los grupos **guest** y **mgmtgroup**.

```
$ EAP_HOME/bin/add-user.sh -u 'mgmtuser1' -p 'password1!' -g
'guest,mgmtgroup'
```

### Especifique un archivo de propiedades alternativas

De forma predeterminada, la información de usuario y grupo creada mediante el script **add-user** se almacena en los archivos de propiedades localizadas en el directorio de configuración de servidor.

La información de usuario se almacena en los siguientes archivos de propiedades:

- **EAP\_HOME/standalone/configuration/mgmt-users.properties**
- **EAP\_HOME/domain/configuration/mgmt-users.properties**

La información de grupo es almacenada en los siguientes archivos de propiedades:

- **EAP\_HOME/standalone/configuration/mgmt-groups.properties**
- **EAP\_HOME/domain/configuration/mgmt-groups.properties**

Estos directorios predeterminados y nombres de archivos pueden sobrescribirse. El siguiente comando agrega un nuevo usuario especificando un nombre diferente y un sitio para los archivos de propiedades de usuario.

```
$ EAP_HOME/bin/add-user.sh -u 'mgmtuser2' -p 'password1!' -sc
'/path/to/standaloneconfig/' -dc '/path/to/domainconfig/' -up
'newname.properties'
```

El nuevo usuario fue agregado a los archivos de propiedades localizados en **/path/to/standaloneconfig/newname.properties** y **/path/to/domainconfig/newname.properties**. Observe que estos archivos ya deben existir o de lo contrario, verá un error.

Para obtener un listado completo de los argumentos **add-user** disponibles y sus propósitos, use el argumento **--help** o consulte la sección [Referencia de los argumentos de la herramienta add-user](#).

## 2.3.2. Interfaces de administración

### 2.3.2.1. CLI administrativa

La interfaz de la línea de comandos de administración (CLI) es una herramienta de administración de la línea de comandos para JBoss EAP.

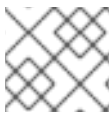
Use la CLI administrativa para iniciar y detener servidores, implementar y desimplementar aplicaciones, configurar parámetros del sistema y realizar otras tareas administrativas. Las operaciones se pueden realizar en modo de lotes (batch) para permitir la ejecución de múltiples tareas como un grupo.

Varios comandos comunes de la terminal están disponibles, tales como **ls**, **cd** y **pwd**. La CLI administrativa también soporta el tabulador de completado.

Para obtener más información sobre el uso de la CLI de administración, incluidos los comandos y operaciones, sintaxis y para ejecución en modo batch, consulte la [Guía de CLI de administración](#) de JBoss EAP.

### Lance la CLI administrativa

```
$ EAP_HOME/bin/jboss-cli.sh
```



#### NOTA

Para Windows Server, use el script **EAP\_HOME\bin\jboss-cli.bat**.

### Conéctese a un servidor en ejecución

```
Conéctese
```

O puede lanzar la CLI de administración y conectarse en un paso mediante el comando **EAP\_HOME/bin/jboss-cli.sh --connect**.

### Mostrar ayuda

Use el siguiente comando para obtener ayuda general.

```
help
```

Use el marcador **--help** en un comando para recibir instrucciones sobre cómo usar ese comando específico. Por ejemplo, para recibir información sobre cómo usar **deploy**, se ejecuta el siguiente comando.

```
deploy --help
```

### Salga de la CLI administrativa

```
quit
```

### Ver parámetros del sistema

El siguiente comando usa la operación **read-attribute** para desplegar si la fuente de datos de ejemplo está habilitada.

```
/subsystem=datasources/data-source=ExampleDS:read-attribute(name=enabled)
{
  "outcome" => "success",
  "result" => true
}
```

■

Cuando se ejecuta en un dominio administrado, debe especificar el perfil que va a actualizar anteponiendo el comando con **/profile=PROFILE\_NAME**.

```
/profile=default/subsystem=datasources/data-source=ExampleDS:read-attribute(name=enabled)
```

### Actualice los parámetros del sistema

El siguiente comando usa la operación **write-attribute** para inhabilitar el ejemplo de fuente de datos.

```
/subsystem=datasources/data-source=ExampleDS:write-attribute(name=enabled,value=false)
```

### Inicie los servidores

La CLI administrativa también puede iniciar y parar servidores cuando se ejecutan en un dominio administrado.

```
/host=HOST_NAME/server-config=server-one:start
```

## 2.3.2.2. Consola administrativa

La consola administrativa es una herramienta de administración basada en la red para JBoss EAP.

Use la consola administrativa para iniciar y detener servidores, implementar y desimplementar aplicaciones, afinar configuraciones del sistema y realizar modificaciones persistentes a la configuración del servidor. La consola de administración también tiene la habilidad de realizar tareas administrativas con notificaciones en vivo cuando algún cambio realizado por el usuario actual requiere que la instancia del servidor sea reiniciada o recargada.

En un dominio administrado, las instancias del servidor y los grupos del servidor en el mismo dominio, se pueden administrar centralmente desde la consola de administración del controlador de dominio.

Para una instancia JBoss EAP que se ejecute en un host local mediante el puerto de administración predeterminado, la consola administrativa, puede ser accedida a través de un navegador en <http://localhost:9990/console/App.html>. Deberá autenticar un usuario que tenga permisos para acceder a la consola administrativa.

La consola administrativa proporciona las siguientes pestañas para navegar y administrar su servidor autónomo JBoss EAP o el dominio administrado.

### Carpeta personal

Aprenda cómo llevar a cabo varias tareas administrativas y de configuración. Haga el tour para familiarizarse con la consola administrativa de JBoss EAP.

### Implementaciones

Agregue, quite y habilite implementaciones. En un dominio administrado, asigne las implementaciones a los grupos de servidores.

### Configuración



Configure los subsistemas disponibles, los cuales proporcionan funcionalidades tales como servicios Web, mensajería o alta disponibilidad. En un dominio administrado, administre los perfiles que contienen diferentes configuraciones de subsistemas.

### Tiempo de ejecución

Vea el tiempo de ejecución, tal como el estado de servidor, el uso de JVM y los registros de servidor. En un dominio administrado, administre sus hosts, los grupos de servidor y los servidores.

### Control de Acceso

Asigne los roles a usuarios y grupos cuando utilice el control de acceso basado en roles.

### Uso de parches

Aplique parches a sus instancias JBoss EAP



#### NOTA

Para realizar un recorrido de su consola de administración, haga clic en el enlace **Tomar el tour** en la página de inicio de la consola.

## 2.3.3. Archivos de configuración

### 2.3.3.1. Archivos de configuración de servidor autónomo

Los archivos de configuración autónomos se localizan en el directorio **EAP\_HOME/standalone/configuration/**. Existe un archivo para cada uno de los cuatro perfiles predefinidos (*default*, *ha*, *full*, *full-ha*, *load-balancer*).

**Tabla 2.1. Archivos de configuración autónomos**

Archivo de configuración	Propósito
<b>standalone.xml</b>	Este archivo de configuración autónomo es la configuración predeterminada utilizada cuando inicia el servidor autónomo. Contiene toda la información sobre el servidor, incluidos subsistemas, redes, implementaciones, enlaces de sockets y otra información configurable. No proporciona los subsistemas necesarios para mensajería o alta disponibilidad.
<b>standalone-ha.xml</b>	Este archivo de configuración autónomo incluye todos los subsistemas predeterminados y agrega los subsistemas <b>modcluster</b> y <b>jgroups</b> para alta disponibilidad. No proporciona los subsistemas necesarios para mensajería.
<b>standalone-full.xml</b>	Este archivo de configuración autónomo incluye todos los subsistemas predeterminados y agrega los subsistemas <b>messaging-activemq</b> y <b>iiop-openjdk</b> . No proporciona los subsistemas necesarios para alta disponibilidad.
<b>standalone-full-ha.xml</b>	Este archivo de configuración incluye soporte para cada subsistema posible, incluidos los de mensajería y alta disponibilidad.

Archivo de configuración	Propósito
<b>standalone-load-balancer.xml</b>	Este archivo de configuración autónomo incluye los subsistemas mínimos necesarios para usar el equilibrador de carga frontal <code>mod_cluster</code> incorporado para equilibrar la carga de otras instancias de JBoss EAP.

El servidor JBoss EAP utiliza de forma predeterminada el archivo **standalone.xml**. Para iniciar JBoss EAP con una configuración diferente, use el argumento **--server-config**. Por ejemplo:

```
$ EAP_HOME/bin/standalone.sh --server-config=standalone-full.xml
```

### 2.3.3.2. Archivos de configuración de dominio administrado

Los archivos de configuración de dominio administrado se localizan en el directorio **EAP\_HOME/domain/configuration/**.

**Tabla 2.2. Archivos de configuración de dominio administrado**

Archivo de configuración	Propósito
<b>domain.xml</b>	Este es el archivo de configuración principal para un dominio administrado. Solamente el maestro de dominio lee este archivo. Este archivo contiene las configuraciones para todos los perfiles ( <i>default, ha, full, full-ha, load-balancer</i> ).
<b>host.xml</b>	Este archivo incluye información de configuración para host físico en un dominio administrado, tal como interfaces, enlaces de socket, el nombre del host y otra información específica de host. El archivo <b>host.xml</b> incluye todas las funcionalidades de <b>host-master.xml</b> y <b>host-slave.xml</b> , las cuales se describen a continuación:
<b>host-master.xml</b>	Este archivo incluye únicamente la información de configuración necesaria para ejecutar un servidor como el controlador de dominio maestro.
<b>host-slave.xml</b>	Este archivo incluye únicamente la información de configuración necesaria para ejecutar un servidor como el controlador de host del dominio administrado.

El servidor JBoss EAP utiliza de forma predeterminada el archivo **host.xml**. Para iniciar JBoss EAP con una configuración diferente, use el argumento **--host-config**. Por ejemplo:

```
$ EAP_HOME/bin/domain.sh --host-config=host-master.xml
```

### 2.3.3.3. Hacer copia de seguridad de los datos de configuración

Con el fin de restaurar más adelante la configuración del servidor JBoss EAP, se debería hacer una copia de seguridad de los siguientes elementos:

- ***EAP\_HOME/standalone/configuration/***
  - Haga una copia de seguridad de todo el directorio para guardar los datos de usuario, la configuración del servidor y los parámetros de ingreso para servidores autónomos.
- ***EAP\_HOME/domain/configuration/***
  - Haga una copia de seguridad de todo el directorio para guardar el usuario y los datos de perfil, el dominio, la configuración de host y los parámetros para dominios administrados.
- ***EAP\_HOME/modules/***
  - Haga una copia de seguridad de los módulos personalizados.
- ***EAP\_HOME/welcome-content/***
  - Haga una copia de seguridad del contenido de bienvenida.
- ***EAP\_HOME/bin/***
  - Haga una copia de seguridad de los scripts personalizados o los archivos de configuración de arranque.

#### 2.3.3.4. Instantáneas de archivos de configuración

Para ayudar en el mantenimiento y la administración del servidor, JBoss EAP crea una versión con marca de tiempo del archivo de configuración original en el momento del arranque. Cualquier cambio de configuración adicional realizado por las operaciones de administración hará que se realice automáticamente una copia de seguridad del archivo original y que se preserve una copia de trabajo de la instancia para futura referencia o para revertir los cambios. Además, se pueden tomar instantáneas de configuración, que son copias en un momento preciso de la configuración actual del servidor. Un administrador puede guardar y cargar estas instantáneas.

Los siguientes ejemplos usan el archivo **standalone.xml**, pero el mismo proceso se aplica a los archivos **domain.xml** y **host.xml**.

##### Tomar una instantánea

Use la CLI administrativa para tomar una instantánea de las configuraciones actuales.

```
:take-snapshot
{
    "outcome" => "success",
    "result" =>
    "EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/2015102
    2-133109702standalone.xml"
}
```

##### Listar las instantáneas

Use la CLI administrativa para listar todas las instantáneas que han sido tomadas.

```
:list-snapshots
```

```
{
  "outcome" => "success",
  "result" => {
    "directory" =>
      "EAP_HOME/standalone/configuration/standalone_xml_history/snapshot",
    "names" => [
      "20151022-133109702standalone.xml",
      "20151022-132715958standalone.xml"
    ]
  }
}
```

### Borre la instantánea

Use la CLI administrativa para borrar una instantánea.

```
:delete-snapshot(name=20151022-133109702standalone.xml)
```

### Inicie el servidor con una instantánea

El servidor puede ser iniciado mediante una instantánea o una versión guardada automáticamente de la configuración.

1. Vaya al directorio **EAP\_HOME/standalone/configuration/standalone\_xml\_history** e identifique la instantánea o el archivo de configuración guardado que desea cargar.
2. Inicie el servidor y señale el archivo de configuración seleccionado. Pase la ruta del archivo relativa al directorio de configuración, **EAP\_HOME/standalone/configuration/**.

```
$ EAP_HOME/bin/standalone.sh --server-
config=standalone_xml_history/snapshot/20151022-
133109702standalone.xml
```



### NOTA

Cuando se ejecuta en un dominio administrado, use el argumento **--host-config** en lugar de especificar el archivo de configuración.

### 2.3.3.5. Reemplazo de propiedad

JBoss EAP le permite usar expresiones para definir propiedades reemplazables en lugar de valores literales en la configuración. Las expresiones usan el formato **\${PARAMETER:DEFAULT\_VALUE}**. Si se establece el parámetro especificado, se utilizará el valor del parámetro. De lo contrario, se utilizará el valor predeterminado provisto.

Las fuentes que reciben soporte para resolver expresiones son propiedades del sistema, variables de entorno y 'vault'. Para las implementaciones únicamente, la fuente puede corresponder a propiedades enumeradas en un archivo **META-INF/jboss.properties** que se encuentra en el archivo de implementación. Para tipos de implementación que admiten subimplementaciones, la resolución abarca todas las subimplementaciones si el archivo de propiedades está en la implementación externa, por ejemplo, el EAR. Si el archivo de propiedades está en la subimplementación, la resolución abarcará únicamente esa subimplementación.

El ejemplo a continuación del archivo de configuración **standalone.xml** establece el valor **inet-address** para la interfaz **public** en **127.0.0.1**, a menos que se establezca el parámetro **jboss.bind.address**.

```
<interface name="public">
  <inet-address value="{jboss.bind.address:127.0.0.1}"/>
</interface>
```

El parámetro **jboss.bind.address** puede ser definido en el inicio de EAP como un servidor autónomo, mediante el siguiente comando:

```
$ EAP_HOME/bin/standalone.sh -Djboss.bind.address=IP_ADDRESS
```

### Expresiones anidadas

Las expresiones pueden ser anidadas, lo que permite el uso más avanzado de las expresiones en lugar de valores fijos. El formato de una expresión anidada es como el de una expresión normal, pero una expresión está incorporada en la otra. Por ejemplo:

```
${SYSTEM_VALUE_1}${SYSTEM_VALUE_2}}
```

Las expresiones anidadas se evalúan de forma recursiva, por lo tanto la expresión *inner* es la primera evaluada, luego lo será la expresión *outer*. Las expresiones también pueden ser recursivas, cuando una expresión resuelve a otra expresión, que es luego resuelta. Las expresiones anidadas están permitidas en cualquier parte que las expresiones estén permitidas, a excepción de los comandos de la CLI administrativa.

Un ejemplo de uso de la expresión anidada es cuando la contraseña en una definición de fuente de datos está enmascarada. La configuración para una fuente de datos podría ser la siguiente línea:

```
<password>${VAULT::ds_ExampleDS::password::1}</password>
```

El valor de **ds\_ExampleDS** podría ser remplazado por una propiedad de sistema (**datasource\_name**), mediante una expresión anidada. La configuración para la fuente de datos podría en su lugar tener la siguiente línea:

```
<password>${VAULT::${datasource_name}::password::1}</password>
```

JBoss EAP primero debe evaluar la expresión **\${datasource\_name}**, luego ingresarla a una expresión más grande y evaluar la expresión resultante. La ventaja de esta configuración es que el nombre de la fuente de datos se abstrae de la configuración corregida.

### Remplazo de las propiedades basadas en el descriptor

La configuración de aplicaciones (como los parámetros de conexión de fuente de datos) generalmente varía entre entornos de desarrollo, prueba y producción. Algunas veces, los scripts de compilación del sistema acomodan esta variación, ya que la especificación Java EE no contiene un método para externalizar estas configuraciones. Gracias a JBoss EAP, puede usar el remplazo de propiedades basado en descriptores para administrar la configuración de modo externo.

El remplazo de la propiedad basada en descriptores, substituye las propiedades basadas en descriptores, lo que le permite suprimir los supuestos concernientes al entorno de la aplicación y la cadena de construcción. Se pueden especificar configuraciones específicas

del entorno en los descriptores de implementación, en lugar de anotaciones o scripts del sistema de construcción. Usted puede proporcionar la configuración en archivos o como parámetros en la línea de comandos.

Existen varios indicadores en el subsistema **ee** que controlan si se aplica el remplazo de propiedades.

El indicador **jboss-descriptor-property-replacement** controla el remplazo del descriptor específico JBoss y se *habilita* por defecto. Cuando está activado, las propiedades pueden ser remplazadas en los siguientes descriptores de implementación:

- **jboss-ejb3.xml**
- **jboss-app.xml**
- **jboss-web.xml**
- **\*-jms.xml**
- **\*-ds.xml**

El siguiente comando de la CLI administrativa puede utilizarse para activar o desactivar el remplazo de propiedad en descriptores específicos JBoss:

```
/subsystem=ee:write-attribute(name="jboss-descriptor-property-replacement",value=VALUE)
```

El indicador de descriptor Java EE controlado por el indicador **spec-descriptor-property-replacement** está *desactivado* por defecto. Cuando se activa, las propiedades pueden ser remplazadas en los siguientes descriptores de implementación:

- **ejb-jar.xml**
- **persistence.xml**
- **application.xml**
- **web.xml**

El siguiente comando de la CLI administrativa puede utilizarse para activar o desactivar el remplazo de propiedad en descriptores Java EE:

```
/subsystem=ee:write-attribute(name="spec-descriptor-property-replacement",value=VALUE)
```

## 2.4. CONFIGURACIÓN DE RED Y PUERTO

### 2.4.1. Interfaces

JBoss EAP hace referencia a interfaces designadas a través de la configuración. Esta acción permite que la configuración haga referencia a las declaraciones de interfaz individuales con nombres lógicos, en lugar de requerir la información total de la interfaz en cada uso.

También facilita la configuración en un dominio administrado donde los detalles de la interfaz de red pueden variar a través de múltiples máquinas. Cada instancia de servidor puede corresponder a un grupo de nombre lógico.

Los archivos **standalone.xml**, **domain.xml** y **host.xml** incluyen declaraciones de interfaz. Existen varios nombres de interfaz preconfigurados, según la configuración predeterminada que se utilice. La interfaz **management** puede utilizarse para todos los componentes y servicios que requieren la capa de administración, incluido el extremo de administración HTTP. La interfaz **public** se puede utilizar para todas las comunicaciones de red relacionadas con aplicaciones. La interfaz **unsecure** se utiliza para sockets IIOP en la configuración estándar. La interfaz **private** se utiliza para sockets JGroups en la configuración estándar.

#### 2.4.1.1. Configuración de interfaz predeterminada

```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address:127.0.0.1}"/>
  </interface>
  <interface name="private">
    <inet-address value="${jboss.bind.address.private:127.0.0.1}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="${jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>
```

JBoss EAP vincula por defecto estas interfaces a **127.0.0.1**, aunque estos valores pueden sobrescribirse en tiempo de ejecución al configurar la propiedad adecuada. Por ejemplo, **inet-address** de la interfaz **pública** puede configurarse al iniciar JBoss EAP como un servidor autónomo con el siguiente comando.

```
$ EAP_HOME/bin/standalone.sh -Djboss.bind.address=IP_ADDRESS
```

De modo alternativo, puede usar la opción **-b** en la línea de comandos de arranque del servidor. Para obtener más información sobre las opciones de arranque del servidor, consulte [Argumentos de tiempo de ejecución de servidor](#).



#### IMPORTANTE

Si modifica las interfaces o los puertos de red predeterminados que usa JBoss EAP, también debe recordar cambiar los scripts que usan las interfaces o los puertos modificados, que incluyen los scripts de servicio de JBoss EAP, así como especificar la interfaz y el puerto correctos al acceder a la consola de administración o la CLI de administración.

#### 2.4.1.2. Configuración de interfaces

Las interfaces de red se declaran al especificar el nombre lógico y los criterios de selección para la interfaz física. Los criterios de selección pueden hacer referencia a una dirección comodín o especificar un grupo de una o más características que una interfaz o dirección deben tener para ser una correspondencia válida. Para obtener un listado de todos los

criterios de selección de interfaces disponibles, consulte la sección [Atributos de interfaces](#).

Las interfaces pueden ser configuradas mediante la consola administrativa o la CLI administrativa. Abajo se presentan varios ejemplos de cómo agregar y actualizar interfaces. El comando de la CLI administrativa aparece primero, seguido de la configuración XML correspondiente.

### Agregue una interfaz con un valor NIC

Agregue una nueva interfaz con un valor NIC de **eth0**.

```
/interface=external:add(nic=eth0)
```

```
<interface name="external">
  <nic name="eth0"/>
</interface>
```

### Agregue la interfaz con varios valores condicionales

Agregue una nueva interfaz que coincida con una interfaz o dirección en la subred correcta si la interfaz está activa, soporta multidifusión y no es punto a punto.

```
/interface=default:add(subnet-
match=192.168.0.0/16,up=true,multicast=true,not={point-to-point=true})
```

```
<interface name="default">
  <subnet-match value="192.168.0.0/16"/>
  <up/>
  <multicast/>
  <not>
    <point-to-point/>
  </not>
</interface>
```

### Actualice el atributo de interfaz

Actualice el valor **inet-address** predeterminado de la interfaz **public**, manteniendo la propiedad **jboss.bind.address** para permitir que este valor sea configurado en tiempo de ejecución.

```
/interface=public:write-attribute(name=inet-
address,value="{jboss.bind.address:192.168.0.0}")
```

```
<interface name="public">
  <inet-address value="{jboss.bind.address:192.168.0.0}"/>
</interface>
```

### Agregue una interfaz a un servidor en un dominio administrado

```
/host=HOST_NAME/server-
config=SERVER_NAME/interface=INTERFACE_NAME:add(inet-address=127.0.0.1)
```

```
<servers>
  <server name="SERVER_NAME" group="main-server-group">
    <interfaces>
      <interface name="INTERFACE_NAME">
```



```

        <inet-address value="127.0.0.1"/>
    </interface>
</interfaces>
</server>
</servers>

```

## 2.4.2. Enlaces de socket

Los enlaces de socket y los grupos de enlaces de sockets le permiten definir puertos de red y su relación con las interfaces de red requeridas para la configuración de JBoss 6 EAP. Un enlace de socket es una configuración designada para un socket. Un grupo de enlaces de sockets es una colección de declaraciones de enlaces de sockets agrupadas en un nombre lógico.

Esto permite a otras secciones de la configuración referirse a los enlaces de socket por su nombre lógico, en lugar de requerir todos los detalles de la configuración del socket en cada uso.

Las declaraciones para estas configuraciones designadas se encuentran en los archivos de configuración **standalone.xml** y **domain.xml**. Un servidor autónomo contiene únicamente un grupo de enlaces de socket, mientras que un dominio administrado puede contener múltiples grupos. Puede crear un grupo de enlaces de sockets para cada grupo de servidor en el dominio administrado o compartir un grupo de enlaces de sockets entre múltiples grupos de servidor.

Los puertos utilizados por JBoss EAP dependen en forma predeterminada de los grupos de enlaces de sockets que se utilizan y de los requisitos de sus implementaciones individuales.

### 2.4.2.1. Puertos de administración

Los puertos de administración se consolidaron en JBoss EAP 7. JBoss EAP 7 usa en forma predeterminada el puerto **9990** para administración nativa, usada por la CLI de administración, y para administración HTTP, usada por la consola de administración basada en la Web. El puerto **9999**, que se usaba como el puerto de administración nativo en JBoss EAP 6, ya no se utiliza pero aún puede habilitarse si se desea.

Si HTTPS está habilitado para consola de administración, entonces se utiliza por defecto el puerto **9993**.

### 2.4.2.2. Enlaces de sockets predeterminados

JBoss EAP se distribuye con un grupo de enlaces de socket para cada uno de los cinco perfiles predefinidos (*default*, *ha*, *full*, *full-ha*, *load-balancer*).

Para obtener información más detallada sobre los enlaces de socket predeterminados, tales como puertos y descripciones predeterminados, consulte la sección [Enlaces de sockets predeterminados](#).



#### IMPORTANTE

Si modifica las interfaces o los puertos de red predeterminados que usa JBoss EAP, también debe recordar cambiar los scripts que usan las interfaces o los puertos modificados, que incluyen los scripts de servicio de JBoss EAP, así como especificar la interfaz y el puerto correctos al acceder a la consola de administración o la CLI de administración.

## Servidor autónomo

Cuando se ejecuta como un servidor autónomo, únicamente se define un grupo de enlaces de socket por archivo de configuración. Cada archivo de configuración autónomo (**standalone.xml**, **standalone-ha.xml**, **standalone-full.xml**, **standalone-full-ha.xml**, **standalone-load-balancer.xml**) define los enlaces de socket para las tecnologías utilizadas por su perfil correspondiente.

Por ejemplo, el archivo de configuración autónomo predeterminado (**standalone.xml**) especifica los siguientes enlaces de sockets:

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:9990}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9993}"/>
  <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
  <socket-binding name="http" port="${jboss.http.port:8080}"/>
  <socket-binding name="https" port="${jboss.https.port:8443}"/>
  <socket-binding name="txn-recovery-environment" port="4712"/>
  <socket-binding name="txn-status-manager" port="4713"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25"/>
  </outbound-socket-binding>
</socket-binding-group>
```

## Dominio administrado

Cuando se ejecuta en un dominio administrado, todos los grupos de enlaces de socket se definen en el archivo **domain.xml**. Existen cinco grupos de enlaces de sockets predefinidos:

- **standard-sockets**
- **ha-sockets**
- **full-sockets**
- **full-ha-sockets**
- **load-balancer-sockets**

Cada grupo de enlaces especifica enlaces de socket para tecnologías utilizadas por su perfil correspondiente. Por ejemplo, el grupo de enlaces de sockets **full-ha-sockets** define varios enlaces de sockets **jgroups**, los cuales son utilizados por el perfil **full-ha** para alta disponibilidad.

```
<socket-binding-groups>
  <socket-binding-group name="standard-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'default' profile -->
    <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
    <socket-binding name="http" port="${jboss.http.port:8080}"/>
    <socket-binding name="https" port="${jboss.https.port:8443}"/>
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
      <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
  </socket-binding-group>
</socket-binding-groups>
```

```

    </outbound-socket-binding>
  </socket-binding-group>
  <socket-binding-group name="ha-sockets" default-interface="public">
    <!-- Needed for server groups using the 'ha' profile -->
    ...
  </socket-binding-group>
  <socket-binding-group name="full-sockets" default-interface="public">
    <!-- Needed for server groups using the 'full' profile -->
    ...
  </socket-binding-group>
  <socket-binding-group name="full-ha-sockets" default-interface="public">
    <!-- Needed for server groups using the 'full-ha' profile -->
    <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
    <socket-binding name="http" port="${jboss.http.port:8080}"/>
    <socket-binding name="https" port="${jboss.https.port:8443}"/>
    <socket-binding name="iiop" interface="unsecure" port="3528"/>
    <socket-binding name="iiop-ssl" interface="unsecure" port="3529"/>
    <socket-binding name="jgroups-mping" interface="private" port="0"
multicast-address="${jboss.default.multicast.address:230.0.0.4}"
multicast-port="45700"/>
    <socket-binding name="jgroups-tcp" interface="private" port="7600"/>
    <socket-binding name="jgroups-udp" interface="private" port="55200"
multicast-address="${jboss.default.multicast.address:230.0.0.4}"
multicast-port="45688"/>
    <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
      <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
  </socket-binding-group>
  <socket-binding-group name="load-balancer-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'load-balancer' profile -->
    ...
  </socket-binding-group>
</socket-binding-groups>

```



## NOTA

La configuración de socket para las interfaces de administración se define en el archivo **host.xml** del controlador de dominio

### 2.4.2.3. Configuración de enlaces de socket

Cuando defina un enlace de socket, puede configurar los atributos **port** e **interface**, así como parámetros multidifusión, como **multicast-address** y **multicast-port**. Para obtener más información sobre todos los atributos de enlaces de sockets disponibles, consulte la sección [Atributos de enlaces de sockets](#).

Los vínculos de sockets se pueden configurar mediante la consola de administración o la CLI de administración. Los siguientes pasos lo guían en la adición de un grupo de enlaces de sockets, en la adición de un enlace de sockets y en la configuración de los parámetros de enlaces de sockets mediante la CLI de administración.

1. Agregue un nuevo grupo de enlace de socket. Observe que este paso no puede realizarse cuando se está ejecutando como servidor autónomo.

```
/socket-binding-group=new-sockets:add(default-interface=public)
```

2. Agregue el enlace de socket.

```
/socket-binding-group=new-sockets/socket-binding=new-socket-binding:add(port=1234)
```

3. Cambie el enlace de socket para usar una interfaz diferente a la predeterminada y establecida por el grupo de enlace de socket.

```
/socket-binding-group=new-sockets/socket-binding=new-socket-binding:write-attribute(name=interface,value=unsecure)
```

El siguiente ejemplo muestra cómo se vería la configuración XML después de completar los pasos anteriores.

```
<socket-binding-groups>
  ...
  <socket-binding-group name="new-sockets" default-interface="public">
    <socket-binding name="new-socket-binding" interface="unsecure"
port="1234"/>
  </socket-binding-group>
</socket-binding-groups>
```

#### 2.4.2.4. Port Offsets

Un valor 'port offset' es una diferencia numérica agregada a todos los valores de puertos especificados en el grupo de enlaces de sockets para dicho servidor. Permite al servidor heredar los valores de puerto definidos en su grupo de enlaces de sockets, con una diferencia para garantizar que no entre en conflicto con ningún otro servidor en el mismo host. Por ejemplo, si el puerto HTTP del grupo de enlaces de sockets es **8080**, y un servidor usa un 'port offset' de **100**, su puerto HTTP será **8180**.

El ejemplo a continuación muestra la configuración de un 'Port Offset' de **250** para un servidor en un dominio administrado mediante la CLI administrativa.

```
/host=master/server-config=server-two/:write-attribute(name=socket-binding-port-offset,value=250)
```

El desfase de puerto de CLI también pueden utilizarse para servidores en un dominio administrado y para ejecutar servidores autónomos múltiples en el mismo host.

Puede pasar en un desfase de puerto cuando inicie el servidor autónomo mediante la propiedad **jboss.socket.binding.port-offset**.

```
$ EAP_HOME/bin/standalone.sh -Djboss.socket.binding.port-offset=100
```

#### 2.4.3. Direcciones IPv6

Por defecto, JBoss EAP se configura para ejecutarse con direcciones IPv4. Los pasos a continuación muestran cómo configurar JBoss EAP para ejecutarse mediante direcciones IPv6.

### Configurar la pila JVM para direcciones IPv6

Actualice la configuración de arranque para preferir direcciones IPv6.

1. Abra el archivo de configuración de arranque.
  - Cuando se ejecuta como un servidor autónomo, modifique el archivo **EAP\_HOME/bin/standalone.conf** (o **standalone.conf.bat** para Windows Server).
  - Cuando se ejecuta en un servidor administrado, modifique el archivo **EAP\_HOME/bin/domain.conf** (o **domain.conf.bat** para Windows Server).
2. Establezca la propiedad **java.net.preferIPv4Stack** como **false**.

```
-Djava.net.preferIPv4Stack=false
```

3. Agregue la propiedad **java.net.preferIPv6Addresses** y defínala como **true**.

```
-Djava.net.preferIPv6Addresses=true
```

El siguiente ejemplo muestra cómo se verían las opciones JVM en el archivo de configuración de arranque después de hacer los cambios anteriores.

```
# Especifique opciones para pasar a la máquina virtual Java.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms1303m -Xmx1303m -Djava.net.preferIPv4Stack=false"
    JAVA_OPTS="$JAVA_OPTS -
Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS -
Djava.awt.headless=true"
    JAVA_OPTS="$JAVA_OPTS -Djava.net.preferIPv6Addresses=true"
else
```

### Actualice las declaraciones de interfaz para direcciones IPv6

Los valores de la interfaz predeterminada en la configuración pueden cambiarse en direcciones IPv6. Por ejemplo, el comando de la CLI administrativa de abajo, establece la interfaz **management** en la dirección loopback IPv6 (: : 1).

```
/interface=management:write-attribute(name=inet-
address,value="${jboss.bind.address.management:::1}]")
```

El siguiente ejemplo muestra cómo se vería la configuración XML después de ejecutar el comando anterior.

```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:::1}"/>
  </interface>
  ....
</interfaces>
```

## CAPÍTULO 3. DESARROLLO DE APLICACIONES MEDIANTE JBOSS EAP

### 3.1. VISIÓN GENERAL

En esta guía se proporciona información sobre cómo iniciar las aplicaciones de desarrollo con la ayuda de Red Hat JBoss Developer Studio y los ejemplos de inicio rápido de JBoss EAP 7.

Red Hat JBoss Developer Studio es un entorno de desarrollo integrado basado en Eclipse (IDE) que integra plugins de desarrollo de aplicaciones JBoss. JBoss Developer Studio puede ayudarlo a desarrollar la aplicación con la disponibilidad de asistentes específicos de JBoss y la capacidad para implementar aplicaciones para servidores JBoss EAP. Se proporcionan muchos ejemplos de código de inicio rápido JBoss EAP 7 para ayudar a los usuarios a iniciar la escritura de aplicaciones mediante diferentes tecnologías diferentes de Java EE 7.

### 3.2. CONFIGURACIÓN DEL ENTORNO DE DESARROLLO

Se recomienda usar JBoss Developer Studio 11.0 o posterior con JBoss EAP 7.1.

1. Descargue e instale JBoss Developer Studio.  
Para obtener instrucciones, consulte [Instalación del servidor autónomo JBoss Developer Studio usando el instalador](#) en la *Guía de instalación* de JBoss Developer Studio.
2. Configure el servidor JBoss EAP en JBoss Developer Studio.  
Para obtener instrucciones, consulte [Uso de la detección del tiempo de ejecución para configurar JBoss EAP desde dentro de IDE](#) en la guía *Introducción a las herramientas de JBoss Developer Studio*.

### 3.3. CÓMO USAR EJEMPLOS DE INICIOS RÁPIDOS

Los ejemplos de inicio rápido proporcionados con JBoss EAP son proyectos Maven.

#### 3.3.1. Acerca de Maven

Apache Maven es una herramienta de automatización de construcción distribuida, que se utiliza en el desarrollo de aplicaciones Java para crear, administrar y construir proyectos software. Maven usa archivos de configuración estándares llamados Modelo de objetos de proyecto (conocidos como POM, por sus siglas en inglés) para definir proyectos y administrar el proceso de construcción. Los POM describen las dependencias de componentes y módulos, el orden de construcción y los destinos para el empaquetado del proyecto resultante y la salida mediante un archivo XML. Esto asegura que el proyecto sea construido de una manera correcta y uniforme.

Maven logra esto con la ayuda de un repositorio. Un repositorio almacena bibliotecas, complementos y otros artefactos de la construcción. El repositorio público predeterminado es el [Maven 2 Central Repository](#), aunque los repositorios pueden ser privados e internos dentro de una compañía con una meta para compartir artefactos comunes entre los equipos de desarrollo. Los repositorios también están disponibles de terceras partes. Para obtener más información, consulte el proyecto [Apache Maven](#) y la guía de [Introducción a repositorios](#).

JBoss EAP incluye el repositorio Maven, el cual contiene requerimientos que los desarrolladores de Java EE suelen emplear para construir aplicaciones en JBoss EAP.

Para obtener más información sobre cómo utilizar Maven con JBoss EAP, consulte la sección [Uso de Maven con JBoss EAP](#) en la *Guía de desarrollo* de JBoss EAP.

### 3.3.2. Uso de Maven con Quickstarts

Los artefactos y dependencias necesarias para construir e implementar aplicaciones para JBoss EAP 6 se alojan en un repositorio público. Para comenzar por los inicios rápidos de JBoss EAP 7, ya no necesita configurar su archivo **settings.xml** de Maven para poder usar dicho repositorio durante la creación de los inicios rápidos. Los repositorios Maven ahora están configurados en los archivos POM del proyecto de inicios rápidos. Este método de configuración se proporciona para facilitarle el inicio con Quickstarts, sin embargo, no es recomendable para proyectos de producción debido a que puede demorar la construcción.

Red Hat JBoss Developer Studio incluye Maven; por lo tanto, no es necesario descargarlo o instalarlo por separado. Se recomienda usar una versión 11.0 o posterior de JBoss Developer Studio.

Si su plan es usar la línea de comandos Maven para construir e implementar sus aplicaciones, entonces primero descargue Maven desde el proyecto [Apache Maven](#) e instálelo siguiendo las instrucciones provistas en la documentación de Maven.

### 3.3.3. Descargar y ejecutar los Quickstarts

#### 3.3.3.1. Descargar los Quickstarts

JBoss EAP viene junto con una serie completa de ejemplos de inicio rápido diseñados para ayudar a los usuarios a empezar a escribir aplicaciones usando la tecnología de Java EE 7. Los inicios rápidos pueden ser descargados desde el Portal del cliente de Red Hat.

1. Ingrese al [Portal del cliente de Red Hat](#)
2. Haga clic en **Downloads**.
3. En la lista **Product Downloads**, haga clic en **Red Hat JBoss Enterprise Application Platform**.
4. Seleccione **7.1** en el menú desplegable **Version** (Versión).
5. Busque la entrada **Red Hat JBoss Enterprise Application Platform 7.1.0 Quickstarts** en la tabla y haga clic en **Download** (Descargar).
6. Guarde el archivo ZIP para el directorio deseado.
7. Extraer el archivo ZIP.

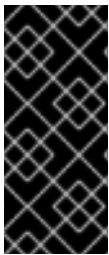
#### 3.3.3.2. Ejecutar los Quickstarts en JBoss Developer Studio

Cuando haya cargado los inicios rápidos, se pueden importar al JBoss Developer Studio y desplegar para JBoss EAP.

### Importe un Quickstart en JBoss Developer Studio



Cada inicio rápido se distribuye con un archivo POM (Modelo de objetos de proyectos) que contiene información de configuración y del proyecto. Al usar este archivo POM, puede importar fácilmente el inicio rápido en JBoss Developer Studio.

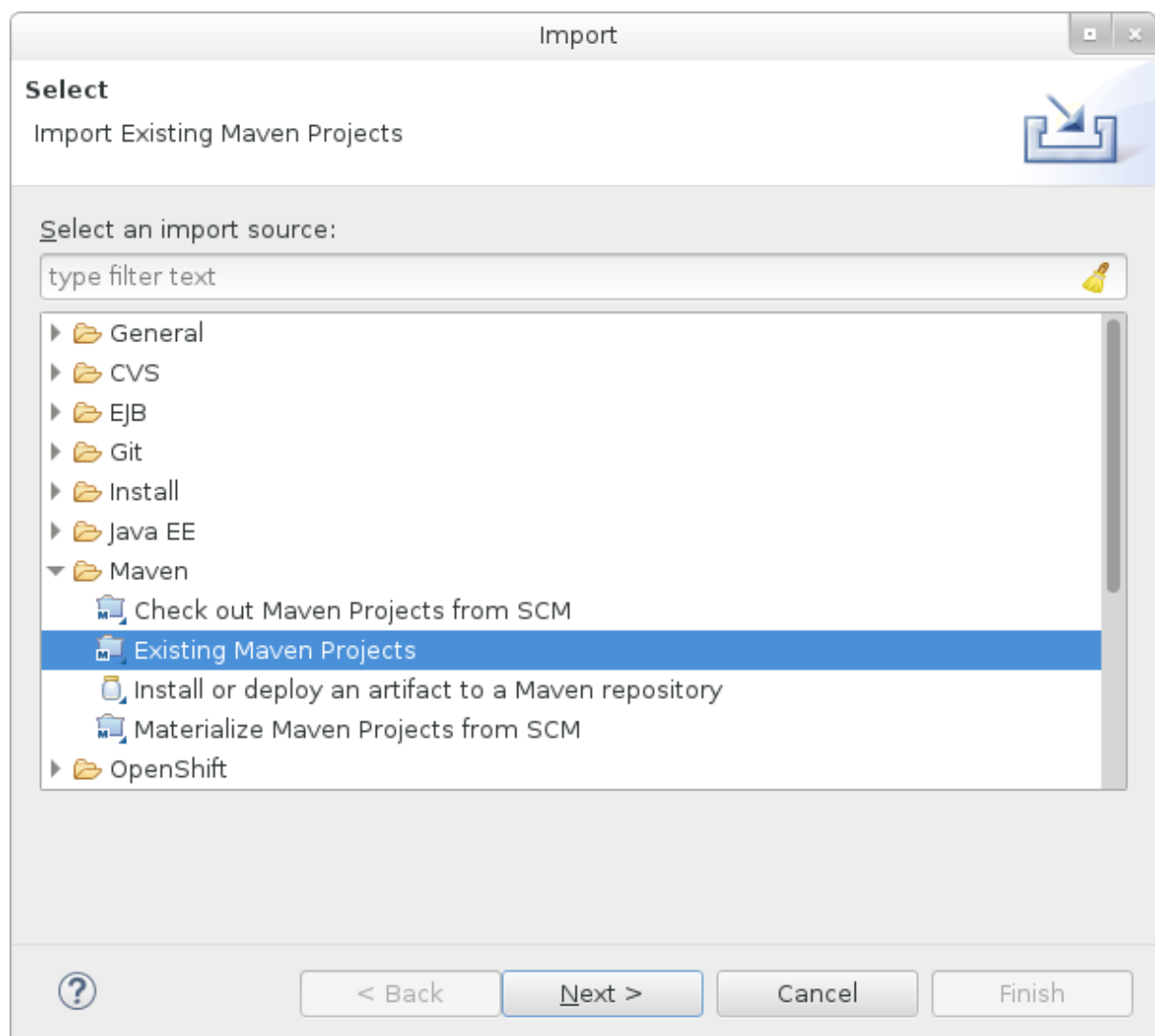


## IMPORTANTE

Si su carpeta de proyectos de inicios rápidos se localiza dentro del espacio de trabajo IDE cuando lo importa a JBoss Developer Studio, el IDE genera un nombre de proyecto inválido y el nombre de archivador WAR. Verifique si su carpeta de proyecto de inicios rápidos se localiza fuera del espacio de trabajo IDE antes de comenzar.

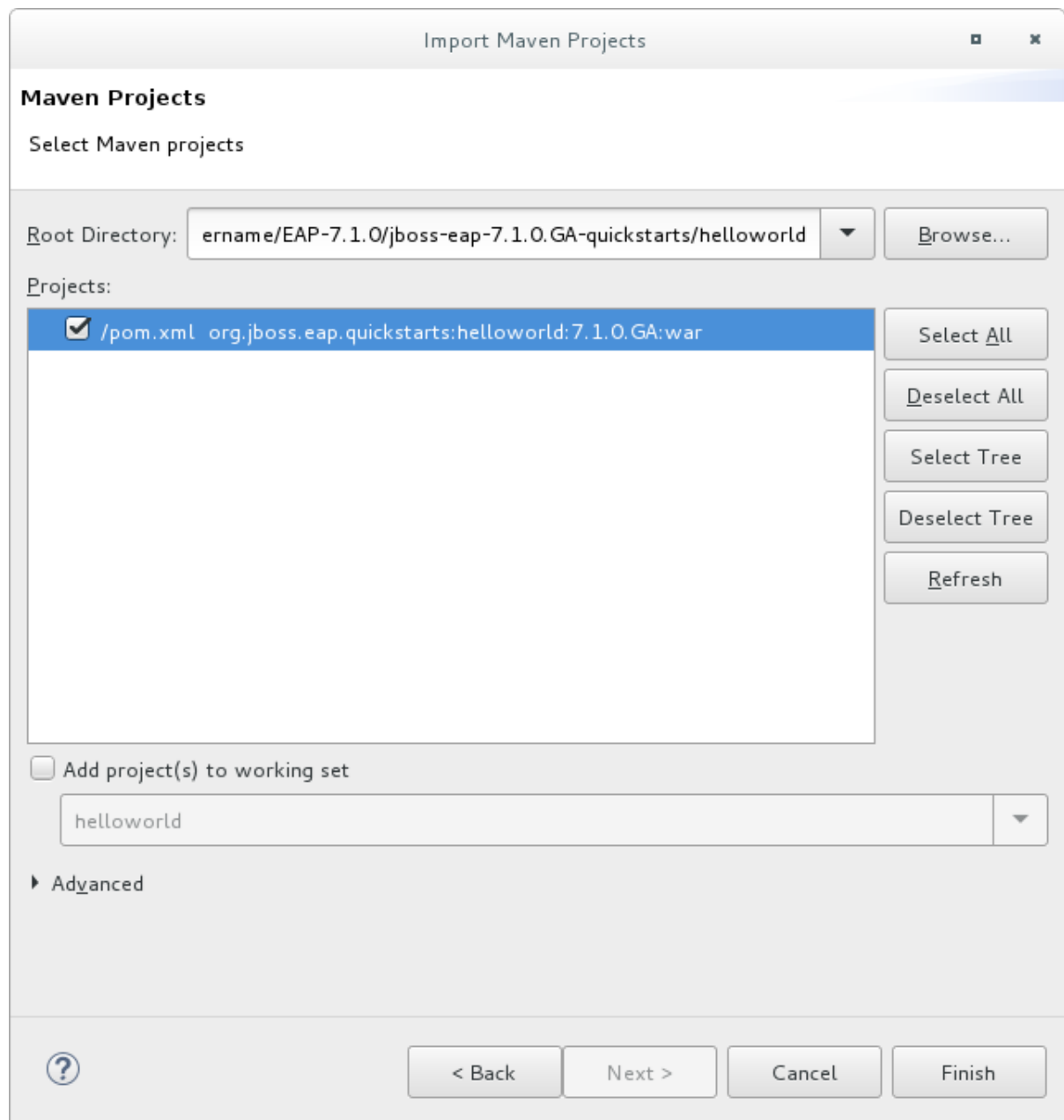
1. Inicie JBoss Developer Studio.
2. Seleccione **File** → **Import**.
3. Seleccione **Maven** → **Existing Maven Projects**, luego haga clic en **Next**.

**Figura 3.1. Importación de proyectos existentes de Maven**



4. Navegue al directorio de inicio rápido (por ejemplo el inicio rápido **helloworld**), y haga clic en **OK**. La caja de listas **Projects** se generará con el archivopom.xml del proyecto de inicio rápido seleccionado.



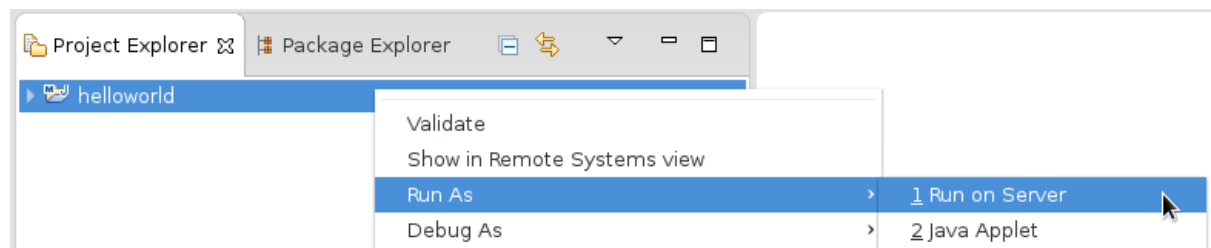
**Figura 3.2. Seleccione proyectos Maven**

5. Haga clic en **Finish**.

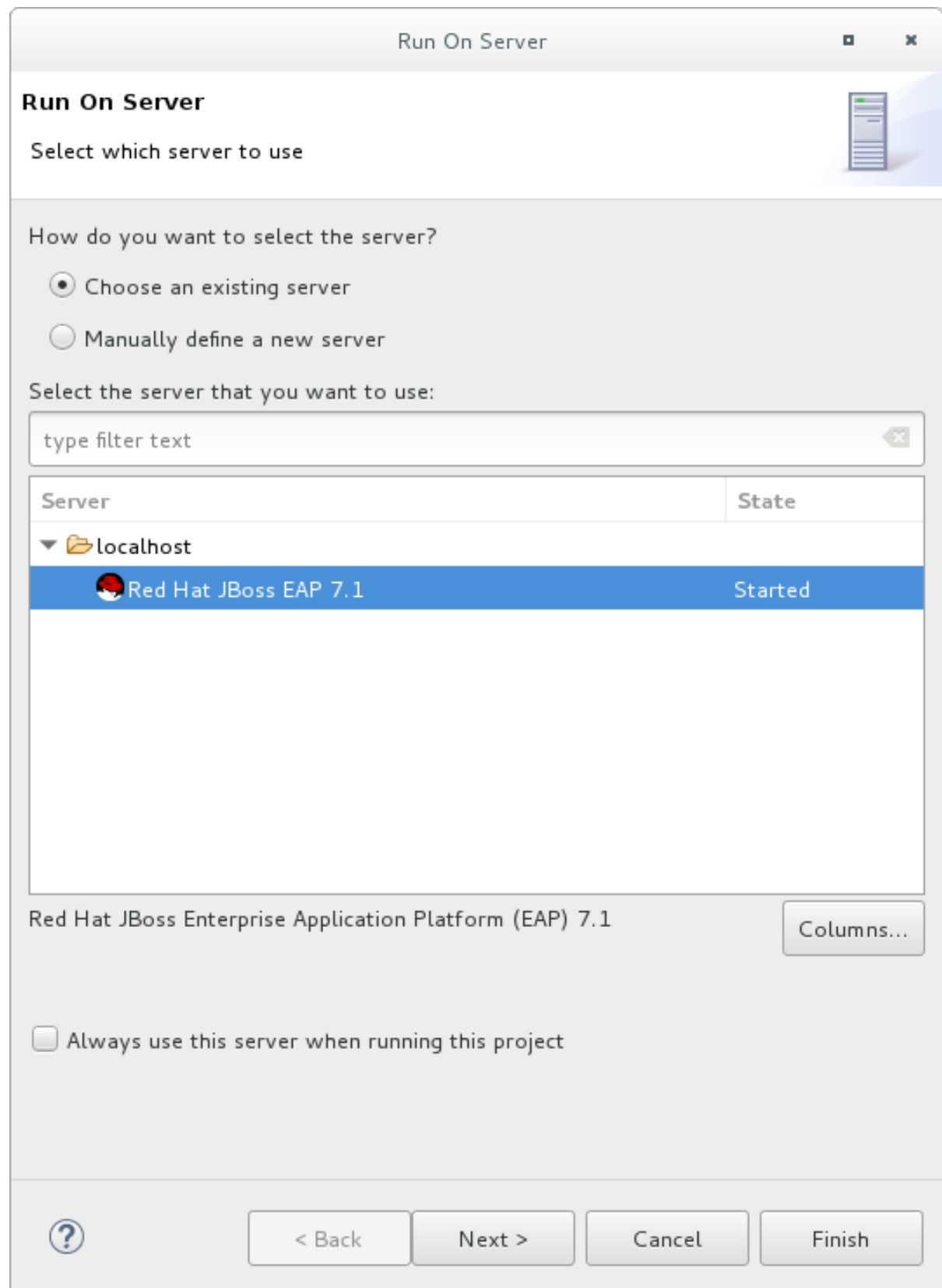
### Ejecute el inicio rápido **helloworld**

La ejecución del inicio rápido **helloworld** es una forma fácil de verificar que el servidor JBoss EAP está configurado y está ejecutándose correctamente.

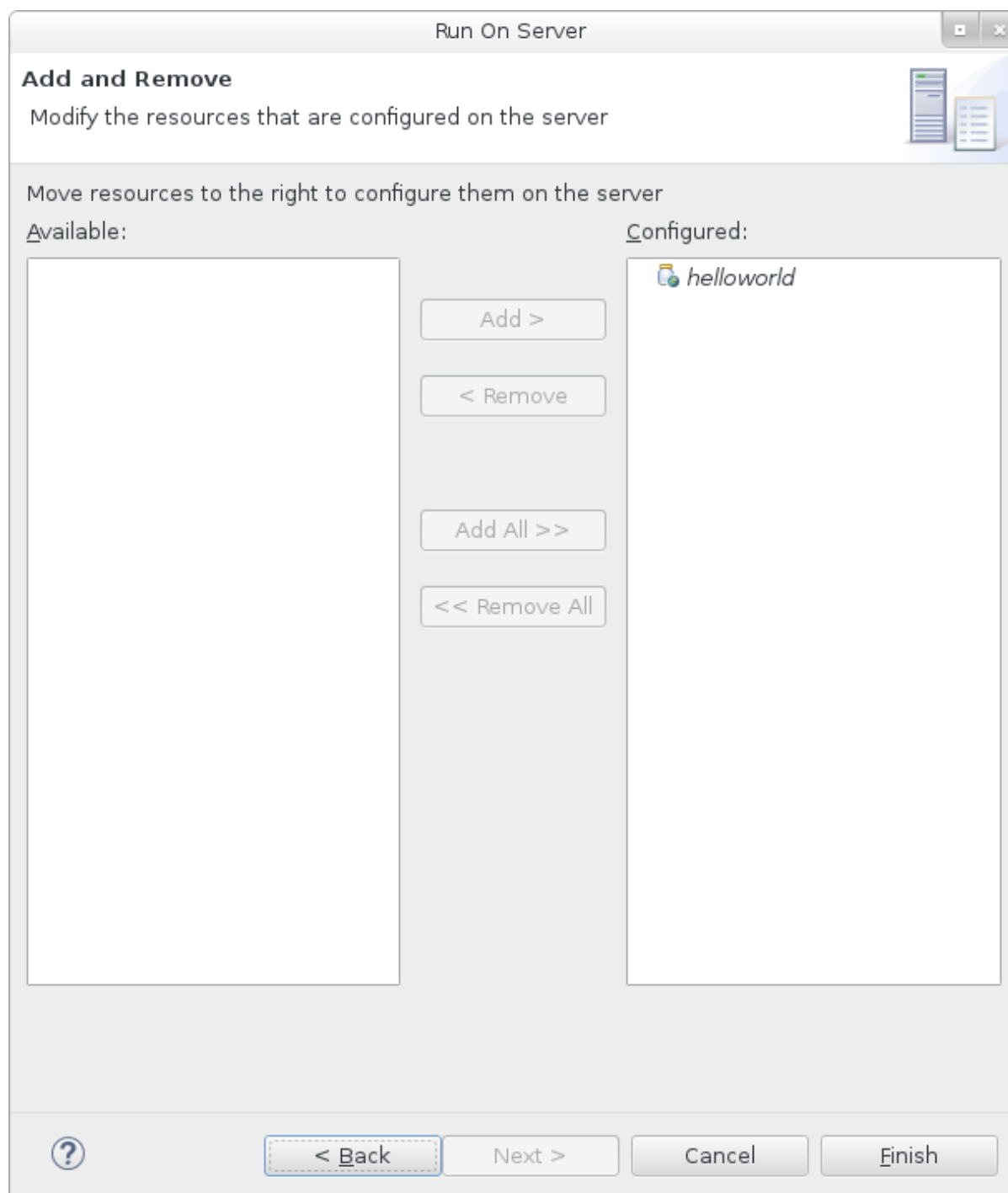
1. Si aún no definió un servidor, añada el servidor JBoss EAP a JBoss Developer Studio. Consulte [Uso de la detección del tiempo de ejecución para configurar JBoss EAP desde dentro de IDE](#) en la guía *Introducción a las herramientas de JBoss Developer Studio*.
2. Haga clic en el proyecto **helloworld** en la pestaña **Project Explorer** (Explorador de proyectos) y seleccione **Run As** (Ejecutar como) → **Run on Server** (Ejecutar en el servidor).

**Figura 3.3. Ejecutar como - Ejecutar en servidor**

3. Seleccione el servidor JBoss EAP 7.1 desde la lista de servidores y haga clic en **Next** (Siguiente).

**Figura 3.4. Ejecute en el servidor**

4. El inicio rápido **helloworld** ya figura como configurado en el servidor. Haga clic en **Finish** (Finish) para implementar el inicio rápido.

**Figura 3.5. Modifique los recursos configurados en el servidor**

#### 5. Verifique los resultados

- En la pestaña **Server** (Servidor), el estado del servidor JBoss EAP 7.1 cambia a **Started** (Iniciado).
- La pestaña **Console** muestra mensajes que detallan el inicio del servidor JBoss EAP y la implementación de inicio rápido **helloworld**.

```
WFLYUT0021: Registered web context: /helloworld
WFLYSRV0010: Deployed "helloworld.war" (runtime-name :
"helloworld.war")
```

- La aplicación **helloworld** está disponible en <http://localhost:8080/helloworld> y muestra el texto **Hello World!** (¡Hola, mundo!).

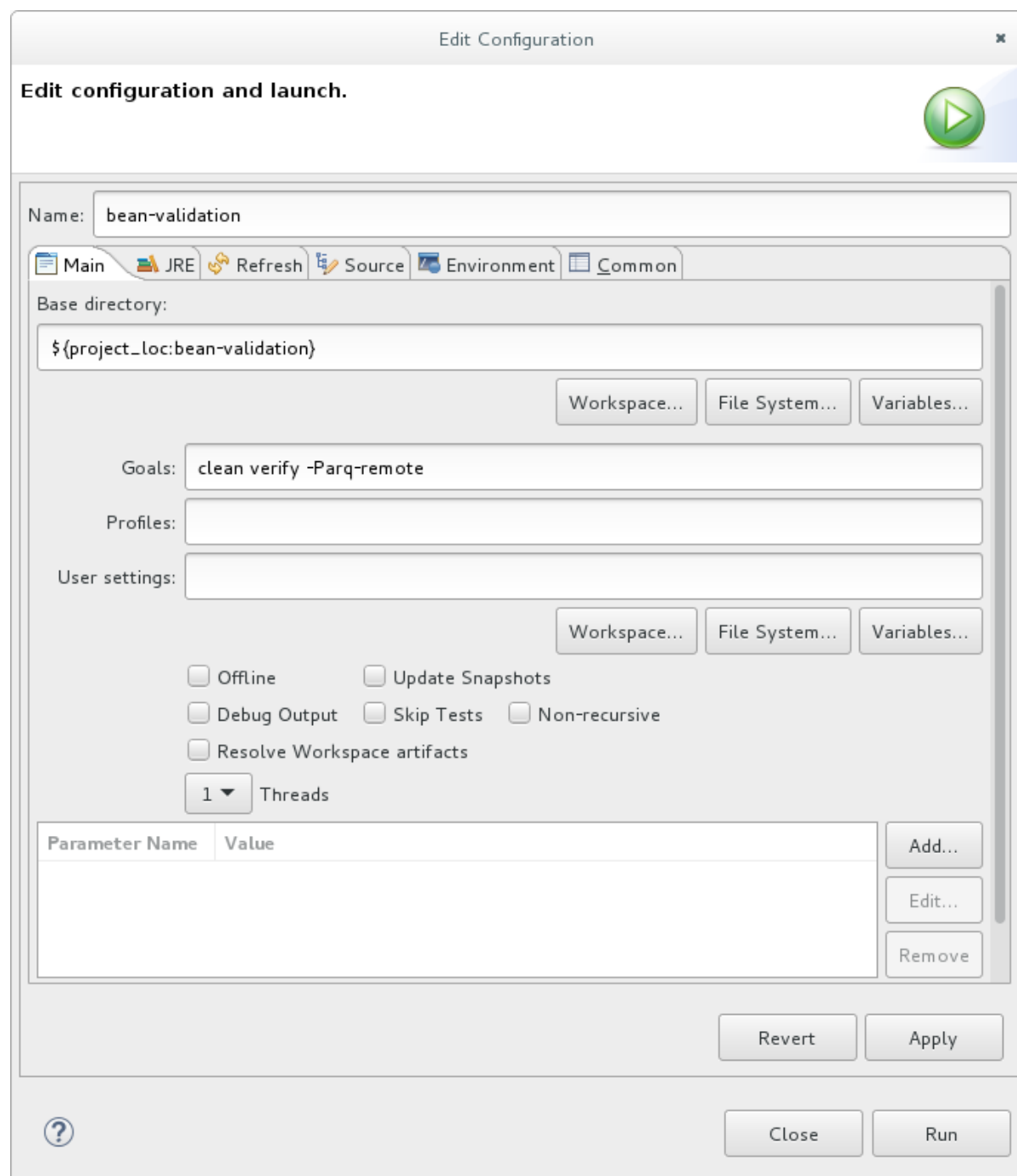
Para obtener más detalles sobre el inicio rápido **helloworld**, consulte [Exploración del inicio rápido helloworld](#).

### Ejecute el inicio rápido *bean-validation*

Algunos inicios rápidos tales como **bean-validation**, no proporcionan una capa de interfaz de usuario y en su lugar, proporcionan pruebas Arquillian para demostrar la funcionalidad.

1. Importe el inicio rápido **bean-validation** a JBoss Developer Studio.
2. En la pestaña **Servers** (Servidores), haga clic derecho en el servidor y elija **Start** (Inicio) para iniciar el servidor JBoss EAP. Si no ve una pestaña **Servers** (Servidores) o aún no definió un servidor, añada el servidor JBoss EAP a JBoss Developer Studio. Consulte [Uso de la detección del tiempo de ejecución para configurar JBoss EAP desde dentro de IDE](#) en la guía *Inicio de las herramientas de JBoss Developer Studio*.
3. Haga clic derecho en el proyecto **bean-validation** en la pestaña **Project Explorer** (Explorador de proyectos) y seleccione **Run As** (Ejecutar como) → **Maven Build**.
4. Entre lo siguiente en el campo de entrada **Goals** y luego haga clic en **Run**.

```
clean verify -Parq-remote
```

**Figura 3.6. Modifique la configuración**

## 5. Verifique los resultados

La pestaña **Console** muestra los resultados de las pruebas Arquillian**bean-validation**:

```
-----
T E S T S
-----
```

```
Running
```

```
org.jboss.as.quickstarts.bean_validation.test.MemberValidationTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:
2.189 sec
```

```
Results :
```

```
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
```

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

### 3.3.3.3. Ejecute los Inicios rápidos mediante la línea de comandos

Puede construir e implementar fácilmente los inicios rápidos desde la línea de comandos usando Maven. Si aún no tiene instalado Maven, por favor consulte el proyecto [Apache Maven](#) para descargarlo e instalarlo.

Se proporciona un archivo **README.md** en el directorio raíz de los inicios rápidos que contiene información general sobre los requisitos del sistema, configuración de Maven, adición de usuarios y ejecución de los inicios rápidos.

Cada inicio rápido contiene su propio archivo **README.md** que proporciona instrucciones específicas y los comandos Maven para ejecutarlo.

### Ejecute el Quickstart *helloworld* desde la línea de comandos

1. Revise el archivo **README.md** en el directorio raíz del inicio rápido *helloworld*.
2. Inicie el servidor JBoss EAP

```
$ EAP_HOME/bin/standalone.sh
```

3. Navegue al directorio de inicio rápido *helloworld*.
4. Compile e implemente el inicio rápido mediante el comando de Maven provisto en el archivo **README.md** del inicio rápido.

```
$ mvn clean install wildfly:deploy
```

5. La aplicación *helloworld* está disponible en <http://localhost:8080/helloworld> y muestra el texto **Hello World!** (¡Hola, mundo!).

## 3.4. REVISAR LOS EJEMPLOS DE INICIOS RÁPIDOS

### 3.4.1. Exploración del inicio rápido *helloworld*

El inicio rápido **helloworld** le muestra cómo implementar un servlet simple en JBoss EAP. La lógica empresarial está encapsulada en un servicio, el cual se ofrece como un bean de Inyección de contextos y dependencia (CDI) y se inyecta en el Servlet. El inicio rápido es un punto de partida para asegurarse de haber configurado e iniciado su servidor correctamente.

Las instrucciones detalladas para compilar e implementar este inicio rápido mediante la línea de comando se puede encontrar en el archivo **README.html** en la raíz del directorio de inicio rápido **helloworld**. Este tema le explica cómo usar Red Hat JBoss Developer Studio para ejecutar el inicio rápido y asume que usted ha instalado Red Hat JBoss Developer Studio, configurado Maven, e importado y ejecutado correctamente el inicio rápido **helloworld**.

## Prerrequisitos

- Instale JBoss Developer Studio. Para obtener instrucciones, consulte [Instalación del servidor autónomo JBoss Developer Studio usando el instalador](#) en la *Guía de instalación* de JBoss Developer Studio.
- Ejecute el inicio rápido **helloworld**. Para obtener instrucciones, consulte [Ejecutar los inicios rápidos en JBoss Developer Studio](#).
- Verifique que el inicio rápido **helloworld** se haya implementado correctamente en JBoss EAP; para ello, abra un explorador web y acceda a la aplicación en <http://localhost:8080/helloworld>.

## Examine la estructura del directorio

El código para el inicio rápido **helloworld** se puede encontrar en el directorio **QUICKSTART\_HOME/helloworld/**. El inicio rápido **helloworld** está compuesto por un Servlet y un bean de CDI. También contiene un archivo **beans.xml** en el directorio **WEB-INF/** de la aplicación, cuyo número de versión es 1.1, y un **bean-discovery-mode** de **all**. Este archivo marcador identifica el WAR como un archivo bean y le ordena a JBoss EAP que busque beans en esta aplicación y que active la CDI.

El directorio **src/main/webapp/** contiene los archivos para el inicio rápido. Todos los archivos de configuración de este ejemplo se encuentran en el directorio **WEB-INF/** dentro de **src/main/webapp/**, incluido el archivo **beans.xml**. El directorio **src/main/webapp/** también incluye un archivo **index.html**, que usa una simple actualización meta para redirigir el navegador del usuario al Servlet, el cual se localiza en <http://localhost:8080/helloworld/HelloWorld>. El inicio rápido no requiere un archivo **web.xml**.

## Examine el código

La declaración del paquete y las importaciones se han excluido de este listado. El listado completo está disponible en el código fuente del inicio rápido.

1. Revise el código **HelloWorldServlet**.

El archivo **HelloWorldServlet.java** se encuentra en el directorio **src/main/java/org/jboss/as/quickstarts/helloworld/**. Este servlet envía la información al navegador.

## Ejemplo: Código de clase HelloWorldServlet

```

42 @SuppressWarnings("serial")
43 @WebServlet("/HelloWorld")
44 public class HelloWorldServlet extends HttpServlet {
45
46     static String PAGE_HEADER = "<html><head>
<title>helloworld</title></head><body>";
47
48     static String PAGE_FOOTER = "</body></html>";
49
50     @Inject
51     HelloService helloService;
52
53     @Override
54     protected void doGet(HttpServletRequest req,
55     HttpServletResponse resp) throws ServletException, IOException {
56         resp.setContentType("text/html");
57         PrintWriter writer = resp.getWriter();
58         writer.println(PAGE_HEADER);

```



```

58         writer.println("<h1>" +
helloService.createHelloMessage("World") + "</h1>");
59         writer.println(PAGE_FOOTER);
60         writer.close();
61     }
62
63 }

```

**Tabla 3.1. Detalles de HelloWorldServlet**

Línea	Nota
43	Todo lo que debe hacer es añadir la anotación <b>@WebServlet</b> y proporcionar una asignación a una URL usada para acceder al servlet.
46-48	Toda página web necesita HTML formado de manera correcta. Este inicio rápido usa cadenas estáticas para escribir la salida de pie de página y de encabezados mínimos.
50-51	Estas líneas inyectan el bean de CDI HelloService, el cual genera el propio mensaje. Mientras no alteremos la API de HelloService, este enfoque nos permite alterar la implementación de HelloService posteriormente, sin modificar la capa de vista.
58	Esta línea llama al servicio para generar el mensaje "Hello World" y escribirlo en la petición HTTP.

## 2. Revise el código **HelloService**.

El archivo **HelloService.java** está localizado en el directorio **src/main/java/org/jboss/as/quickstarts/helloworld/**. Este servicio simplemente devuelve un mensaje. No se necesita registro de XML o anotación.

### Ejemplo: Código de clase HelloService

```

public class HelloService {

    String createHelloMessage(String name) {
        return "Hello " + name + "!";
    }

}

```

## 3.4.2. Explore el inicio rápido de **numberguess**

El inicio rápido **numberguess** le explica cómo crear e implementar una aplicación no persistente simple en JBoss EAP. La información se muestra usando una vista JSF y la lógica empresarial se encapsula en dos beans de CDI. En el inicio rápido **numberguess**, tiene diez intentos de adivinar un número entre 1 y 100. Después de cada intento, se le indica si su tentativa fue muy alta o muy baja.

El código para el inicio rápido **numberguess** se puede encontrar en el directorio **QUICKSTART\_HOME/numberguess/**, donde **QUICKSTART\_HOME** es el directorio en el que

descargó y descomprimió los inicios rápidos de JBoss EAP. El inicio rápido **numberguess** está compuesto por un número de beans, archivos de configuración, y vistas de Facelets (JSF), y viene empaquetado como un módulo WAR.

Las instrucciones detalladas para desarrollar e implementar este inicio rápido usando la línea de comando se pueden encontrar en el archivo **README.html** en la raíz del directorio de inicio rápido **numberguess**. Los siguientes ejemplos usan Red Hat JBoss Developer Studio para ejecutar el inicio rápido.

## Prerrequisitos

- Instale JBoss Developer Studio. Para obtener instrucciones, consulte [Instalación del servidor autónomo JBoss Developer Studio usando el instalador](#) en la *Guía de instalación* de JBoss Developer Studio.
- Ejecute el inicio rápido **numberguess**. Para obtener instrucciones, consulte [Ejecutar los inicios rápidos en JBoss Developer Studio](#) y reemplace **helloworld** con **numberguess** en las instrucciones.
- Verifique que el inicio rápido **numberguess** se haya implementado correctamente en JBoss EAP abriendo un explorador web y accediendo a la aplicación en esta URL: <http://localhost:8080/numberguess>.

## Examine los archivos de configuración

Todos los archivos de configuración para este ejemplo se localizan en el directorio **QUICKSTART\_HOME/numberguess/src/main/webapp/WEB-INF/** del inicio rápido.

1. Examine el archivo **faces-config.xml**.

Este inicio rápido usa la versión JSF 2.2 del nombre del archivo **faces-config.xml**. El controlador de vista predeterminado en JSF 2.2 es una versión estandarizada de Facelets, por lo que no necesita configuración. Este archivo consiste del elemento raíz solamente y es simplemente un archivo marcador para indicar que JSF se debe habilitar en la aplicación.

```
<faces-config version="2.2"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">

</faces-config>
```

2. Examine el archivo **beans.xml**.

El archivo **beans.xml** cuenta con un número de versión 1.1 y un **bean-discovery-mode** de **all**. Este archivo marcador identifica el WAR como un archivo bean y le ordena a JBoss EAP que busque beans en esta aplicación y que active la CDI.

```
<beans xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/beans_1_1.xsd"
  bean-discovery-mode="all">

</beans>
```

**NOTA**

El inicio rápido no necesita el archivo `web.xml`.

**3.4.2.1. Examine el código JSF**

JSF usa la extensión del archivo `.xhtml` para archivos de origen, pero entrega las vistas mostradas con la extensión `.jsf`. El archivo `home.xhtml` se localiza en el directorio `src/main/webapp/`.

**Ejemplo: Código de origen JSF**

```

19<html xmlns="http://www.w3.org/1999/xhtml"
20 xmlns:ui="http://java.sun.com/jsf/facelets"
21 xmlns:h="http://java.sun.com/jsf/html"
22 xmlns:f="http://java.sun.com/jsf/core">
23
24 <head>
25 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
26 />
27 <title>Numberguess</title>
28 </head>
29 <body>
30 <div id="content">
31 <h1>Guess a number...</h1>
32 <h:form id="numberGuess">
33
34 <!-- Feedback for the user on their guess -->
35 <div style="color: red">
36 <h:messages id="messages" globalOnly="false" />
37 <h:outputText id="Higher" value="Higher!"
38     rendered="#{game.number gt game.guess and game.guess ne 0}" />
39 <h:outputText id="Lower" value="Lower!"
40     rendered="#{game.number lt game.guess and game.guess ne 0}" />
41 </div>
42
43 <!-- Instructions for the user -->
44 <div>
45 I'm thinking of a number between <span
46 id="numberGuess:smallest">#{game.smallest}</span> and <span
47 id="numberGuess:biggest">#{game.biggest}</span>. You have
48 #{game.remainingGuesses} guesses remaining.
49 </div>
50
51 <!-- Input box for the users guess, plus a button to submit, and reset
52 -->
53 <!-- These are bound using EL to our CDI beans -->
54 <div>
55 Your guess:
56 <h:inputText id="inputGuess" value="#{game.guess}"
57     required="true" size="3"
58     disabled="#{game.number eq game.guess}"
59     validator="#{game.validateNumberRange}" />
60 <h:commandButton id="guessButton" value="Guess"
61     action="#{game.check}"

```

```

61 disabled="#{game.number eq game.guess}" />
62 </div>
63 <div>
64 <h:commandButton id="restartButton" value="Reset"
65 action="#{game.reset}" immediate="true" />
66 </div>
67 </h:form>
68
69 </div>
70
71 <br style="clear: both" />
72
73 </body>
74</html>

```

Los siguientes números de línea corresponden a los que se ven al visualizar el archivo en JBoss Developer Studio.

**Tabla 3.2. Detalles de JSF**

Línea	Nota
36-40	Estos son los mensajes que se pueden enviar al usuario: "Higher!" y "Lower!"
45-48	Cuando el usuario empieza a adivinar, el rango de números se empieza a volver más pequeño. Esta oración cambia para asegurarse de que sepan el rango válido de números a adivinar.
55-58	Este campo de entrada está enlazado a una propiedad bean usando una expresión de valor.
58	Un enlace validador se utiliza para asegurarse de que el usuario no ingrese de manera accidental un número por fuera del rango de adivinanza. Si el validador no está presente entonces puede que el usuario use un intento por fuera de los límites de la adivinanza.
59-61	Debe haber una manera de que el usuario envíe su intento al servidor. Aquí enlazamos a un método de acción al bean.

### 3.4.2.2. Examine los archivos de clases

Todos los archivos de origen del inicio rápido **numberguess** se pueden encontrar en el directorio

**QUICKSTART\_HOME/numberguess/src/main/java/org/jboss/as/quickstarts/numberguess/**. La declaración del paquete y las importaciones se han excluido de este listado. El listado completo está disponible en el código fuente del inicio rápido.

1. Revise el Código del calificador **Random.java**.

Se utiliza un calificador para eliminar la ambigüedad entre dos beans, cuando ambos son elegibles para inyección según su tipo. Para obtener más información sobre los calificadores, consulte [Usar un calificador para resolver una inyección ambigua](#) en la

Guía de desarrollo de JBoss EAP. El calificador **@Random** se usa para inyectar un número aleatorio.

```
@Target({ TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@Qualifier
public @interface Random {

}
```

2. Revise el Código del calificador **MaxNumber.java**.

El **calificador @MaxNumber** se usa para inyectar la cantidad máxima permitida.

```
@Target({ TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@Qualifier
public @interface MaxNumber {

}
```

3. Revise el Código **Generator.java**.

La clase **Generator** crea el número aleatorio a través de un método de productor, exponiendo el máximo número posible a través de este. Esta clase tiene el alcance de la aplicación, por lo que no obtiene un número aleatorio diferente cada vez.

```
@SuppressWarnings("serial")
@ApplicationScoped
public class Generator implements Serializable {

    private java.util.Random random = new
    java.util.Random(System.currentTimeMillis());

    private int maxNumber = 100;

    java.util.Random getRandom() {
        return random;
    }

    @Produces
    @Random
    int next() {
        // a number between 1 and 100
        return getRandom().nextInt(maxNumber - 1) + 1;
    }

    @Produces
    @MaxNumber
    int getMaxNumber() {
        return maxNumber;
    }

}
```

4. Revise el Código **Game.java**.

La clase **Game** con alcance de la sesión es el principal punto de entrada de la aplicación. Es responsable de configurar o reconfigurar el juego, capturar y validar los intentos del usuario, y proporcionar una retroalimentación al usuario mediante un **FacesMessage**. Usa el método de ciclo de vida posterior al desarrollo para iniciar el juego, obteniendo un número aleatorio del bean **@Random Instance<Integer>**.

Observe la anotación **@Named** en la clase. Esta anotación solo es requerida cuando desea que el bean sea accesible a una vista JSF mediante el lenguaje de expresión (Expression Language, EL), en este caso, **#{game}**.

```
@SuppressWarnings("serial")
@Named
@SessionScoped
public class Game implements Serializable {

    /**
     * The number that the user needs to guess
     */
    private int number;

    /**
     * The users latest guess
     */
    private int guess;

    /**
     * The smallest number guessed so far (so we can track the valid
     guess range).
     */
    private int smallest;

    /**
     * The largest number guessed so far
     */
    private int biggest;

    /**
     * The number of guesses remaining
     */
    private int remainingGuesses;

    /**
     * The maximum number we should ask them to guess
     */
    @Inject
    @MaxNumber
    private int maxNumber;

    /**
     * The random number to guess
     */
    @Inject
    @Random
    Instance<Integer> randomNumber;

    public Game() {
```

```

    }

    public int getNumber() {
        return number;
    }

    public int getGuess() {
        return guess;
    }

    public void setGuess(int guess) {
        this.guess = guess;
    }

    public int getSmallest() {
        return smallest;
    }

    public int getBiggest() {
        return biggest;
    }

    public int getRemainingGuesses() {
        return remainingGuesses;
    }

    /**
     * Check whether the current guess is correct, and update the
     * biggest/smallest guesses as needed. Give feedback to the user
     * if they are correct.
     */
    public void check() {
        if (guess > number) {
            biggest = guess - 1;
        } else if (guess < number) {
            smallest = guess + 1;
        } else if (guess == number) {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Correct!"));
        }
        remainingGuesses--;
    }

    /**
     * Reset the game, by putting all values back to their defaults,
     * and getting a new random number. We also call this method
     * when the user starts playing for the first time using
     * {@linkplain PostConstruct @PostConstruct} to set the initial
     * values.
     */
    @PostConstruct
    public void reset() {
        this.smallest = 0;
        this.guess = 0;
        this.remainingGuesses = 10;
        this.biggest = maxNumber;
    }

```

```
        this.number = randomNumber.get();
    }

    /**
     * A JSF validation method which checks whether the guess is
     * valid. It might not be valid because there are no guesses left,
     * or because the guess is not in range.
     */
    public void validateNumberRange(FacesContext context,
        UIComponent toValidate, Object value) {
        if (remainingGuesses <= 0) {
            FacesMessage message = new FacesMessage("No guesses
left!");
            context.addMessage(toValidate.getClientId(context),
message);
            ((UIInput) toValidate).setValid(false);
            return;
        }
        int input = (Integer) value;

        if (input < smallest || input > biggest) {
            ((UIInput) toValidate).setValid(false);

            FacesMessage message = new FacesMessage("Invalid
guess");
            context.addMessage(toValidate.getClientId(context),
message);
        }
    }
}
```



## APÉNDICE A. MATERIAL DE REFERENCIA

### A.1. ARGUMENTOS DE TIEMPO DE EJECUCIÓN DEL SERVIDOR

El script de arranque del servidor de aplicaciones acepta argumentos y cambia en tiempo de ejecución. Esto permite al servidor iniciarse en configuraciones alternativas a aquellas definidas en los archivos de configuración **standalone.xml**, **domain.xml** y **host.xml**

Las configuraciones alternativas podrían incluir el inicio del servidor con un set de enlace de socket alternativo o una configuración secundaria.

La lista de los parámetros disponibles puede accederse al pasar la opción de ayuda **-h** o **--help** en el inicio.

**Tabla A.1. Opciones y argumentos de ejecución**

Argumento u opción	Modo operativo	Descripción
--admin-only	Autónomo	Definir el tipo de ejecución del servidor como <b>ADMIN_ONLY</b> . Esto lo hará abrir las interfaces administrativas y aceptar las peticiones de administración, pero no podrá iniciar otros servicios de tiempo de ejecución ni aceptar peticiones de usuarios finales. Observe que, en cambio, se recomienda usar <b>--start-mode=admin-only</b> .
--admin-only	Dominio	Configure el tipo de ejecución del controlador host como <b>ADMIN_ONLY</b> para que abra interfaces administrativas y acepte peticiones de administración, pero no podrá iniciar servidores o, si este controlador host es el maestro del dominio, aceptar conexiones entrantes de controladores hosts esclavos.
-b=<value>, -b <value>	Autónomo, dominio	Establecer la propiedad <b>jboss.bind.address</b> del sistema, que se utiliza para configurar la dirección de enlace para la interfaz pública. Si no se especifica un valor, se establecerá de forma predeterminada en <b>127.0.0.1</b> . Consulte la entrada <b>-b&lt;interface&gt;=&lt;value&gt;</b> para configurar la dirección de enlace para otras interfaces.
-b<interface>=<value>	Autónomo, dominio	Establecer la propiedad <b>jboss.bind.address.&lt;interface&gt;</b> del sistema con el valor proporcionado. Por ejemplo, <b>-bmanagement=IP_ADDRESS</b>

Argumento u opción	Modo operativo	Descripción
--backup	Dominio	Mantenga una copia de la configuración de dominio persistente incluso si este host no es el controlador de dominios.
-c=<config>, -c <config>	Autónomo	Nombre del archivo de configuración del servidor que se va a utilizar. El predeterminado es <b>standalone.xml</b> .
-c=<config>, -c <config>	Dominio	Nombre del archivo de configuración del servidor a utilizar. El predeterminado es <b>domain.xml</b> .
--cached-dc	Dominio	Si el host no es el controlador de dominios y no puede contactar el controlador de dominios en el arranque, inicie utilizando una copia en caché local de la configuración del dominio.
--debug [<port>]	Autónomo	Active el modo de depuración con un argumento opcional para especificar el puerto. Solo funciona si el script de lanzamiento lo soporta.
-D<name>[=<value>]	Autónomo, dominio	Establecer una propiedad del sistema.
--domain-config=<config>	Dominio	Nombre del archivo de configuración del servidor a utilizar. El predeterminado es <b>domain.xml</b> .
-h, --help	Autónomo, dominio	Presente el mensaje de ayuda y salga.
--host-config=<config>	Dominio	Nombre del archivo de configuración host a utilizar. El valor predeterminado es <b>host.xml</b> .
--interprocess-hc-address=<address>	Dominio	Dirección en la que el controlador host debe escuchar la comunicación del controlador de procesos.
--interprocess-hc-port=<port>	Dominio	Puerto en el que el controlador host debe escuchar la comunicación del controlador de procesos.

Argumento u opción	Modo operativo	Descripción
--master-address=<address>	Dominio	Establezca la propiedad del sistema <b>jboss.domain.master.address</b> con el valor dado. En una configuración predeterminada del controlador host esclavo, esto se utiliza para configurar la dirección del controlador host maestro.
--master-port=<port>	Dominio	Establezca la propiedad del sistema <b>jboss.domain.master.port</b> con el valor dado. En una configuración predeterminada del controlador host esclavo, esto se utiliza para configurar el puerto utilizado para la comunicación de administración nativa por parte del controlador host maestro.
--read-only-server-config=<config>	Autónomo	Nombre del archivo de configuración del servidor a utilizar. Difiere de <b>--server-config</b> y <b>-c</b> en que el archivo original nunca se sobrescribe.
--read-only-domain-config=<config>	Dominio	Nombre del archivo de configuración de dominio a utilizar. Difiere de <b>--domain-config</b> y <b>-c</b> en que el archivo inicial nunca se sobrescribe.
--read-only-host-config=<config>	Dominio	Nombre del archivo de configuración host a utilizar. Difiere de <b>--host-config</b> en que el archivo inicial nunca se sobrescribe.
-P=<url>, -P <url>, --properties=<url>	Autónomo, dominio	Cargar las propiedades del sistema desde la URL dada.
--pc-address=<address>	Dominio	Dirección en la que el controlador de procesos escucha la comunicación de los procesos que controla.
--pc-port=<port>	Dominio	Puerto en el que el controlador de procesos escucha la comunicación de los procesos que controla.
-S<name>[=<value>]	Autónomo	Establecer una propiedad de seguridad.
-secmgr	Autónomo, dominio	Ejecuta el servidor con un gestor de seguridad instalado.
--server-config=<config>	Autónomo	Nombre del archivo de configuración del servidor que se va a utilizar. El predeterminado es <b>standalone.xml</b> .

Argumento u opción	Modo operativo	Descripción
--start-mode=<mode>	Autónomo	<p>Establezca el modo de inicio del servidor. Esta opción no se puede usar junto con <b>--admin-only</b>. Los valores válidos son:</p> <ul style="list-style-type: none"> <li><b>normal</b>: El servidor se iniciará normalmente.</li> <li><b>admin-only</b>: El servidor solo abrirá interfaces administrativas y aceptará solicitudes de administración, pero no ejecutará otros servicios de tiempo de ejecución ni aceptará solicitudes de usuario final.</li> <li><b>suspend</b>: El servidor se iniciará en modo suspendido y no atenderá solicitudes hasta que se haya reanudado.</li> </ul>
-u=<value>, -u <value>	Autónomo, dominio	<p>Establecer la propiedad <b>jboss.default.multicast.address</b> del sistema, que se utiliza en la configuración de la dirección multidifusión de los elementos de vinculación de sockets en los archivos de configuración. Si no se especifica ningún valor, se establecerá de forma predeterminada en <b>230.0.0.4</b>.</p>
-v, -V, --version	Autónomo, dominio	<p>Presenta la versión del servidor de aplicaciones y sale.</p>



### AVISO

Los archivos de configuración que se distribuyen con JBoss EAP se configuran para manejar el comportamiento de los switches (por ej., **-b** y **-u**). Si usted cambia los archivos de configuración para que no utilicen más la propiedad de sistemas controlada por el switch, no afectará en nada añadirlo al comando de lanzamiento.

## A.2. ARGUMENTOS DE LA HERRAMIENTA ADD-USER

La siguiente tabla describe los argumentos disponibles para el script **add-user.sh** o **add-user.bat**, una herramienta para agregar nuevos usuarios al archivo de propiedades para una autenticación fuera de lo común.

**Tabla A.2. Argumentos para el comando Add-User**

Argumento de línea de comandos	Descripción
-a	Cree un usuario en el dominio de aplicaciones. Si se omite, se creará de forma predeterminada un usuario en el dominio de administración.
-dc <value>	El directorio de configuración de dominio que contendrá los archivos de propiedades. Si se omite, el directorio predeterminado es <b>EAP_HOME/domain/configuration/</b> .
-sc <value>	Un directorio de configuración de servidor autónomo alternativo que contendrá los archivos de propiedades. Si se omite, el directorio predeterminado es <b>EAP_HOME/standalone/configuration/</b> .
-up, --user-properties <value>	El nombre del archivo alternativo de propiedades de usuario. Puede ser una ruta absoluta o puede ser un nombre de archivo utilizado junto con el argumento <b>-sc</b> o <b>-dc</b> que especifica el directorio de configuración alternativo.
-g, --group <value>	Una lista de grupos separada por comas para asignar este usuario.
-gp, --group-properties <value>	El nombre del archivo alternativo de propiedades del grupo. Puede ser una ruta absoluta o un nombre de archivo utilizado junto con el argumento <b>-sc</b> o <b>-dc</b> que especifica el directorio de configuración alternativo.
-p, --password <value>	La contraseña del usuario
-u, --user <value>	El nombre del usuario. Los nombres de usuarios solo pueden contener los siguientes caracteres, en cualquier cantidad y orden: <ul style="list-style-type: none"> <li>• Caracteres alfanuméricos (a-z, A-Z, 0-9)</li> <li>• Guiones (-), puntos (.), comas (,), arroba (@)</li> <li>• Barra diagonal inversa (\)</li> <li>• Signo igual (=)</li> </ul>
-r, --realm <value>	El nombre del dominio utilizado para proteger las interfaces de administración. Si se omiten, el valor predeterminado será: <b>ManagementRealm</b> .
-s, --silent	Ejecute el script <b>add-user</b> sin salida a la consola.
-e, --enable	Habilite el usuario.

Argumento de línea de comandos	Descripción
-d, --disable	Inhabilite el usuario
-cw, --confirm-warning	Confirma las advertencia automáticamente en modo interactivo.
-h, --help	Mostrar la información de uso para el script <b>add-user</b> .
-ds, --display-secret	Imprima el valor secreto en modo no interactivo

### A.3. ATRIBUTOS DE INTERFAZ



#### NOTA

Los nombres de atributos en esta tabla se enumeran como aparecen en el modelo de administración, por ejemplo, al usar la CLI de administración. Consulte el archivo de definición del esquema localizado en **EAP\_HOME/docs/schema/wildfly-config\_5\_0.xsd** para ver los elementos como aparecen en el XML, ya que puede haber diferencias en relación con el modelo de administración.

**Tabla A.3. Valores y atributos de la interfaz**

Elemento interfaz	Descripción
any	Elemento vacío que indica que parte del criterio de la selección para una interfaz debe ser que cumpla con alguno de los criterios anidados, no necesariamente con todos.
any-address	Elemento vacío que indica que los sockets que hacen uso de esta interfaz se deben enlazar a una dirección comodín. Se utilizará la dirección comodín IPv6 (: : ) a menos que la propiedad del sistema <b>java.net.preferIPv4Stack</b> esté configurada como verdadera, en cuyo caso se utilizará la dirección comodín IPv4 (0.0.0.0). Si un socket está enlazado a una dirección IPv6 anylocal en una máquina con doble pila, podrá aceptar tráfico IPv6 e IPv4; si está enlazado a una dirección IPv4 anylocal (mapeada para IPv4), solamente podrá aceptar tráfico IPv4.
inet-address	Una dirección IP en IPv6 o IPv4 con una notación decimal con punto o un nombre de host que pueda ser resuelto como una dirección IP.
link-local-address	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si una dirección está asociada con su enlace local.

Elemento interfaz	Descripción
loopback	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si es una interfaz loopback.
loopback-address	Una dirección loopback que puede que no esté configurada en la interfaz loopback de la máquina. Difiere de tipo de inet-address, ya que el valor especificado será utilizado, incluso si no se puede encontrar un NIC que tenga la dirección IP asociada.
multicast	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si soporta multicast o no.
Nombre	El nombre de la interfaz.
nic	El nombre de una interfaz de red (por ejemplo, eth0, eth1, lo).
nic-match	Una expresión regular frente a la cual los nombres de las interfaces de red disponibles en la máquina pueden coincidir para encontrar una interfaz aceptable.
not	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser que no cumpla con ninguno de los criterios establecidos.
point-to-point	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si es una interfaz punto-a-punto.
public-address	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si tiene una dirección enrutable públicamente o no.
site-local-address	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si una dirección está asociada con su sitio local.
subnet-match	Una dirección IP de red y el número de bits en el prefijo de red de la dirección, escrito en <i>notación con barras</i> , por ejemplo, <b>192.168.0.0/16</b> .
up	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si está funcionando o no.
Virtual	Elemento vacío que indica que parte del criterio de selección para una interfaz debe ser si es una interfaz virtual o no.

## A.4. ATRIBUTOS DE ENLACES DE SOCKETS



## NOTA

Los nombres de atributos en esta tabla se enumeran como aparecen en el modelo de administración, por ejemplo, al usar la CLI de administración. Consulte el archivo de definición del esquema localizado en **EAP\_HOME/docs/schema/wildfly-config\_5\_0.xsd** para ver los elementos como aparecen en el XML, ya que puede haber diferencias en relación con el modelo de administración.

**Tabla A.4. Atributos de enlaces de sockets**

Atributo	Descripción
client-mappings	Especifica la asignación de clientes para este enlace de socket. Un cliente que se conecte a este socket debe usar la dirección de destino especificada en el mapa que corresponda con su interfaz de salida deseada. Así permite que las topologías de red avanzadas, usen la traducción de dirección de redes o tengan enlaces en múltiples interfaces de red para funcionar. Cada mapeo debe ser evaluado en orden declarado con la primera correspondencia utilizada para determinar el destino.
fixed-port	Si el valor del puerto debería permanecer fijo incluso cuando las diferencias numéricas son aplicadas a los demás sockets en el grupo de sockets.
interface	Nombre de la interfaz a la cual se debe vincular el socket, o para los sockets multidifusión, la interfaz en la cual se debe escuchar. Esta debe ser una de las interfaces declaradas. Si no está definido, se utilizará el valor del atributo <b>default-interface</b> del grupo de enlaces de sockets vinculado.
multicast-address	Dirección multidifusión en la cual el socket debería recibir el tráfico de multidifusión. Si no se especifica, el socket no será configurado para recibir multidifusión.
multicast-port	Puerto en el cual el socket debe recibir tráfico multidifusión. Se debe configurar si <b>multicast-address</b> está configurado.
Nombre	El nombre del socket. Los servicios que necesiten acceder a la información de configuración de socket la encontrarán por el nombre. Este atributo es obligatorio.
port	El número de puerto al cual el socket debería enlazarse. Observe que este valor puede sobrescribirse si los servidores aplican un port-offset para incrementar o disminuir todos los valores de puerto.

## A.5. ENLACES DE SOCKETS PREDETERMINADOS

En las tablas siguientes se muestran los enlaces de sockets predeterminados para cada grupo de enlaces de sockets.



- [standard-sockets](#)
- [ha-sockets](#)
- [full-sockets](#)
- [full-ha-sockets](#)
- [load-balancer-sockets](#)

**Tabla A.5. standard-sockets**

Enlace de sockets	Puerto	Descripción
ajp	8009	Protocolo Apache JServ. Utilizado para balanceo de carga y clústers HTTP.
http	8080	El puerto predeterminado para las aplicaciones web implementadas.
https	8443	Conexión SSL encriptada entre las aplicaciones web implementadas y los clientes.
management-http	9990	Utilizado para comunicación HTTP con la capa de administración.
management-https	9993	Utilizado para la comunicación HTTPS con la capa de administración.
txn-recovery-environment	4712	El administrador de recuperación de transacciones JTA.
txn-status-manager	4713	El administrador de transacciones JTA / JTS.

**Tabla A.6. ha-sockets**

Enlace de sockets	Puerto	Puerto multidifusión	Descripción
ajp	8009		Protocolo Apache JServ. Utilizado para balanceo de carga y clústers HTTP.
http	8080		El puerto predeterminado para las aplicaciones web implementadas.
https	8443		Conexión SSL encriptada entre las aplicaciones web implementadas y los clientes.

Enlace de sockets	Puerto	Puerto multidifusión	Descripción
jgroups-mping		45700	Multidifusión. Utilizado para descubrir la membresía inicial en un clúster HA.
jgroups-tcp	7600		Descubrimiento de pares unidifusión en clústeres HA mediante TCP.
jgroups-udp	55200	45688	Descubrimiento de pares multidifusión en clústeres HA mediante UDP.
management-http	9990		Utilizado para comunicación HTTP con la capa de administración.
management-https	9993		Utilizado para la comunicación HTTPS con la capa de administración.
modcluster		23364	Puerto multicast para la comunicación entre JBoss EAP y el balanceador de carga HTTP.
txn-recovery-environment	4712		El administrador de recuperación de transacciones JTA.
txn-status-manager	4713		El administrador de transacciones JTA / JTS.

**Tabla A.7. full-sockets**

Enlace de sockets	Puerto	Descripción
ajp	8009	Protocolo Apache JServ. Utilizado para balanceo de carga y clústers HTTP.
http	8080	El puerto predeterminado para las aplicaciones web implementadas.
https	8443	Conexión SSL encriptada entre las aplicaciones web implementadas y los clientes.
iiop	3528	Servicios CORBA para transacciones JTS y otros servicios dependientes ORB.

Enlace de sockets	Puerto	Descripción
iiop-ssl	3529	Servicios CORBA SSL encriptados.
management-http	9990	Utilizado para comunicación HTTP con la capa de administración.
management-https	9993	Utilizado para la comunicación HTTPS con la capa de administración.
txn-recovery-environment	4712	El administrador de recuperación de transacciones JTA.
txn-status-manager	4713	El administrador de transacciones JTA / JTS.

**Tabla A.8. full-ha-sockets**

Name	Puerto	Puerto multidifusión	Descripción
ajp	8009		Protocolo Apache JServ. Utilizado para balanceo de carga y clústers HTTP.
http	8080		El puerto predeterminado para las aplicaciones web implementadas.
https	8443		Conexión SSL encriptada entre las aplicaciones web implementadas y los clientes.
iiop	3528		Servicios CORBA para transacciones JTS y otros servicios dependientes ORB.
iiop-ssl	3529		Servicios CORBA SSL encriptados.
jgroups-mping		45700	Multidifusión. Utilizado para descubrir la membresía inicial en un clúster HA.
jgroups-tcp	7600		Descubrimiento de pares unidifusión en clústeres HA mediante TCP.
jgroups-udp	55200	45688	Descubrimiento de pares multidifusión en clústeres HA mediante UDP.

Name	Puerto	Puerto multidifusión	Descripción
management-http	9990		Utilizado para comunicación HTTP con la capa de administración.
management-https	9993		Utilizado para la comunicación HTTPS con la capa de administración.
modcluster		23364	Puerto multicast para la comunicación entre JBoss EAP y el balanceador de carga HTTP.
txn-recovery-environment	4712		El administrador de recuperación de transacciones JTA.
txn-status-manager	4713		El administrador de transacciones JTA / JTS.

**Tabla A.9. load-balancer-sockets**

Name	Puerto	Puerto multidifusión	Descripción
http	8080		El puerto predeterminado para las aplicaciones web implementadas.
https	8443		Conexión SSL encriptada entre las aplicaciones web implementadas y los clientes.
management-http	9990		Utilizado para comunicación HTTP con la capa de administración.
management-https	9993		Utilizado para la comunicación HTTPS con la capa de administración.
mcmp-management	8090		El puerto para la conexión del Protocolo de administración Mod-Cluster (MCMP) para transmitir los eventos de ciclo de vida.
modcluster		23364	Puerto multicast para la comunicación entre JBoss EAP y el balanceador de carga HTTP.

*Revised on 2018-01-11 05:29:29 EST*