



# Proyecto Adalid, Plataforma jee1

## Guía de Instalación y Configuración en ambientes Linux

### Tabla de Contenido

Archivos necesarios para la instalación de Adalid y jee1 .....	2
Requisitos para la instalación .....	2
Instalación de librerías de terceras partes .....	3
Instalación de Adalid.....	3
Instalación de complementos para NetBeans.....	3
Creación de meta-proyectos con la plantilla jee1.....	6
Extensión de meta-proyectos creados con la plantilla jee1 .....	7
Configuración de NetBeans para meta-proyectos creados con la plantilla jee1 .....	7
Librerías Ant para NetBeans.....	8
Configuración de eclipse para meta-proyectos creados con la plantilla jee1 .....	8
Librerías de Usuario para eclipse .....	16
Configuración de NetBeans para proyectos generados con la plataforma jee1 .....	16
Configuración de eclipse para proyectos generados con la plataforma jee1 .....	18
Generación de proyectos con la plataforma jee1 .....	18
Configuración de proyectos generados con la plataforma jee1 .....	21
Configuración de PostgreSQL para proyectos generados con jee1 .....	21
Configuración de GlassFish para proyectos generados con jee1 .....	21
Configuración de WildFly para proyectos generados con jee1 .....	22
Ejecución de proyectos generados con jee1.....	27



## Archivos necesarios para la instalación de Adalid y jee1

Para instalar **Adalid** y su plataforma **jee1** son necesarios los siguientes archivos:

- third-party.zip
- wildfly.zip
- adalid.zip
- jee1.zip

Para instalar, descargue los archivos que se encuentran en el directorio **zip** del proyecto **adalid-jee1**. La URL es <https://github.com/proyecto-adalid/adalid-jee1/tree/master/zip>.

Para obtener la documentación del producto, descargue los archivos que se encuentran en el directorio **documentos**. La URL es <https://github.com/proyecto-adalid/adalid-jee1/tree/master/documentos>.

Alternativamente, puede descargar los archivos de instalación la documentación completa del producto y la aplicación de ejemplo **xyz1** haciendo un *clone* del proyecto **adalid-jee1**. La URL del proyecto es <https://github.com/proyecto-adalid/adalid-jee1.git>.

**Nota:** el directorio **zip** podría contener múltiples archivos con extensión **.zip** y con nombres que comienzan por **adalid** y por **jee1**, seguidos de la versión y el número de revisión del producto. Utilice los archivos correspondientes a la versión y revisión más reciente.

## Requisitos para la instalación

Los siguientes productos son necesarios para el funcionamiento de Adalid y/o de los meta-proyectos creados con la plantilla **jee1** y/o de los proyectos (aplicaciones) generados con la plataforma **jee1**. Las versiones corresponden a las versiones mínimas requeridas. Teóricamente debería funcionar con versiones superiores.

- Red Hat Linux, Ubuntu o similar.
- Java jdk1.8.0\_77 o superior.  
URL: <http://www.oracle.com/technetwork/java/javase/downloads>
- NetBeans 8.0.2, opción Java EE (incluye GlassFish 4.1) o superior.  
URL: <https://netbeans.org/downloads>  
Alternativamente, eclipse Mars o superior.  
URL: <https://www.eclipse.org/downloads>
- GlassFish 4.1 o superior.  
URL: <https://glassfish.java.net/download.html>  
Alternativamente, WildFly 10.0.0.Final o superior.  
URL: <http://wildfly.org/downloads/>
- PostgreSQL 9.2 o superior.  
URL: <http://www.enterprisedb.com/products-services-training/pgdownload>
- Mozilla Firefox 34.0 o superior.  
URL: <http://www.mozilla.org/es-ES/firefox/new>  
Alternativamente, Microsoft Internet Explorer 11.0 o superior.  
URL: <http://windows.microsoft.com/es-ve/internet-explorer/download-ie>



## Instalación de librerías de terceras partes

Para instalar las librerías de terceras partes siga los siguientes pasos:

- Crear el directorio **workspace**

**Nota:** el directorio **workspace** puede ser creado en cualquier directorio cuya ruta no contenga espacios en blanco; en lo sucesivo se da por hecho que el directorio **workspace** se encuentra en el directorio **/opt**.

- Descomprimir el archivo **third-party.zip** en **/opt/workspace**.

Esta acción crea el directorio **/opt/workspace/third-party**.

- Descomprimir el archivo **wildfly.zip** en **/opt/workspace/third-party**.

Esta acción crea el directorio **/opt/workspace/third-party/wildfly**.

## Instalación de Adalid

Después de instalar las librerías de terceras partes, para instalar Adalid solo es necesario descomprimir el archivo **adalid.zip** en **/opt/workspace**. Esta acción crea el directorio **/opt/workspace/adalid**.

## Instalación de complementos para NetBeans

Adalid incluye algunos complementos para facilitar el trabajo con NetBeans. Para instalar estos complementos siga los siguientes pasos:

- Instalar el módulo de plantillas de código de Adalid.

En el menú de barra de NetBeans, seleccione **Tools**.

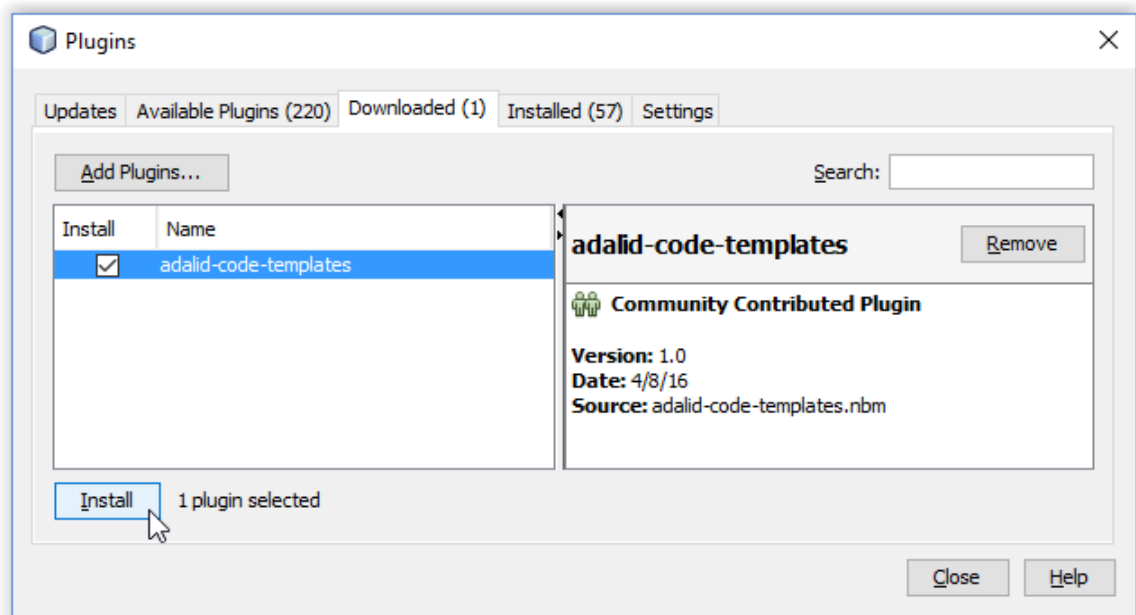
En el menú desplegable **Tools** seleccione **Plugins**.

En la ventana **Plugins**, seleccione la pestaña **Downloaded**.

En la pestaña **Downloaded**, haga clic en el botón **Add Plugins...**

En la ventana **Add Plugins**, busque el directorio **/opt/workspace/adalid/lib**, seleccione el archivo **adalid-code-templates.nbm** y haga clic en el botón **Open**.

De regreso en la ventana **Plugins**, haga clic en el botón **Install**.



- Agregar las plantillas de archivos.

En el menú de barra de NetBeans, seleccione **Tools**.

En el menú desplegable **Tools** seleccione **Templates**.

En la ventana **Template Manager**, seleccione la primera de las carpetas de la lista de plantillas y haga clic en el botón **New Folder**. Esta acción crea una carpeta **New Folder** dentro de la carpeta previamente seleccionada.

En la ventana **Template Manager**, haga clic en el botón **Rename**.

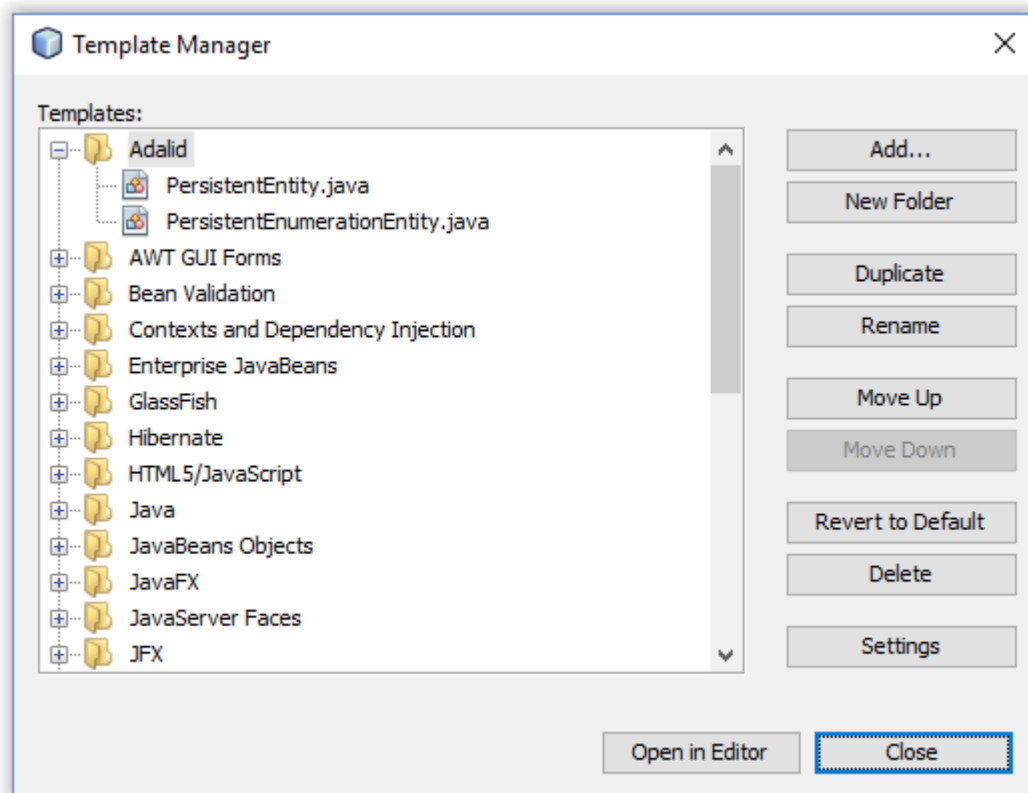
En la ventana **Rename template**, escriba **Adalid** en todos los campos y haga clic en el botón **OK**.

De regreso en la ventana **Template Manager**, seleccione la carpeta **Adalid** en la lista de plantillas, arrástrela fuera de la carpeta que la contiene y suéltela entre otras dos carpetas cualesquiera.

En la ventana **Template Manager**, seleccione la carpeta **Adalid** en la lista de plantillas (debería estar muy cerca del comienzo de la lista, ya que ésta se encuentra en orden alfabético) y haga clic en el botón **Add...**

En la ventana **Add Existing Template**, busque el directorio **/opt/workspace/adalid/source/development/resources/freemarker/file-templates/entities**, seleccione uno cualquiera de los archivos y haga clic en el botón **Add**.

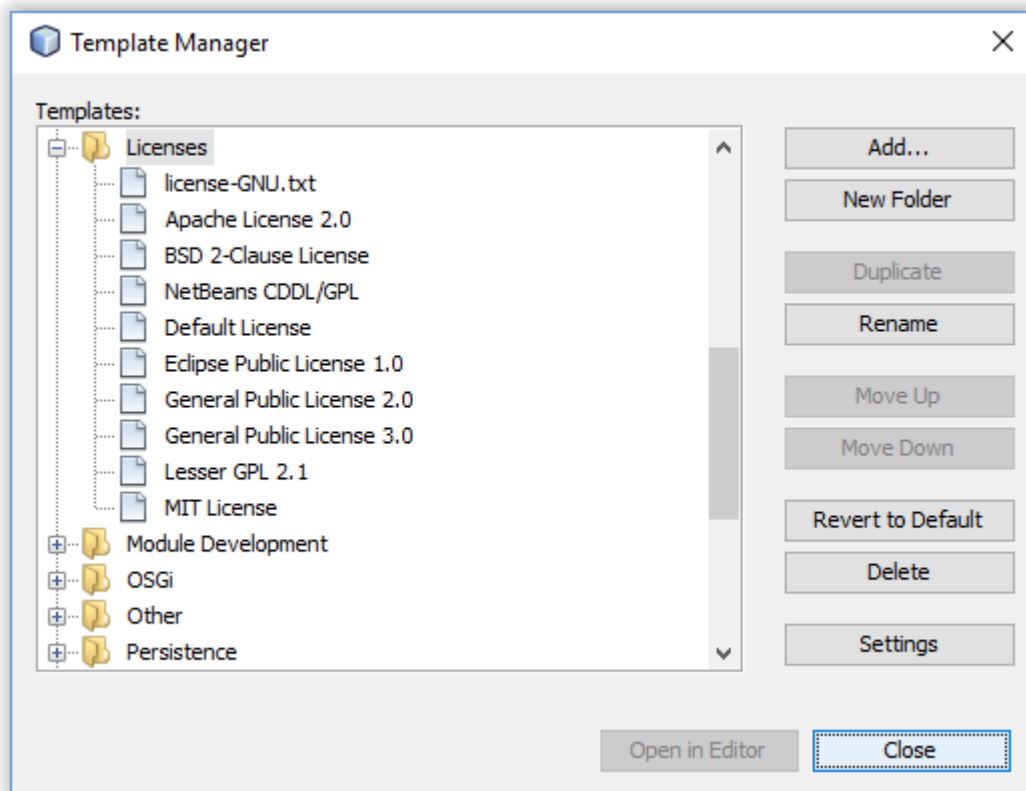
Repita los pasos anteriores para agregar el resto de las plantillas que se encuentran en **/home/workspace/adalid/source/development/resources/freemarker/file-templates/entities**.



- Agregar la plantilla de licencia.

En la ventana **Template Manager**, seleccione la carpeta **Licenses** en la lista de plantillas y haga clic en el botón **Add...**

En la ventana **Add Existing Template**, busque el directorio `/opt/workspace/adalid/source/development/resources/freemarker/file-templates/licenses`, seleccione el archivo `license-GNU.txt` y haga clic en el botón **Add**.



- Reiniciar NetBeans.

## Creación de meta-proyectos con la plantilla jee1

Para crear un meta-proyecto con la plantilla **jee1** siga los siguientes pasos:

- Descomprimir el archivo **jee1.zip** en **/opt/workspace**.

Esta acción crea el directorio **/opt/workspace/jee1**, el cual contiene el meta-proyecto **meta-jee1** en el directorio **/opt/workspace/xyz1/source/meta**.

- Renombrar el directorio **jee1** creado en **/opt/workspace**.

El nuevo nombre debe identificar su proyecto. Solo puede contener letras minúsculas y números, debe comenzar por una letra y no puede ser **workspace**.

**Nota:** en lo sucesivo se da por hecho que el nombre de su proyecto y el nuevo nombre del directorio **jee1** es **xyz1**.

- Revisar y, en caso de ser necesario, ajustar las variables de entorno que se establecen en el archivo **/opt/workspace/xyz1/variables-home.sh**.
- Ejecutar el script **/opt/workspace/xyz1/refactor.sh**.

Esta acción modifica los archivos que se encuentran en el directorio **/opt/workspace/xyz1**; específicamente cambia **jee1** por el nombre de su proyecto (**xyz1** para los efectos de este



documento) y cambia **/opt/workspace/** por la ruta de su directorio **workspace**, en caso de que no se encuentre en **/opt**. Una de las consecuencias de estas acciones es que el meta-proyecto **meta-jee1** en el directorio **/opt/workspace/xyz1/source/meta** es renombrado como **meta-xyz1**.

**Nota:** debe ejecutar el script **refactor.sh** como súper-usuario.

## Extensión de meta-proyectos creados con la plantilla jee1

Los meta-proyectos creados con la plantilla **jee1** sirven de base para el modelaje de sus aplicaciones. La ejecución de los meta-proyectos genera otros proyectos, que implementan las aplicaciones modeladas.

Los meta-proyectos creados con la plantilla **jee1** incluyen los módulos predefinidos de la plataforma **jee1**: Control de Acceso, Auditoría y Control de Pruebas. Por lo tanto, después de finalizar la configuración, es posible generar y ejecutar una aplicación con solo esos módulos, tal como se describe más adelante en esta misma guía.

Para obtener información detallada sobre cómo extender los meta-proyectos creados con la plantilla **jee1** con módulos propios de su aplicación, consulte la [Guía de Meta-Programación](#).

## Configuración de NetBeans para meta-proyectos creados con la plantilla jee1

Si desea construir (*build*) y ejecutar meta-proyectos creados con la plantilla **jee1** utilizando NetBeans, entonces siga los siguientes pasos:

- Definir una variable Ant con la ubicación del directorio de librerías de terceras partes.

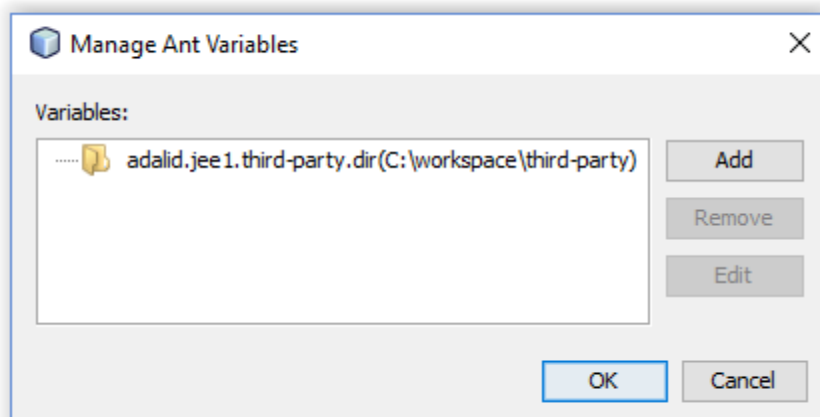
En el menú de barra de NetBeans, seleccione **Tools**.

En el menú desplegable **Tools** seleccione **Ant Variables**.

En la ventana **Manage Ant Variables**, haga clic en el botón **Add**.

En la ventana **Add New Variable** escriba **adalid.jee1.third-party.dir** en el campo **Name** y la ruta completa del directorio third-party, creado en la [Instalación de librerías de terceras partes](#) (usualmente **/opt/workspace/third-party**) en el campo **Location**, y haga clic en el botón **OK**.

De regreso en la ventana **Manage Ant Variables**, haga clic en el botón **OK**.





- Copiar librerías Ant.

Copie los archivos XML que se encuentran en el directorio **/opt/workspace/xyz1/source/development/resources/libraries/netbeans/linux** al subdirectorio **config/org-netbeans-api-project-libraries/Libraries** del directorio de configuración personalizada de NetBeans, conocido como **userdir**. Por omisión, **userdir** está dentro del subdirectorio NetBeans que se encuentra en el directorio de datos de programa del usuario. Para mayor información, consulte la página <http://wiki.netbeans.org/FaqWhatIsUserdir>.

**Nota:** no debe copiar todos los archivos. Algunos de ellos son opcionales, ya que son específicos para GlassFish o WildFly. Copie solo aquellos archivos que correspondan a la versión del producto que va a utilizar. Consulte la sección [Librerías Ant para NetBeans](#) para conocer cuáles archivos son específicos para algún producto.

- Reiniciar NetBeans.

### Librerías Ant para NetBeans

Archivo	Librería	Requerida para ...
adalid-compile-lib-1.xml	adalid-compile-lib-1	Compilar todos los proyectos generados
adalid-compile-meta-lib-1.xml	adalid-compile-meta-lib-1	Compilar y ejecutar todos los meta-proyectos
adalid-runtime-lib-1.xml	adalid-runtime-lib-1	Ejecutar todos los proyectos generados
glassfish-core-modules-4.x.x.xml	glassfish-core-modules	Compilar solo los proyectos generados que utilicen <b>GlassFish versión 4.x.x</b>
glassfish-eclipselink-4.x.x.xml	glassfish-eclipselink	Compilar <b>solo</b> los proyectos generados que utilicen <b>GlassFish versión 4.x.x</b>
jdbc-drivers.xml	jdbc-drivers	Compilar y ejecutar todos los meta-proyectos
vw-jsf12-support.xml	jsf12-support	Compilar todos los proyectos generados
vw-woodstock-components.xml	woodstock-components	Compilar todos los proyectos generados
wildfly-core-modules-10.x.x.xml	wildfly-core-modules	Compilar solo los proyectos generados que utilicen <b>WildFly versión 10.x.x</b>
wildfly-eclipselink-10.x.x.xml	wildfly-eclipselink	Compilar solo los proyectos generados que utilicen <b>WildFly versión 10.x.x</b>

## Configuración de eclipse para meta-proyectos creados con la plantilla jee1

Si desea construir (*build*) y ejecutar meta-proyectos creados con la plantilla **jee1** utilizando eclipse, entonces siga los siguientes pasos:

- Definir un JRE utilizando el JDK de Java SE 1.8 y seleccionarlo como JRE por *default*.

En el menú de barra de eclipse, seleccione **Window**.

En el menú desplegable **Window**, seleccione **Preferences**.

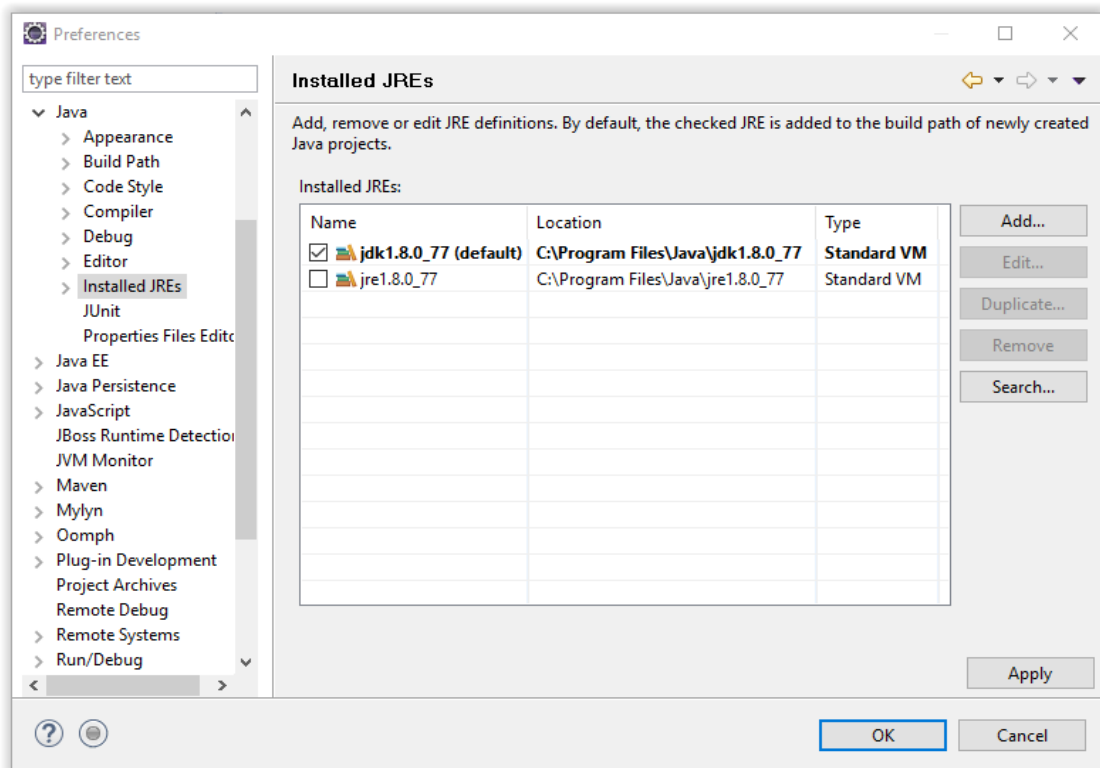
En la ventana **Preferences**, expanda el nodo **Java**; dentro del nodo **Java** seleccione el nodo **Installed JREs**; finalmente el haga clic en el botón **Add...**

En la ventana **Add JRE** seleccione **Standard VM** y haga clic en el botón **Next**. Escriba la ruta

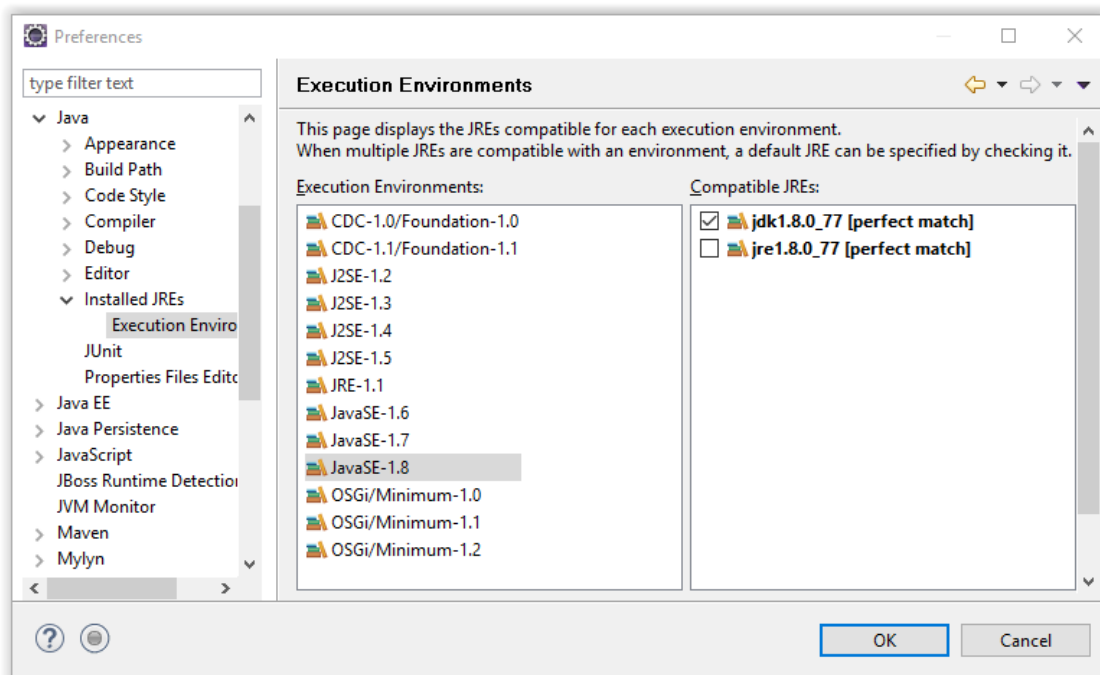


completa del JDK en el campo **JRE home**, y haga clic en el botón **Finish**.

De regreso en la ventana **Preferences**, en el cuadro **Installed JREs** seleccione la casilla de verificación que precede el JRE recién agregado y haga clic en el botón **Apply**.



Ahora expanda el nodo **Installed JREs**; dentro del nodo **Installed JREs** seleccione el nodo **Execution Environments**; en el cuadro **Execution Environments** seleccione **JavaSE-1.8**; el cuadro **Compatible JREs** seleccione la casilla de verificación que precede el JRE recién agregado y haga clic en el botón **OK**.

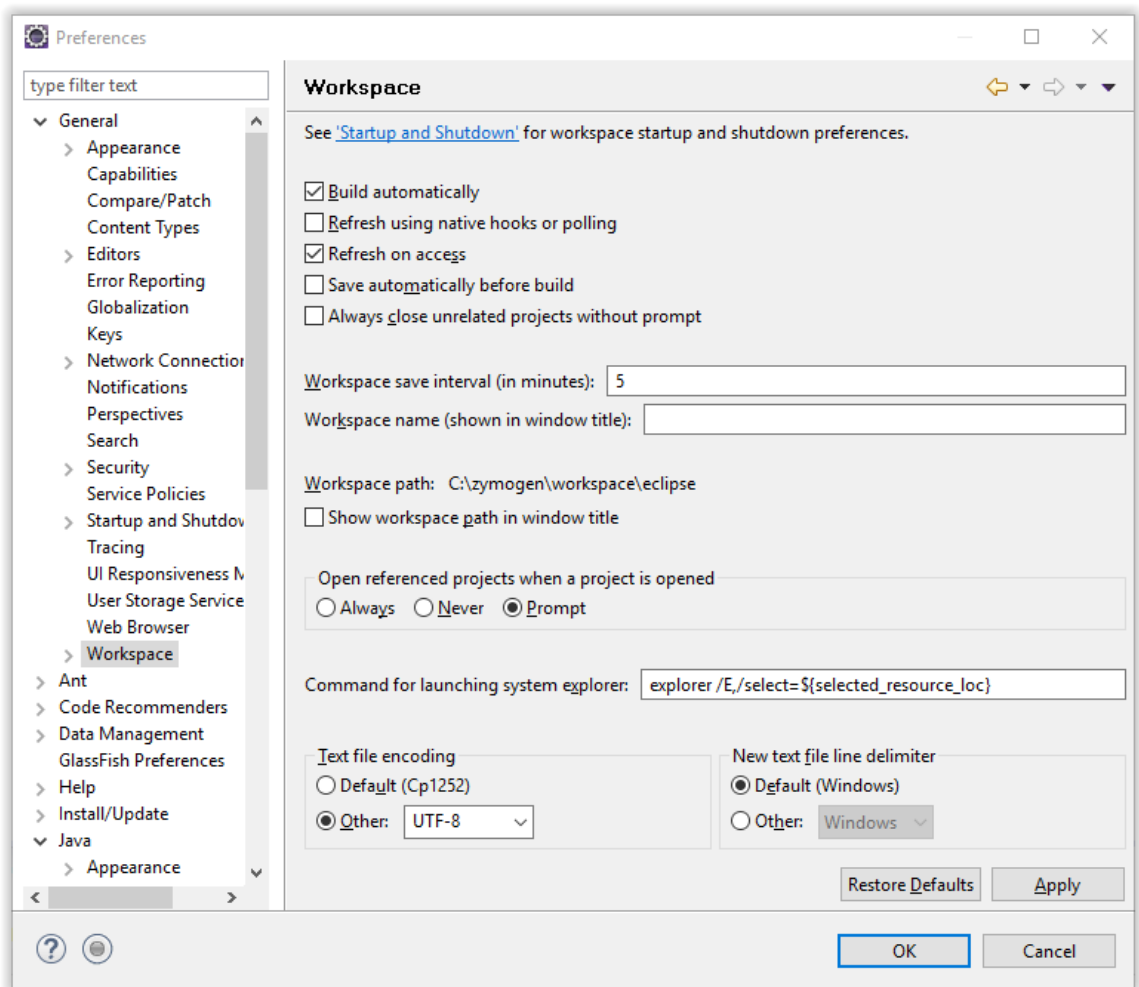


- Ajustar el *encoding* del *workspace*.

En el menú de barra de eclipse, seleccione **Window**.

En el menú desplegable **Window**, seleccione **Preferences**.

En la ventana **Preferences**, expanda el nodo **General**; dentro del nodo **General** seleccione el nodo **Workspace**; entre las opciones que aparecen seleccione la opción **Other** del grupo **Text file encoding** y luego la opción **UTF-8** en la lista desplegable contigua; finalmente el haga clic en el botón **OK**.



- Definir una propiedad Ant con la ubicación del directorio de librerías de terceras partes.

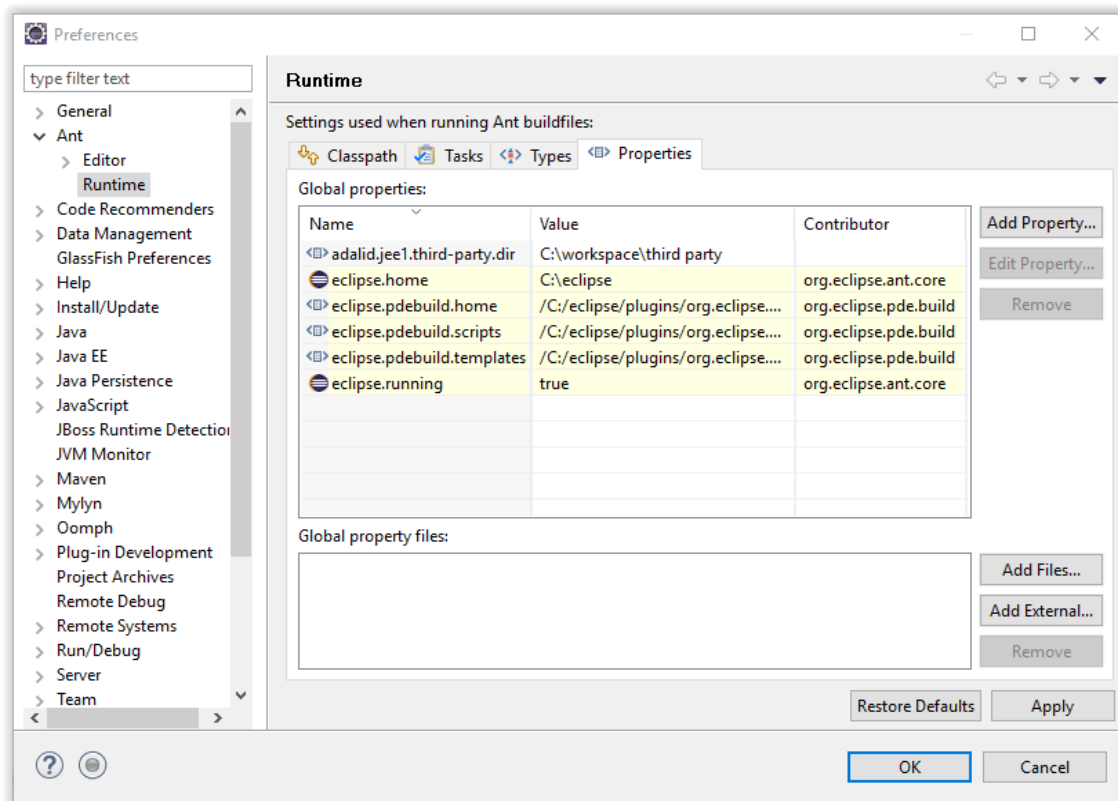
En el menú de barra de eclipse, seleccione **Window**.

En el menú desplegable **Window**, seleccione **Preferences**.

En la ventana **Preferences**, expanda el nodo **Ant**; dentro del nodo **Ant** seleccione el nodo **Runtime** y luego la pestaña **Properties**; finalmente el haga clic en el botón **Add Property...**

En la ventana **Add Property** escriba **adalid.jee1.third-party.dir** en el campo **Name** y la ruta completa del directorio third-party, creado en la [Instalación de librerías de terceras partes](#) (usualmente **/opt/workspace/third-party**) en el campo **Value**, y haga clic en el botón **OK**.

De regreso en la ventana **Preferences**, haga clic en el botón **OK**.



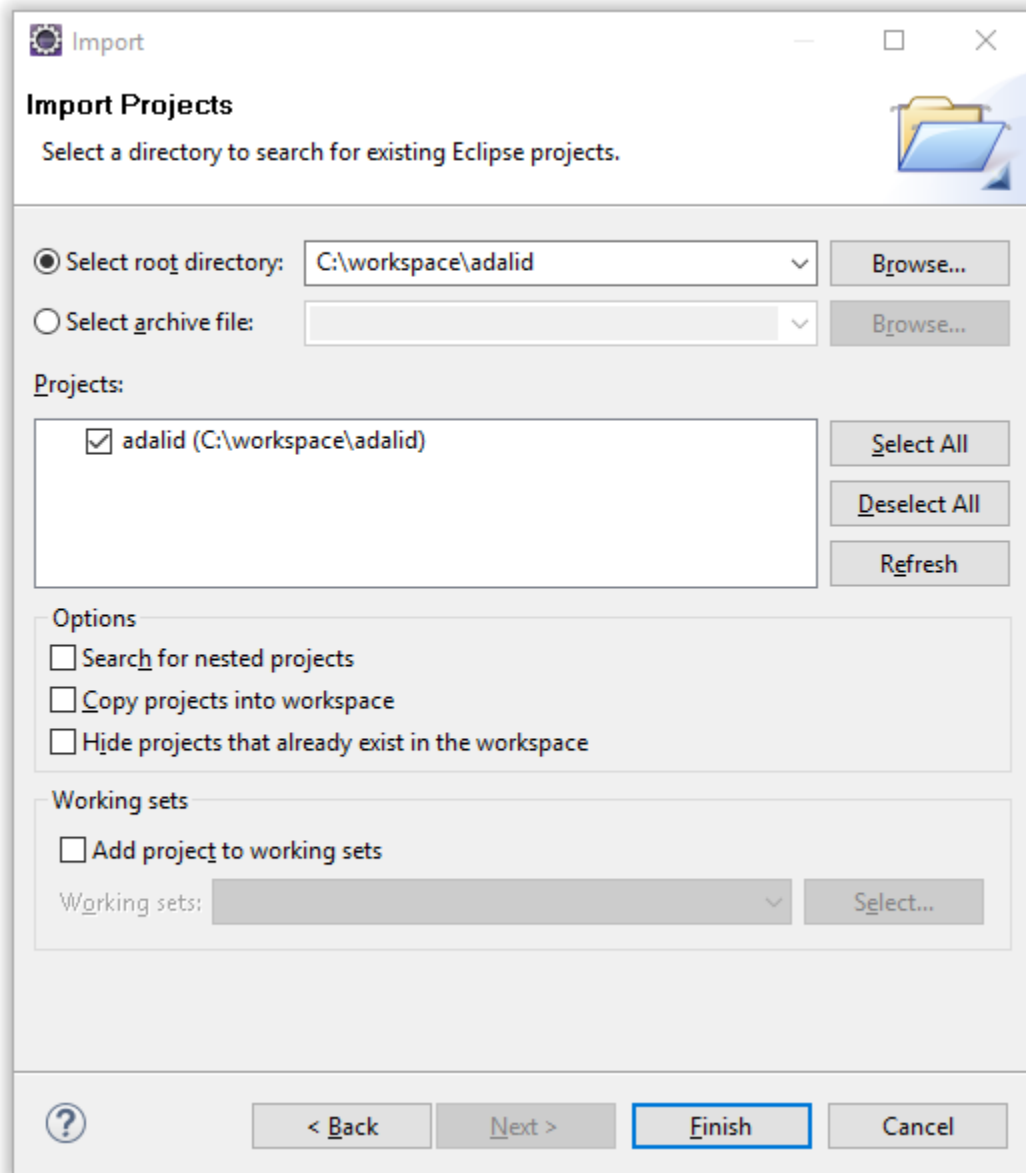
- Importar el directorio **/opt/workspace/adalid** al *workspace*.

En el menú de barra de eclipse, seleccione **File**.

En el menú desplegable **File**, seleccione **Import...**

En la ventana **Import**, seleccione la opción **Existing Projects into Workspace** dentro de la carpeta **General**; haga clic en el botón **Next**.

En la ventana **Import**, seleccione **Select root directory** y busque (haciendo clic en el botón **Browse**) el directorio **/opt/workspace/adalid**. Sin seleccionar ninguna de las opciones, haga clic en el botón **Finish**.



- Importar el directorio **/opt/workspace/third-party** al *workspace*.

Repita el procedimiento anterior para importar el directorio **/opt/workspace/third-party**.

- Importar el directorio HOME de GlassFish al *workspace*.

Si desea utilizar GlassFish como servidor de aplicaciones, repita el procedimiento anterior para importar el directorio HOME de GlassFish.

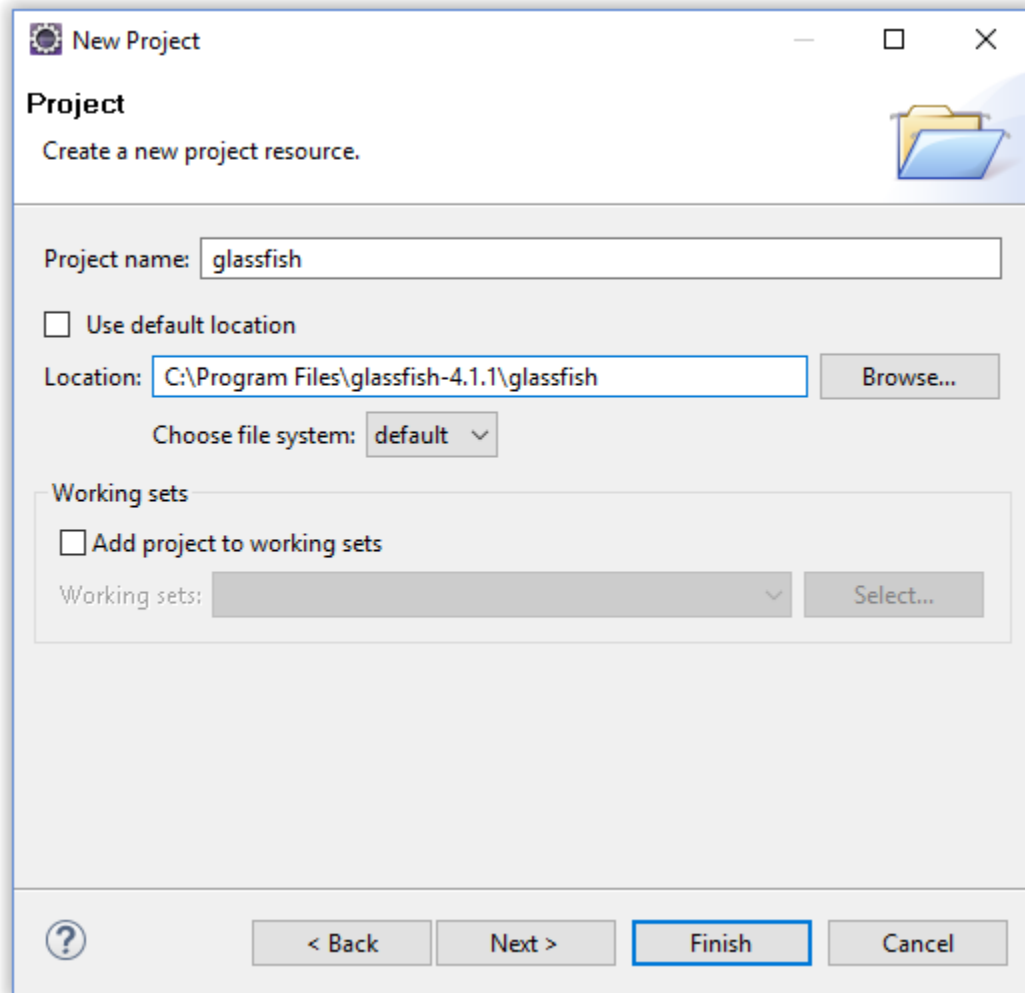
**Nota:** el script **refactor.sh**, ejecutado como parte de la [Creación de meta-proyectos con la plantilla jee1](#), copia un archivo **.project** al directorio HOME de GlassFish para permitir su importación. Esta copia podría fallar si el script **refactor.sh** es ejecutado por un usuario sin autorización para escritura en el directorio HOME de GlassFish. En este caso se debe agregar el directorio HOME de GlassFish al *workspace* como un nuevo proyecto llamado **glassfish**. Para agregarlo, siga los siguientes pasos:

En el menú de barra de eclipse, seleccione **File**.

En el menú desplegable **File**, seleccione **New** y luego **Project...**

En la ventana **New Project**, seleccione la opción **Project** dentro de la carpeta **General**; haga clic en el botón **Next**.

En la ventana **New Project**, escriba **glassfish** en **Project name**, desmarque la opción **Use default location** y escriba la ruta completa del directorio HOME de GlassFish en **Location** (o búsquela haciendo clic en el botón **Browse**); haga clic en el botón **Finish**.



- Importar el directorio HOME de WildFly al *workspace*.

Si desea utilizar WildFly como servidor de aplicaciones, repita el procedimiento anterior para importar el directorio HOME de WildFly.

**Nota:** el script **refactor.sh**, ejecutado como parte de la [Creación de meta-proyectos con la plantilla jee1](#), copia un archivo **.project** al directorio HOME de WildFly para permitir su importación. Esta copia podría fallar si el script **refactor.sh** es ejecutado por un usuario sin autorización para escritura en el directorio HOME de WildFly. En este caso se debe agregar el directorio HOME de WildFly al *workspace* como un nuevo proyecto llamado **wildfly**. Para agregarlo, siga los pasos indicados en el punto anterior.

- Importar las librerías de usuario de Adalid y **jee1** al *workspace*.

En el menú de barra de eclipse, seleccione **Window**.

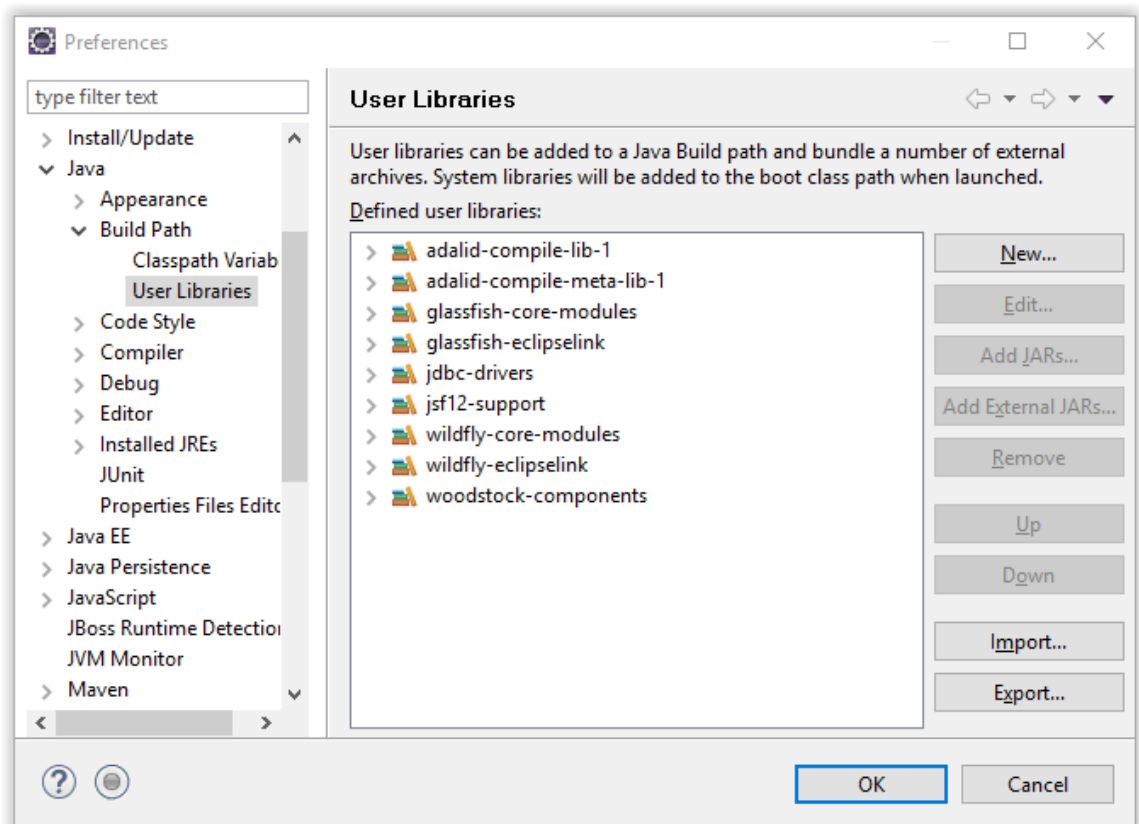
En el menú desplegable **Window**, seleccione **Preferences**.

En la ventana **Preferences**, expanda el nodo **Java**; dentro del nodo **Java** expanda el nodo **Build Path**; dentro del nodo **Build Path** seleccione el nodo **User Libraries**; haga clic en el botón **Import...**

En la ventana **Import User Libraries**, haga clic en el botón **Browse** y busque el directorio **/opt/workspace/xyz1/source/development/resources/libraries/eclipse**, seleccione el archivo **adalid.userlibraries** y haga clic en el botón **Open**. Al regresar a la ventana **Import User Libraries**, haga clic en el botón **OK**.

Repita los pasos anteriores para importar las librerías definidas en el resto de los archivos que se encuentran en **/opt/workspace/xyz1/source/development/resources/libraries/eclipse**.

**Nota:** no debe importar todos los archivos. Algunos de ellos son opcionales, ya que son específicos para GlassFish o WildFly. Importe solo aquellos archivos que correspondan a la versión del producto que va a utilizar. Consulte la sección [Librerías de Usuario para eclipse](#) para conocer cuáles archivos son específicos para algún producto.



- Importar el meta-proyecto **meta-xyz1** al *workspace*.



Repita el procedimiento descrito anteriormente para importar el directorio **/opt/workspace/xyz1/source/meta**.

### Librerías de Usuario para eclipse

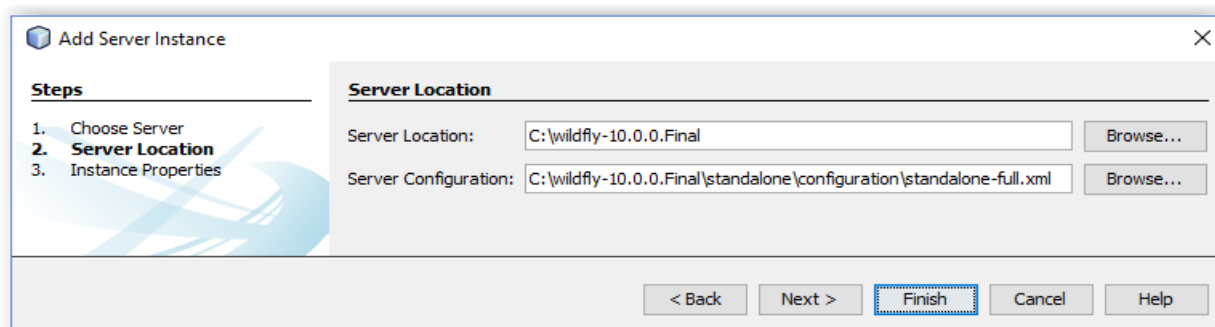
Archivo	Librería	Requerida para ...
adalid.userlibraries	adalid-compile-lib-1	Compilar todos los proyectos generados
	adalid-compile-meta-lib-1	Compilar y ejecutar todos los meta-proyectos
glassfish-4.x.x.userlibraries	glassfish-core-modules	Compilar solo los proyectos generados que utilicen <b>GlassFish versión 4.x.x</b>
	glassfish-eclipselink	Compilar <b>solo</b> los proyectos generados que utilicen <b>GlassFish versión 4.x.x</b>
jdbc-drivers.userlibraries	jdbc-drivers	Compilar y ejecutar todos los meta-proyectos
Visualweb2.userlibraries	jsf12-support	Compilar todos los proyectos generados
	woodstock-components	Compilar todos los proyectos generados
wildfly-10.x.x.userlibraries	wildfly-core-modules	Compilar solo los proyectos generados que utilicen <b>WildFly versión 10.x.x</b>
	wildfly-eclipselink	Compilar solo los proyectos generados que utilicen <b>WildFly versión 10.x.x</b>

## Configuración de NetBeans para proyectos generados con la plataforma jee1

Si desea construir (*build*) y ejecutar los proyectos generados con la plataforma **jee1** utilizando NetBeans, entonces deberá instalar y agregar al menos un servidor de aplicaciones a la configuración de NetBeans, preferiblemente antes de generar el proyecto.

La instalación de NetBeans con Java EE incluye la instalación de GlassFish y además agrega el servidor instalado a la configuración de NetBeans. Si prefiere trabajar con WildFly debe instalarlo por separado. Consulte la documentación de WildFly para obtener información sobre su instalación. Después de instalar WildFly, debe agregarlo a la configuración de NetBeans. Consulte la documentación de NetBeans para obtener información sobre como agregar un nuevo servidor de aplicaciones.

Cuando se utiliza WildFly como servidor de aplicaciones, los proyectos generados con la plataforma **jee1** necesitan utilizar la configuración completa y, por lo tanto, al agregar el servidor a NetBeans es necesario especificar el archivo de configuración **standalone-full.xml**.

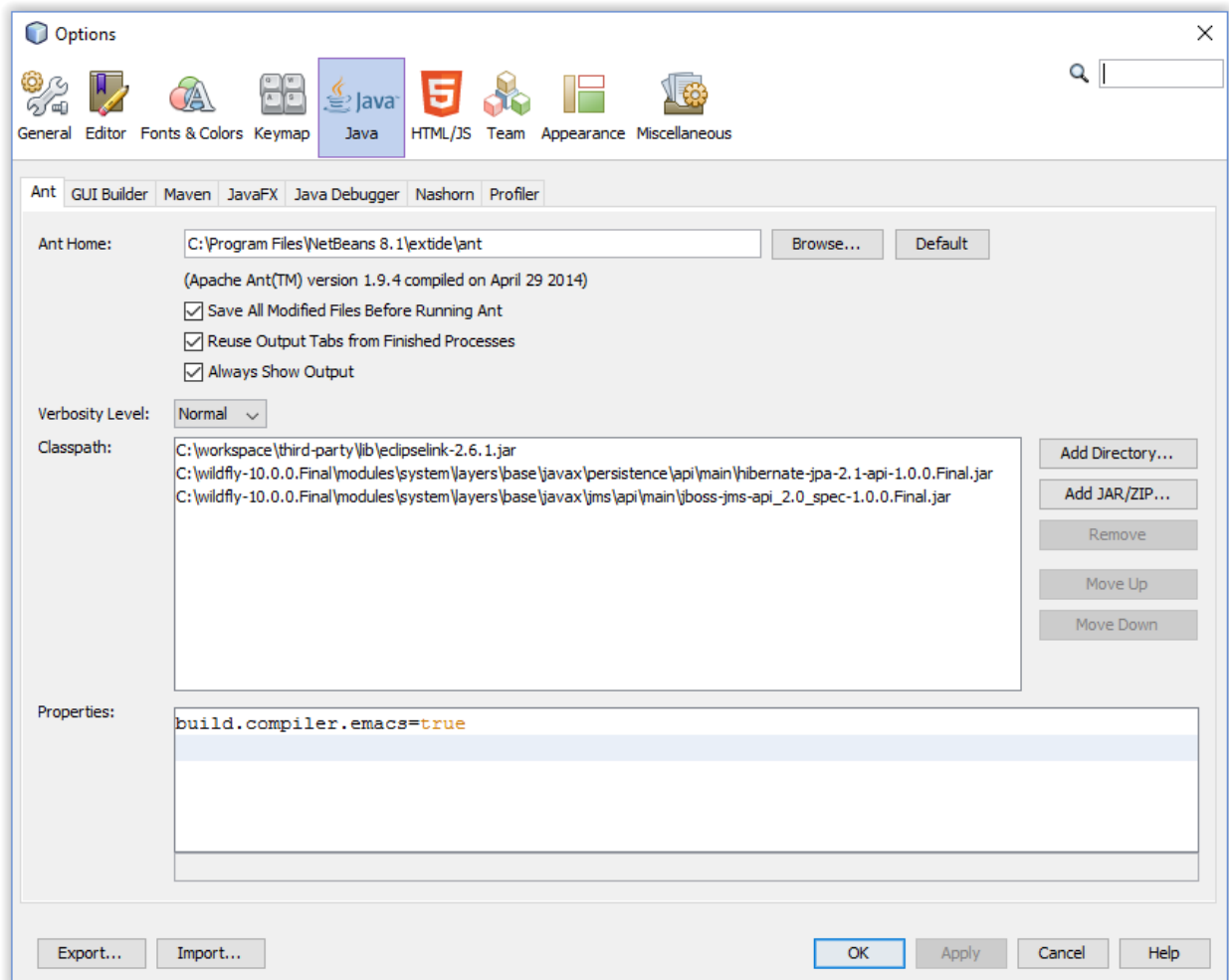






Cuando se utiliza WildFly también es necesario configurar el **Classpath** de **Ant**. Para configurarlo, siga los siguientes pasos:

- En el menú de barra de NetBeans, seleccione **Tools**.
- En el menú desplegable **Tools** seleccione **Options**.
- En la ventana **Options**, seleccione la opción **Java** y luego la pestaña **Ant**.
- En la pestaña **Ant**, haga clic en el botón **Add JAR/ZIP...** que se encuentra a la derecha del cuadro **Classpath**.
- En la ventana **Add JAR File**, busque el directorio **/opt/workspace/third-party/lib**, seleccione el archivo **eclipseLink-2.6.1.jar** y haga clic en el botón **OK**.
- De regreso en la pestaña **Ant**, nuevamente haga clic en el botón **Add JAR/ZIP...**
- En la ventana **Add JAR File**, busque el directorio del módulo **javax.persistence.api** de WildFly (subdirectorio **javax/persistence/api/main** del directorio de módulos de WildFly), seleccione el archivo **hibernate-jpa-2.x-api-1.x.x.Final.jar** y haga clic en el botón **OK**.
- De regreso en la pestaña **Ant**, nuevamente haga clic en el botón **Add JAR/ZIP...**
- En la ventana **Add JAR File**, busque el directorio del módulo **javax.jms.api** de WildFly (subdirectorio **javax/jms/api/main** del directorio de módulos de WildFly), seleccione el archivo **jboss-jms-api\_2.0\_spec-1.x.x.Final.jar** y haga clic en el botón **OK**.
- De regreso en la pestaña **Ant**, haga clic en el botón **OK**.





## Configuración de eclipse para proyectos generados con la plataforma jee1

Si desea construir (*build*) y ejecutar los proyectos generados con la plataforma **jee1** utilizando eclipse, entonces deberá instalar y agregar al menos un servidor de aplicaciones a la configuración de eclipse, preferiblemente antes de generar el proyecto.

A diferencia de NetBeans, la instalación de eclipse no incluye la de ningún servidor de aplicaciones. Debe instalar GlassFish o, alternativamente, WildFly. Consulte la documentación del servidor escogido para obtener información sobre su instalación. Después de instalar el servidor, debe agregarlo a la configuración de eclipse. Consulte la documentación de eclipse para obtener información sobre como agregar un servidor de aplicaciones.

Cuando se utiliza WildFly como servidor de aplicaciones, los proyectos generados con la plataforma **jee1** necesitan utilizar la configuración completa y, por lo tanto, al agregar el servidor a eclipse es necesario especificar el archivo de configuración **standalone-full.xml**.

## Generación de proyectos con la plataforma jee1

Para generar proyectos con la plataforma **jee1** a partir de meta-proyectos creados con la plantilla **jee1** siga los siguientes pasos:

- Especificar la versión de los productos requeridos instalados para generar el proyecto (Java, GlassFish, WildFly y PostgreSQL).

Abra el meta-proyecto **meta-xyz1** ubicado en el directorio **/opt/workspace/xyz1/source/meta**.

En la clase **meta.proyecto.Maestro**, las siguientes instrucciones:

```
maestro.putEnvironmentVariable(VERSION_JAVA, "1.8.0_77");  
maestro.putEnvironmentVariable(VERSION_GLASSFISH, "4.1.1");  
maestro.putEnvironmentVariable(VERSION_WILDFLY, "10.0.0.Final");  
maestro.putEnvironmentVariable(VERSION_POSTGRESQL, "9.5");
```

establecen la versión instalada de Java, GlassFish, WildFly y PostgreSQL, respectivamente.

No es necesario especificar tanto la versión de GlassFish como la de WildFly. Basta con especificar la del producto que corresponde a la variante de la plataforma **jee1** que va a utilizar para generar el proyecto. La variante se establece tal como se explica más adelante, en esta misma sección.

Si no se establece la versión de alguno de los productos, entonces se utilizará la versión mínima requerida del mismo. La sección [Requisitos para la instalación](#) documenta la versión mínima requerida de cada producto.

- Especificar el nombre del proyecto a generar.

En la clase **meta.proyecto.Maestro**, la siguiente instrucción:

```
maestro.setAlias("xyz1ap101");
```

establece el alias del proyecto **Maestro**, el cual se utilizará como nombre del proyecto a generar por la siguiente instrucción:



```
maestro.generate (PLATAFORMA_NETBEANS_POSTGRESQL_GLASSFISH) ;
```

Si lo desea, puede cambiar el alias para generar el proyecto con un nombre diferente. El alias solo puede contener letras minúsculas y números, debe comenzar por una letra, y no puede ser **jee1ap101**, **meta** o **workspace**. Se recomienda utilizar un alias que tenga el nombre de su proyecto como prefijo.

**Nota:** en lo sucesivo se da por hecho que el alias y, por lo tanto, el nombre del proyecto generado es **xyz1ap101**.

- Especificar la variante de la plataforma **jee1** a utilizar para generar el proyecto.

Si desea utilizar WildFly en lugar de GlassFish, entonces deberá cambiar la instrucción:

```
maestro.generate (PLATAFORMA_NETBEANS_POSTGRESQL_GLASSFISH) ;
```

por la instrucción:

```
maestro.generate (PLATAFORMA_NETBEANS_POSTGRESQL_WILDFLY) ;
```

Si desea utilizar eclipse en lugar de NetBeans, entonces deberá cambiarla por una de las siguientes instrucciones, dependiendo del servidor de aplicaciones que desea utilizar. Para GlassFish deberá cambiarla por la instrucción:

```
maestro.generate (PLATAFORMA_ECLIPSE_POSTGRESQL_GLASSFISH) ;
```

y para WildFly, por la instrucción:

```
maestro.generate (PLATAFORMA_ECLIPSE_POSTGRESQL_WILDFLY) ;
```

- Ejecutar el meta-proyecto **meta-xyz1**.

Esta acción genera el proyecto **xyz1ap101** en el directorio **/opt/workspace/xyz1ap101/source/xyz1ap101**

Consulte la documentación del IDE seleccionado (NetBeans o eclipse) para obtener información sobre cómo ejecutar un proyecto.

- Construir (build) el proyecto **xyz1ap101**.

Consulte la documentación del IDE seleccionado (NetBeans o eclipse) para obtener información sobre cómo construir un proyecto.

- **Antes de construir por primera vez el proyecto utilizando NetBeans**

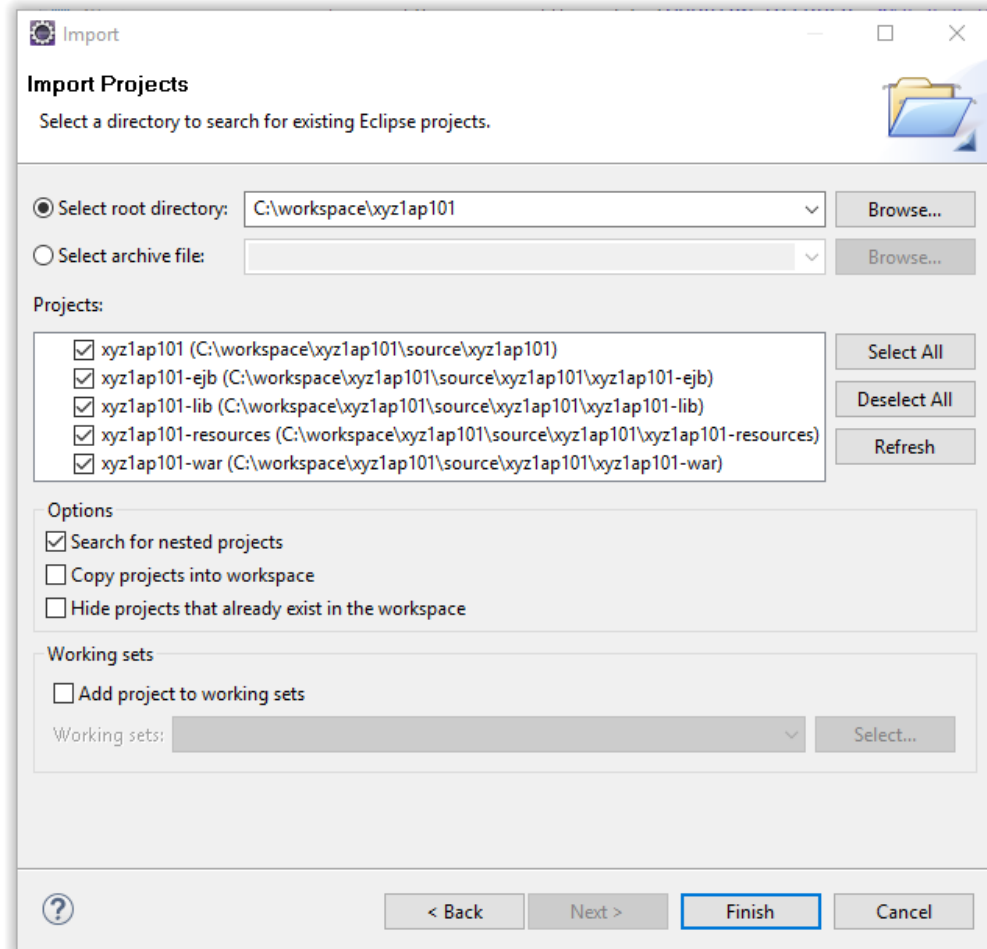
Abra el proyecto **xyz1ap101** (directorio **/opt/workspace/xyz1ap101/source/xyz1ap101**).

En la ventana **Projects**, haga clic-derecho al proyecto **xyz1ap101** seleccione la opción **Open Required Projects** y luego **Open All Projects**. Esta acción abre todos los subproyectos y crea archivos necesarios para la construcción.

Finalmente, si utiliza WildFly es necesario cambiar el script **Ant** para construir el proyecto **xyz1ap101-ejb**. En el directorio raíz de ese proyecto encontrará los archivos **build.xml** y **build-for-linux.xml**. Elimine el archivo **build.xml** y renombre **build-for-linux.xml** como **build.xml**.

- **Antes de construir por primera vez el proyecto utilizando eclipse**

Importe el proyecto **xyz1ap101** al *workspace*, siguiendo un procedimiento muy similar al descrito anteriormente, esta vez para importar el directorio **/opt/workspace/xyz1ap101/source/xyz1ap101**. Es necesario seleccionar la opción **Search for nested projects** para así importar también los subproyectos de **xyz1ap101**.



Finalmente, si utiliza WildFly es necesario agregar un nuevo Ant Builder al proyecto **xyz1ap101-ejb**. En el directorio raíz del proyecto **xyz1ap101-ejb** encontrará el archivo **weaving-for-linux.xml**. Al agregar el nuevo Ant Builder, use **weaving-for-linux.xml** como **Buildfile**. Asegúrese de que en los **Targets** del nuevo Ant Builder solo se especifique el *default target* para **Manual Build** (debe eliminar el *default target* seleccionado de **After a "Clean"**). También es necesario configurar el **Classpath** del nuevo Ant Builder; agregue el archivo **eclipselink-2.6.1.jar** del directorio **/opt/workspace/third-party/lib**, el archivo **hibernate-jpa-2.x-api-1.x.x.Final.jar** del módulo **javax.persistence.api** de WildFly (subdirectorio **javax/persistence/api/main** del directorio de módulos de WildFly) y el archivo **jboss-jms-api\_2.0\_spec-1.x.x.Final.jar** del módulo **javax.jms.api** de WildFly (subdirectorio **javax/jms/api/main** del directorio de módulos de WildFly) al **Classpath**. Consulte la documentación del eclipse para obtener información sobre cómo agregar un Ant Builder a un proyecto.



## Configuración de proyectos generados con la plataforma jee1

Para configurar los proyectos generados con la plataforma **jee1** siga los siguientes pasos:

- Ejecutar el script **/opt/workspace/xyz1ap101/home-setup.sh**.
- Revisar y, en caso de ser necesario, ajustar las variables de entorno que se establecen en los siguientes archivos del directorio **/opt/workspace/xyz1ap101/source/management**:

**variables-conf.sh**  
**variables-home.sh**  
**variables-postgresql.sh**

Si utiliza GlassFish:

**asadmin.password**  
**variables-glassfish.sh**

Si utiliza WildFly:

**variables-jboss.sh**

**Nota:** normalmente solo las variables definidas en el script **variables-home.sh** necesitan ser ajustadas a su ambiente de trabajo ya que los valores de las variables definidas en los demás archivos coinciden con los valores predeterminados de los correspondientes productos.

## Configuración de PostgreSQL para proyectos generados con jee1

Para configurar PostgreSQL para los proyectos generados con la plataforma **jee1** siga los siguientes pasos:

- Iniciar PostgreSQL.
- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/postgresql/createdb.sh**.
- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/postgresql/makedb.sh**.

## Configuración de GlassFish para proyectos generados con jee1

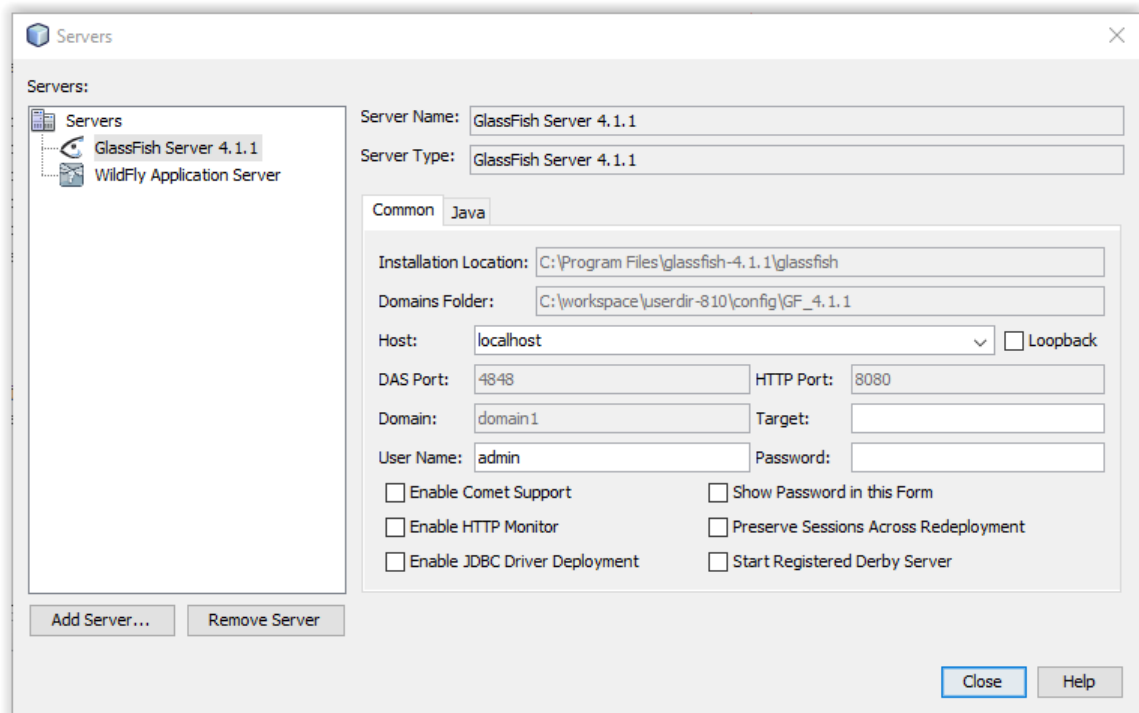
Para configurar GlassFish para los proyectos generados con la plataforma **jee1** siga los siguientes pasos:

- Detener GlassFish y hacer una copia de respaldo de los siguientes archivos:

**domain1/config/default-web.xml**  
**domain1/config/domain.xml**

**Nota:** utilizando NetBeans, puede conocer la ruta y el nombre del subdirectorio **domain1**. En el menú de barra de NetBeans, seleccione **Window**; en el menú desplegable **Window**, seleccione **Services**; en la ventana **Services**, expanda el nodo **Servers**, haga clic-derecho en el nodo **GlassFish Server 4.x.x** y seleccione **Properties**; en la pestaña **Common**, las propiedades

**Domains folder** y **Domain Name** muestran la ruta y el nombre del subdirectorio **domain1**, respectivamente.



- Copiar el archivo **log4j.properties** del directorio **/opt/workspace/xyz1ap101/source/management/setup/config/log4j** al directorio **domain1/config**.
- Iniciar GlassFish.
- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/glassfish/add-libraries.sh**.
- Detener y reiniciar GlassFish.
- Ejecutar los siguientes scripts, que se encuentran en el directorio **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/glassfish**:
  - **create-jdbc.sh**
  - **create-jms.sh**
  - **create-mail.sh**
  - **create-jvm-options.sh**
  - **create-jvm-options.es\_VE.sh**
  - **create-system-properties.sh**
- Detener y reiniciar GlassFish.

## Configuración de WildFly para proyectos generados con jee1

Para configurar WildFly para los proyectos generados con la plataforma **jee1** siga los siguientes pasos. La mayor parte de la configuración se lleva a cabo modificando al archivo **JBOSS\_HOME/standalone/configuration/standalone-full.xml**. El directorio **/opt/workspace/xyz1ap101/source/management/**



**setup/config/wildfly/standalone/configuration/10.0.0/standalone-full** y sus subdirectorios contienen archivos de donde puede copiar lo que se debe agregar al archivo **standalone-full.xml**. En cada paso se indica el nombre del archivo correspondiente.

- Detener WildFly y hacer una copia de respaldo de los siguientes archivos:

**JBOSS\_HOME/standalone/configuration/standalone-full.xml**  
**JBOSS\_HOME/bin/standalone.conf**

- Especificar las propiedades para el **logging** de la aplicación en la sección **<subsystem xmlns="urn:jboss:domain:logging:3.0">** del archivo **standalone-full.xml** (archivo 020-logging.xml).

```
<subsystem xmlns="urn:jboss:domain:logging:3.0">

    <logger category="xyz1ap101">
        <level name="DEBUG"/>
    </logger>

</subsystem>
```

- Especificar los recursos JDBC de la aplicación en la sección **<datasources>** del archivo **standalone-full.xml** (archivos 030-datasource.xml y 035-drivers.xml del subdirectorio postgresql).

```
<datasources>

    <datasource
        jta="true"
        jndi-name="java:/jdbc/xyz1ap101"
        pool-name="xyz1ap101-pool"
        enabled="true"
        use-java-context="true"
        spy="false"
        use-ccm="true">
        <connection-url>
            jdbc:postgresql://localhost:5432/xyz1ap101
        </connection-url>
        <driver>pgjdbc4</driver>
        <security>
            <user-name>postgres</user-name>
            <password>postgres</password>
        </security>
    </datasource>

    <drivers>

        <driver name="pgjdbc4" module="org.postgresql.pgjdbc4"/>

    </drivers>

</datasources>
```



- Especificar los recursos de JavaMail de la aplicación en la sección **<subsystem xmlns="urn:jboss:domain:mail:2.0">** del archivo **standalone-full.xml** (archivo 040-mail-session.xml).

```
<subsystem xmlns="urn:jboss:domain:mail:2.0">

    <mail-session
        name="xyz1ap101-mail-session"
        jndi-name="java:jboss/mail/xyz1ap101MailerBeanSession">
        <smtp-server
            ssl="true"
            outbound-socket-binding-ref="google-mail-smtp"
            username="xyz1ap101@gmail.com"
            password="xyz1ap101-password"/>
        </mail-session>

    </subsystem>
```

- Especificar los recursos de Google Mail de la aplicación en la sección **<socket-binding-group>** del archivo **standalone-full.xml** (archivo 045-outbound-socket-binding.xml).

```
<socket-binding-group>

    <outbound-socket-binding name="google-mail-smtp">
        <remote-destination host="smtp.gmail.com" port="465"/>
    </outbound-socket-binding>

</socket-binding-group>
```



- Especificar los recursos JMS de la aplicación en la sección **<server name="default">** de la sección **<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">** del archivo **standalone-full.xml** (archivo 050-messaging-activemq-server.xml).

```
<server name="default">

    <connection-factory
        name="xyz1ap101BusinessProcessMessageFactory"
        connectors="in-vm"
        entries="jms/xyz1ap101BusinessProcessMessageFactory"/>
    <connection-factory
        name="xyz1ap101UtilityProcessMessageFactory"
        connectors="in-vm"
        entries="jms/xyz1ap101UtilityProcessMessageFactory"/>
    <jms-queue
        name="xyz1ap101BusinessProcessRequestMessageQueue"
        entries="jms/xyz1ap101BusinessProcessRequestMessageQueue
java:jboss/exported/jms/xyz1ap101BusinessProcessRequestMessageQueue"/>
    <jms-queue
        name="xyz1ap101BusinessProcessReplyMessageQueue"
        entries="jms/xyz1ap101BusinessProcessReplyMessageQueue
java:jboss/exported/jms/xyz1ap101BusinessProcessReplyMessageQueue"/>
    <jms-queue
        name="xyz1ap101BusinessProcessReturnMessageQueue"
        entries="jms/xyz1ap101BusinessProcessReturnMessageQueue
java:jboss/exported/jms/xyz1ap101BusinessProcessReturnMessageQueue"/>
    <jms-queue
        name="xyz1ap101UtilityProcessRequestMessageQueue"
        entries="jms/xyz1ap101UtilityProcessRequestMessageQueue
java:jboss/exported/jms/xyz1ap101UtilityProcessRequestMessageQueue"/>
    <jms-queue
        name="xyz1ap101UtilityProcessReplyMessageQueue"
        entries="jms/xyz1ap101UtilityProcessReplyMessageQueue
java:jboss/exported/jms/xyz1ap101UtilityProcessReplyMessageQueue"/>
    <jms-queue
        name="xyz1ap101UtilityProcessReturnMessageQueue"
        entries="jms/xyz1ap101UtilityProcessReturnMessageQueue
java:jboss/exported/jms/xyz1ap101UtilityProcessReturnMessageQueue"/>

</server>
```

- Especificar el dominio de seguridad de la aplicación en la sección **<security-domains>** del archivo **standalone-full.xml** (archivo 060-security-domain.xml).

```
<security-domains>

    <security-domain name="xyz1ap101-policy">
        <authentication>
            <login-module code="Database" flag="required">
                <module-option
                    name="dsJndiName"
                    value="java:/jdbc/xyz1ap101"/>
                <module-option
                    name="hashAlgorithm"
                    value="MD5"/>
                <module-option
                    name="hashEncoding"
                    value="rfc2617"/>
                <module-option
                    name="principalsQuery"
                    value="select password_usuario from
vista_autenticacion_1 where codigo_usuario=?"/>
                <module-option
                    name="rolesQuery"
                    value="select codigo_rol, 'Roles' from
vista_autenticacion_3 where codigo_usuario=?"/>
            </login-module>
        </authentication>
    </security-domain>

</security-domains>
```

- Comentar o eliminar la línea **<default-security-domain value="other"/>** de la sección **<subsystem xmlns="urn:jboss:domain:ejb3:4.0">** del archivo **standalone-full.xml**.

- Ajustar los parámetros de la JVM en el archivo **standalone.conf**.

En la línea que establece la variable **JAVA\_OPTS**, cambiar **-Xms64m** por **-Xms128m**; en esa misma línea, cambiar **-Xmx512m** por **-Xmx1024m**.

- Si utiliza NetBeans, ajustar los parámetros de la JVM en las propiedades del servidor WildFly.

Cambiar **-Xms64M** por **-Xms128M** y **-Xmx512M** por **-Xmx1024M**.

- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/jboss/add-modules.sh**.
- Iniciar WildFly.
- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/jboss/create-system-properties.sh**.
- Detener y reiniciar WildFly.

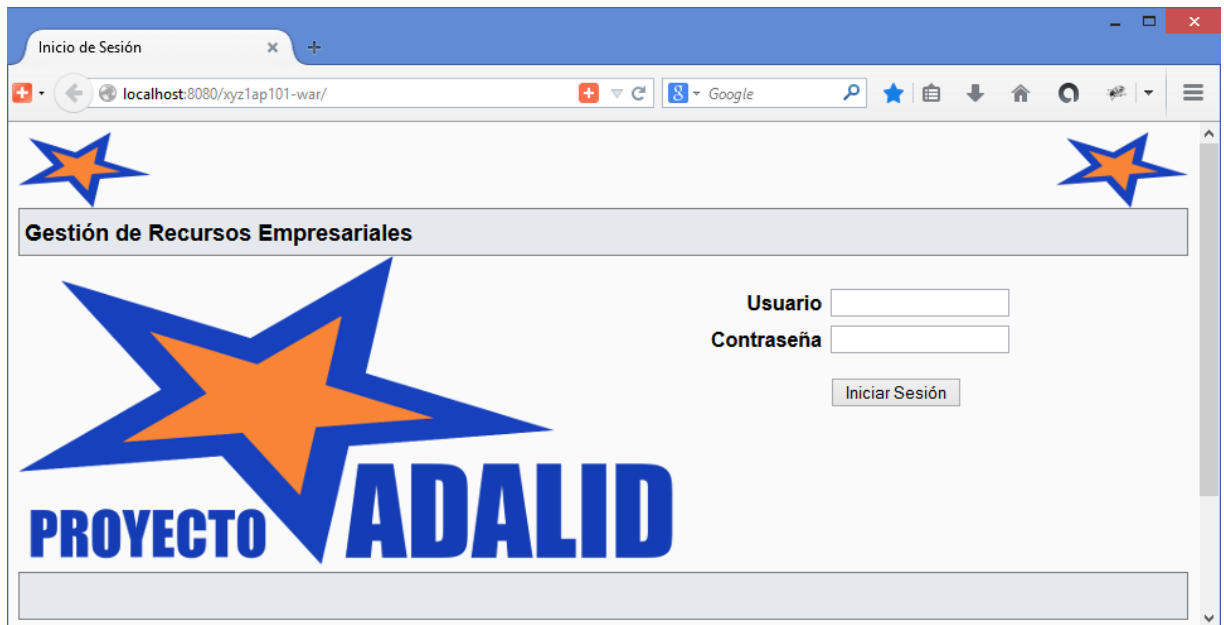


## Ejecución de proyectos generados con jee1

Para ejecutar los proyectos generados con la plataforma **jee1** siga los siguientes pasos:

- Ejecutar el proyecto **xyz1ap101** utilizando el IDE (NetBeans o eclipse).

El IDE debería iniciar el navegador web y abrir la página **Inicio de Sesión** de la aplicación. Si esto no sucede, inicie el navegador y abra la página **<http://localhost:8080/xyz1ap101-war/>**.



- Iniciar sesión en la aplicación

Utilice el usuario **admin** y la contraseña **sesamo**. Al comenzar la sesión de trabajo se abre la página **Menú Principal**.

