



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	6
Configuración de NetBeans para meta-proyectos creados con la plantilla jee1	6
Configuración de eclipse para meta-proyectos creados con la plantilla jee1	7
Configuración de NetBeans para proyectos generados con la plataforma jee1.....	11
Configuración de eclipse para proyectos generados con la plataforma jee1	13
Generación de proyectos con la plataforma jee1	13
Configuración de proyectos generados con la plataforma jee1	16
Configuración de PostgreSQL para proyectos generados con jee1	16
Configuración de GlassFish para proyectos generados con jee1	17
Configuración de JBoss para proyectos generados con jee1	19
Configuración de JBoss AS 7.1.1.Final	19
Configuración de WildFly 9.0.0.Final.....	24
Ejecución de proyectos generados con jee1.....	29



Archivos necesarios para la instalación de Adalid y jee1

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

- third-party.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.

- Java jdk1.7.0_25 o jdk1.8.0_25
URL: <http://www.oracle.com/technetwork/java/javase/downloads>
- NetBeans 7.3.1, opción Java EE (incluye GlassFish 4.0), o
NetBeans 8.0.2, opción Java EE (incluye GlassFish 4.1)
URL: <https://netbeans.org/downloads>
Alternativamente, eclipse Kepler
URL: <https://www.eclipse.org/downloads>
- GlassFish 4.0 o 4.1
URL: <https://glassfish.java.net/download.html>
Alternativamente, JBoss AS 7.1.1.Final o WildFly 9.0.0.Final
URL: <http://jbossas.jboss.org/downloads>
- PostgreSQL 9.2 o 9.3
URL: <http://www.enterprisedb.com/products-services-training/pgdownload>
- Mozilla Firefox 21.0
URL: <http://www.mozilla.org/es-ES/firefox/new>

Alternativamente, Microsoft Internet Explorer 11.0

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; en lo sucesivo se da por hecho que el directorio **workspace** se encuentra en **/opt**.

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

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

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**.

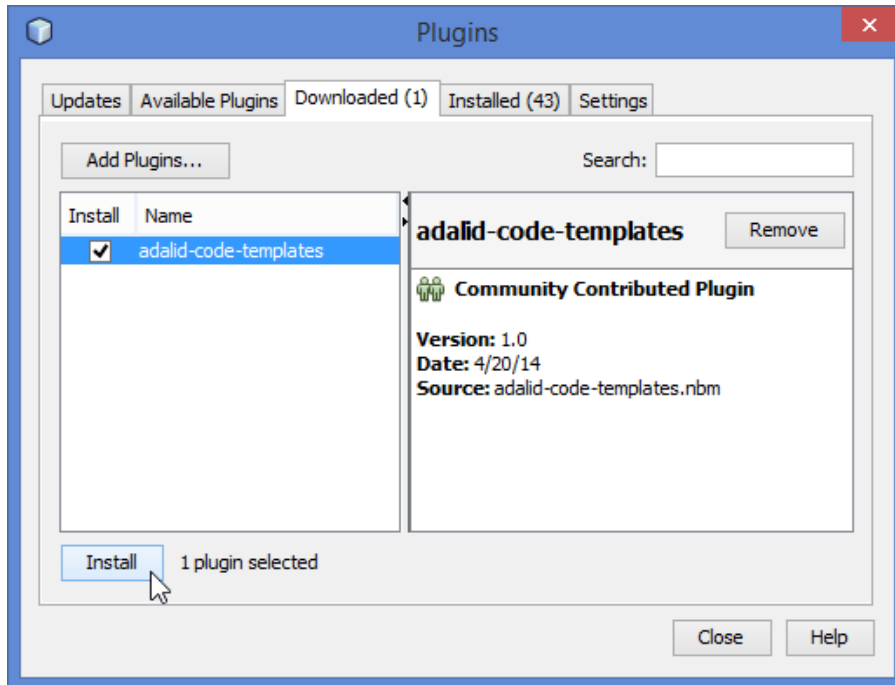
En el menú desplegable **Tools** seleccione **Plugins**.

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**.

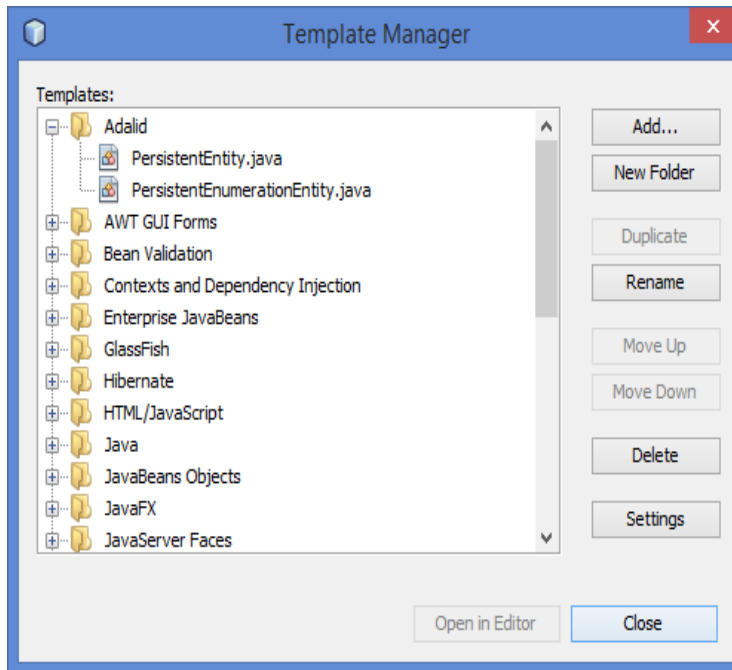
En la ventana **Rename template**, escriba **Adalid** en todos los campos y haga clic en el botón **OK**.

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**.

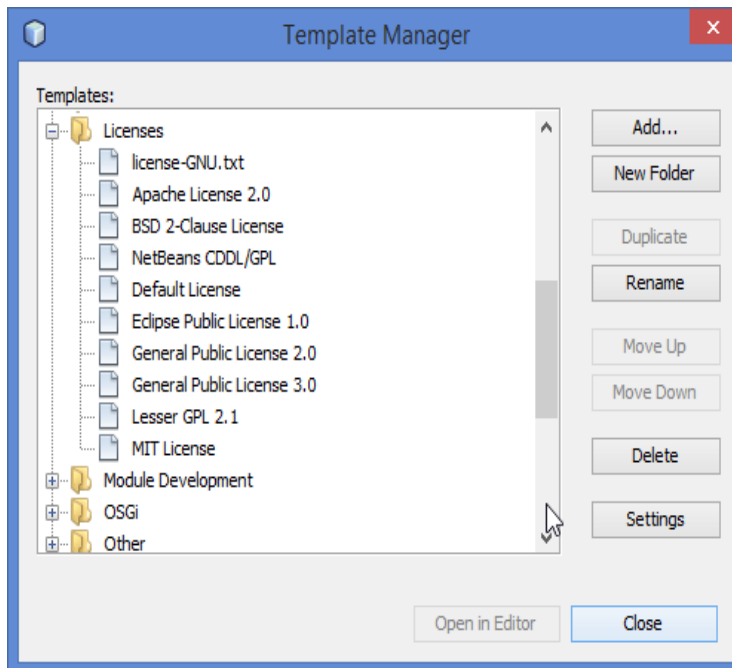
Repita los pasos anteriores para agregar el resto de las plantillas que se encuentran en **/home/workspace/adalid/source/development/resources/freemarker/file-templates/entities**.



- 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** 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 **/home**.

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:

- Copiar librerías Ant.

Copie todos 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/FaqWhatsUserdir>

- Reiniciar NetBeans.

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:

- Si desea utilizar JBoss AS 7.1.1 debe definir un JRE de Java SE 1.7.

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

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

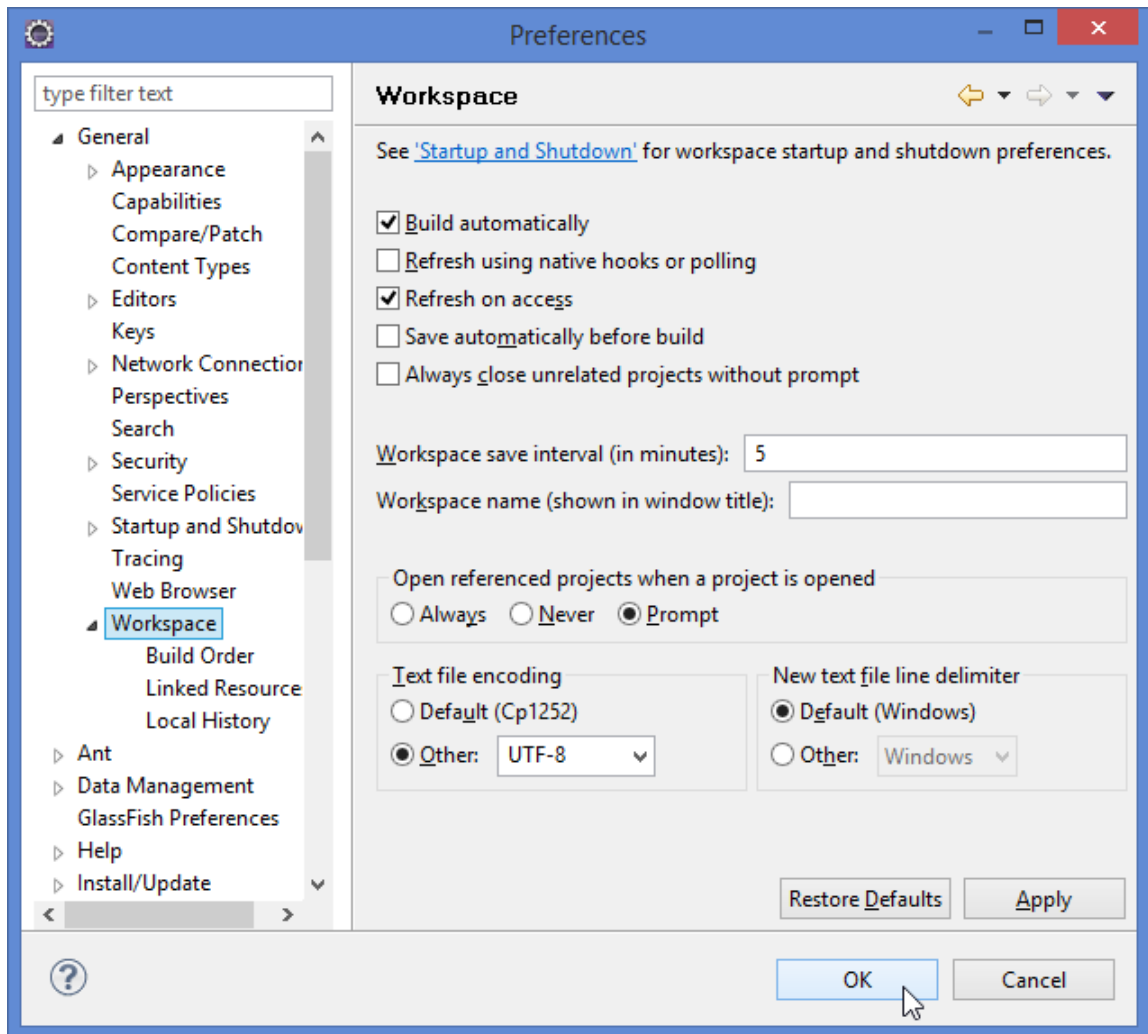
En la ventana **Preferences**, expanda el nodo **Java**; dentro del nodo **Java** seleccione el nodo **Installed JREs**; si un JRE de nombre **jre7** no aparece registrado, el haga clic en el botón **Add...** que se encuentra a la derecha de la lista **Installed JREs**. En la ventana **Add JRE** seleccione **Standard VM** y haga clic en el botón **Next >**. Especifique el directorio HOME del JRE en **JRE home** (búsquelo haciendo clic en el botón **Directory...**). Especifique **jre7** en **JRE name**. Finalmente haga clic en el botón **Finish**.

- Si desea utilizar WildFly 9.0.0 debe definir un JRE de Java SE 1.7 o 1.8.
- 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**.



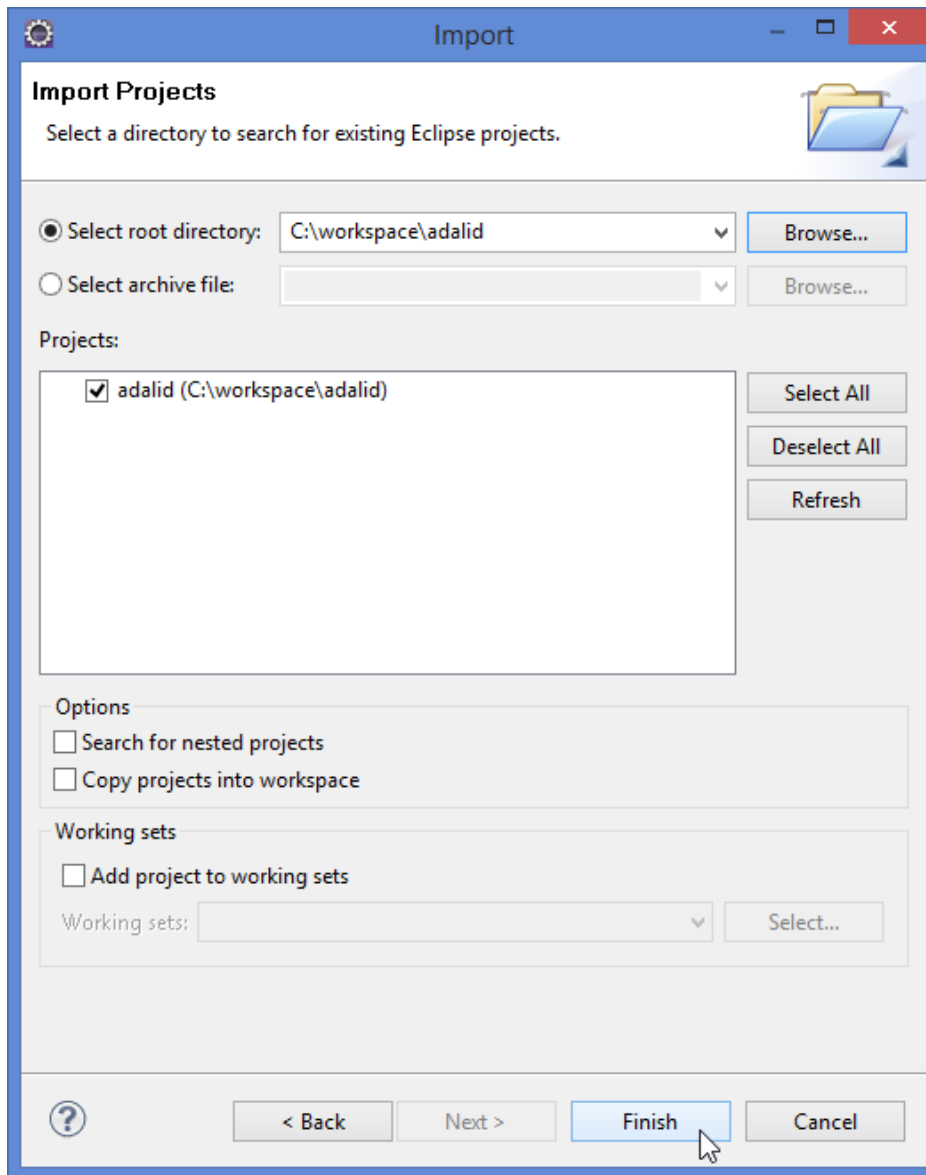
- Importar el directorio `/opt/workspace/adalid` al *workspace*.

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.

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

Si desea utilizar JBoss (JBoss AS o WildFly) como servidor de aplicaciones, repita el procedimiento anterior para importar el directorio HOME de JBoss.

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

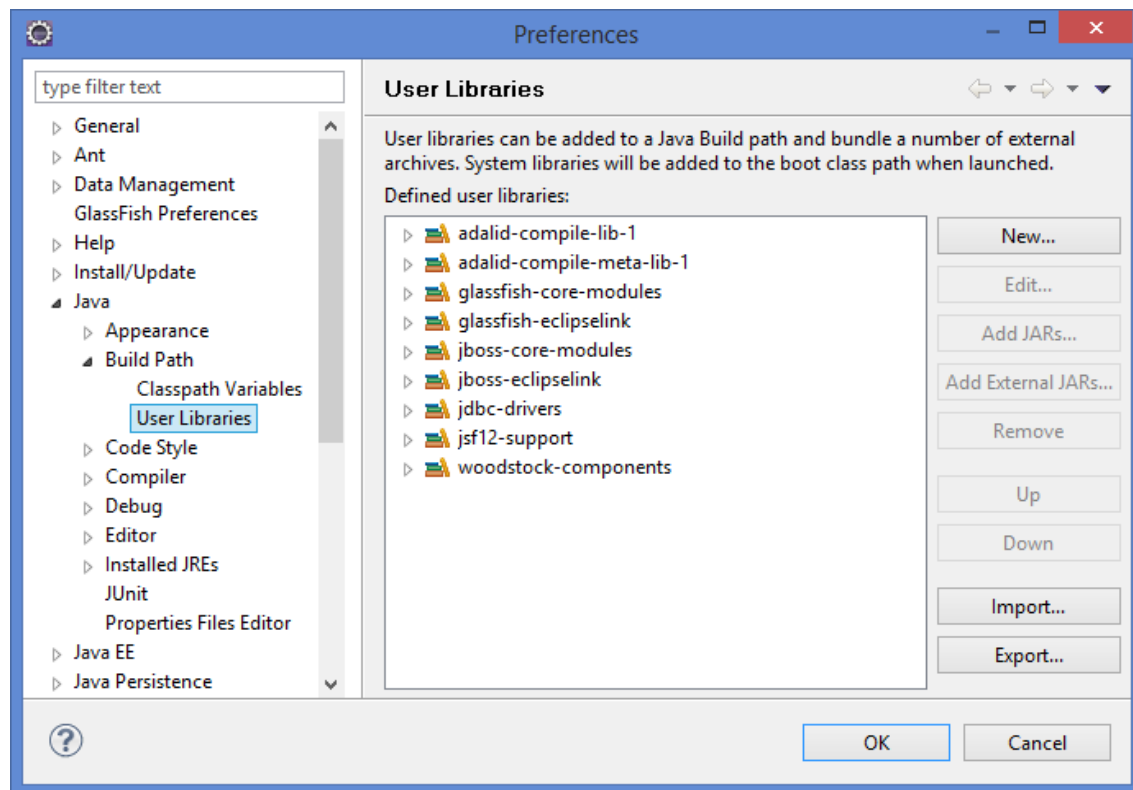
En el menú de barra de eclipse, seleccione **Window**.

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**. Importe los archivos con prefijo **glassfish-**, **jboss-** y **wildfly-** dependiendo del servidor de aplicaciones que desea utilizar. Debe importar todos los demás archivos.



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

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

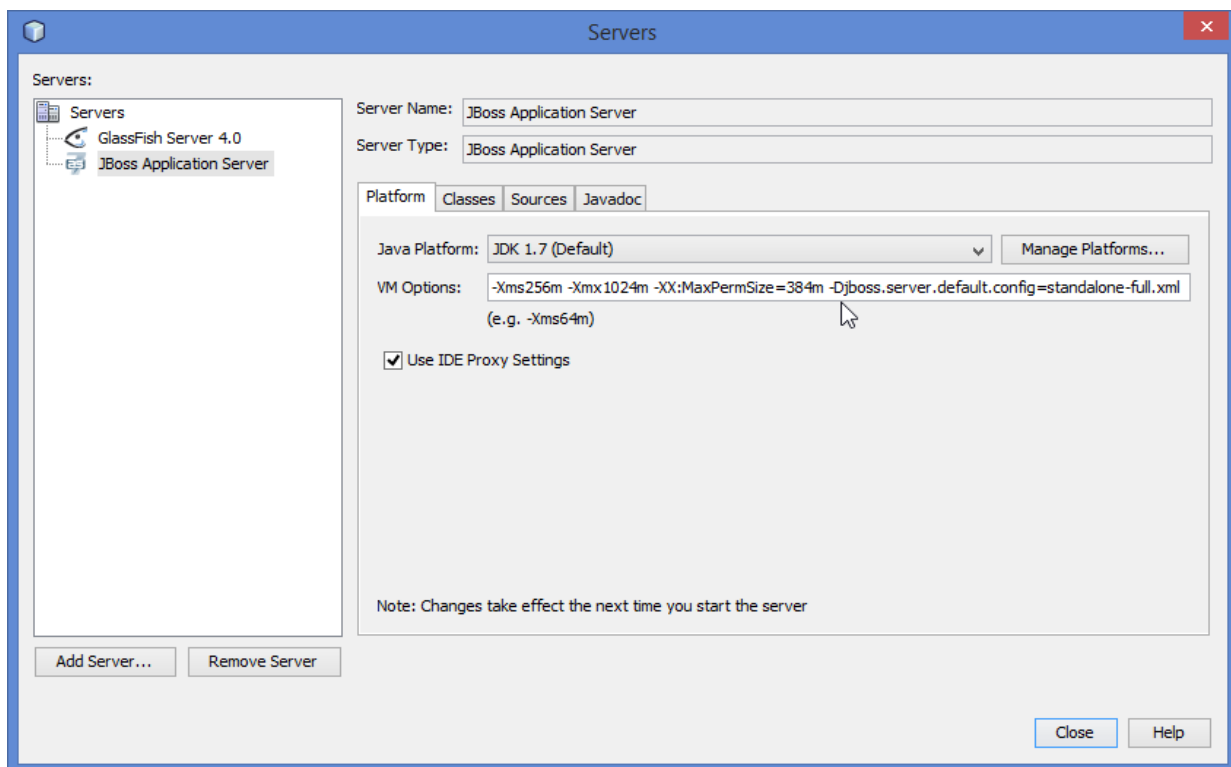
Configuración de NetBeans para proyectos generados con la plataforma jee1

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 JBoss (JBoss AS o WildFly) debe instalarlo por separado. Consulte la documentación de JBoss para obtener información sobre su instalación. Después de instalar JBoss, debe agregarlo a la configuración de NetBeans. Consulte la documentación de NetBeans para obtener información sobre como agregar un nuevo servidor de aplicaciones.

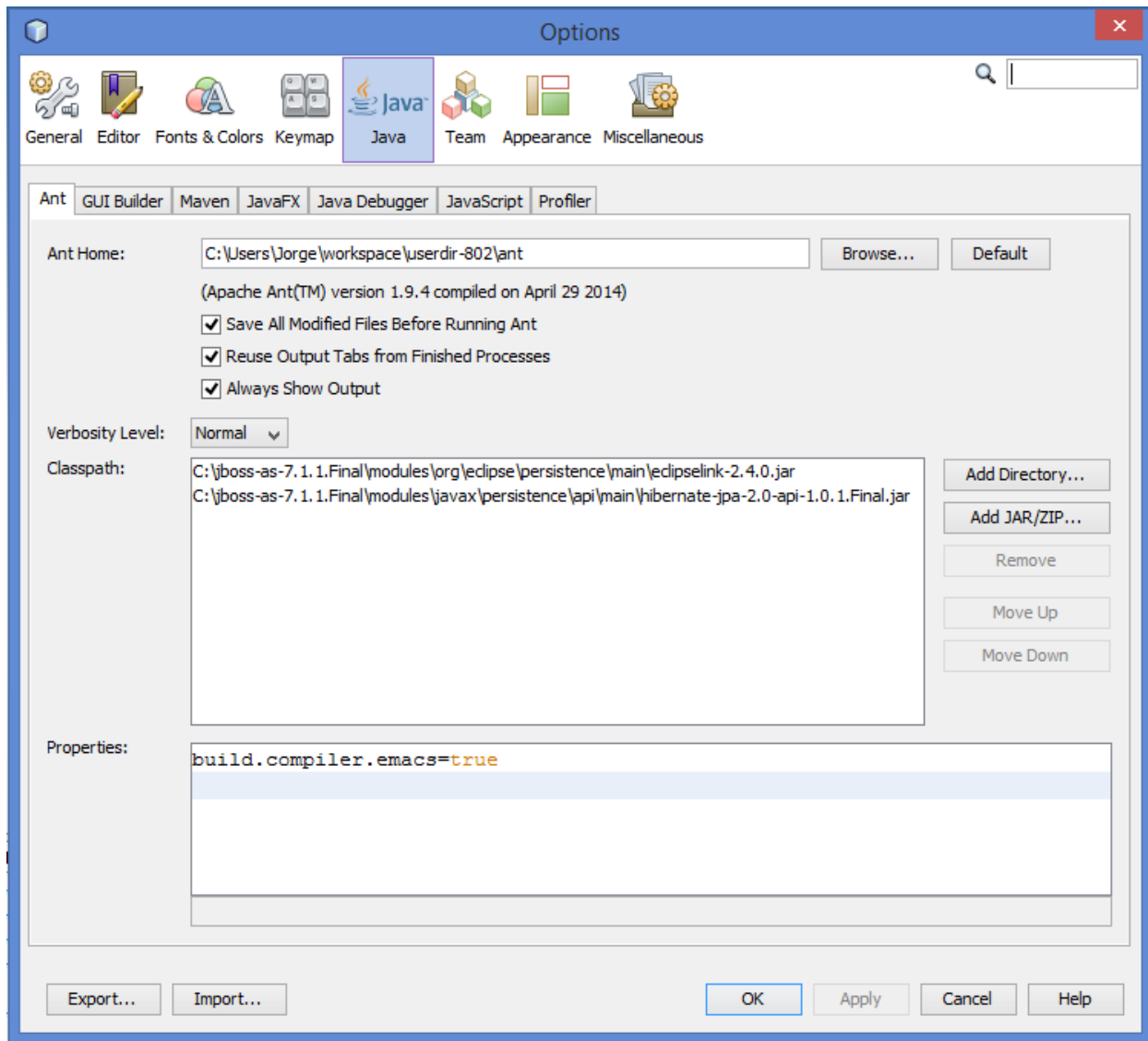
Cuando se utiliza JBoss como servidor de aplicaciones, los proyectos generados con la plataforma **jee1** necesitan utilizar la configuración completa y, por lo tanto, al agregar el servidor a NetBeans es necesario especificar el archivo de configuración **standalone-full.xml**.

Advertencia: la versión 7.3.1 de NetBeans permite agregar un servidor JBoss pero no permite especificar el archivo de configuración que se desea utilizar. Para hacerlo, debe modificar las propiedades del servidor agregado, incluyendo la opción **jboss.server.default.config** entre las opciones de la máquina virtual de Java, y especificar el valor **standalone-full.xml**, tal como se muestra en la siguiente imagen.



También es necesario configurar el **Classpath** de **Ant**, pero esto debe hacerse después de terminar la [Configuración de JBoss para proyectos generados con jee1](#). Para hacerlo, 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 del módulo **org.eclipse.persistence** de JBoss (subdirectorio **org/eclipse/persistence/main** del directorio de módulos de JBoss), seleccione el archivo **eclipselink-2.4.0.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 JBoss (subdirectorio **javax/persistence/api/main** del directorio de módulos de JBoss), seleccione el archivo **hibernate-jpa-2.x-api-1.0.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, JBoss (JBoss AS o 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 JBoss 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, JBoss y PostgreSQL).

Abra el meta-proyecto **meta** 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_25");
maestro.putEnvironmentVariable(VERSION_GLASSFISH, "4.1");
maestro.putEnvironmentVariable(VERSION_JBOSS, "9.0.0.Final");
maestro.putEnvironmentVariable(VERSION_POSTGRESQL, "9.3");
```

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

No es necesario especificar tanto la versión de GlassFish como la de JBoss. 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 las siguientes instrucciones:



```
maestro.generate(PLATAFORMA_BASE);  
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 JBoss en lugar de GlassFish, entonces deberá cambiar la instrucción:

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

por la instrucción:

```
maestro.generate(PLATAFORMA_NETBEANS_POSTGRESQL_JBOSS);
```

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 JBoss, por la instrucción:

```
maestro.generate(PLATAFORMA_ECLIPSE_POSTGRESQL_JBOSS);
```

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

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 **/opt/workspace/xyz1ap101/source/xyz1ap101**.

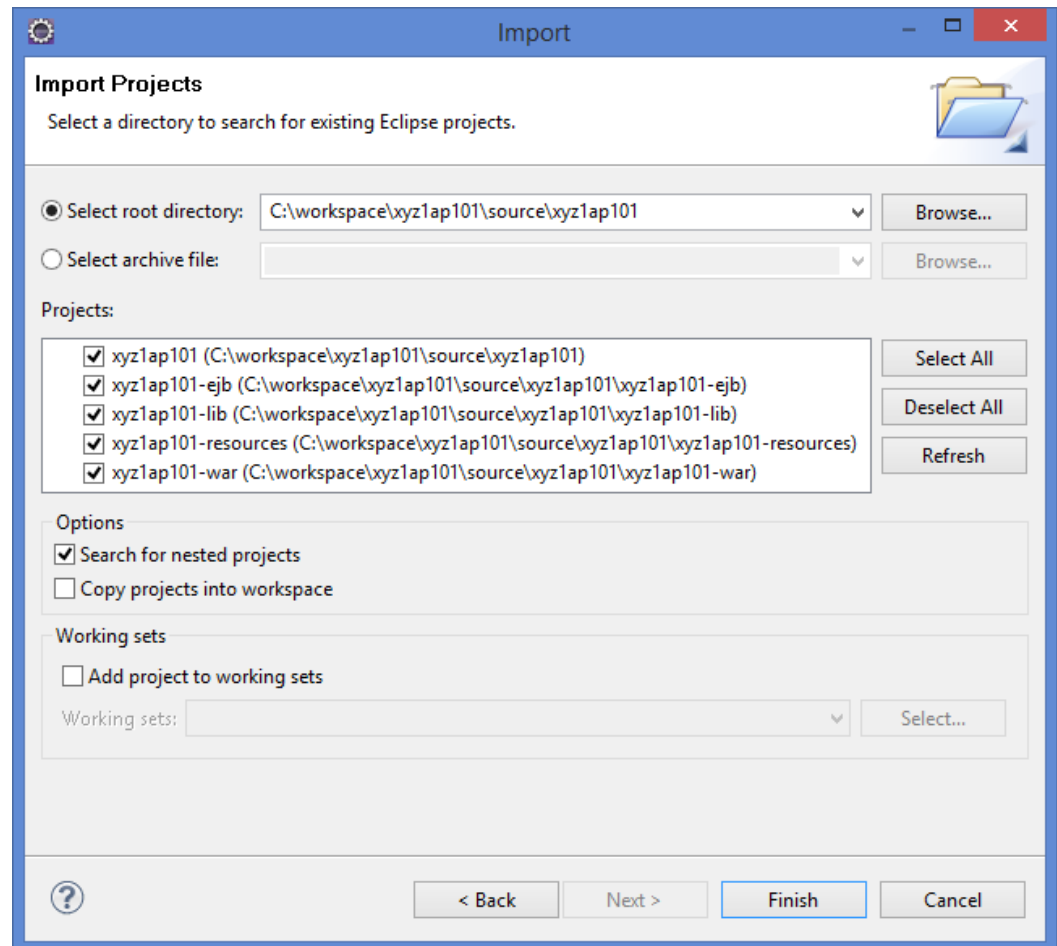
En la ventana **Projects**, haga clic-derecho al proyecto **xyz1ap101** y seleccione la opción **Open Required Projects**. Esta acción abre todos los subproyectos y crea los archivos necesarios para la construcción.

Finalmente, si utiliza JBoss es necesario cambiar el script **Ant** para construir el proyecto **xyz1ap101-ejb**. En el directorio raíz de ese proyecto encontrará los archivos **build.xml**, **build-for-linux.xml** y **build-for-windows.xml**. Revise el archivo **build-for-linux.xml**; compruebe que las rutas que aparecen en los **pathelement** del **classpath** de la tarea

weave se ajusten a su ambiente de trabajo. 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 **/opt/workspace/xyz1ap101/source/xyz1ap101** al *workspace*, siguiendo un procedimiento muy similar al descrito anteriormente, esta vez para importar el directorio **/opt/workspace/xyz1ap101/source/xyz1ap101**. En esta oportunidad es necesario seleccionar la opción **Search for nested projects** para así también importar los subproyectos de **xyz1ap101**.



Advertencia: actualmente las herramientas de GlassFish (GlassFish tools) para eclipse no incluyen una librería de *runtime*. Por esta razón, si utiliza GlassFish, después de importar el proyecto es necesario agregar explícitamente la librería externa **javaee.jar** de GlassFish a los módulos **ejb** y **web**, es decir, a los subproyectos **xyz1ap101-ejb** y **xyz1ap101-war**, respectivamente. Consulte la documentación de eclipse para obtener información sobre como agregar una librería externa. La librería **javaee.jar** debería encontrarse en el subdirectorio **lib** del directorio raíz de Glassfish (GLASSFISH_HOME). Por otra parte, las herramientas de JBoss (JBoss tools) si incluyen una librería de *runtime*. Por lo tanto, si utiliza JBoss, no es necesario agregar librerías externas adicionales a ninguno de los subproyectos.

Finalmente, si utiliza JBoss es necesario agregar un nuevo **Ant Builder** al proyecto **xyz1ap101-ejb**, pero esto debe hacerse después de terminar la [Configuración de JBoss](#)



[para proyectos generados con jee1](#). En el directorio raíz del proyecto **xyz1ap101-ejb** encontrará los archivos **weaving-for-linux.xml** y **weaving-for-windows.xml**. Revise el archivo **weaving-for-linux.xml**; compruebe que las rutas que aparecen en los **pathelement** del **classpath** de la tarea **weave** se ajusten a su ambiente de trabajo. 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 los **jar** de los módulos **org.eclipse.persistence** y **javax.persistence.api** de JBoss al Classpath (subdirectorios **org/eclipse/persistence/main** y **javax/persistence/api/main** del directorio de módulos de JBoss, respectivamente). 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 JBoss:

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**.

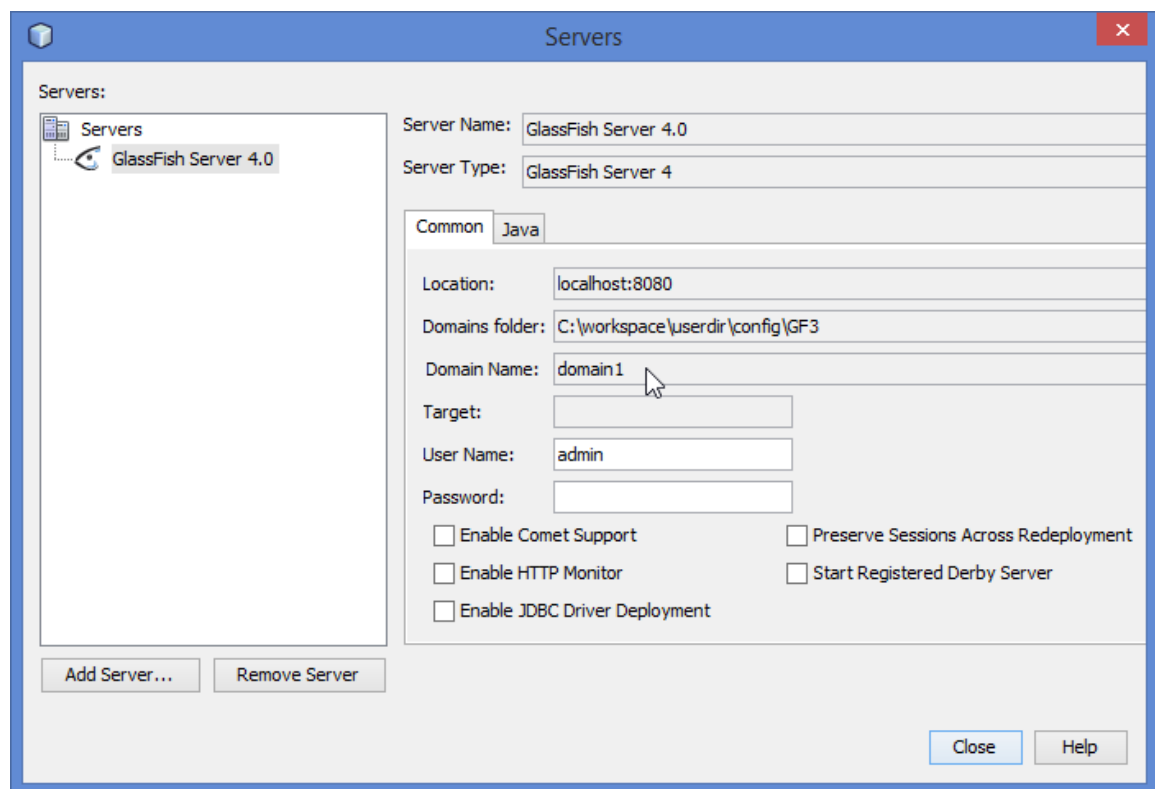
Configuración de GlassFish para proyectos generados con jee1

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 4.0** y seleccione **Properties**; en la pestaña **Common**, las propiedades **Domains folder** y **Domain Name** muestran la ruta y el nombre del subdirectorio **domain1**, respectivamente.



- En el archivo **domain1/config/default-web.xml**, agregar:

En la sección **Default MIME Type Mappings**, entre las extensiones **css** y **dib**:

```
<mime-mapping>
  <extension>csv</extension>
  <mime-type>application/vnd.ms-excel</mime-type>
</mime-mapping>
```

En la sección **Default MIME Type Mappings**, entre las extensiones **latex** y **m3u**:

```
<mime-mapping>
```

```
<extension>log</extension>
<mime-type>text/plain</mime-type>
</mime-mapping>
```

- En el archivo **domain1/config/default-web.xml**, establecer la propiedad **listings** del **servlet** predeterminado (default) como **false**, como se muestra a continuación:

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.DefaultServlet
  </servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

- En el archivo **domain1/config/domain.xml**, en la sección **<config name="server-config">**, establecer la propiedad **sso-enabled** del **http-service** como **true**, como se muestra a continuación:

```
<http-service sso-enabled="true">
```

- En el archivo **domain1/config/domain.xml**, en la sección **<config name="server-config">**, subsección **<java-config>**, ajustar los parámetros de la JVM, añadiendo y/o modificando las opciones **Xms**, **Xmx** y **XX:MaxPermSize**, como se muestra a continuación:

```
<jvm-options>-Xms128m</jvm-options>
<jvm-options>-Xmx1024m</jvm-options>
<jvm-options>-XX:MaxPermSize=384m</jvm-options>
```

- 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-jvm-options.sh**
 - **create-jvm-options.es_VE.sh**
 - **create-system-properties.sh**



- Detener y reiniciar GlassFish.

Configuración de JBoss para proyectos generados con jee1

Configuración de JBoss AS 7.1.1.Final

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

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

JBOSS_HOME/standalone/configuration/standalone-full.xml

JBOSS_HOME/bin/standalone.sh

- Agregar propiedades del sistema en la sección **<system-properties>** del archivo **standalone-full.xml**.

```
<system-properties>

  <property name="xyz1ap101.home.dir"
    value="/opt/workspace/xyz1ap101/source/management"/>
  <property name="xyz1ap101.jdbc.url"
    value="jdbc:postgresql://localhost:5432/xyz1ap101"/>
  <property name="xyz1ap101.jdbc.driver"
    value="org.postgresql.Driver"/>
  <property name="xyz1ap101.jdbc.user"
    value="postgres"/>
  <property name="xyz1ap101.jdbc.password"
    value="postgres"/>
  <property name="xyz1ap101.jndi.ejb.lookup.pattern"
    value="java:global/xyz1ap101/xyz1ap101-ejb/{0}"/>
  <property name="xyz1ap101.project.stage"
    value="Development"/>
  <property name="xyz1ap101.project.mailing"
    value="enabled"/>
  <property name="org.apache.el.parser.COERCE_TO_ZERO"
    value="false"/>
  <property name="org.apache.catalina.connector.URI_ENCODING"
    value="UTF-8"/>
  <property
name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING"
    value="true"/>
  <property name="eclipselink.deploy-on-startup"
    value="true"/>
  <property name="eclipselink.id-validation"
    value="NULL"/>
  <property name="eclipselink.target-server"
    value="JBoss"/>
  <property name="eclipselink.archive.factory"
value="id.au.ringerc.as7.eclipselinkintegration.JBossArchiveFactoryImpl"
/>
  <property name="eclipselink.logging.logger"
    value="id.au.ringerc.as7.eclipselinkintegration.JBossLogger"/>

</system-properties>
```

- **Nota:** la sección **<system-properties>** se debe colocar inmediatamente después de la sección **<extensions>**.
- Especificar las propiedades para el **logging** de la aplicación en la sección **<subsystem xmlns="urn:jboss:domain:logging:1.1">** del archivo **standalone-full.xml**.

```
<subsystem xmlns="urn:jboss:domain:logging:1.1">

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

</subsystem>
```

- Especificar los recursos de JavaMail de la aplicación en las secciones **<subsystem xmlns="urn:jboss:domain:mail:1.0">** y **<socket-binding-group>** del archivo **standalone-full.xml**.

```
<subsystem xmlns="urn:jboss:domain:mail:1.0">

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

</subsystem>

<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 JDBC de la aplicación en la sección **<datasources>** del archivo **standalone-full.xml**.

```
<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 JMS de la aplicación en las secciones **<jms-connection-factories>** y **<jms-destinations>** del archivo **standalone-full.xml**.

```
<jms-connection-factories>

    <connection-factory name="xyz1ap101BusinessProcessMessageFactory">
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="jms/xyz1ap101BusinessProcessMessageFactory"/>
        </entries>
    </connection-factory>

    <connection-factory name="xyz1ap101UtilityProcessMessageFactory">
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="jms/xyz1ap101UtilityProcessMessageFactory"/>
        </entries>
    </connection-factory>

</jms-connection-factories>
```

```
<jms-destinations>

  <jms-queue name="xyz1ap101BusinessProcessRequestMessageQueue">
    <entry name="jms/xyz1ap101BusinessProcessRequestMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessRequestMessageQueue"/>
  </jms-queue>
  <jms-queue name="xyz1ap101BusinessProcessReplyMessageQueue">
    <entry name="jms/xyz1ap101BusinessProcessReplyMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessReplyMessageQueue"/>
  </jms-queue>
  <jms-queue name="xyz1ap101BusinessProcessReturnMessageQueue">
    <entry name="jms/xyz1ap101BusinessProcessReturnMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessReturnMessageQueue"/>
  </jms-queue>
  <jms-queue name="xyz1ap101UtilityProcessRequestMessageQueue">
    <entry name="jms/xyz1ap101UtilityProcessRequestMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessRequestMessageQueue"/>
  </jms-queue>
  <jms-queue name="xyz1ap101UtilityProcessReplyMessageQueue">
    <entry name="jms/xyz1ap101UtilityProcessReplyMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessReplyMessageQueue"/>
  </jms-queue>
  <jms-queue name="xyz1ap101UtilityProcessReturnMessageQueue">
    <entry name="jms/xyz1ap101UtilityProcessReturnMessageQueue"/>
    <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessReturnMessageQueue"/>
  </jms-queue>

</jms-destinations>
```



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

```
<security-domains>

    <security-domain name="xyz1ap101-policy">
        <authentication>
            <login-module code="Database" flag="required">
                <module-option name="multi-threaded" value="true"/>
                <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>
```

- Agregar **<sso reauthenticate="false"/>** a la definición del virtual-server **default-host** en el archivo **standalone-full.xml**.

```
<subsystem xmlns="urn:jboss:domain:web:1.1"
    default-virtual-server="default-host" native="false">
    <connector name="http" protocol="HTTP/1.1" scheme="http"
        socket-binding="http"/>
    <virtual-server name="default-host" enable-welcome-root="true">
        <alias name="localhost"/>
        <alias name="example.com"/>
        <sso reauthenticate="false"/>
    </virtual-server>
</subsystem>
```

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

Cambiar la línea `set "JAVA_OPTS=-Xms64M -Xmx512M -XX:MaxPermSize=256M"`
por `set "JAVA_OPTS=-Xms128M -Xmx1024M -XX:MaxPermSize=384M"`

- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/jboss/add-modules.sh**.
- Reiniciar JBoss AS.

Nota: el directorio **/opt/workspace/xyz1ap101/source/management/setup/config/jboss/standalone/configuration/7.1.1** contiene un archivo **standalone-full.xml** de donde puede copiar las secciones mencionadas anteriormente.



Configuración de WildFly 9.0.0.Final

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

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

JBOSS_HOME/standalone/configuration/standalone-full.xml

JBOSS_HOME/bin/standalone.sh

- Agregar propiedades del sistema en la sección **<system-properties>** del archivo **standalone-full.xml**.

```
<system-properties>

  <property name="xyz1ap101.home.dir"
    value="/opt/workspace/xyz1ap101/source/management"/>
  <property name="xyz1ap101.jdbc.url"
    value="jdbc:postgresql://localhost:5432/xyz1ap101"/>
  <property name="xyz1ap101.jdbc.driver"
    value="org.postgresql.Driver"/>
  <property name="xyz1ap101.jdbc.user"
    value="postgres"/>
  <property name="xyz1ap101.jdbc.password"
    value="postgres"/>
  <property name="xyz1ap101.jndi.ejb.lookup.pattern"
    value="java:global/xyz1ap101/xyz1ap101-ejb/{0}"/>
  <property name="xyz1ap101.project.stage"
    value="Development"/>
  <property name="xyz1ap101.project.mailing"
    value="enabled"/>
  <property name="org.apache.el.parser.COERCE_TO_ZERO"
    value="false"/>
  <property name="org.apache.catalina.connector.URI_ENCODING"
    value="UTF-8"/>
  <property
name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING"
    value="true"/>
  <property name="eclipselink.deploy-on-startup"
    value="true"/>
  <property name="eclipselink.id-validation"
    value="NULL"/>
  <property name="eclipselink.target-server"
    value="JBoss"/>
  <property name="eclipselink.archive.factory"
value="org.jipijapa.eclipselink.JBossArchiveFactoryImpl"/>
  <property name="eclipselink.logging.logger"
    value="org.jipijapa.eclipselink.JBossLogger"/>

</system-properties>
```

- **Nota:** la sección **<system-properties>** se debe colocar inmediatamente después de la sección **<extensions>**.

- 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**.

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

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

</subsystem>
```

- Especificar los recursos de JavaMail de la aplicación en las secciones **<subsystem xmlns="urn:jboss:domain:mail:2.0">** y **<socket-binding-group>** del archivo **standalone-full.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>

<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 JDBC de la aplicación en la sección **<datasources>** del archivo **standalone-full.xml**.

```
<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 JMS de la aplicación en las secciones **<jms-connection-factories>** y **<jms-destinations>** del archivo **standalone-full.xml**.

```
<jms-connection-factories>

    <connection-factory name="xyz1ap101BusinessProcessMessageFactory">
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="jms/xyz1ap101BusinessProcessMessageFactory"/>
        </entries>
    </connection-factory>

    <connection-factory name="xyz1ap101UtilityProcessMessageFactory">
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="jms/xyz1ap101UtilityProcessMessageFactory"/>
        </entries>
    </connection-factory>

</jms-connection-factories>
```

```
<jms-destinations>

    <jms-queue name="xyz1ap101BusinessProcessRequestMessageQueue">
        <entry name="jms/xyz1ap101BusinessProcessRequestMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessRequestMessageQueue"/>
    </jms-queue>
    <jms-queue name="xyz1ap101BusinessProcessReplyMessageQueue">
        <entry name="jms/xyz1ap101BusinessProcessReplyMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessReplyMessageQueue"/>
    </jms-queue>
    <jms-queue name="xyz1ap101BusinessProcessReturnMessageQueue">
        <entry name="jms/xyz1ap101BusinessProcessReturnMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101BusinessProcessReturnMessageQueue"/>
    </jms-queue>
    <jms-queue name="xyz1ap101UtilityProcessRequestMessageQueue">
        <entry name="jms/xyz1ap101UtilityProcessRequestMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessRequestMessageQueue"/>
    </jms-queue>
    <jms-queue name="xyz1ap101UtilityProcessReplyMessageQueue">
        <entry name="jms/xyz1ap101UtilityProcessReplyMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessReplyMessageQueue"/>
    </jms-queue>
    <jms-queue name="xyz1ap101UtilityProcessReturnMessageQueue">
        <entry name="jms/xyz1ap101UtilityProcessReturnMessageQueue"/>
        <entry
name="java:jboss/exported/jms/xyz1ap101UtilityProcessReturnMessageQueue"/>
    </jms-queue>

</jms-destinations>
```



- Especificar el dominio de seguridad de la aplicación en la sección **<security-domains>** del archivo **standalone-full.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>
```

- Eliminar **<default-security-domain value="other"/>** de la sección **<subsystem xmlns="urn:jboss:domain:ejb3:3.0">** del archivo **standalone-full.xml**.
- Ajustar los parámetros de la JVM en el archivo **standalone.sh**.

Cambiar la línea `set "JAVA_OPTS=-Xms64M -Xmx512M -XX:MaxPermSize=256M"`
por `set "JAVA_OPTS=-Xms128M -Xmx1024M -XX:MaxPermSize=384M"`

- Ejecutar el script **/opt/workspace/xyz1ap101/source/management/resources/scripts/linux/jboss/add-modules.sh**.
- Reiniciar WildFly.

Nota: el directorio **/opt/workspace/xyz1ap101/source/management/setup/config/wildfly/standalone/configuration/9.0.0** contiene un archivo **standalone-full.xml** de donde puede copiar las secciones mencionadas anteriormente.

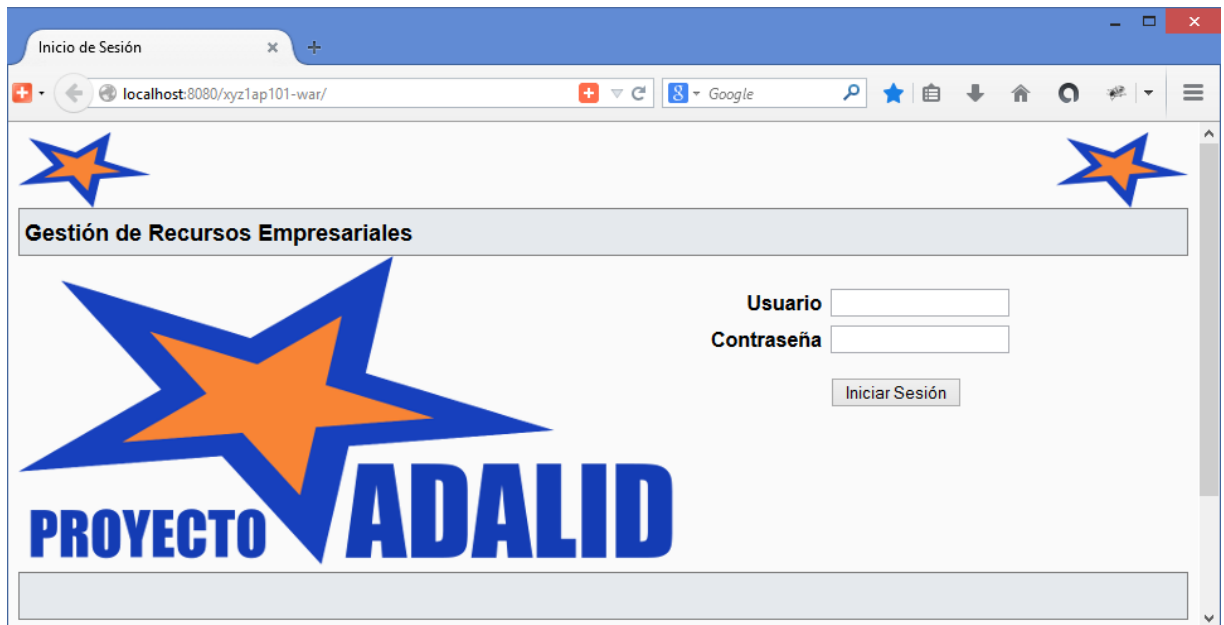


Ejecución de proyectos generados con jee1

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**.

