



Proyecto Adalid, Plataforma jee2

Guía de Instalación y Configuración en ambientes Linux

Tabla de Contenido

Archivos necesarios para la instalación de Adalid y jee2	2
Requisitos para la instalación	2
Instalación de librerías de terceras partes	2
Instalación de Adalid	3
Instalación de complementos para NetBeans	3
Creación de meta-proyectos con meta-jee2-archetype	5
Extensión de meta-proyectos creados con meta-jee2-archetype	8
Configuración de NetBeans para proyectos generados con jee2	9
Configuración de eclipse para proyectos generados con la plataforma jee2	9
Generación de proyectos con la plataforma jee2	10
Configuración de proyectos generados con la plataforma jee2	11
Configuración de PostgreSQL para proyectos generados con jee2	12
Configuración de Oracle para proyectos generados con jee2	12
Configuración de GlassFish para proyectos generados con jee2	12
Configuración de WildFly para proyectos generados con jee2	13
Ejecución de proyectos generados con jee2	19



Archivos necesarios para la instalación de Adalid y jee2

Para instalar **Adalid** y su plataforma **jee2** son necesarios los siguientes archivos:

- adalid-2.0.zip
- third-party-2.0.1.zip

Estos archivos se encuentran en el directorio raíz del disco de instalación.

Requisitos para la instalación

Los siguientes productos son necesarios para el funcionamiento de Adalid y/o de los meta-proyectos creados con el arquetipo **meta-jee2-archetype** y/o de los proyectos (aplicaciones) generados con la plataforma **jee2**. Las versiones corresponden a las versiones mínimas requeridas. Teóricamente debería funcionar con versiones superiores.

- Red Hat Linux, Ubuntu o similar. Debe tener instalado el utilitario **dos2unix**.
- Java jdk1.8.0_112 o superior.
URL: <http://www.oracle.com/technetwork/java/javase/downloads>
- NetBeans 8.2, opción Java EE (incluye GlassFish 4.1.1) o superior.
URL: <https://netbeans.org/downloads>
Alternativamente, eclipse Mars o superior.
URL: <https://www.eclipse.org/downloads>
- GlassFish 4.1.1 o superior
URL: <https://glassfish.java.net/download.html>
Alternativamente, WildFly 10.1.0.Final o 12.0.0.Final.
URL: <http://wildfly.org/downloads/>
- PostgreSQL 9.6 o superior.
URL: <http://www.enterprisedb.com/products-services-training/pgdownload>
- URL: <http://wildfly.org/downloads/>
- Oracle Express 11.2.0 o superior.
URL: <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>
- Mozilla Firefox 54.0 o superior.
URL: <http://www.mozilla.org/es-ES/firefox/new>
Alternativamente, Google Chrome 59.0 o superior
URL: <https://www.google.com/chrome>
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:

- Descomprimir el archivo **third party-2.0.1.zip** en **/opt**.

Esta acción crea el directorio **/opt/third-party-2.0**.



Instalación de Adalid

Para instalar Adalid 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 **\$HOME**.

- Descomprimir el archivo **adalid-2.0.zip** en **\$HOME/workspace**.

Esta acción crea el directorio **\$HOME/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**.

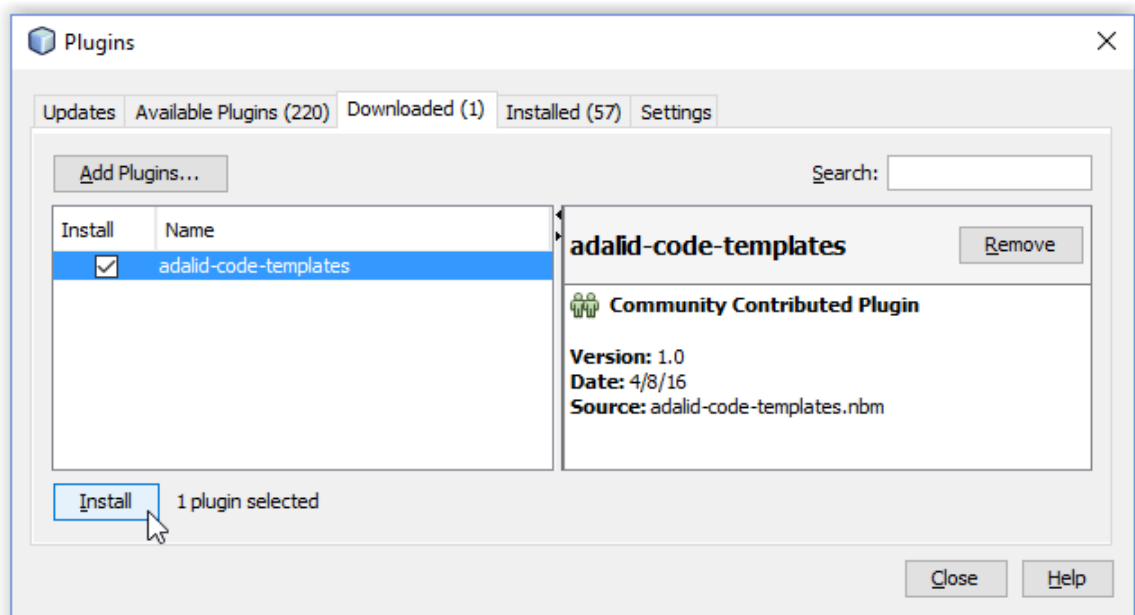
En el menú desplegable **Tools** seleccione **Plugins**.

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 **\$HOME/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**.

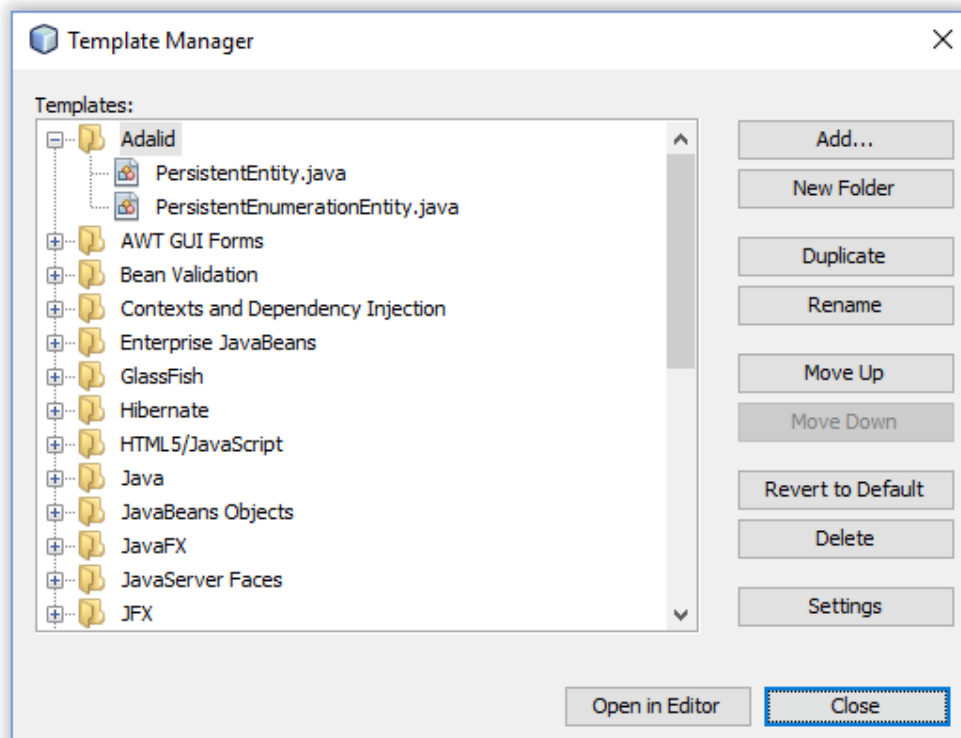
En la ventana **Rename template**, escriba **Adalid** en todos los campos y haga clic en el botón **OK**.

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 **\$HOME/workspace/adalid/source/development/resources/freemarker/file-templates/entities**, seleccione uno cualquiera de los archivos y haga clic en el botón **Add**.

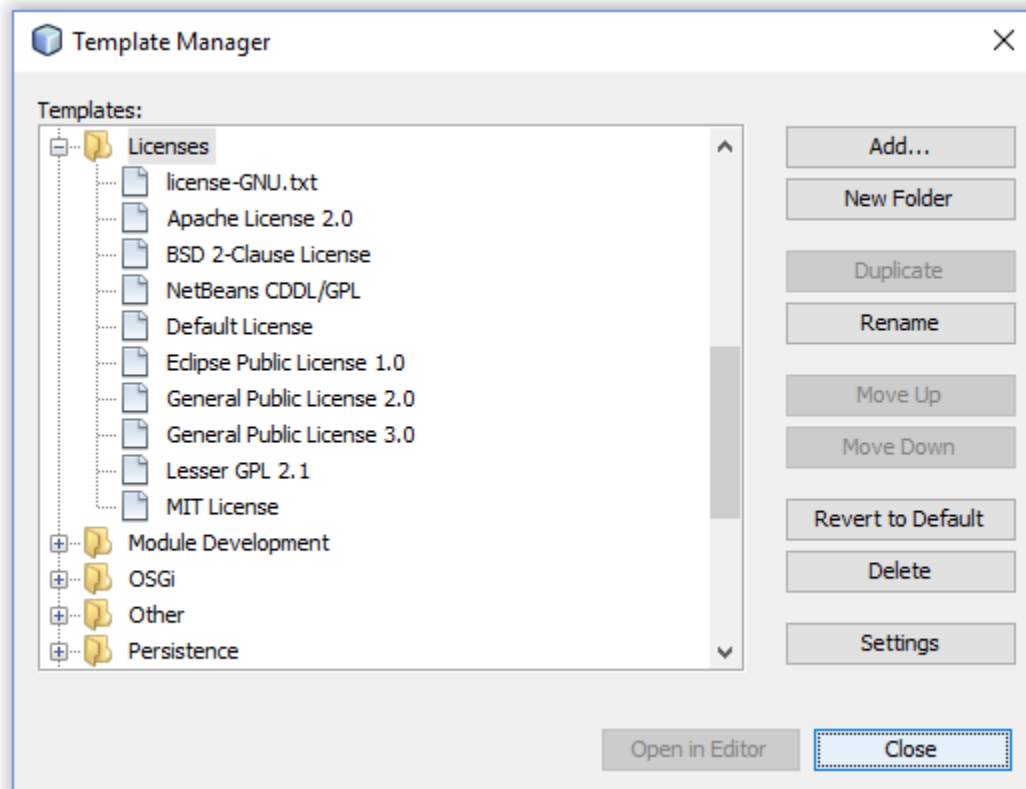
Repita los pasos anteriores para agregar el resto de las plantillas que se encuentran en **\$HOME/workspace/adalid/source/development/resources/freemarker/file-templates/entities**.



- 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 **\$HOME/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 meta-jee2-archetype

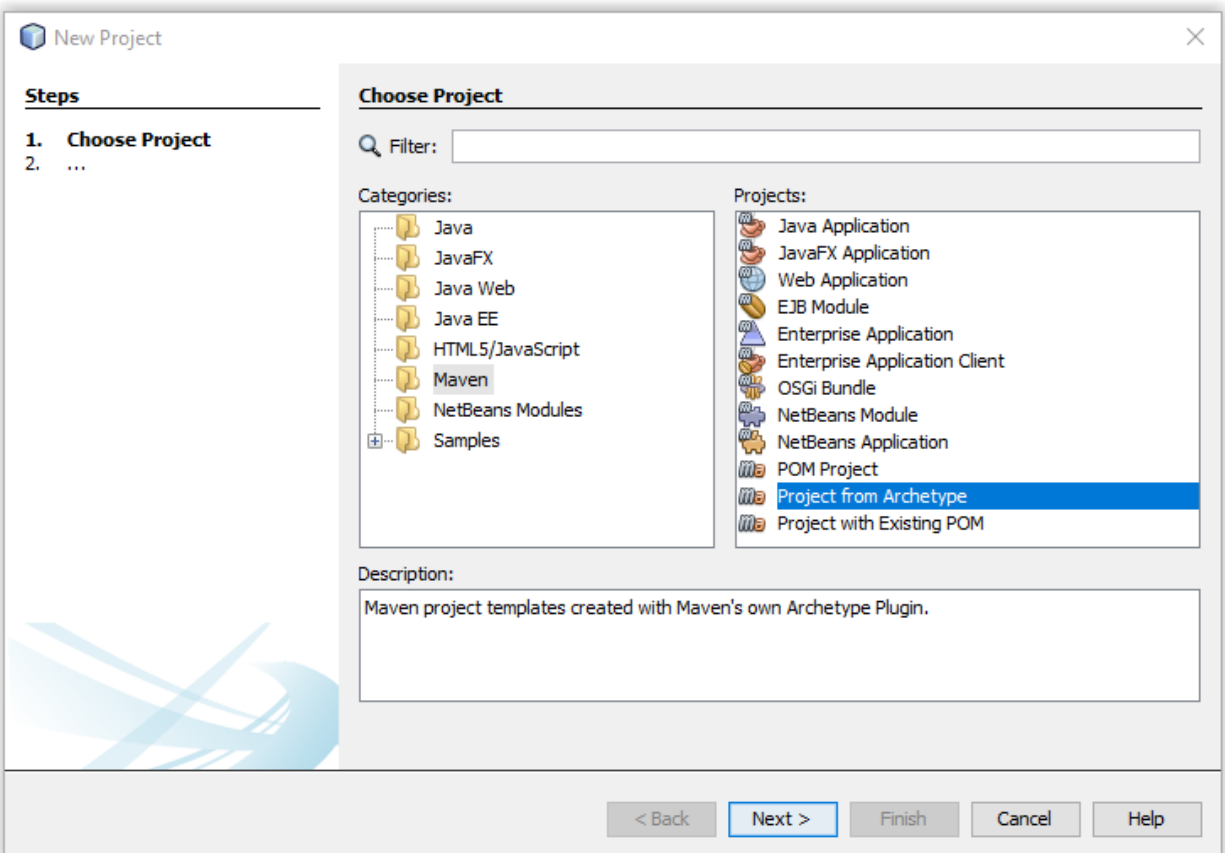
Utilice el IDE para crear un proyecto utilizando el arquetipo **meta-jee2-archetype**, cuyas coordenadas Maven se muestran en la siguiente tabla.

Group Id	org.proyecto-adalid.meta
Artifact Id	meta-jee2-archetype
Version	2.0
Repository	http://repo.maven.apache.org/maven2/

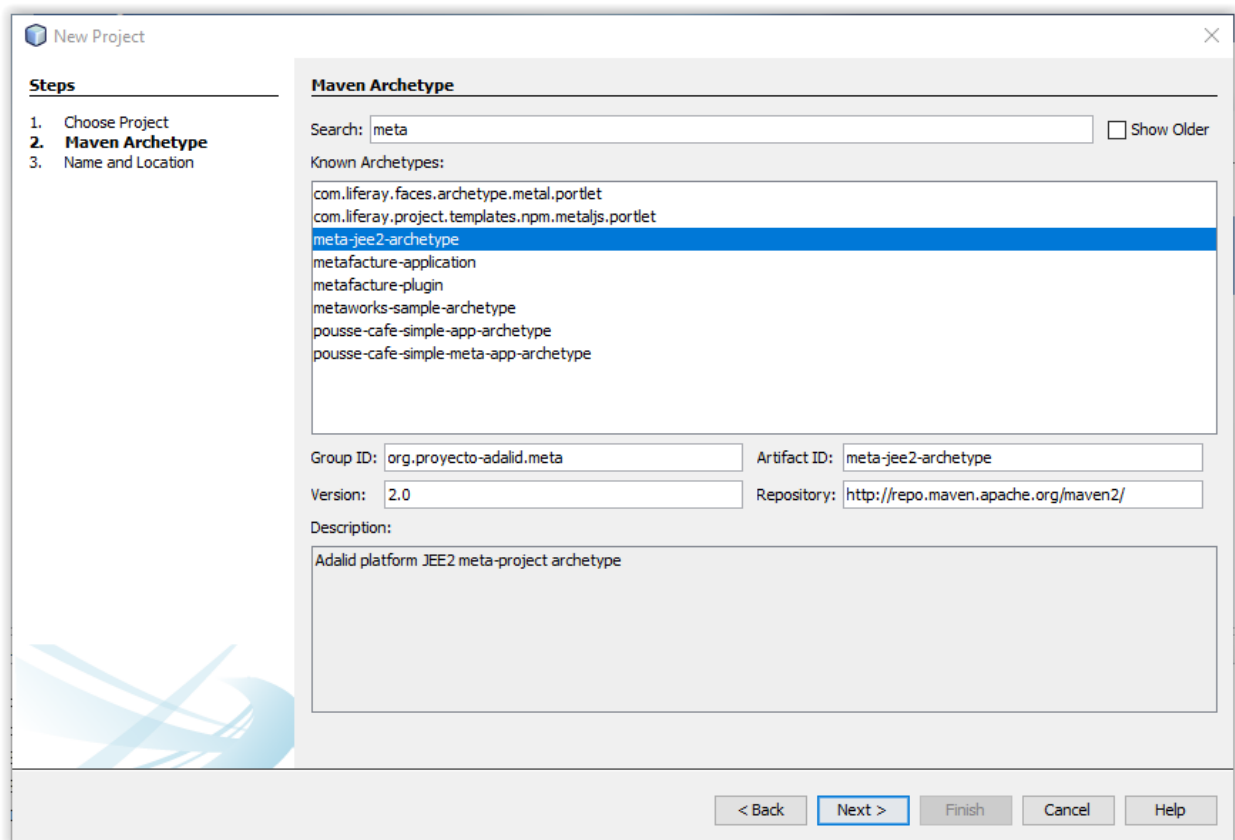
Por ejemplo, utilizando NetBeans se deben ejecutar los siguientes pasos:

- En el menú de barra de NetBeans, seleccione **File**.

- En el menú desplegable **File** seleccione **New Project...**
 - Esta acción da inicio a la secuencia de pasos para la creación de proyectos:
 - Paso 1: **Choose Project**
 - Paso 2: **Maven Archetype**
 - Paso 3: **Name and Location**
- En el paso 1, **Choose Project**
 - En la lista **Categories**, seleccione **Maven**.
 - En la lista **Projects**, seleccione **Project from Archetype**.
 - Haga clic en el botón **Next**.



- En el paso 2, **Maven Archetype**
 - Busque **meta-jee2-archetype** (basta con escribir **meta-** en el campo **Search**) y selecciónelo en la lista **Known Archetypes**; esta acción asigna valores a los siguientes campos:
 - **Group ID:** org.proyecto-adalid.meta
 - **Artifact ID:** meta-jee2-archetype
 - **Version:** 2.0
 - **Repository:** http://repo.maven.apache.org/maven2/
 - Haga clic en el botón **Next**.



New Project

Steps

1. Choose Project
2. **Maven Archetype**
3. Name and Location

Maven Archetype

Search: meta ☐ Show Older

Known Archetypes:

- com.liferay.faces.archetype.metal.portlet
- com.liferay.project.templates.npm.metaljs.portlet
- meta-jee2-archetype**
- metafacture-application
- metafacture-plugin
- metaworks-sample-archetype
- pousse-cafe-simple-app-archetype
- pousse-cafe-simple-meta-app-archetype

Group ID: org.proyecto-adalid.meta Artifact ID: meta-jee2-archetype

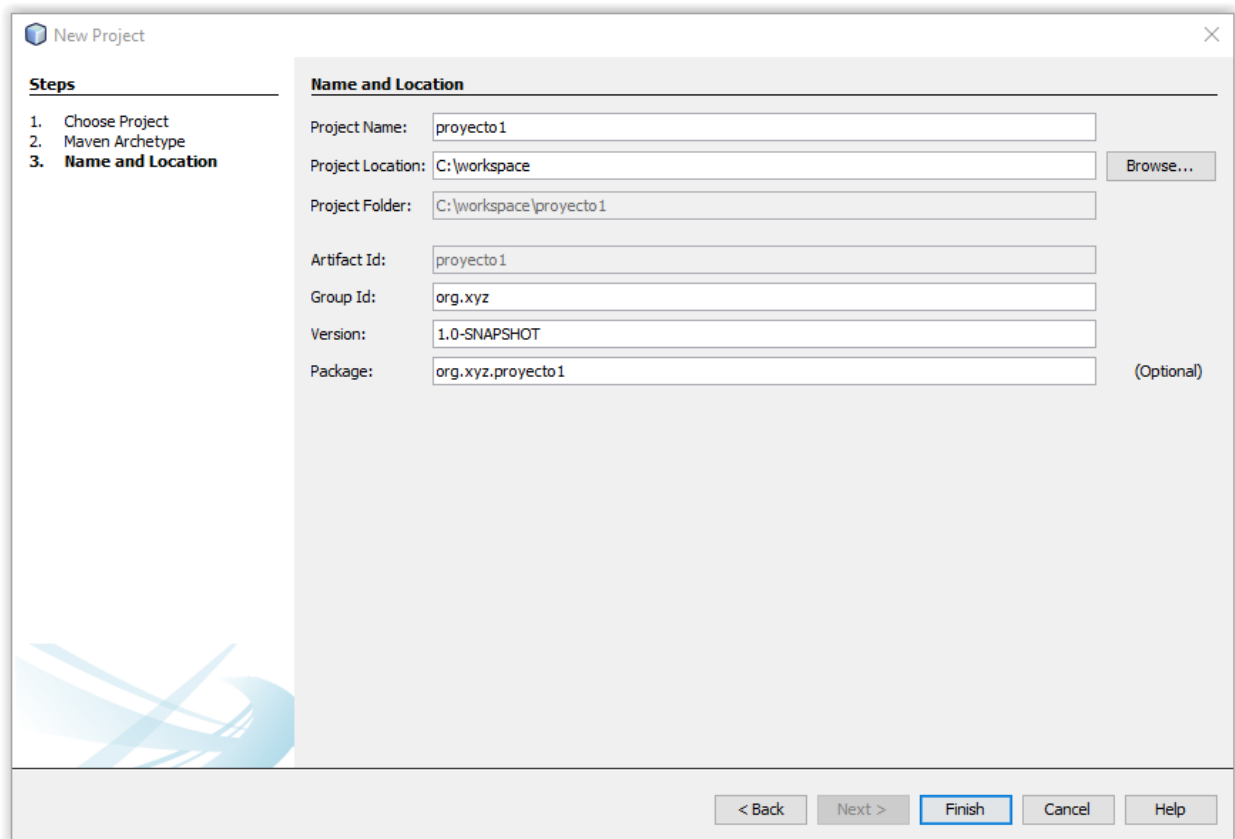
Version: 2.0 Repository: http://repo.maven.apache.org/maven2/

Description:

Adalid platform JEE2 meta-project archetype

< Back Next > Finish Cancel Help

- En el paso 3, **Name and Location**
 - En el campo **Project Name**, escriba el nombre de su proyecto. El nombre determina el nombre del subdirectorio donde se creará el meta-proyecto y también el **Artifact Id** de **Maven**; por lo tanto, tiene que cumplir con las convenciones de nomenclatura de Linux y también con las [convenciones de nomenclatura de Maven](#). Preferiblemente, el nombre comienza por una letra y solo contiene letras minúsculas y números.
 - En el campo **Project Location**, escriba la ruta del directorio **workspace** creado en la [Instalación de Adalid](#).
 - En el campo **Group Id**, escriba el nombre de su grupo. El nombre tiene que cumplir con las convenciones de nomenclatura de **Maven**.
 - En el campo **Version**, escriba la versión de su proyecto. La versión tiene que cumplir con las convenciones de nomenclatura de **Maven**. Normalmente se comienza por la versión 1.0-SNAPSHOT.
 - En el campo **Package**, escriba el nombre del paquete raíz de su proyecto. El nombre tiene que cumplir con las convenciones de nomenclatura de **Java**.
 - Haga clic en el botón **Finish**; esta acción genera el meta proyecto en el directorio **workspace**.



New Project

Steps

1. Choose Project
2. Maven Archetype
3. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

Package: (Optional)

< Back Next > **Finish** Cancel Help

Extensión de meta-proyectos creados con meta-jee2-archetype

Los meta-proyectos creados con el arquetipo **meta-jee2-archetype** 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 el arquetipo **meta-jee2-archetype** incluyen los módulos predefinidos de la plataforma **jee2**: 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.

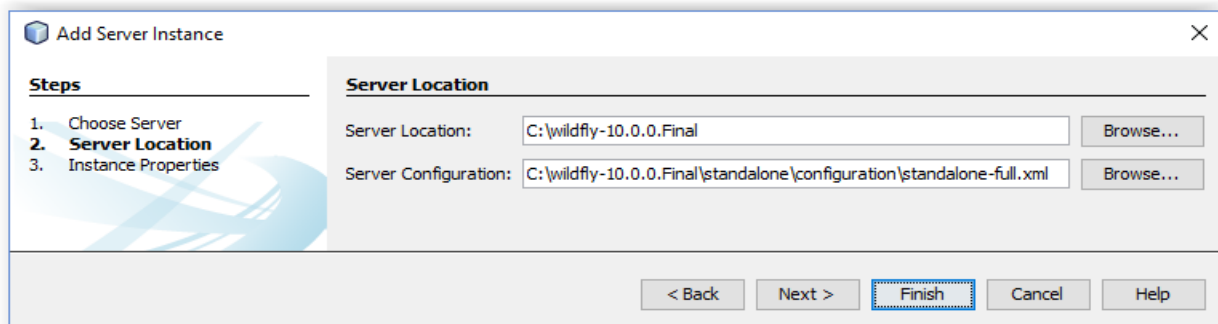
Para obtener información detallada sobre cómo extender los meta-proyectos creados con el arquetipo **meta-jee2-archetype** con módulos propios de su aplicación, consulte la [Guía de Meta-Programación](#).

Configuración de NetBeans para proyectos generados con jee2

Si desea construir (*build*) y ejecutar los proyectos generados con la plataforma **jee2** 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 **jee2** 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**.



Configuración de eclipse para proyectos generados con la plataforma jee2

Si desea construir (*build*) y ejecutar los proyectos generados con la plataforma **jee2** 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 **jee2** 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 jee2

Para generar proyectos con la plataforma **jee2** a partir de meta-proyectos creados con el arquetipo **meta-jee2-archetype** siga los siguientes pasos:

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

Abra el meta-proyecto creado utilizando el arquetipo **meta-jee2-archetype**. El meta-proyecto incluye la clase **root-package.meta.proyecto.Maestro** para generar proyectos con PostgreSQL y la clase **root-package.meta.proyecto.oracle.Maestro** para generar proyectos con Oracle. En ambos casos, **root-package** corresponde al nombre del paquete raíz de su proyecto, el cual especificó al crear el meta-proyecto utilizando el arquetipo **meta-jee2-archetype**.

En la clases **Maestro**, las siguientes instrucciones:

```
maestro.putEnvironmentVariable(VERSION_JAVA, "1.8.0_112");
maestro.putEnvironmentVariable(VERSION_GLASSFISH, "4.1.1");
maestro.putEnvironmentVariable(VERSION_WILDFLY, "10.1.0.Final");
maestro.putEnvironmentVariable(VERSION_ORACLE, "11.2.0");
maestro.putEnvironmentVariable(VERSION_POSTGRESQL, "9.6");
```

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

No es necesario especificar la versión de cada producto. Basta con especificar las versiones de los productos correspondientes a la variante de la plataforma **jee2** 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 **Maestro**, la siguiente instrucción:

```
maestro.setAlias("xyz2ap101");
```

establece el alias del proyecto **Maestro**, el cual se utilizará como nombre del proyecto.

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 **jee2ap101**, **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 **xyz2ap101**.

- Especificar la variante de la plataforma **jee2** a utilizar para generar el proyecto, mediante el argumento del método `generate` en la siguiente instrucción:

```
maestro.generate(plataforma);
```

Las variantes disponibles son:

En la clase **meta.proyecto.Maestro**:



PLATAFORMA_MAVEN_POSTGRESQL_GLASSFISH
PLATAFORMA_MAVEN_POSTGRESQL_WILDFLY

En la clase **meta.proyecto.oracle.Maestro**:

PLATAFORMA_MAVEN_ORACLE_GLASSFISH
PLATAFORMA_MAVEN_ORACLE_WILDFLY

- Ejecutar la clase **Maestro**.

Esta acción genera el proyecto **xyz2ap101** en el directorio **\$HOME/workspace/xyz2ap101/source/xyz2ap101**

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

- Construir (build) el proyecto **xyz2ap101**.

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

Configuración de proyectos generados con la plataforma jee2

Para configurar los proyectos generados con la plataforma **jee2** siga los siguientes pasos:

- Ejecutar el script **\$HOME/workspace/xyz2ap101\$HOME-setup.sh**.
- Revisar y, en caso de ser necesario, ajustar las variables de entorno que se establecen en los siguientes archivos del directorio **\$HOME/workspace/xyz2ap101/source/management**:

variables-conf.sh
variables-home.sh

Si utiliza GlassFish:

asadmin.password
variables-glassfish.sh

Si utiliza WildFly:

variables-jboss.sh

Si utiliza PostgreSQL:

variables-postgresql.bat

Si utiliza Oracle:

variables-oracle.bat

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 jee2

Para configurar PostgreSQL para los proyectos generados con la plataforma **jee2** siga los siguientes pasos:

- Iniciar PostgreSQL.
- Ejecutar el script `$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/postgresql/createdb.sh`.
- Ejecutar el script `$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/postgresql/makedb.sh`.

Configuración de Oracle para proyectos generados con jee2

Para configurar Oracle para los proyectos generados con la plataforma **jee2** siga los siguientes pasos:

- Iniciar Oracle.
- Ejecutar el script `$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/oracle/createdb.sh`.
- Ejecutar el script `$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/oracle/makedb.sh`.

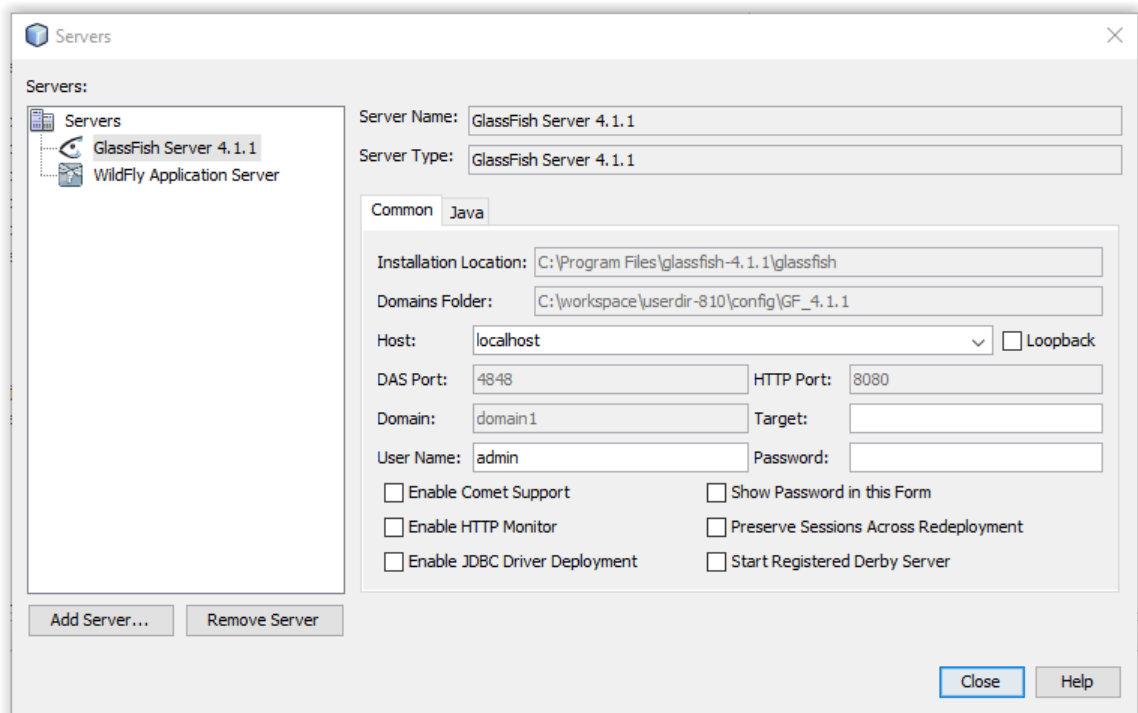
Configuración de GlassFish para proyectos generados con jee2

Para configurar GlassFish para los proyectos generados con la plataforma **jee2** 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 **\$HOME/workspace/xyz2ap101/source/management/setup/config/log4j** al directorio **domain1/config**.
- Iniciar GlassFish.
- Ejecutar el script **\$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/glassfish/add-libraries.sh**.
- Detener y reiniciar GlassFish.
- Ejecutar los siguientes scripts, que se encuentran en el directorio **\$HOME/workspace/xyz2ap101/ 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 jee2

Para configurar WildFly para los proyectos generados con la plataforma **jee2** 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 **\$HOME/workspace/xyz2ap101/source/management/setup/config/wildfly/standalone/configuration/versión-wildfly/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:versión-urn">** del archivo **standalone-full.xml** (archivo 020-logging.xml).

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

    <logger category="xyz2ap101">
        <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** o del subdirectorio **oracle**, según corresponda).

Para PostgreSQL:

```
<datasources>

    <datasource
        jta="true"
        jndi-name="java:/jdbc/xyz2ap101"
        pool-name="xyz2ap101-pool"
        enabled="true"
        use-java-context="true"
        spy="false"
        use-ccm="true">
        <connection-url>
            jdbc:postgresql://localhost:5432/xyz2ap101
        </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>
```

Para Oracle:

```
<datasources>

  <datasource
    jta="true"
    jndi-name="java:/jdbc/xyz2ap101"
    pool-name="xyz2ap101-pool"
    enabled="true"
    use-java-context="true"
    spy="false"
    use-ccm="true">
    <connection-url>
    jdbc:oracle:thin:@localhost:1521:XE
    </connection-url>
    <driver>ojdbc6</driver>
    <security>
    <user-name>XYZ2AP101</user-name>
    <password>oracle</password>
    </security>
  </datasource>

  <drivers>

    <driver name="ojdbc6" module="com.oracle.ojdbc6"/>

  </drivers>

</datasources>
```

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

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

  <mail-session
    name="xyz2ap101-mail-session"
    jndi-name="java:jboss/mail/xyz2ap101MailerBeanSession">
    <smtp-server
      ssl="true"
      outbound-socket-binding-ref="google-mail-smtp"
      username="xyz2ap101@gmail.com"
      password="xyz2ap101-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:versión-urn">** del archivo **standalone-full.xml** (archivo 050-messaging-activemq-server.xml).

```
<server name="default">

    <connection-factory
        name="xyz2ap101BusinessProcessMessageFactory"
        connectors="in-vm"
        entries="jms/xyz2ap101BusinessProcessMessageFactory"/>
    <connection-factory
        name="xyz2ap101UtilityProcessMessageFactory"
        connectors="in-vm"
        entries="jms/xyz2ap101UtilityProcessMessageFactory"/>
    <jms-queue
        name="xyz2ap101BusinessProcessRequestMessageQueue"
        entries="jms/xyz2ap101BusinessProcessRequestMessageQueue
java:jboss/exported/jms/xyz2ap101BusinessProcessRequestMessageQueue"/>
    <jms-queue
        name="xyz2ap101BusinessProcessReplyMessageQueue"
        entries="jms/xyz2ap101BusinessProcessReplyMessageQueue
java:jboss/exported/jms/xyz2ap101BusinessProcessReplyMessageQueue"/>
    <jms-queue
        name="xyz2ap101BusinessProcessReturnMessageQueue"
        entries="jms/xyz2ap101BusinessProcessReturnMessageQueue
java:jboss/exported/jms/xyz2ap101BusinessProcessReturnMessageQueue"/>
    <jms-queue
        name="xyz2ap101UtilityProcessRequestMessageQueue"
        entries="jms/xyz2ap101UtilityProcessRequestMessageQueue
java:jboss/exported/jms/xyz2ap101UtilityProcessRequestMessageQueue"/>
    <jms-queue
        name="xyz2ap101UtilityProcessReplyMessageQueue"
        entries="jms/xyz2ap101UtilityProcessReplyMessageQueue
java:jboss/exported/jms/xyz2ap101UtilityProcessReplyMessageQueue"/>
    <jms-queue
        name="xyz2ap101UtilityProcessReturnMessageQueue"
        entries="jms/xyz2ap101UtilityProcessReturnMessageQueue
java:jboss/exported/jms/xyz2ap101UtilityProcessReturnMessageQueue"/>

</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="xyz2ap101-policy">
        <authentication>
            <login-module code="Database" flag="required">
                <module-option
                    name="dsJndiName"
                    value="java:/jdbc/xyz2ap101"/>
                <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:versión-urn">** del archivo **standalone-full.xml**.
- Ajustar los parámetros de la JVM en el archivo **standalone.conf**. Se recomienda dar a WildFly tanta memoria como sea posible sin afectar negativamente los demás procesos que ejecutan en el mismo servidor. Utilice los argumentos de JVM **-Xms** y **-Xmx** para configurar el mínimo y el máximo de memoria RAM disponible para WildFly. Se recomienda configurarlos con el mismo valor.

En la línea que establece la variable **JAVA_OPTS**, cambiar **-Xms64m** y **-Xmx512m** al menos por **-Xms1024m** y **-Xmx1024m**, respectivamente.

Adicionalmente se recomienda agregar las siguientes opciones:

-XX:+UseParallelGC -XX:+AggressiveOpts -XX:+UseFastAccessorMethods

- Si utiliza NetBeans, ajustar los parámetros de la JVM en las propiedades del servidor WildFly.
- Ejecutar el script **\$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/jboss/add-modules.sh**.
- Iniciar WildFly.



- Ejecutar el script **\$HOME/workspace/xyz2ap101/source/management/resources/scripts/linux/jboss/create-system-properties.sh**.
- Detener y reiniciar WildFly.



Ejecución de proyectos generados con jee2

Para ejecutar los proyectos generados con la plataforma **jee2** siga los siguientes pasos:

- Ejecutar el proyecto **xyz2ap101** 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/xyz2ap101-web/>**.

- 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**.