



Eusko Jaurlaritzaren
Informatika Elkarte

Sociedad Informática
del Gobierno Vasco



Instalación en PC local

Eclipse y WebLogic Server 19c (14.1.1.0.0)

Este documento es propiedad de Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, S.A. (EJIE) y su contenido es CONFIDENCIAL. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Versión	Fecha	Resumen de cambios	Elaborado por:	Aprobado por:
1.0.0	15/02/2024	Primera versión	Marcos Llorente	
1.1.0	22/03/2024	Incluir sección de integración con XLNetS	Xabier Agustín	

Índice

1	Introducción	4
2	Instalación del software base	5
2.1	Servidor de aplicaciones y dominio.....	5
2.1.1	Instalación de WebLogic	5
2.1.2	Generación de nuevo dominio.....	10
2.2	Integración del servidor de aplicaciones con XLNetS	14
2.3	Parametrización específica en WebLogic 14.....	15
2.4	Eclipse IDE	15
2.4.1	Codificación de ficheros	16
2.4.2	Librería de usuario UDAWLS14Classpath.....	16
2.4.3	Plugin de UDA	17
2.4.4	Desinstalar plugin de UDA	18
2.4.5	Integración del servidor de aplicaciones con Eclipse	19
2.4.6	Workspace.....	23
2.4.7	Configuración de las aplicaciones y uso de librerías.....	23
2.4.8	Errores detectados.....	24
3	Estructura de directorios de aplicaciones en PC local	26
3.1	Estructura en PC local.....	26
3.1.1	Proyecto Classes (bbbEARClasses)	30
3.1.2	Proyecto bbbNombreWAR	31
3.1.3	Proyecto bbbModuloEJB.....	33
3.1.4	Proyecto bbbEAR	34
3.1.5	Estructura en Servidor de Desarrollo.....	35
3.1.6	Correspondencia de aplicaciones entre PC local y Servidor de Desarrollo.....	37

1 Introducción

El alcance de este documento se limita a la configuración del entorno de Desarrollo en PC local para aplicativos Java con WebLogic Server 19c (14.1.1.0.0). En los entornos de Desarrollo, Pruebas (Preproducción) y Producción, se mantendrá la estructura de directorios planteada en el documento Normativa de Albergue.

En este documento se detalla:

- El proceso de instalación del software base.
 - WebLogic Server 19c con el dominio: dominio_wls14110.
 - Eclipse 2020-03 (4.15.0) como IDE.
- El proceso de configuración de dicho software para su integración entre sí y con el sistema de seguridad.
- La estructura de directorios para el desarrollo en PC local.
- La correspondencia entre las estructuras de directorios existentes en PC local y el Servidor de Desarrollo.

2 Instalación del software base

Descargar el servidor de aplicaciones de [Oracle WebLogic Server 19c \(14.1.1.0.0\)](#) seleccionando la versión genérica o desde las [descargas de UDA](#).

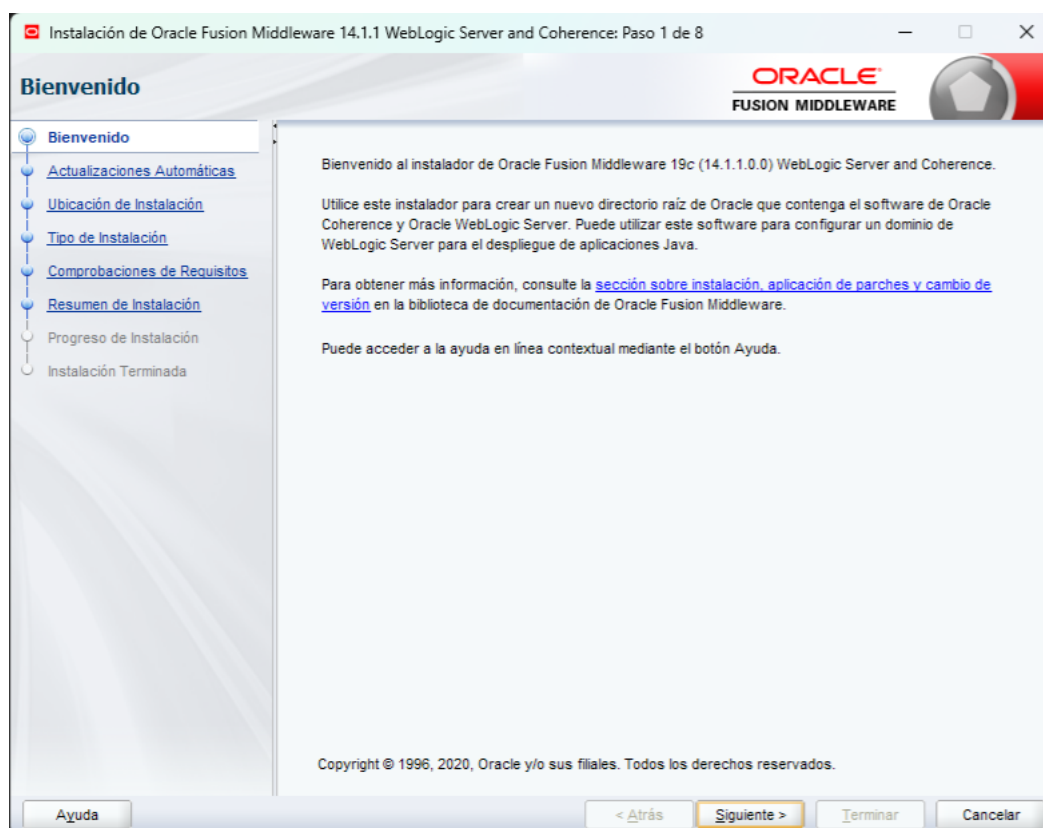
Después, seguir los pasos detallados a continuación para llevar a cabo su instalación:

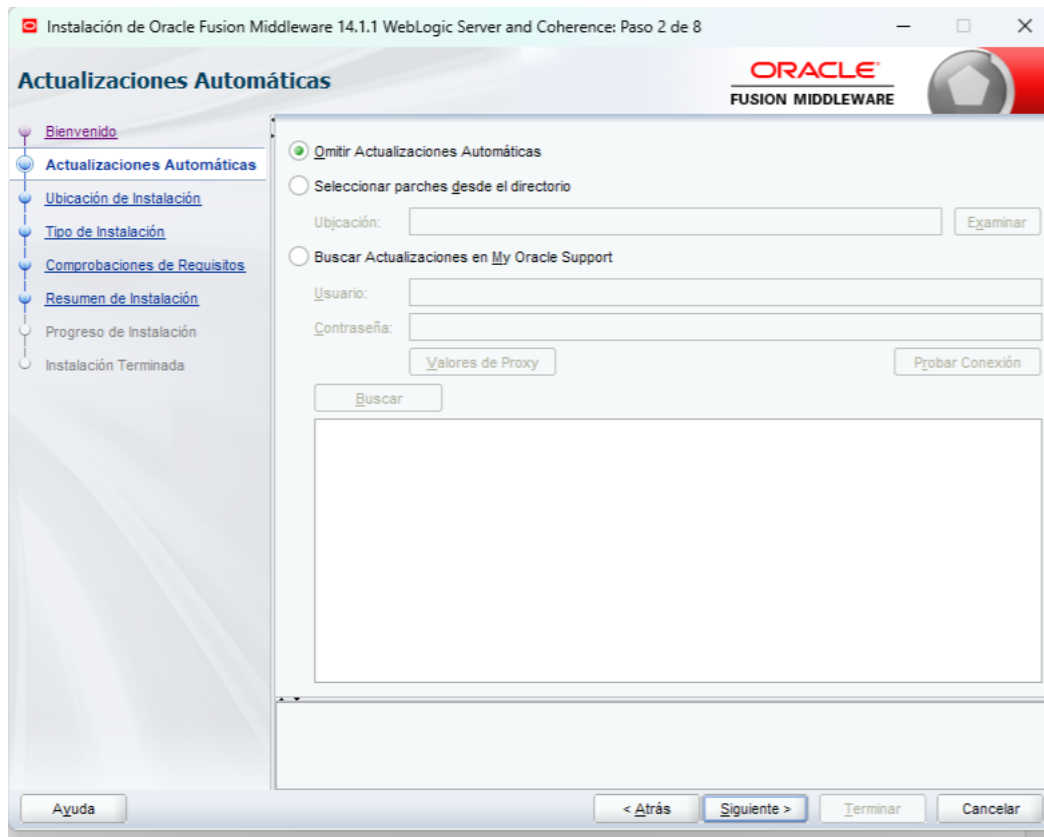
2.1 Servidor de aplicaciones y dominio

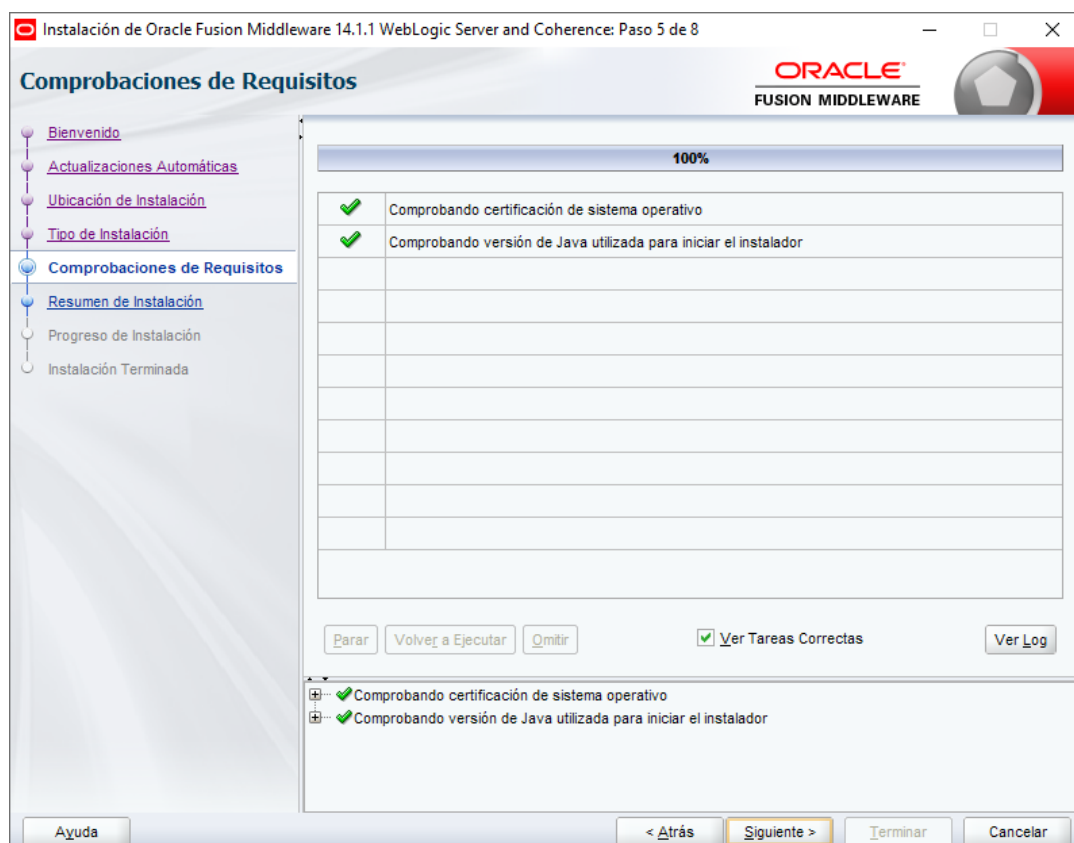
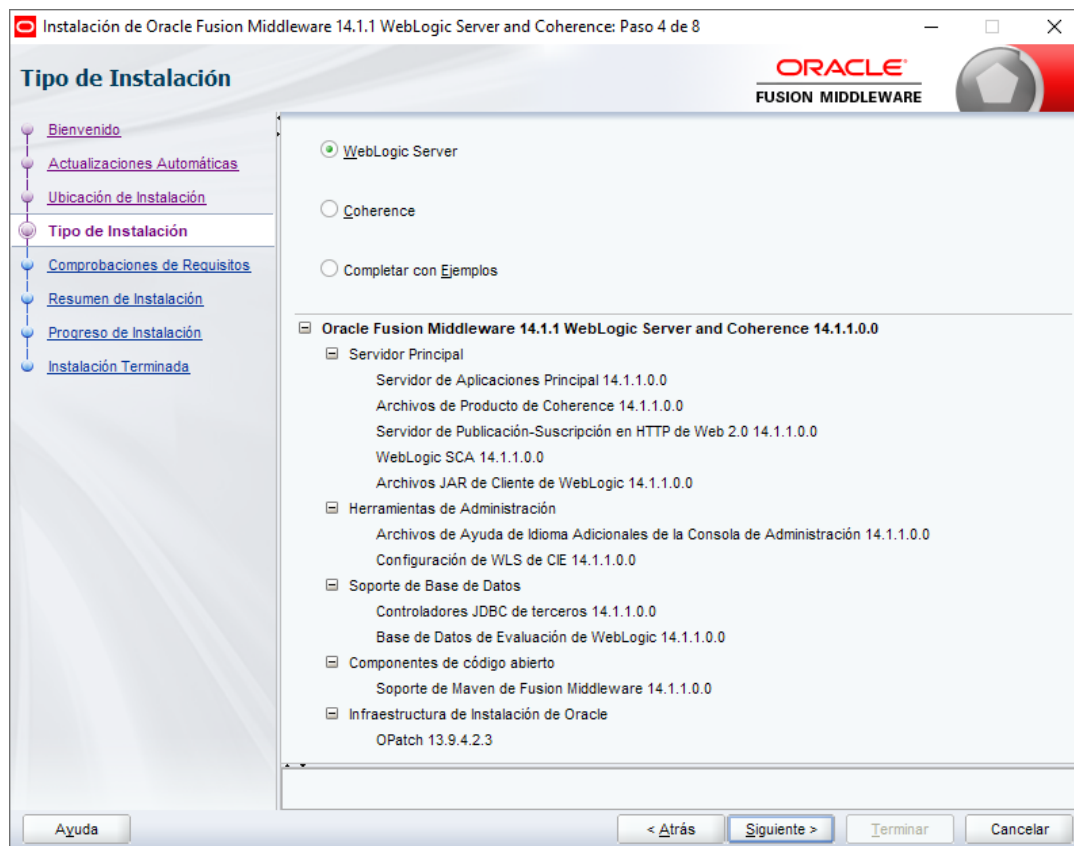
Puntos a tener en cuenta durante la ejecución del instalador:

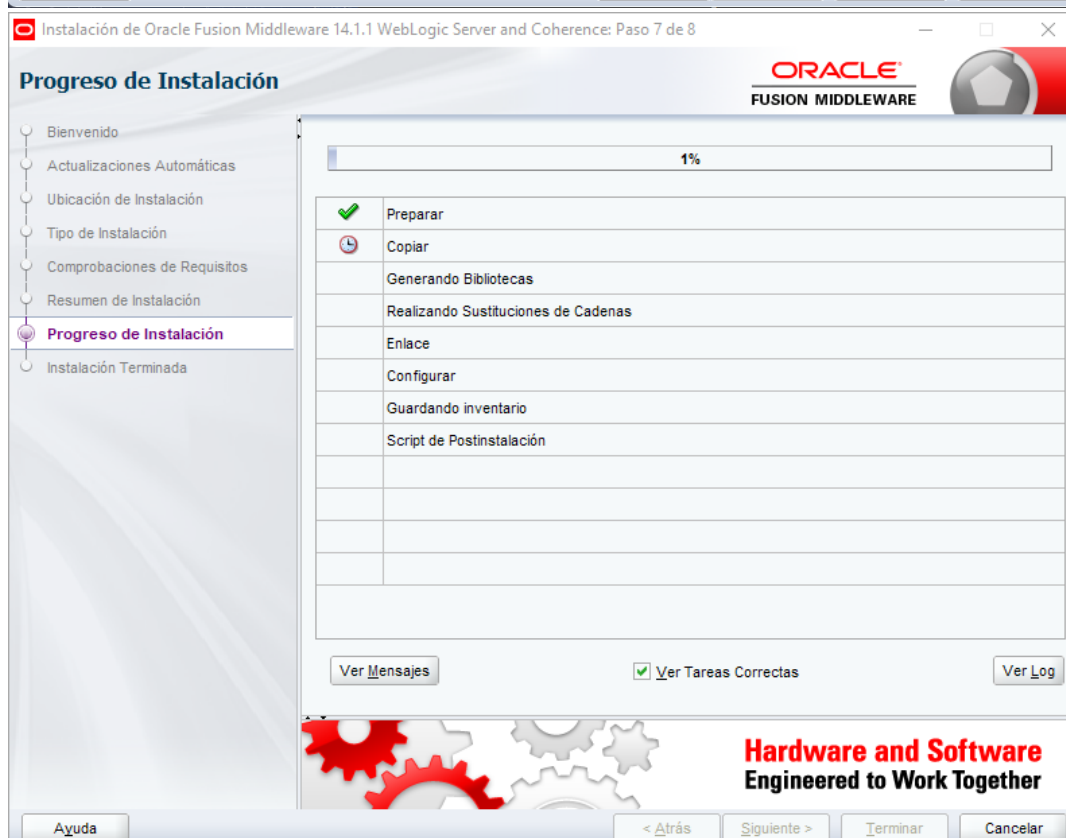
- Se instala un nuevo servidor.
- Se crea un nuevo dominio.
- Utilizar la JDK 11 de Oracle.

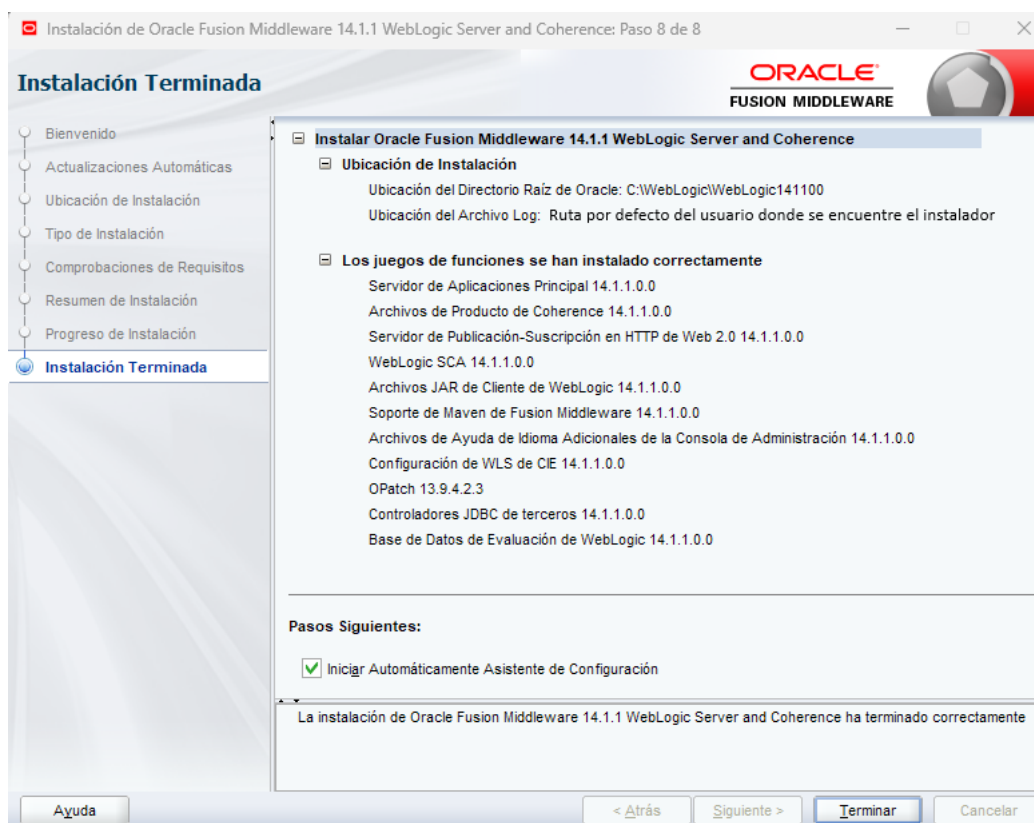
2.1.1 Instalación de WebLogic



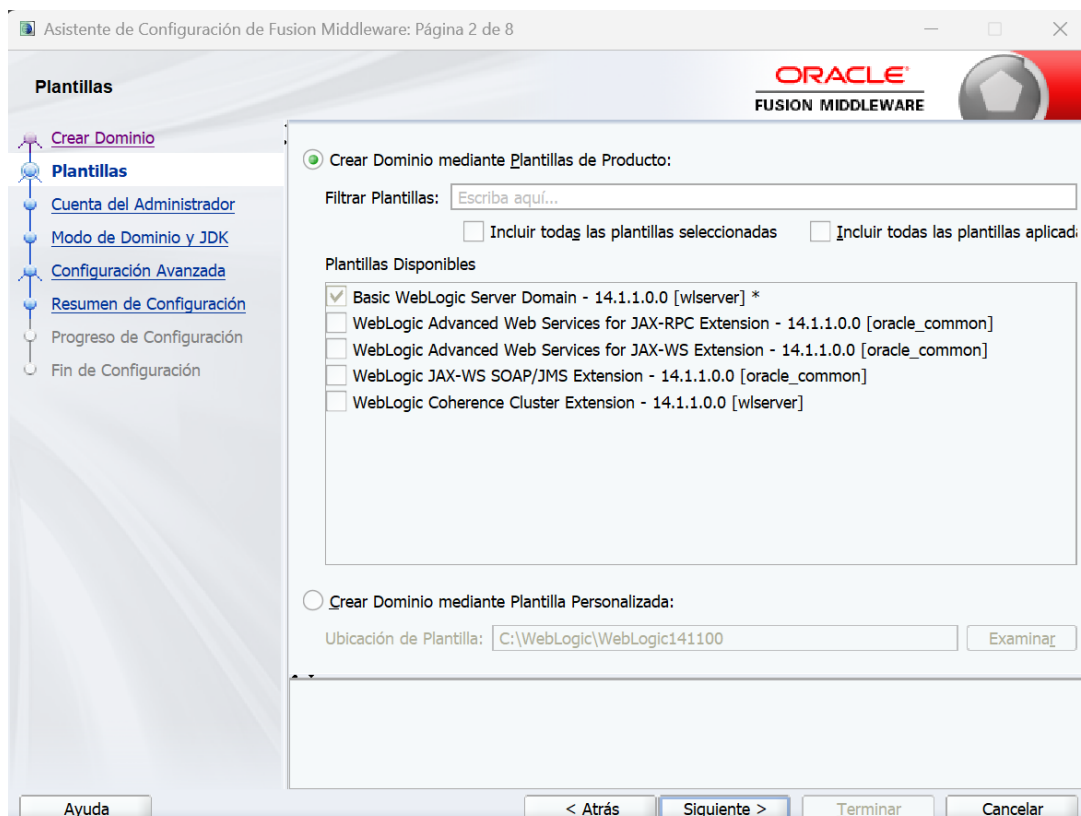
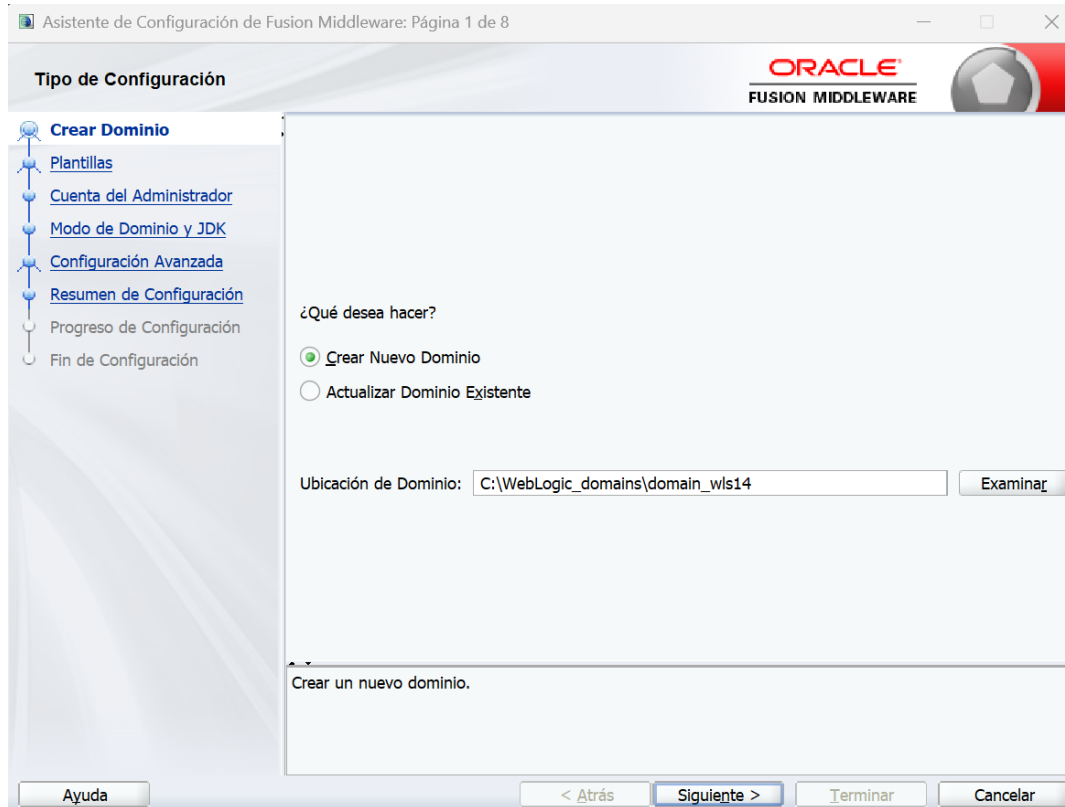








2.1.2 Generación de nuevo dominio



Asistente de Configuración de Fusion Middleware: Página 3 de 8

Cuenta del Administrador

ORACLE
FUSION MIDDLEWARE

Crear Dominio
Plantillas
Cuenta del Administrador
Modo de Dominio y JDK
Configuración Avanzada
Resumen de Configuración
Progreso de Configuración
Fin de Configuración

Nombre:

Contraseña:

Confirmar Contraseña:

Debe ser igual que la contraseña. La contraseña debe tener al menos 8 caracteres alfanuméricos con un número o carácter especial como mínimo.

Ayuda < Atrás **Siguiente >** Terminar Cancelar

Asistente de Configuración de Fusion Middleware: Página 4 de 8

Modo de Dominio y JDK

ORACLE
FUSION MIDDLEWARE

Crear Dominio
Plantillas
Cuenta del Administrador
Modo de Dominio y JDK
Configuración Avanzada
Resumen de Configuración
Progreso de Configuración
Fin de Configuración

Modo de Dominio

☒ Desarrollo
Utiliza boot.properties para el nombre de usuario y la contraseña y sondea las aplicaciones que desplegar.

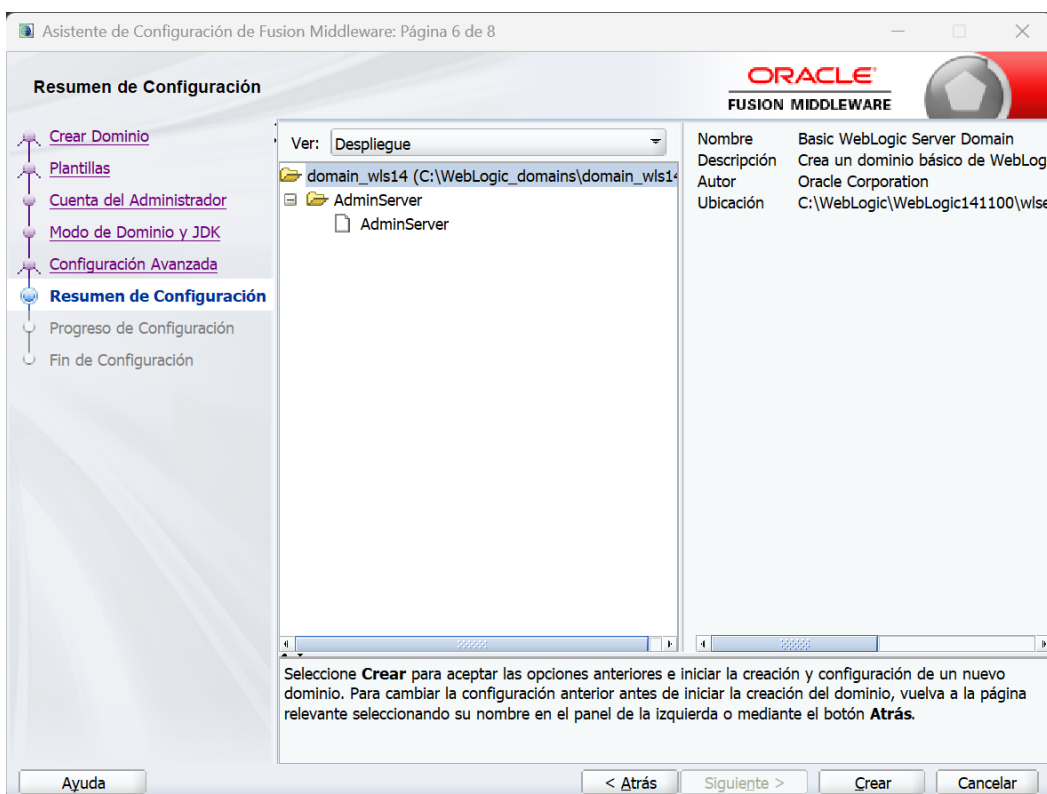
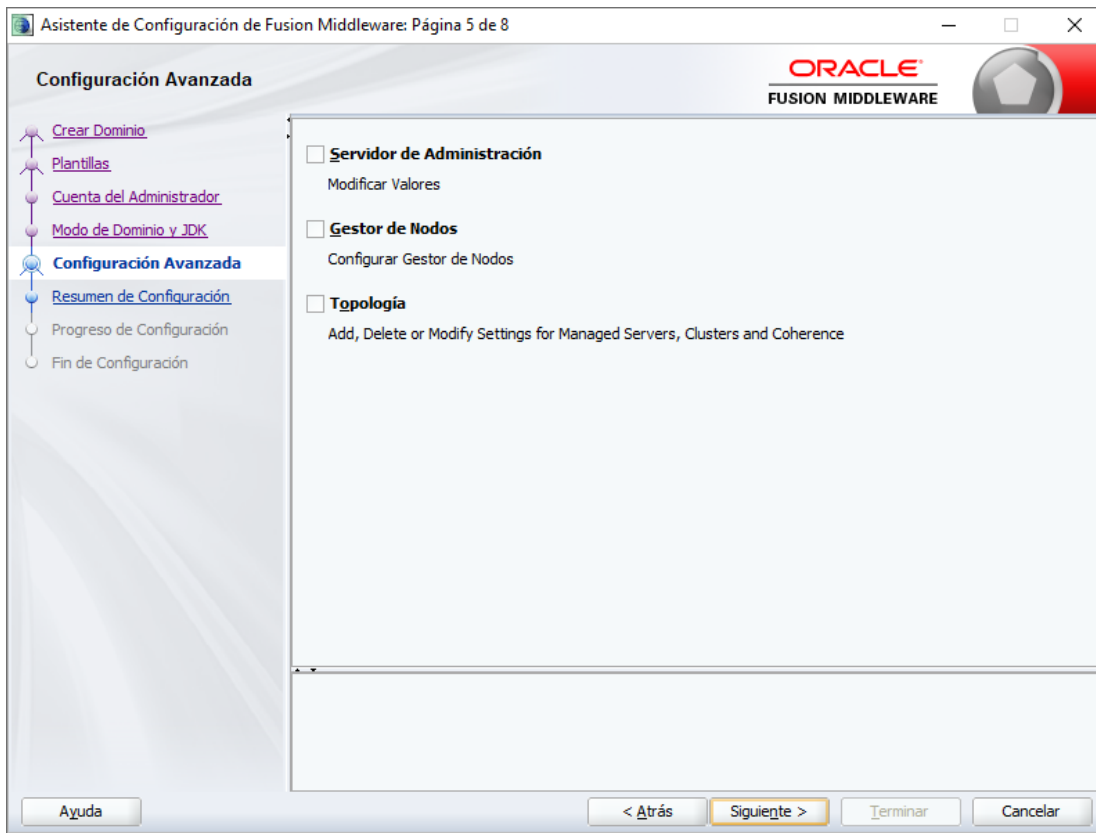
☐ Producción
Necesita la introducción de un nombre de usuario y una contraseña y no sondea las aplicaciones que desplegar.

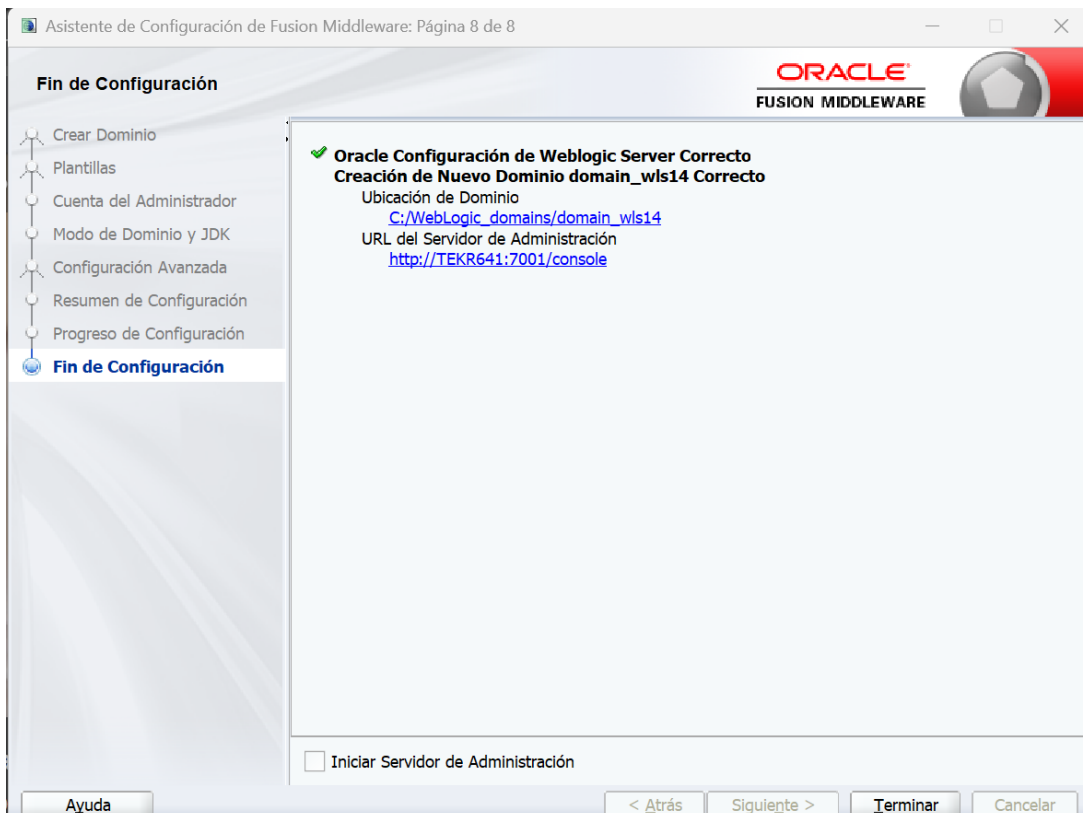
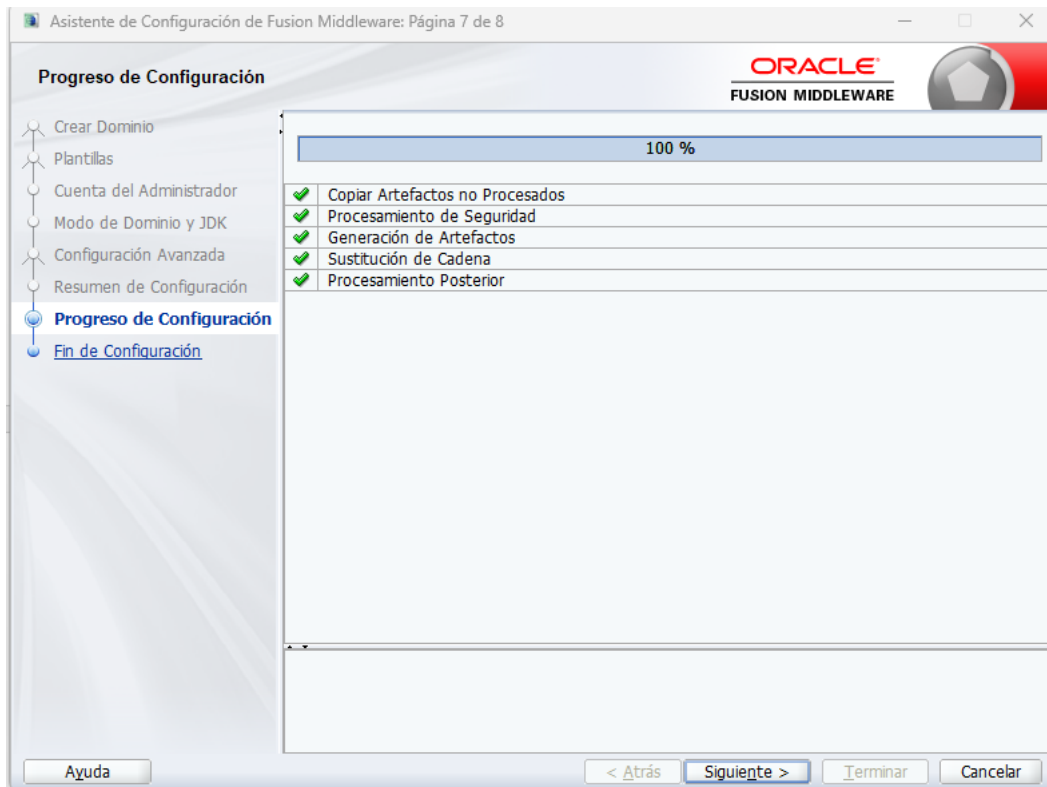
JDK

☐ Oracle HotSpot 11.0.16.1 C:\Users\MLLORE~1\Desktop\ENTORN~1\JDK\jdk-11~1.1

☒ Ubicación de JDK de Otros:

Ayuda < Atrás **Siguiente >** Terminar Cancelar





2.2 Integración del servidor de aplicaciones con XLNetS

XLNetS es el sistema de seguridad utilizado en el entorno de EJIE-GV, por tanto, todo lo relacionado con su configuración solo es necesario para desarrollos en dicho entorno. A continuación, los pasos a seguir para la integración:

1. Descargar las [configuraciones y librerías de XLNetS](#).
2. Copiar las carpetas en una ruta local.
3. Configurar el t3 de la instancia WebLogic. En cada fichero properties de las n38* sustituir el t3 por el utilizado en local, por ejemplo:

```
<parametro id="n38ProveedorURL"><valor>t3://localhost:7111</valor></parametro>
```

4. Configurar el log en cada fichero properties de las n38*.
 1. Sustituir la ruta del fichero configLog a utilizar, por ejemplo:

```
<parametro
id="n38FicheroLog"><valor>/config/xlnets/wl14_7111/config/n38n/configLog.x
ml</valor></parametro><parametro>
```

2. Configurar el path del DTD y el destino del log:

```
<!DOCTYPE log4j:configuration SYSTEM "/aplic/xlnets/lib/log4j.dtd">
<param name="File" value="c:/datos/n38a/n38a_7111.log" />
```

3. Para activar el log, poner el id n38Log a true (false para desactivar):

```
<parametro id="n38Log"><valor>>true</valor></parametro>
```

5. En el arranque de weblogic, en el script startWebLogic, poner las librerías en el classpath, por ejemplo en Windows:

```
@REM ##### n38 #####
set N38_LIB=%UNIDAD_XLNETS%\aplic\n38\lib\log4j\log4j-
1.2.12.jar;%UNIDAD_XLNETS%\aplic\n38\lib\cryptix\cryptix32.jar;%UNIDAD_XLNETS%\ap
lic\n38\lib\iws\iws.jar;%UNIDAD_XLNETS%\aplic\n38\lib\n38i.jar;%UNIDAD_XLNETS%\ap
lic\n38\lib\n38a.jar;%UNIDAD_XLNETS%\aplic\n38\lib\n38n.jar;%UNIDAD_XLNETS%\aplic
\n38\lib\n38c.jar
set N38OPTIONS="-Dfile.encoding=iso8859-1"
@REM ##### export values (PRE_CLASSPATH) #####
set EXT_PRE_CLASSPATH=%CONFIG%;%N38_LIB%;
```

6. Añadir variable de entorno con la ruta de la configuración. Hay que tener en cuenta que a partir de esta ruta cada componente espera después su carpeta con su config n38c/n38c.properties, n38a/n38a.properties etc. Igualmente, para usar el login de aplicación, en el arranque de WebLogic, poner la variable con los tres últimos números de la IP local:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -DN38ENTORNO=%UNIDAD_XLNETS% -DN38IPServidor=0.0.1
%N38OPTIONS%
```

7. Desplegar n38i.ear. Arrancar el WebLogic y desplegar el componente n38i.ear como aplicación con un orden de despliegue prioritario.

2.3 Parametrización específica en WebLogic 14

A continuación, se provee un ejemplo de la parametrización específica de WebLogic 14 que se encuentra ubicada en **startWeblogic.cmd**:

```
@ECHO OFF

@REM WARNING: This file is created by the Configuration Wizard.
@REM Any changes to this script may be lost when adding extensions to this
configuration.

SETLOCAL

@REM *****
@REM *                                CONFIGURACION EJIE                                *
@REM *****

@REM ##### configuration #####
set ENTORNO=C:\Users\nombreUsuario\Desktop\Entornos_UDA
set DOMAIN_HOME=%ENTORNO%\domains\domain_wls141
set CONFIG=%ENTORNO%\config\domain_wls141
set UNIDAD_XLNETS=%ENTORNO%\xlnets\wl14_7111

@REM ##### n38 #####
set N38_LIB=%UNIDAD_XLNETS%\aplic\n38\lib\log4j\log4j-
1.2.12.jar;%UNIDAD_XLNETS%\aplic\n38\lib\cryptix\cryptix32.jar;%UNIDAD_XLNETS%\aplic\n3
8\lib\iws\iws.jar;%UNIDAD_XLNETS%\aplic\n38\lib\n38i.jar;%UNIDAD_XLNETS%\aplic\n38\lib\
n38a.jar;%UNIDAD_XLNETS%\aplic\n38\lib\n38n.jar;%UNIDAD_XLNETS%\aplic\n38\lib\n38c.jar
set N38OPTIONS="-Dfile.encoding=iso8859-1"
set JAVA_OPTIONS=%JAVA_OPTIONS% -DN38ENTORNO=%UNIDAD_XLNETS% -DN38IPServidor=0.0.1
%N38OPTIONS%

@REM ##### allow external access to DTD, Schema and Stylesheet #####
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djavax.xml.accessExternalDTD=all -
Djavax.xml.accessExternalSchema=all -Djavax.xml.accessExternalStylesheet=all

@REM ##### The variable to override the standard memory arguments passed to java
#####
@REM set USER_MEM_ARGS=-Xms4g -Xmx4g -XX:MaxPermSize=1g
@REM set WLS_MEM_ARGS_64BIT=-Xms4096m -Xmx4096m

@REM ##### export values (PRE CLASSPATH) #####
set EXT_PRE_CLASSPATH=%CONFIG%;%N38_LIB%;

@REM *****
@REM *                                CONFIGURACION EJIE                                *
@REM *****

call "%DOMAIN_HOME%\bin\startWebLogic.cmd" %*

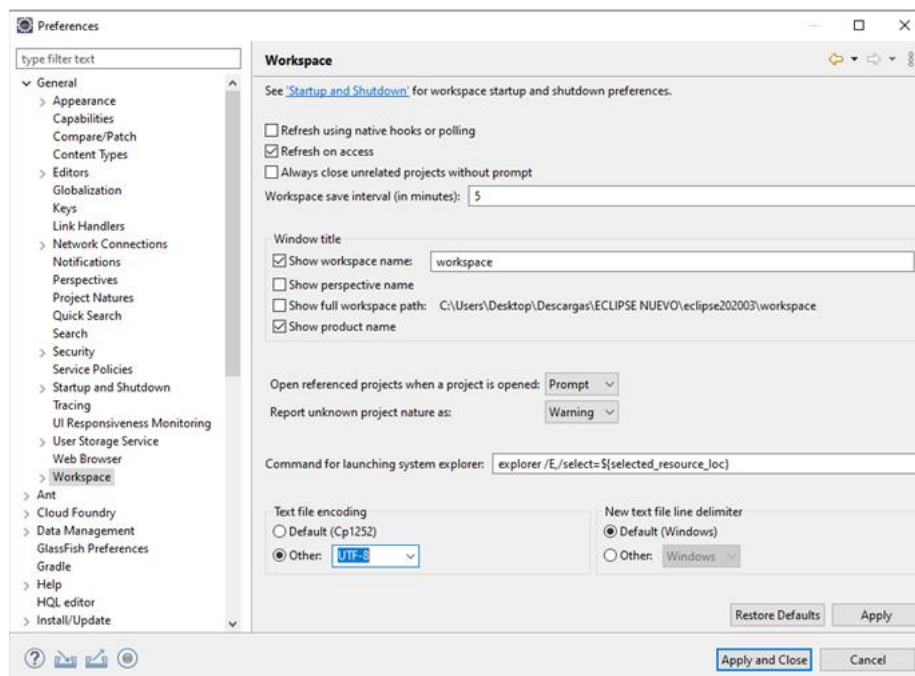
ENDLOCAL
```

2.4 Eclipse IDE

Como entorno de desarrollo integrado (IDE), se utiliza el Eclipse liberado por el equipo de UDA que está basado en la versión [2020-03](#) (4.15.0) e incluye todo lo necesario para poder comenzar con el desarrollo de aplicaciones UDA.

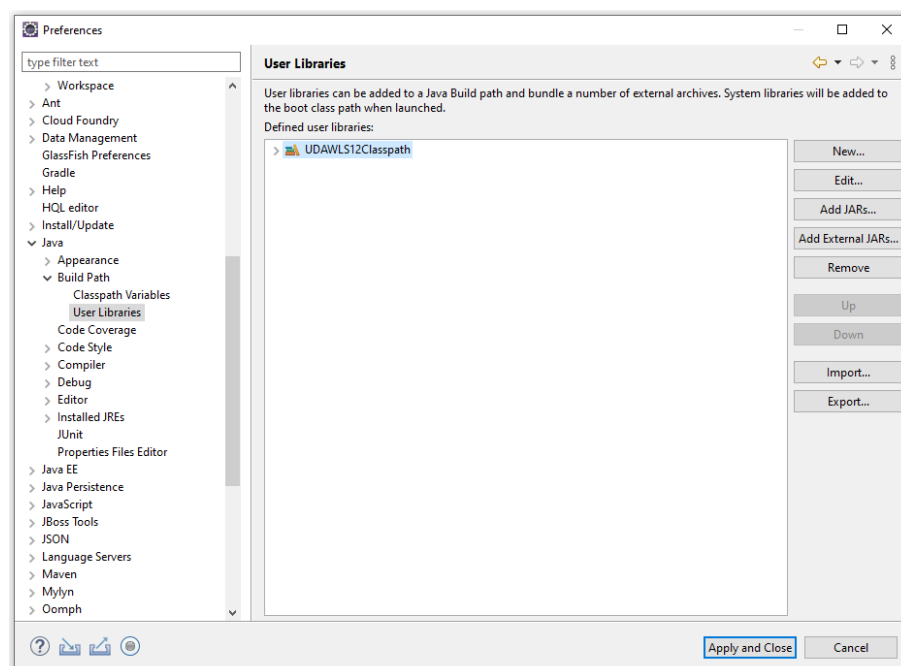
2.4.1 Codificación de ficheros

La codificación de los ficheros creados ha de ser cambiada a UTF-8 desde el menú de preferencias: *Window > Preferences > General > Workspace*.



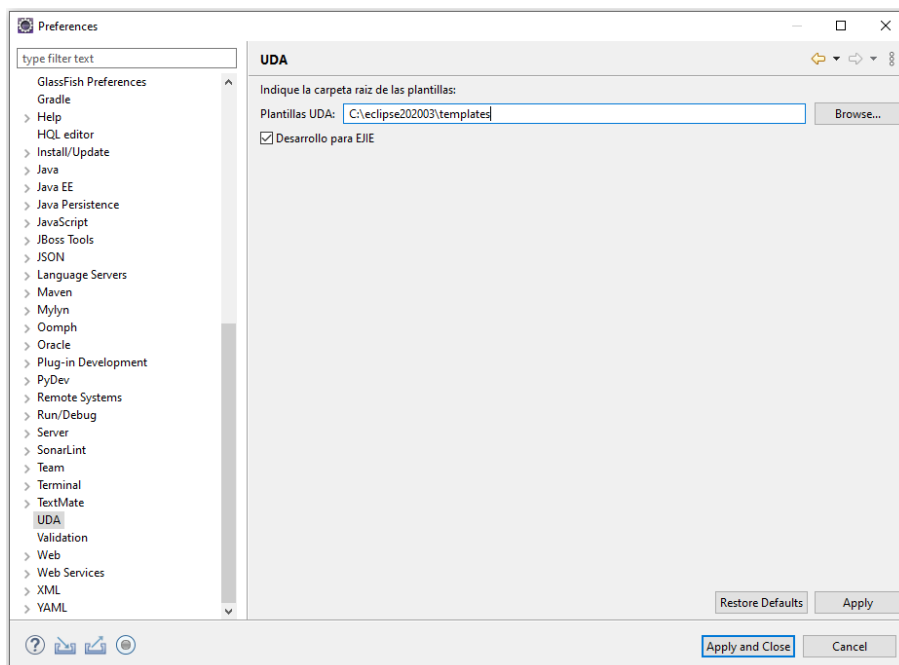
2.4.2 Librería de usuario UDAWLS14Classpath

Es necesario configurar la librería de usuario que contiene las referencias a las librerías del classpath de WebLogic (descargadas en el apartado 2.2).

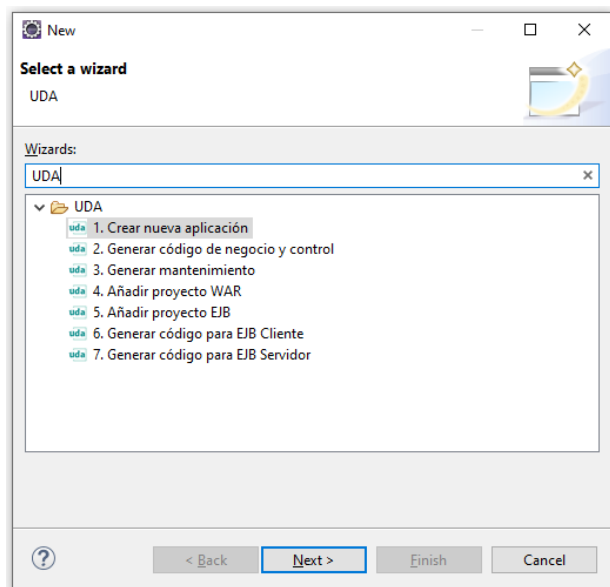


2.4.3 Plugin de UDA

Aunque el Eclipse provisto contiene el plugin de UDA, es necesario indicarle la ubicación de las plantillas (<ECLIPSE_HOME>\templates) además de indicar si se va a trabajar sobre un entorno EJIE, para ello, se deberá realizar a través de las preferencias del plugin: *Window > Preferences > UDA*.



Una vez completados los pasos anteriores, el entorno estará listo para generar, desarrollar y desplegar aplicaciones de UDA. Para usar el plugin se puede hacer desde *New > UDA*.



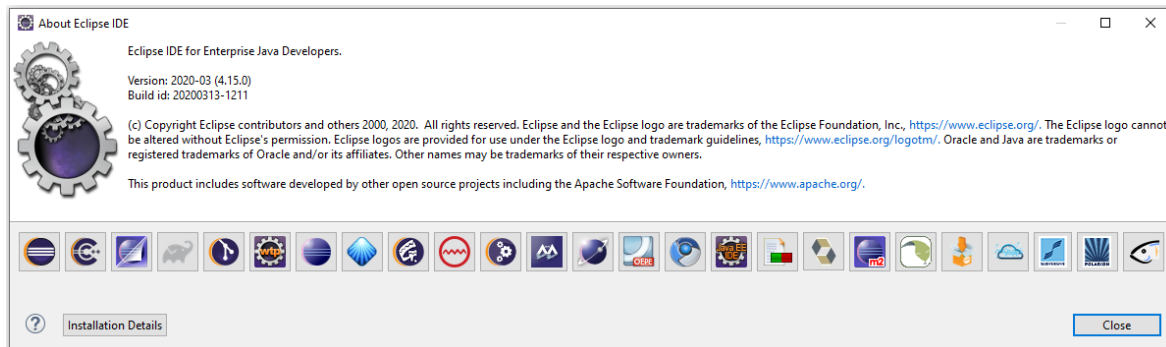
El plugin UDA se caracteriza por proveer a los desarrolladores de utilidades tales como generación de proyectos o la generación de código. A fin de agilizar el desarrollo, UDA se ha creado como conjunto de utilidades para desarrolladores.

Algunos asistentes del plugin utilizarán Maven para ejecutar tareas como, por ejemplo, descargarse las librerías de la aplicación. Por ello se ha de configurar las variables de entorno M2_REPO y MAVEN_HOME como se han indicado previamente.

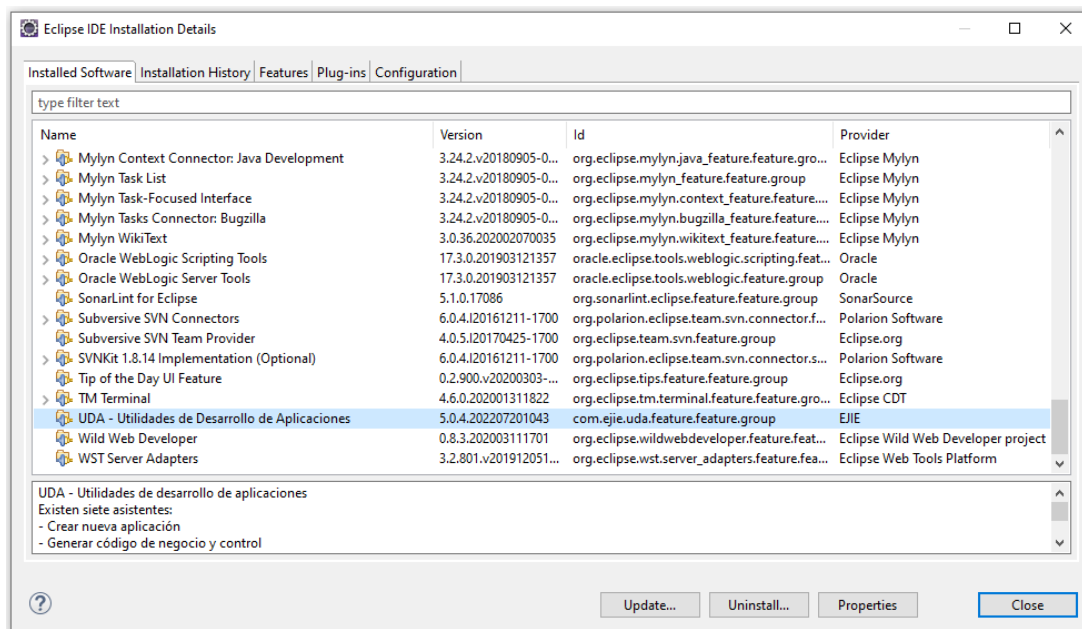
Para más información acerca del uso del plugin, se recomienda leer el siguiente documento: [Plugin UDA-Guia de uso del plugin](#)

2.4.4 Desinstalar plugin de UDA

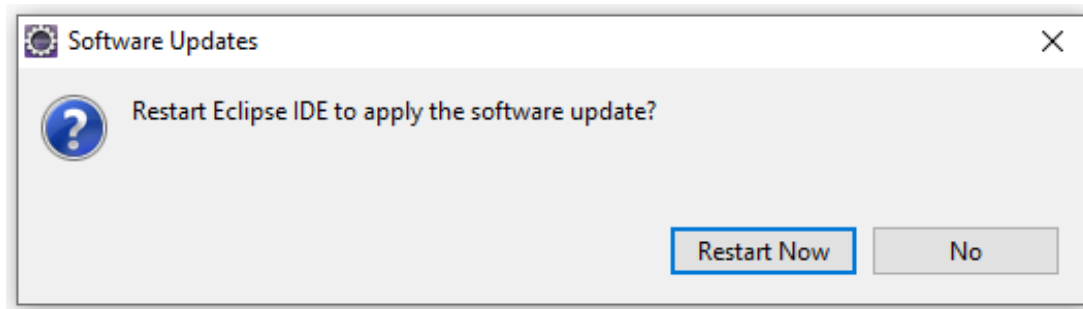
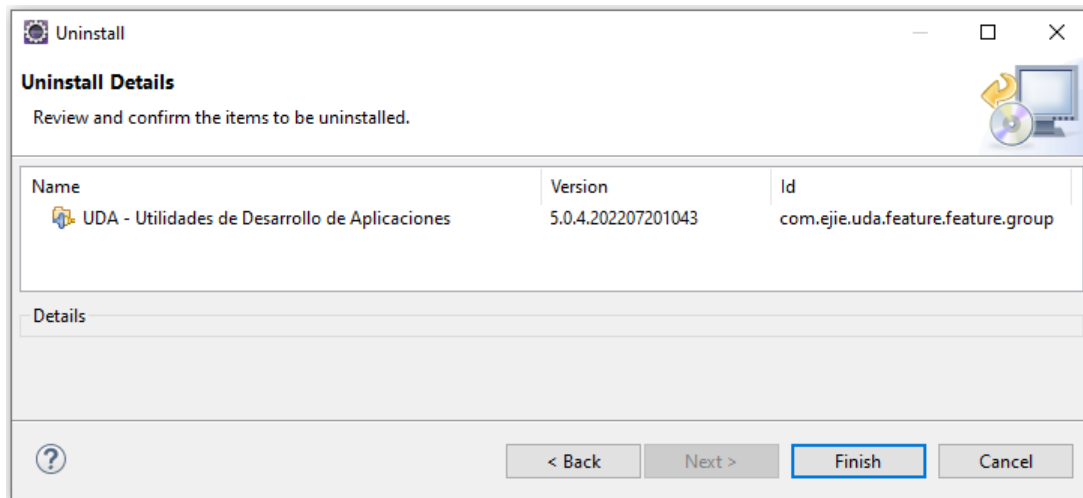
La desinstalación del plugin se realiza a través de *Help > About Eclipse IDE*



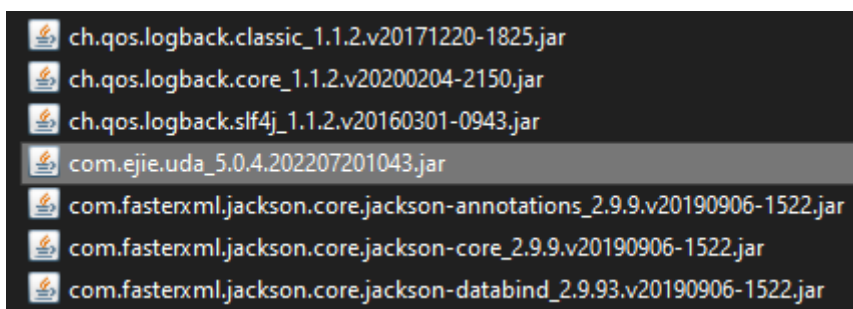
Pulsando el botón “Installation Details” y a continuación en la pestaña “Installed Software” seleccionando **UDA – Utilidades de Desarrollo de Aplicaciones** para después pulsar sobre “Uninstall”:



En la siguiente pantalla se deberá pulsar sobre el botón “Finish” y una vez completado el proceso, Eclipse solicitará reiniciarse para aplicar los cambios:

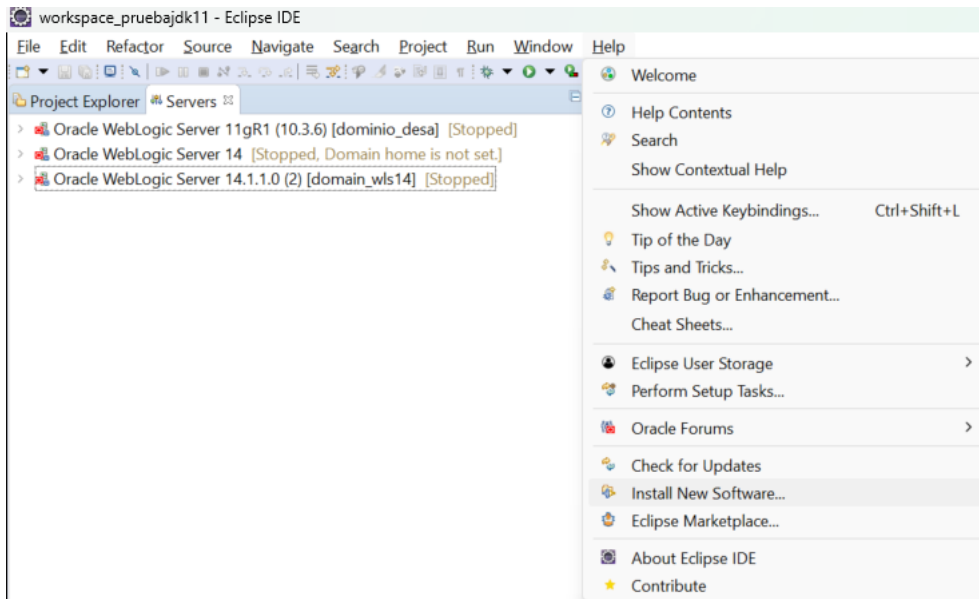


Una vez se haya reiniciado, hay que eliminar de la carpeta plugins de Eclipse **com.ejje.uda_XXXX.jar** manualmente:

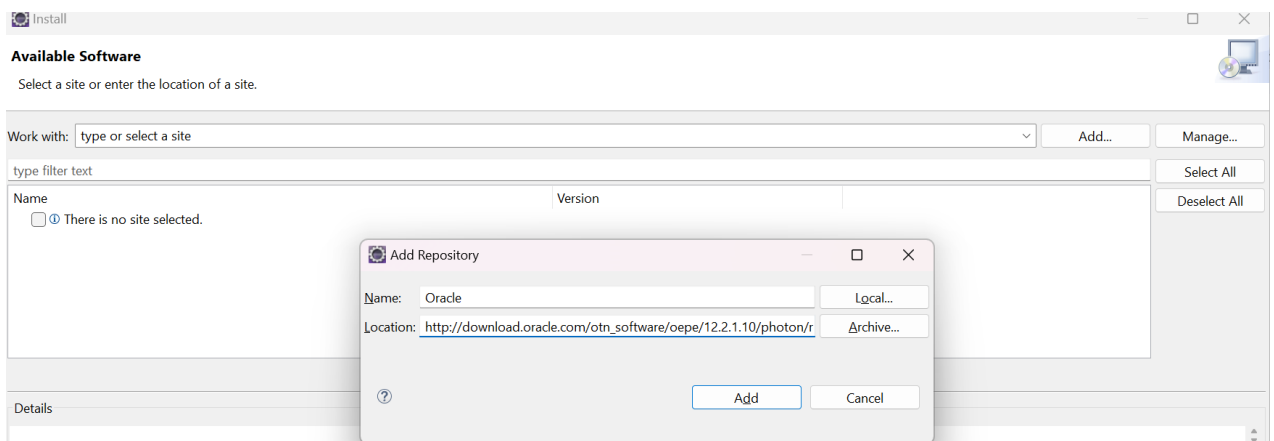


2.4.5 Integración del servidor de aplicaciones con Eclipse

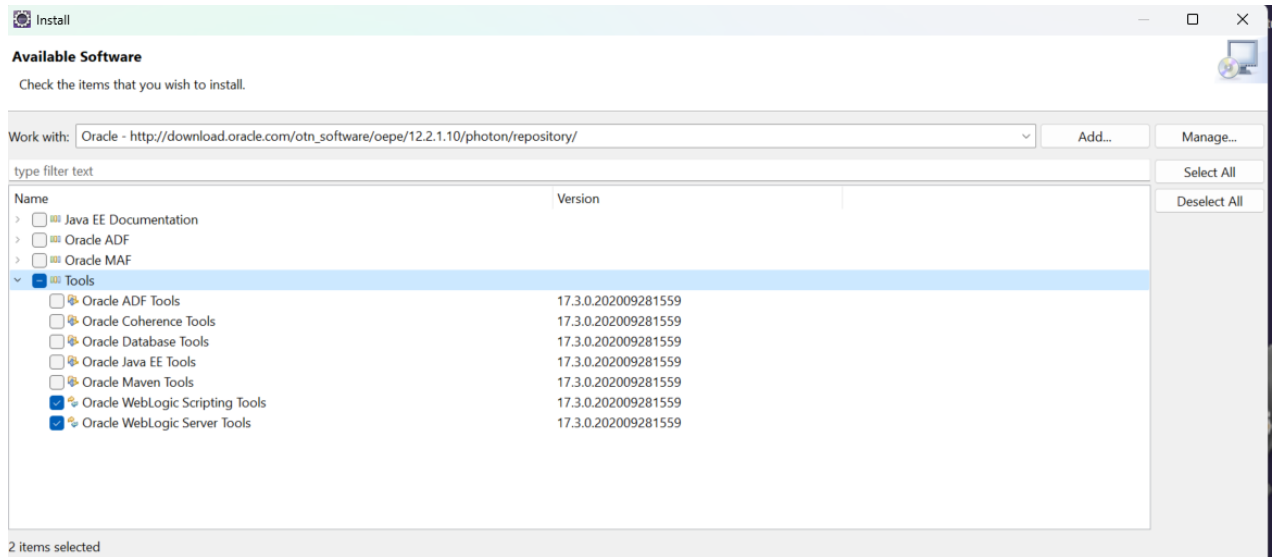
Antes de lanzar un servidor WebLogic 14 con JDK si se está usando un eclipse personal o nuevo es necesario tener instalado un OEPE de la siguiente forma:



Se añade un nuevo software, poniendo “Oracle” en el “Name” y la siguiente URL “http://download.oracle.com/otn_software/oepe/12.2.1.10/photon/repository/” en “Location”.

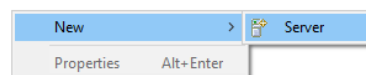
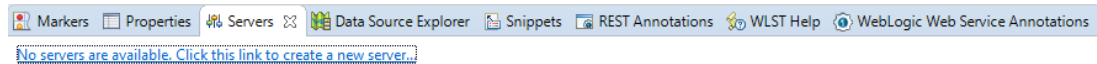


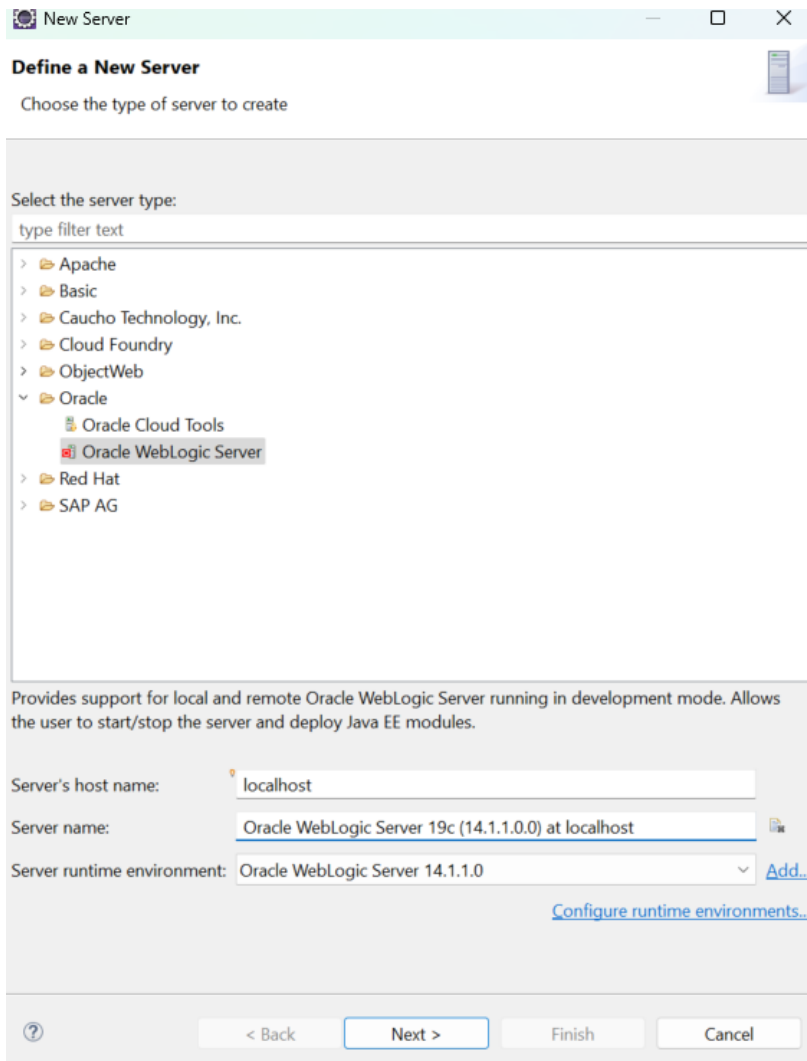
En el apartado de Tools basta con seleccionar “Oracle WebLogic Scripting Tools” y “Oracle WebLogic Server Tools”, con esto sería suficiente para poder seleccionar la JDK 11.



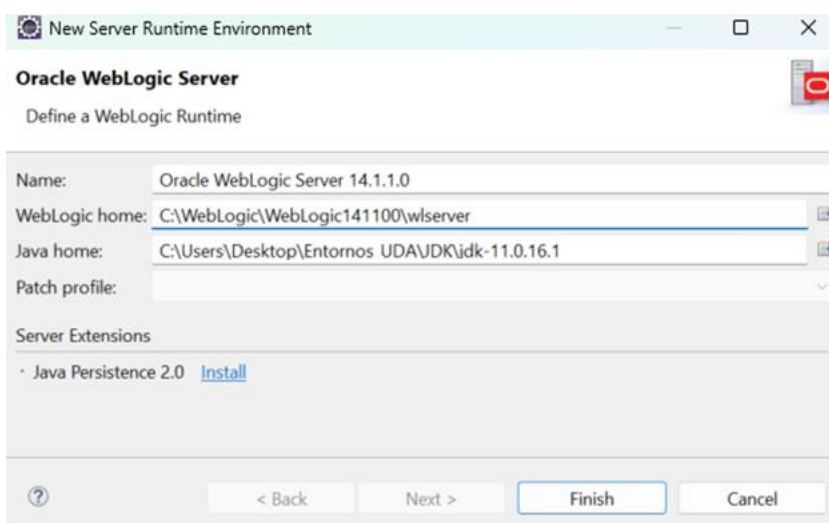
Debido a que el Eclipse proporcionado tiene las Oracle WebLogic Server Tools instaladas, el trabajo de integración de las aplicaciones con WebLogic Server se resume a:

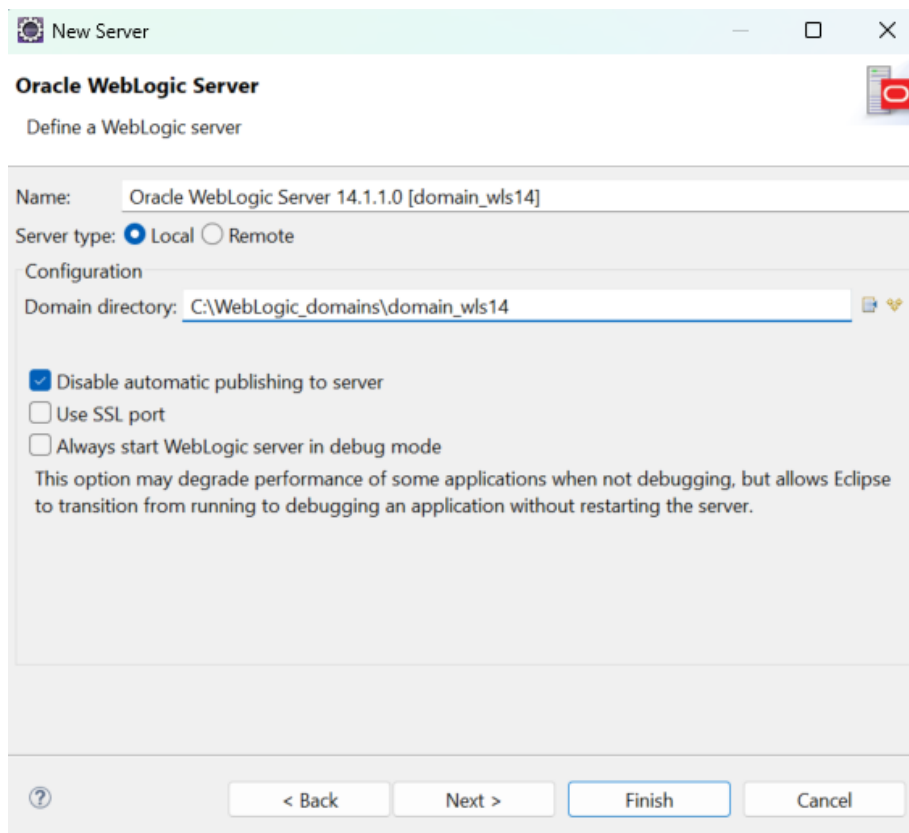
Desde Eclipse, ir a la pestaña “Servers” y seleccionar con el botón derecho *New > Server*. Añadir uno nuevo que sea del tipo **Oracle WebLogic Server**.





Se añade un nuevo “Server runtime enviroment”:





2.4.6 Workspace

Los proyectos se crearán en el workspace del eclipse. No se especifica una ubicación por defecto, pero se recomienda usar /aplic

2.4.7 Configuración de las aplicaciones y uso de librerías

Existen tres carpetas base en la unidad de disco C:\ que intervienen en el desarrollo de las aplicaciones:

- config: Contiene una carpeta por cada aplicación con sus ficheros de configuración dependientes del entorno:
 - c:\config\dominio_desa\bbb\
 - bbb.properties
- datos: Contiene las trazas generados por la aplicación:
 - c:\datos\bbb\log
- usr, con las librerías necesarias para la compilación de las aplicaciones:
 - c:\usr\

Para el uso de los ficheros de configuración en \config\dominio_desa\, se añade al script de arranque de WebLogic Server su ruta, tal y como se hizo anteriormente con las librerías de XLNetS. La variable %CONFIG% sirve para establecer una base de ficheros de configuración para todas las aplicaciones.

A continuación, se muestran las líneas añadidas al script startWeblogic.cmd:

```
...
@REM ##### FICHEROS DE CONFIGURACIÓN #####

set CONFIG = %UNIDAD_USR%/config/dominio_desa
.....

set EXT_PRE_CLASSPATH=%N38_LIB%;%CONFIG%;
```

2.4.8 Errores detectados

Error en función del datasource de conexión

WebLogic Server 14.1.1.0.0 se entrega con el controlador Oracle 14.1.1.0.0. Al conectarse Weblogic Server con este controlador a una base de datos que no tiene GRID da error. Y para ello se recomienda deshabilitar el FAN (Fast Application Notification)

Más información:

<https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/jdbca/using-active-gridlink-data-sources.html>

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Doracle.jdbc.fanEnabled=false -
Dweblogic.configuration.schemaValidationEnabled=false
```

Error en la carga e instanciación de beans de la aplicación.

CAUSE

1. Implicit CDI Bean scanning is enabled in WLS 12.2.1.2.0 by default, as per the CDI 1.1 specification. This can be disabled.
2. Customer's code-base/shared lib has some components that require CDI dependency and some do not. It is better to refactor the codes that has, for example, an annotation @Resource containing a java:module to corresponding EJB codes out of the shared lib/code-base component.

SOLUTION

1. Turn off CDI in the library jar via beans.xml. If there is not a beans.xml in the file then add one like this one:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/beans_1_1.xsd"
version="1.1" bean-discovery-mode="none" />
```

2. Disable implicit cdi for the application but not all of cdi. I.e. the cdi-descriptor in the weblogic-application.xml file should contain a section like below:

```
<cdi-descriptor> <wls:implicit-bean-discovery-enabled>false</wls:implicit-bean-discovery-enabled> </cdi-descriptor>
```

3. For base code/shared lib that needs CDI dependency, adjust/refactor the codes with the field, for example, refactor an annotation @Resource containing a java:module referenced by EJB codes to the corresponding EJB codes/component out of the base code/shared lib class.

REFERENCES

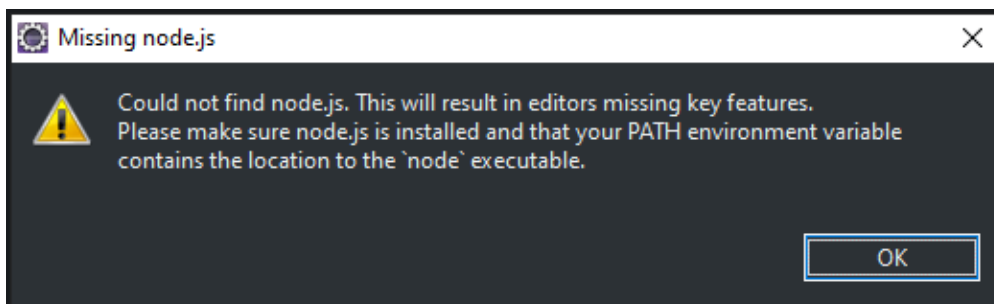
[BUG:27186441](#) - POJOANNOTATIONPROCESSINGFLOW CAUSES ERROR-JAVA:COMP AND JAVA:MODULE CAN NOT BE
[NOTE:2146096.1](#) - Dependency Injection Fails when Deploying Application with Multiple EJB Modules on WLS 12.2.1
[NOTE:2149796.1](#) - Jersey 1.1 Style Servlet-Mapping Ignored and Unable to Start WLS 12.2.1 with "CDI definition failure:java.lang.NullPointerException"

Didn't find what you are looking for? [Ask in Community...](#)

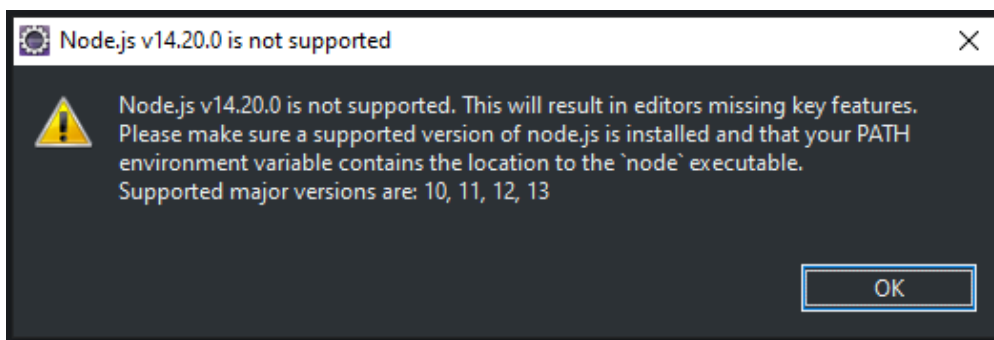
```
@REM ### weblogic.management.DeploymentException: java:comp and java:module can not be
applied in class of the EAR scope. ###
```

```
<wls:cdi-descriptor>
<wls:implicit-bean-discovery-enabled>false</wls:implicit-bean-discovery-enabled>
<wls:policy>Disabled</wls:policy>
</wls:cdi-descriptor>
```


Aviso de eclipse



Si node.js no está instalado, Eclipse mostrará una alerta cada vez que se abra un workspace que contenga proyectos. La solución es sencilla, instalar cualquiera de las versiones que soporta (10, 11, 12 o 13), aunque [se recomienda la 12](#) porque de las cuatro opciones, es la que más actualizada está a pesar de que desde abril del 2022 no haya recibido más actualizaciones por estar descontinuada.



Si no hay una versión de node.js 10, 11, 12 o 13 instalada, Eclipse mostrará una alerta cada vez que se abra un workspace que contenga proyectos. La solución es sencilla, instalar cualquiera de las versiones que solicita, aunque [se recomienda la 12](#) porque de las cuatro opciones, es la que más actualizada está a pesar de que desde abril del 2022 no haya recibido más actualizaciones por estar descontinuada.

3 Estructura de directorios de aplicaciones en PC local

La estructura de directorios en PC local ha sido orientada al proceso de desarrollo con el IDE de Eclipse.

Por otra parte, los desarrolladores contarán con un Repositorio centralizado de código y gestión de versiones SVN, que les permitirá trabajar desde los PCs de forma coordinada.

En el workspace se encontrarán los distintos proyectos de Eclipse:

- Java Projects – bbbEARClasses (como clases comunes)
- Web Application Projects – bbbNombreWAR
- Enterprise Java Beans – bbbModuloEJB
- Enterprise Application Project – bbbEAR

3.1 Estructura en PC local

Los campos seleccionados en la columna Estructura de SVN los almacenará el plug-in Subversive de forma automática en el repositorio.

Se refleja la estructura de desarrollo para una aplicación común.

Directorio PC	Descripción	Estr. SVN
/aplic/bbb		
/bbbEARClasses	Existirá 1 proyecto de tipo 'Java Project' que incluirá las clases y recursos comunes.	✓
Ficheros y carpetas ocultos formato: .fichero (ej. .classpath , .project , ...) .carpeta (.settings)	Son ficheros y carpetas de configuración del proyecto	✓
/bin	Incluirá las clases compiladas (.class)	
/lib	Incluirá las librerías de la propia aplicación (.jar)	✓
/src	Incluirá las carpetas con las fuentes de las clases generales (.java).	✓
/test-unit	Incluirá las carpetas con las fuentes de las pruebas unitarias (.java).	✓
/test-integration	Incluirá las carpetas con las fuentes de las pruebas de integración (.java).	✓
/test-system	Incluirá las carpetas con las fuentes de las pruebas de sistema (.java).	✓

/resources	Incluirá los ficheros de configuración (.properties) no dependientes del entorno, así como recursos varios de la aplicación (.xsl, .xml, .fo, etc.).	✓
\bbbNombreWAR	Existirán n proyectos de tipo 'Web Project' que incluirán los componentes web (XHTMLs, Servlets, ...). <i>Nota: Si sólo existiera un único proyecto WAR en la aplicación, su nombre quedaría de la forma: bbbWAR</i>	✓
Ficheros y carpetas ocultos formato: .fichero (ej. .classpath, .project, ...) .carpeta (.settings)	Son ficheros y carpetas de configuración del proyecto	✓
\src	Incluirá las carpetas con las fuentes de las clases del módulo web y de los servlets (.java)	✓
\META-INF	Fichero 'persistence.xml'	✓
\test-unit	Incluirá las carpetas con las fuentes de las pruebas unitarias (.java).	✓
\test-integration	Incluirá las carpetas con las fuentes de las pruebas de integración (.java).	✓
\test-system	Incluirá las carpetas con las fuentes de las pruebas de sistema (.java).	✓
\resources	Incluirá los ficheros de configuración, internacionalización, etc (los .properties) no dependientes del entorno, así como los recursos varios de la aplicación (.xml, etc.)	✓
/build	Se crea por defecto al crear un nuevo War con Eclipse. (para que incluya las clases compiladas)	
\WebContent		✓
\templates	Incluirá las plantillas XHTML o JSP para la maquetación de las páginas de la aplicación con facelets o tiles.	✓
\bbbdirectorio_de_xhtmlXHTML	Carpeta con las fuentes de los XHTML (.java)	✓
\bbbdirectorio_de_jspJSP	Carpeta con las fuentes de los JSP (.java)	✓
\META-INF	Incluirá el 'MANIFEST.MF'	✓
\WEB-INF	Ficheros 'web.xml' y 'weblogic.xml'	✓
\classes	Incluirá las clases compiladas (.class)	

\lib	Directorio de librerías de War	✓
/bbbNombreEJB	Existirán m proyectos de tipo <i>'EJB Project'</i> que incluirán componentes EJBs	✓
Ficheros y carpetas ocultos formato: .fichero (ej. .classpath , .project , ...) .carpeta (.settings)	Son ficheros y carpetas de configuración del proyecto	✓
/build	Incluirá las clases compiladas (.class)	
/ejbModule		✓
/bbbModuloEJB	carpetas con las fuentes de los EJBs (.java)	✓
/META-INF	Ficheros <i>"ejb-jar.xml"</i> , <i>"weblogic-ejb-jar.xml"</i> y <i>"MANIFEST.MF"</i>	✓
/test-unit	Incluirá las carpetas con las fuentes de las pruebas unitarias (.java).	✓
/test-integration	Incluirá las carpetas con las fuentes de las pruebas de integración (.java).	✓
/test-system	Incluirá las carpetas con las fuentes de las pruebas de sistema (.java).	✓
/resources	Incluirá los ficheros de configuraciónno dependientes del entorno, así como los recursos varios de la aplicación (.properties, .xml, etc.)	
\bbbEAR	Existirá 1 proyecto de tipo <i>'Enterprise Application Project'</i> que será el nexo de unión de todos los anteriores	✓
Ficheros y carpetas ocultos formato: .fichero (ej. .classpath , .project , ...) .carpeta (.settings)	Son ficheros y carpetas de configuración del proyecto	✓
\EarContent		✓
\APP-INF		✓
\classes	Incluirá las clases compiladas	
\lib	Directorio de librerías de EAR	✓

\META-INF	Con los ficheros <i>'application.xml'</i> y <i>'weblogic-application.xml'</i> del EAR	✓
------------------	---	---

Se refleja la estructura de desarrollo para una aplicación de tipo librería (idéntica a bbbEARClasses):

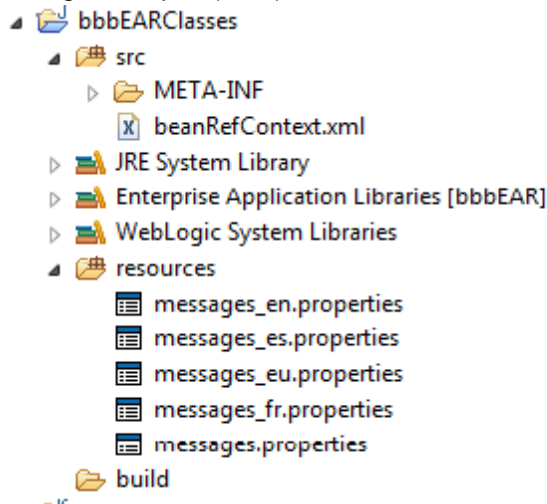
Directorio PC	Descripción	Estr. Servidor
c:\aplic\bbb		
\bbbShLibClasses	Existirán un proyecto de tipo <i>'Java Project'</i> para cada librería.	✓
Ficheros y carpetas ocultos formato: .fichero (ej. .classpath, .project, ...) .carpeta (.settings)	Son ficheros y carpetas de configuración del proyecto	✓
/bin	Incluirá las clases compiladas (.class)	
/lib	Incluirá las librerías de la propia aplicación (.jar)	✓
/src	Incluirá las carpetas con las fuentes de las clases generales (.java).	✓
/test-unit	Incluirá las carpetas con las fuentes de las pruebas unitarias (.java).	✓
/test-integration	Incluirá las carpetas con las fuentes de las pruebas de integración (.java).	✓
/test-system	Incluirá las carpetas con las fuentes de las pruebas de sistema (.java).	✓
/resources	Incluirá los ficheros de configuración (.properties) no dependientes del entorno, así como recursos varios de la aplicación (.xsl, .xml, .fo, etc.).	✓

La estructura de directorios de PC, 'filesystem', es ligeramente distinta a la Visión Lógica de proyectos que proporciona la herramienta Eclipse, ya que, entre otras cosas, Eclipse no muestra las carpetas de archivos generados (classes). En el repositorio de subversión se almacenan prácticamente todos los archivos y directorios exceptuando los compilados y las librerías.

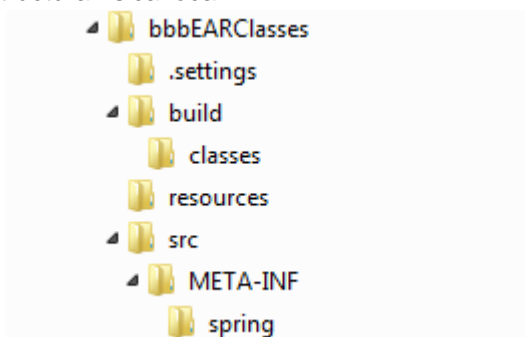
A continuación, se muestra la estructura de directorios del aplicativo en local (visión física y lógica de Eclipse).

3.1.1 Proyecto Classes (bbbEARClasses)

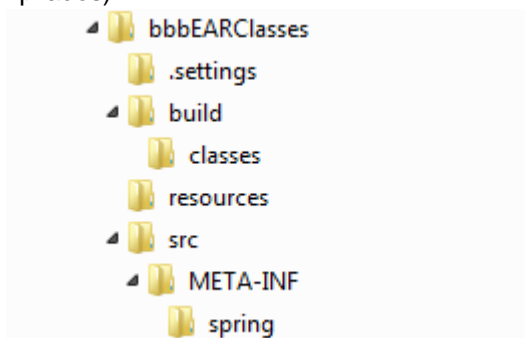
1 Visión lógica Eclipse (local):



2 Estructura física local:

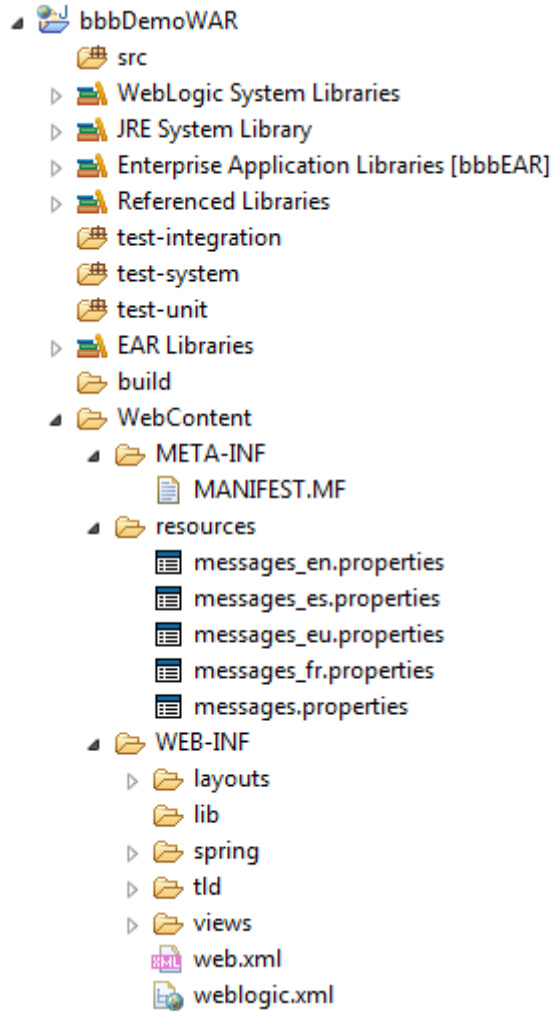


3 Estructura SVN (misma estructura que local contiendo todos los ficheros menos la carpeta con los compilados):

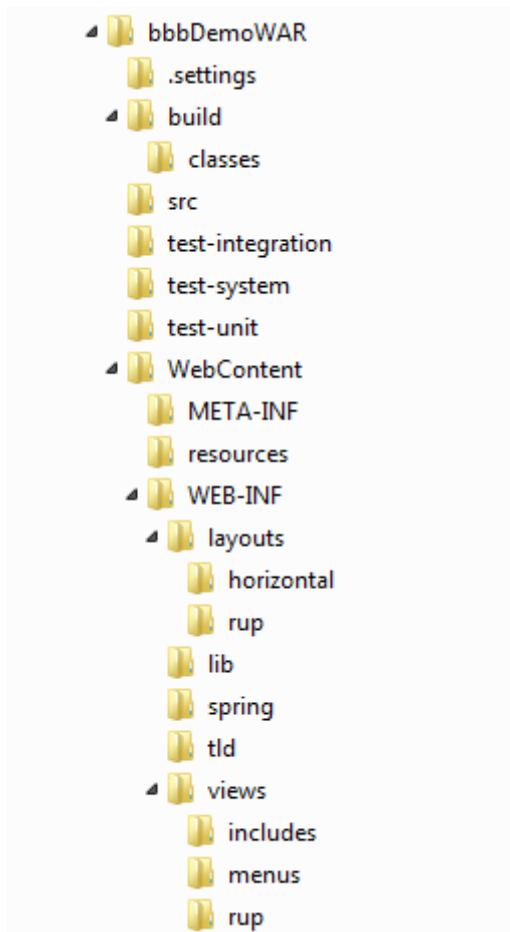


3.1.2 Proyecto bbbNombreWAR

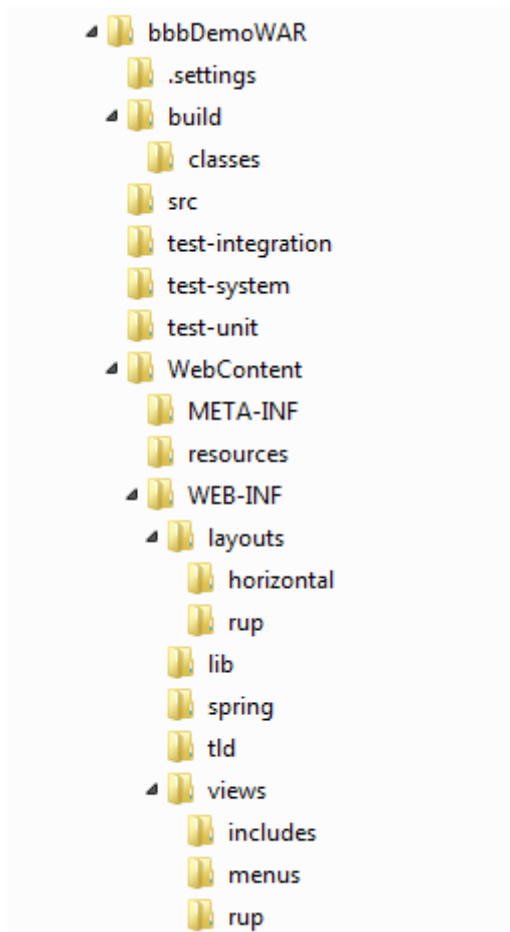
4 Visión lógica Eclipse (local):



5 Estructura física local:

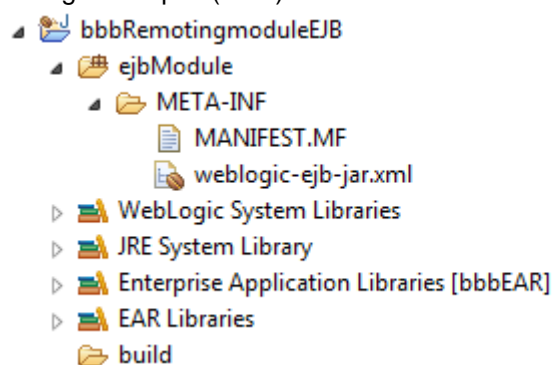


- 6 Estructura SVN (misma estructura que local contiendo todos los ficheros menos la carpeta con los compilados)

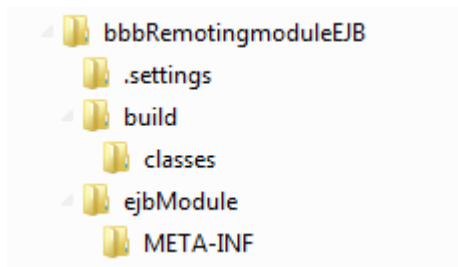


3.1.3 Proyecto bbbModuloEJB

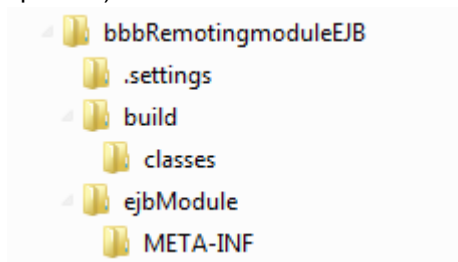
7 Visión lógica Eclipse (local):



8 Estructura física local:

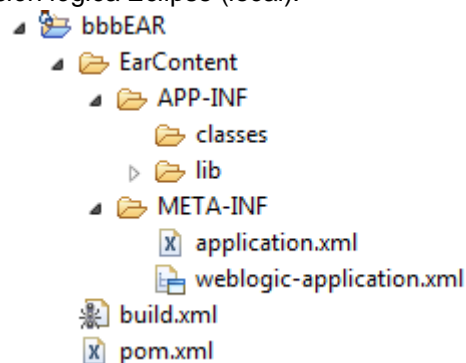


- 9 Estructura SVN (misma estructura que local contiendo todos los ficheros menos la carpeta con los compilados):

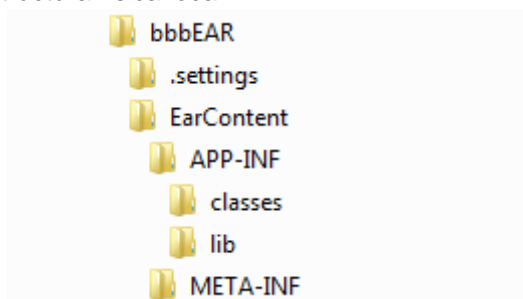


3.1.4 Proyecto bbbEAR

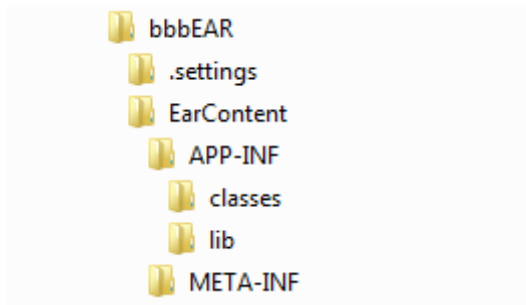
- 10 Visión lógica Eclipse (local):



- 11 Estructura física local:



- 12 Estructura SVN (misma estructura que local contiendo todos los ficheros menos la carpeta con los compilados):



3.1.5 Estructura en Servidor de Desarrollo

Se refleja ahora la misma estructura de directorios de desarrollo, esta vez en el Servidor de Desarrollo:

/aplic

```
>--- bbb
    +--- ant
    +--- dist
    +--- bin
    >--- srctest
        >--- unit
        >--- integration
        >--- system
    >--- bintest
        >--- unit
        >--- integration
        >--- system
    >--- src
        >--- bbbEAR
    >--- bbbEARClasses
        src
            --- .java
        >--- ejbs
            >--- bbbNombre1EJB
            >--- src
                --- .java
            >--- resources
            >--- META-INF
                --- ejb-jar.xml
                --- weblogic-ejb-jar.xml
            >--- bbbNombreNEJB
```

```

>---src
    --- .java
>--- resources
>--- META-INF
    --- ejb-jar.xml
    --- weblogic-ejb-jar.xml
>--- wars
>--- bbbNombre1War
    >--- WebContent
    >--- resources
    >--- WEB-INF
    >--- classes
    >-- lib
    --- web.xml
    --- weblogic.xml
>--- bbbNombreNWar
>--- WebContent
>--- resources
    >--- WEB-INF
    >--- classes
    >-- lib
    --- web.xml
    --- weblogic.xml
>--- META-INF
    --- application.xml
    --- weblogic-application.xml
>-- APP-INF
>--- lib
    >-- resources

>--- bbbShLibClasses
src
    --- .java
resources
lib
META-INF
MANIFEST.MF

```

3.1.6 Correspondencia de aplicaciones entre PC local y Servidor de Desarrollo

Directorio Local	Directorio Servidor
c:\aplic\bbb\bbbEAR\EarContent\META-INF application.xml weblogic-application.xml	/aplic/bbb/src/bbbEAR/META-INF application.xml weblogic-application.xml
c:\aplic\bbb\bbbEAR\EarContent\APP-INF\lib	/aplic/bbb/src/bbbEAR/APP-INF/lib
c:\aplic\bbb\bbbEARClasses\src	/aplic/bbb/src/bbbEAR/bbbEARClasses/src
c:\aplic\bbb\bbbEARClasses\test-unit	/aplic/bbb/src/test/unit
c:\aplic\bbb\bbbEARClasses\test-integration	/aplic/bbb/src/test/integration
c:\aplic\bbb\bbbEARClasses\test-system	/aplic/bbb/src/test/system
c:\aplic\bbb\bbbEARClasses\resources	/aplic/bbb/src/bbbEAR/resources
c:\aplic\bbb\bbbEARClasses\lib	/aplic/bbb/src/bbbEARClasses/lib
c:\aplic\bbb\bbbNOMBREWar\WebContent\ xhtml, *.jsp, *.jsf	/aplic/bbb/src/wars/bbbNOMBREWar/WebContent/ *.xhtml, *.jsp, *.jsf
c:\aplic\bbb\bbbNOMBREWar\WebContent\WEB-INF\ web.xml weblogic.xml	/aplic/bbb/src/wars/bbbNOMBREWar/WebContent\WEB-INF/ web.xml weblogic.xml
c:\aplic\bbb\bbbNOMBREWar\WebContent\META-INF	/aplic/bbb/src/wars/bbbNOMBREWar/WebContent\META-INF
c:\aplic\bbb\bbbNOMBREWar\WebContent\WEB-INF\lib	/aplic/bbb/src/wars/bbbNOMBREWar/WebContent\WEB-INF/lib
c:\aplic\bbb\bbbNOMBREWar\src	/aplic/bbb/src/wars/bbbNOMBREWar/WebContent\WEB-INF/classes

c:\aplic\bbb\bbb NOMBREWar \test-unit	/aplic/bbb/src/test/unit
c:\aplic\bbb\bbb NOMBREWar \test-integration	/aplic/bbb/src/test/integration
c:\aplic\bbb\bbb NOMBREWar \test-system	/aplic/bbb/src/test/system
c:\aplic\bbb\bbb NOMBREWar \resources	/aplic/bbb/src//warsbbbNOMBREWAR/resources
c:\aplic\bbb\bbb MODULO EJB \resources	/aplic/bbb/src/ejbs/bbbEAR/bbbMODULO EJB /resources
c:\aplic\bbb\bbb MODULO EJB \ejbModule	/aplic/bbb/src/ejbs/bbbEAR/bbbMODULO EJB /ejbModule
c:\aplic\bbb\bbb MODULO EJB \test-unit	/aplic/bbb/src/test/unit
c:\aplic\bbb\bbb MODULO EJB \test-integration	/aplic/bbb/src/test/integration
c:\aplic\bbb\bbb MODULO EJB \test-system	/aplic/bbb/src/test/system
c:\aplic\bbb\bbb MODULO EJB \ejbModule\META-INF\ ejb-jar.xml weblogic-ejb-jar.xml	/aplic/bbb/src//ejbsbbbMODULO EJB /ejbModule/META-INF/ ejb-jar.xml weblogic-ejb-jar.xml
c:\aplic\bbb\bbb ShLibClasses \src	/aplic/bbb/src/bbb EAR /bbb ShLibClasses /src
c:\aplic\bbb\bbb ShLibClasses \resources	/aplic/bbb/src/bbb EAR /bbb ShLibClasses /resources
c:\aplic\bbb\bbb ShLibClasses \lib	/aplic/bbb/src/bbb EAR /bbb ShLibClasses /bbbEARClasses/lib
c:\aplic\bbb\bbb ShLibClasses \test-unit	/aplic/bbb/src/test/unit
c:\aplic\bbb\bbb ShLibClasses \test-integration	/aplic/bbb/src/test/integration
c:\aplic\bbb\bbb ShLibClasses \test-system	/aplic/bbb/src/test/system