Índice

1 Instalación y puesta en marcha del servidor	2
2 Proyecto cliente	3
3 Creación del enterprise bean Saludo	5
4 Prueba de los clientes del bean	8
5 (*) Creación de la aplicación Web	8
6 (*) Creación y despliegue del EAR	10

En esta sesión de ejercicios vamos a crear el mismo ejb de sesión que en la clase pasada, pero utilizando el entorno de desarrollo BEA Workshop y el servidor de aplicaciones BEA WebLogic.

WebLogic permite, ya en la versión 2.1, el uso de anotaciones propias en el fichero de implementación del enterprise bean. Estas anotaciones son procesdas por las herramientas de despliegue de WebLogic y el EJB se construye de forma automática a partir de ellas. La especificación EJB 3.0 utiliza una estrategia similar, pero con anotaciones estándar.

Pasar de la forma de trabajar con WebLogic a la especificación 3.0 es inmediato.

1. Instalación y puesta en marcha del servidor

1. Lanza el Workshop y crea el espacio de trabajo Workshop en el directorio que desees. Ese va a ser el espacio de trabajo en el que vas a desarrollar los proyectos en lo que queda de módulo.

2. Añade el *Runtime* de BEA Weblogic 9.2 con la opción "Window > Preferences > Server > Installed Runtimes > Add...". Selecciona el runtime que aparece:

W Preferences			
type filter text 🗨	Installed Server Runtime	e Environments	↓ ↓ ↓ ↓
 General Ant Data File Templates 	Add, remove, or edit installed se The checked runtime will be used Installed server runtimes:	rver runtime definitions. I by default when creating new	projects.
. Help	Name	Туре	Add
 Install/Update Internet Java NetUI Page Flow 	EA WebLogic v9.2	BEA WebLogic v9.2	Edit Remove
NetUI Page Flow Plug-in Development Run/Debug Server			Search
		ОК	Cancel

3. Una vez añadido el runtime, pulsa el botón derecho en el panel de servidores para añadir un nuevo servidor ("Botón derecho > New > Server"). Introduce el dominio "ejb" que creaste

Copyright © 2006-2007 Depto. CCIA All rights reserved.

en los ejercicios anteriores ("Next > Enter Domain home (Browse...) C:\bea\user_projects\domains\ejb"). Pulsa "Finish".

Pulsa en el botón "Play" para poner en marcha el servidor. Debe quedar en estado de "Started":

			41 4	-	17 A	_	
Problems M Servers X Javado	: Tasks Console JUnit		\$F (2	\$ ⁰ ~ %		<u> </u>
Server	Status	State					
G BEA WebLogic v9.2 Server [ejb]	🔒 Started	Republish					
				_			

Aparecerá un terminal en el que se muestra la salida estándar del servidor de aplicaciones:

		_
🕰 WebLogic Server - 9.2.1.0 [ejb]	- 🗆	×
s opened. All server side log events will be written to this file.		•
<22-mar-2007 06H12' CET> <notice> <security> <bea-090082> <security initial<="" p=""></security></bea-090082></security></notice>	izing	-
using security realm more alm. $>$		
<pre><22-mar-2007 06H12' CET> <notice> <weblogicserver> <bea-000365> <server pre="" sta<=""></server></bea-000365></weblogicserver></notice></pre>	te ch	
anged to STANDRY>		
<22-mar-2007 06H12' CET> <notice> <weblogicserver> <bea-000365> <server p="" sta<=""></server></bea-000365></weblogicserver></notice>	te ch	
anged to STABTING>		
K22-mar-2007 06H13' CET> <notice> <log management=""> <bea-170027> <the p="" server<=""></the></bea-170027></log></notice>	init	
ialized the domain log broadcaster successfully. Log messages will now be b	roadc	
asted to the domain log.>		
K22-mar-2007 06H13' CET> <notice> <weblogicserver> <bea-000365> <server p="" sta<=""></server></bea-000365></weblogicserver></notice>	te ch	
anged to ADMIN>		
<22-mar-2007 06H13' CET> <notice> <weblogicserver> <bea-000365> <server p="" sta<=""></server></bea-000365></weblogicserver></notice>	te ch	
anged to RESUMING>		
<22-mar-2007 06H13' CET> <notice> <server> <bea-002613> <channel "default[1<="" td=""><td>]" is</td><td></td></channel></bea-002613></server></notice>]" is	
now listening on 127.0.0.1:7001 for protocols iiop, t3, ldap, http.>		
<pre><22-mar-2007 06H13' CET> <notice> <server> <bea-002613> <channel "default"<="" pre=""></channel></bea-002613></server></notice></pre>	is no	
w listening on 10.211.55.3:7001 for protocols iiop, t3, ldap, http.>		
<22-mar-2007 06H13' CET> <notice> <weblogicserver> <bea-000331> <Štarted We</bea-000331></weblogicserver></notice>	bLogi	
c Admin Server "AdminServer" for domain [®] ejb" running in Development Mode>	_	
<22-mar-2007 06H13' CET> <notice> <weblogicserver> <bea-000365> <server p="" sta<=""></server></bea-000365></weblogicserver></notice>	te ch	
anged to RUNNING>		
<22-mar-2007 06H13' CET> <notice> <weblogicserver> <bea-000360> <server p="" sta<=""></server></bea-000360></weblogicserver></notice>	rted	
in RUNNING mode>		

2. Proyecto cliente

Vamos ahora a importar el proyecto cliente de la sesión anterior (creado con Eclipse) al entorno de trabajo Workhsop.

1. Copia la carpeta ejb-cliente-saludo de la carpeta donde está el espacio de trabajo de Eclipse a la carpeta donde has creado el espacio de trabajo de Workshop.

2. Ya en Workshop, crea un nuevo proyecto java ("New > Project... > Java > Java Project"). Introduce como nombre del proyecto el mismo del proyecto que has copiado (ejb-cliente-saludo). Aparecerá un mensaje en el que el Workshop te informa de que el proyecto ya existe y que va a intentar configurar el build path. Pulsa "Finish" y cambia a la perspectiva de Java. El entorno mostrará el siguiente aspecto:

File Edit Source Refactor Navigate Search Project Run Window Help Image: Search Project Run Kindow He	W Java - BEA Workshop for WebLogic Platform				
Package Explore	File Edit Source Refactor Navigate Search Project Run Window Help				
Padage Explorer X Herarchy Couline S Couline] 🗈 • 🖫 🗁] 🎄 • Ø • 隆 •] 🖑 📽 •] 🥭 🖋] 🥹] *	⇔ • ⇔ •		😭 🐉 Java	**
Providence satures Providence sature	📕 Package Explorer 🗙 Hierarchy 🗇 🗘 🕞 🤹 🍟 🗖		🗖 🗖 🗄 Outlin	ne 🛛	- 8
Zerrors, U warnings, U infos Description Description Without a service annot be built until build path errors are resolved who und asspath container: 'org.edipse.jdt.junt.JUNIT_CONTAINER/3.8.1' elb-clente	Bit electrote-saudo B: electrote-saudo <td< td=""><td>Problems 8 Javador Decla</td><td>An outline</td><td>is not available.</td><td></td></td<>	Problems 8 Javador Decla	An outline	is not available.	
Description Resource In Folder Image: The project cannot be built until build path errors are resolved ejb-clente ejb-clente Unbound dasspath container: 'org.edipse.jdt.junit.JUNIT_CONTAINER/3.8.1'I ejb-clente ejb-clente Unbound dasspath container: 'org.edipse.jdt.junit.JUNIT_CONTAINER/3.8.1'I ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente ejb-clente Image: The project cannot be built until build path errors are resolved. ejb-clente.		2 errors, 0 warnings, 0 infos			
		Description		Resource	In Folder
Unbound dasspath container: 'org.edipse.jdt.junt.JUNIT_CONTAINER/3.8.1'1 etb-cliente		O The project cannot be built until I	ouild path errors are resolved	ejb-cliente	
			googoo jarijan Jone Loon Analysis i r		

3. Aparece un error debido a que no se encuentran las librerías de JUnit. Haz que Workshop las instale automáticamente creando un nuevo test y cancelando su creación después de haber añadido la librería al build path. El aspecto del proyecto debe ser el siguiente:



4. Prueba ahora a ejecutar el cliente y el test de JUnit. Verás que aparece un error, debido a que el enterprise bean no está desplegado en el servidor.



Aviso:

Puede ser que el bean haya quedado desplegado de la sesión anterior. En ese caso, elimínalo utilizando la consola de administración.

3. Creación del enterprise bean Saludo

Una vez instalado el servidor de aplicaciones y el proyecto cliente vamos a crear y desplegar el bean Saludo.

1. Comenzamos creando un proyecto EJB WebLogic ("Botón derecho en el panel del explorador de proyectos > New > Project... > EJB > WebLogic EJB Project"). Escribimos como nombre de proyecto ejb-saludo terminamos ("Next > Finish"). El entorno cambia a la perspectiva Workshop.

2. Creamos el paquete es.ua.jtech.ejb.beans y en él creamos un WebLogic SessionBean ("Botón derecho sobre el paquete > New > WebLogic SessionBean"). Ponemos como nombre de fichero SaludoBean:



3. Vamos a escribir el código del bean de sesión. Será el mismo que tenía el fichero de implementación del proyecto Eclipse, añadiéndole algunas anotaciones. Abre el fichero SaludoBean. java del proyecto Eclipse ("File > Open File > ...") y copia la definición del array de saludos y los métodos saluda() y getSaludo(). No tienes que copiar los métodos vacíos que implementan la interfaz SessionBean porque ya están implementados en la clase GenericSessionBean extendida por la clase SaludoBean.

4. Arregla los errores del build path importando las clases necesarias (java.util.Date y java.util.ArrayList).

5. Crea la clase SaludoTO en el mismo paquete que el bean y copia su contenido del fichero del proyecto Eclipse.

6. Vamos ahora a modificar las anotaciones, para adaptarlas a los nombres que usamos en las sesiones anteriores. En primer lugar, en la clase SaludoBean, debe aparecer como nombre remoto JNDI SaludoBean:

```
@JndiName(remote="SaludoBean")
```

Después, los nombres de las interfaces remota y home debemos cambiarlos a "Saludo" y "SaludoHome":

@FileGeneration(...

remoteClassName="Saludo",
remoteHomeName="SaludoHome")

Por último, anota los métodos saluda() y getSaludo() con la anotación @RemoteMethod()

Con esto ya está definido el enterprise bean.

7. Antes de desplegar el bean, vamos a definir el nombre del fichero cliente ejbjar necesario para acceder al bean desde un cliente remoto. Pulsa "Botón derecho en ejb-saludo > Properties > WebLogic EJB"). Quita el *tic* "Use default" y escribe el nombre ejb-saludo-clientjar.jar.

👿 Properties for ejb-sal	ludo	
type filter text 💌	WebLogic EJB 🗢	• <> -
Info BeanInfo Path Builders 2016 Java Build Path Java Build Path 3 Java Code Style Java Congiler Java Congiler Java Congiler Aava Congiler Profile Coordian Profile Coordian Profile Coordian Profile Coordiant Server Task Tags Validation WebLog(EIB XDCdet XMLBeans	EJBC Flags: V Us Jar settings - create tables: CREATE_ONLY V US Jar settings - EJB client jar: ejb-saludo-clientjar.jar V Us Jar settings - enable bean class redeploy: WASPECIFIED V V Us Jar settings - disable warning: V US Jar settings - disable warning: V US NOTE: Project-level EJB preferences are only used during the IDE build. For exported build script. He @JarSettings annotation must be used and EJBC flags must be adder to the Ant build script.	e default e default e default e default e default Ant d
<	Restore Defaults A	Apply
	ОК Са	ancel

8. Vamos ya a desplegar el bean. Pulsa en el servidor BEA WebLogic con el botón derecho y escoge la opción "Add and Remove Projects...". Verás el proyecto ejb-saludo. Añádelo y finaliza el despliegue. El entorno debe quedar con el siguiente aspecto:



9. Abre la consola de administración y comprueba que el bean está deplegado.

4. Prueba de los clientes del bean

1. Actualiza la librería clientjar del bean. Para ello borra la librería

ejb-saludo-clientjar.jar del proyecto cliente con la opción "Botón derecho > Build Path > Remove from Build Path". Copia el fichero ejb-saludo-clientjar.jar del proyecto EJB al proyecto cliente y añádela al build path ("Botón derecho > Build Path > Add to Build Path").

2. Limpia el proyecto cliente y prueba la aplicación cliente y los tests de JUnit.

5. (*) Creación de la aplicación Web

Vamos ahora a crear la aplicación web que se conecta al bean de forma local. Para ello deberemos:

- Crear el proyecto Web
- Modificar el proyecto EJB para añadir las interfaces locales al bean

Vamos a ello.

1. Crea un nuevo proyecto Web ("New > Projet... > Web > Dynamic Web Project") con el nombre de ejb-saludoWar. Pulsa "Next" y deja marcadas sólo las opciones "Dynamic Web Module" y "Java" (ver figura). Pulsa "Finish" para finalizar.

🐨 New Dynamic Web Project	
Select Project Facets Facets are units of functionality that can be added to a pr	roject.
Presets: <a href="https://www.customscore.</th> <th>Save Delete</th>	Save Delete
Project Facet	Version
🖂 🐼 Dynamic Web Module	2.4
Java	5.0
	1.0
	1.0
🔲 🔲 Java Annotation Processing Support	5.0
	1.1
	1.1
E Struts	1.2
Web Bervices	122
WebLogic Control Extensions	920
	9.2.0
WebLogic Web App Extensions	9.2.0
1	
	<< Show Runtimes
< Back	Next > Finish Cancel

2. Crea el paquete es.ua.jtech.ejb.delegates en el que copiarás el JavaBean que accederá al EJB. Para ello crea la clase SaludoBean en el paquete y copia el contenido de la clase SaludoBean del proyecto Eclipse.

3. Arregla los errores de librerías añadiendo el proyecto EJB en el build path ("Build Path > Configure Build Path" y en la pestaña "Projects" pulsar "Add... ejb-saludo").

4. Para añadir el carácter local al EJB basta con añadir las siguientes anotaciones:

• En la definición de la clase SaludoBean, añade el nombre local JNDI y los nombres de las clases locales:

```
@JndiName(...
local="SaludoBeanLocal")
@FileGeneration(...
localClass=TRUE
localClassName="SaludoLocal"
localHome=TRUE
localHomeName="SaludoLocalHome")
```

• En los métodos saluda() y getSaludo() añade la anotación @LocalMethod().

5. Graba el bean y verás cómo desaparecen los errores en ejb-saludoWar (no hace falta crear de nuevo el ejbjar cliente porque se va acceder al bean de forma local y los ficheros class están disponibles en el servidor).

6. Copia el fichero index. jsp del proyecto Eclipse al directorio WebContent.

6. (*) Creación y despliegue del EAR

Una vez creado el proyecto War vamos a guardar ambos en un proyecto EAR y a desplegarlos en el servidor de aplicaciones.

1. Creamos un proyecto EAR ("New > Project > J2EE > Enterprise Application Project") con el nombre de ejb-saludoWebEAR. Añadimos al proyecto los módulos EJB (ejb-saludo) y WAR (ejb-saludoWar):

W New EAR Applicat	ion Project		×
J2EE Modules to Ad Select the J2EE modules Select New Module butto	d to the EAR to add to the new EAR app n to create a new J2EE moo	lication from the list. Jule.	
☐ ejb-cliente-saludo ☑ ejb-saludo ☑ ejb-saludoWar			
Select All Content Directory: EarContent	Deselect All	New Module]
	< Back Next >	Finish	Cancel

- 2. Elimina el módulo EJB del servidor.
- 3. Añade el módulo EAR ejb-saludoWebEAR al servidor.



4. Comprueba que funciona la aplicación Web conectándote a la URL http://localhost:7001/ejb-saludoWar/:



5. Por último, prueba el cliente remoto y el test JUnit para comprobar que la interfaz remota del EJB sigue funcionando.