



CC68J APLICACIONES EMPRESARIALES CON JEE

# APLICACIONES WEB

Estructura, desarrollo y despliegue

Profesores:  
■ Andrés Farías



v1.0

---

---

---

---

---

---

---

---

## Objetivos: aprender a...

1. La estructura y estándar de una aplicación web.
2. Estándar de almacenamiento de una aplicación web (WAR).
3. Desarrollar una aplicación web con BEA WebLogic Workshop.
4. Configurar distintos parámetros de una aplicación web:
  1. Establecer /cambiar el nombre de contexto.
  2. Definir el o los archivos de bienvenida (welcome-file list).
5. Desplegar la aplicación web en WLS vía Workshop.
6. Desplegar la aplicación web en WLS vía Consola de Administración del Dominio.

v1.1

---

---

---

---

---

---

---

---

# APLICACIONES WEB

v1.1

---

---

---

---

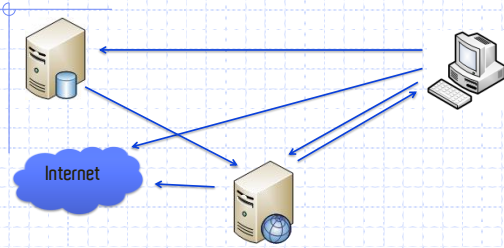
---

---

---

---

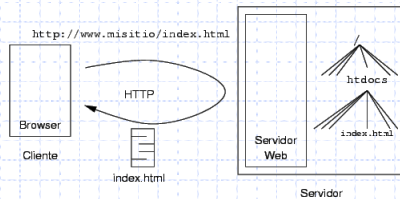
## Protocolo para intercambiar documentos de hipermmedia



Para ello hay una conversación entre el que pide un documento y el que lo "sirve" (invisible para el usuario del browser). Esta conversación se realiza según el protocolo HTTP.

## Arquitectura Web

- Para abrir una página Web en un navegador, normalmente se teclea el correspondiente URL o se "clickea" sobre un "link" oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de directorios y la envía de vuelta al navegador que la solicitó.



## Arquitectura Web

### Composición de una petición de un recurso web

- Una URL, como por ejemplo `http://www.dcc.uchile.cl/~afarias/index.html`, para solicitar un recurso a un servidor web se compone de cuatro partes:
  - ✓ Protocolo. Aunque obvio, se indica el protocolo en el navegador: `http`.
  - ✓ Dirección del Servidor. Puede ser indicada de dos modos:
    - La dirección IP del servidor, por ejemplo `192.80.24.4`.
    - Un nombre asociado a la dirección (`www.dcc.uchile.cl`).
  - ✓ Nombre del contexto. Todo recurso accedido en una petición http se encuentra disponible en el contexto de un módulo web, que posee un nombre de contexto. El nombre del contexto puede ser `"/afarias"`, `"/tienda"`, o el caso especial sin nombre: `"/"`.
  - ✓ Ubicación del recurso. El recurso se encuentra en un path específico (aunque lógico) dentro del módulo web. Por ejemplo, `/index.html`.

## El servidor web

### Un proceso del OS

- El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface.
- Dependiendo del tipo de la petición:
  - ✓ el servidor Web buscará el recurso solicitado o,
  - ✓ ejecutará un programa en el servidor para generar la respuesta a la petición solicitada.
  - ✓ De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente que realizó la petición.
- El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor.
  - ✓ La programación de una aplicación web puede aprovechar cada una de las ventajas propias (propietarias) que distinguen a ese servidor.
- Un servidor web es, fundamentalmente, un proceso del OS asociado a una dirección IP de la máquina asociada al OS y a un puerto particular.

### ¿De dónde lo saco, cómo lo instalo?

- Hay varios gratis que se pueden bajar de la Internet.
  - ✓ JBOSS,
  - ✓ Apache / Apache Tomcat
- Otros menos "gratis"
  - ✓ (Oracle) BEA WebLogic Server.
  - ✓ Oracle IAS.
  - ✓ IBM WebSphere.
- Obviamente, el computador debe estar en la internet y tener una dirección IP fija y un nombre de dominio (DNS).

## Aplicaciones Web J2EE

- Una aplicación web
  - ✓ está organizada en una estructura jerárquica de directorios
    - dos partes
      - Un directorio privado WEB-INF
        - contiene los recursos que no son descargables para el cliente
        - Un directorio público que contiene los recursos públicos
    - Ejemplo: miaplicación\
 

```

index.html
login.jsp
images\ logo.gif
doc\ tutorial.pdf
WEB-INF\
      web.xml (Deployment Descriptor)
      classes\ ServletCompras.class
      lib\ cualquierOtraApi.jar
              
```
- Una aplicación web puede ser empaquetada en un fichero WAR.

## Aplicaciones WEB

### Empaquetamiento en WAR

- Web Application Archive
- Permiten empaquetar en una sola unidad aplicaciones web java completas.
  - ✓ Servlets y JSPs
  - ✓ Contenido estático
    - Html
    - Imágenes
    - etc.)
  - ✓ Otros recursos web

---

---

---

---

---

---

---

---

## Aplicaciones WEB

### Empaquetamiento en WAR

- Son una extensión del archivo JAR
- Se introdujeron en la especificación 2.2 de los servlets.
- Multiplataforma
- MultiVendor

---

---

---

---

---

---

---

---

## Aplicaciones WEB

### Empaquetamiento en WAR

- Ventajas:
  - ✓ Simplifican el despliegue de aplicaciones web.
    - Facilidad de instalación
    - Un solo fichero para cada servidor en un cluster.
  - ✓ Seguridad
    - No permite el acceso entre aplicaciones web distintas
- Desventajas
  - ✓ La principal desventaja del despliegue de archivos WAR es que pequeños cambios en la aplicación no pueden ser realizados en tiempo real (run-time desde ahora).
  - ✓ Cada cambio requiere generar un nuevo WAR.
- Casi todos los contenedores web JEE permiten desplegar aplicaciones web como directorios en vez de un único archivo. Esto es conocido como un despliegue expandido (exploded deployment).

---

---

---

---

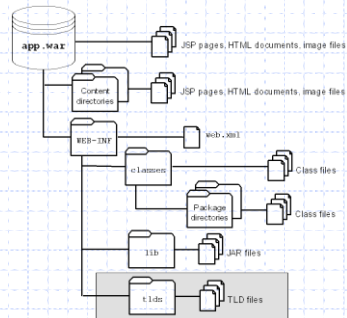
---

---

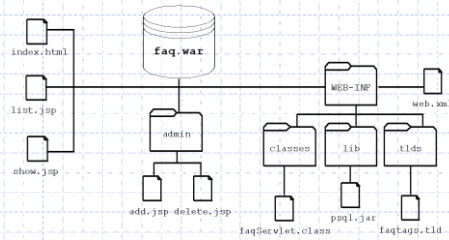
---

---

## Estructura



## Estructura Ejemplo



## Archivo Web.xml

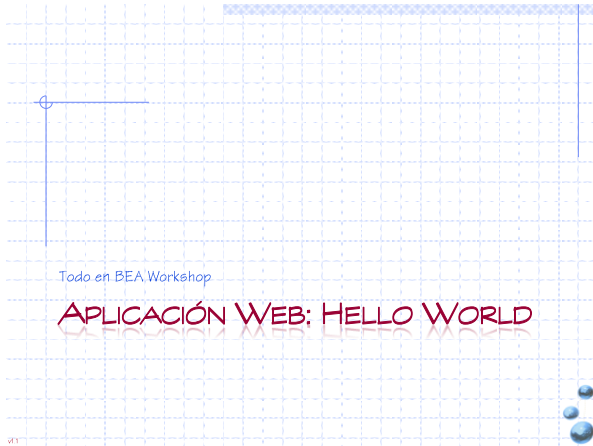
### Descriptor de Despliegue

- WEB-INF/web.xml
- Documento XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc./DTD Web Application 2.2/EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
```

- En él se dan de alta

- ✓ Servlets
- ✓ Parámetros del contexto
- ✓ TLDs
- ✓ Filtros
- ✓ Etc.




---

---

---

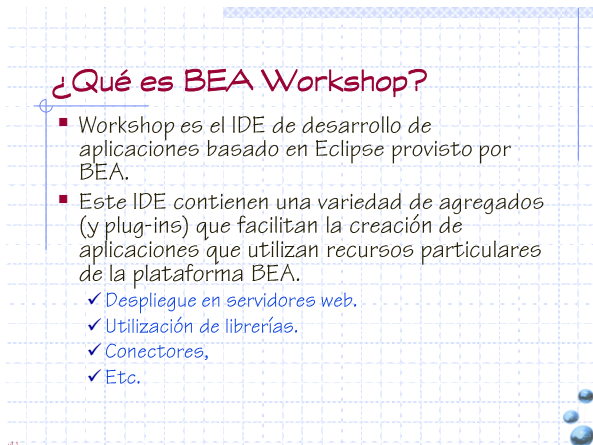
---

---

---

---

---




---

---

---

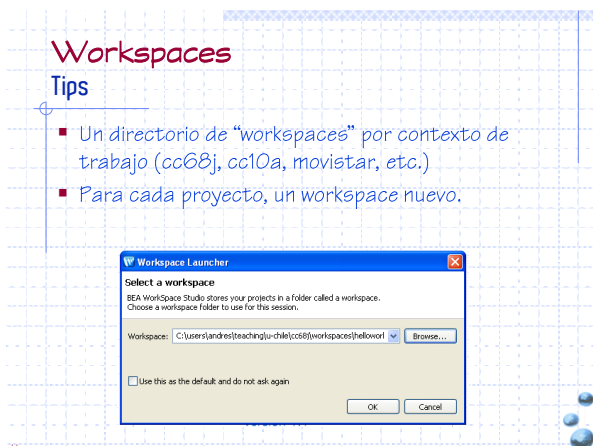
---

---

---

---

---




---

---

---

---

---

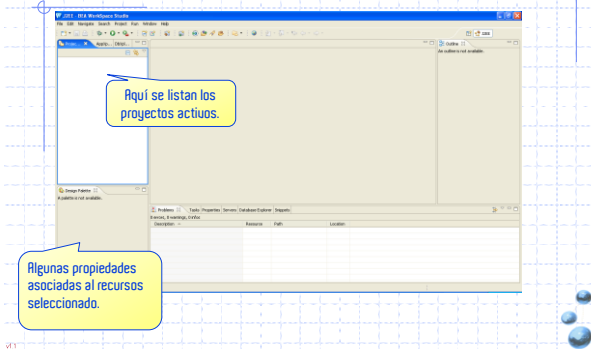
---

---

---

## Vista General

### Perspectiva J2EE




---

---

---

---

---

---

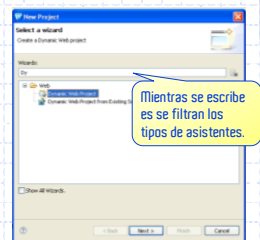
---

---

## Proyecto Web

### Wizard: Dynamic Web Project

- Vía menú:
  - ✓ File → New → Dynamic Web Project.
  - ✓ File → New → Project. Luego Dynamic Web Project.
- Vía shortcuts:
  - ✓ ALT-SHIFT-N




---

---

---

---

---

---

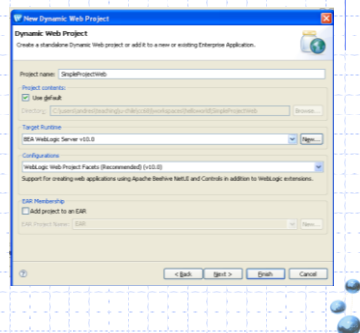
---

---

## Configuración

### Wizard: Dynamic Web Project

- Nombre del proyecto:
  - ✓ Notación húngara.
  - ✓ No utilizar espacios (uno nunca sabe).
- Almacenar en el workspace.




---

---

---

---

---

---

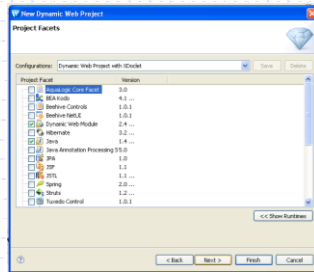
---

---

## Librerías a incluir

### Wizard: Dynamic Web Project

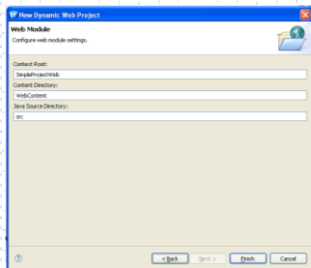
- En esta pantalla se incluyen las librerías disponibles para desarrollar.



## Librerías a incluir

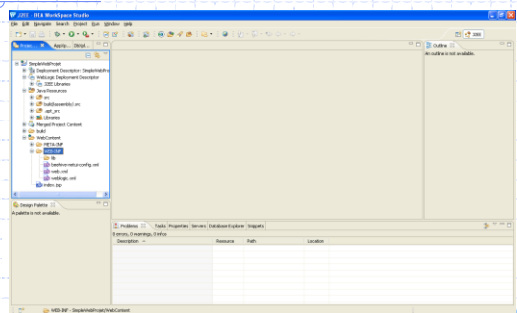
### Wizard: Dynamic Web Project

- Context Root:
  - ✓ El nombre del módulo web.
  - ✓ `http://uhile.cl/portal`.
- Context Directory:
  - ✓ Donde están los archivos públicos.
  - ✓ Puedo tener cualquier nombre.
  - ✓ Evitar los nombres con espacios (nunca se sabe).
  - ✓ La estructura interna, excluyendo el directorio WEB-INF.
- Java Source Directory:
  - ✓ Donde se utilizan los archivos Java.
  - ✓ No las clases, éstas se encuentran en el directorio WEB-INF.



## Resultado final

### Wizard: Dynamic Web Project







## Archivos de configuración

### El web.xml y weblogic.xml

- Un aplicación web puede ser configurada en diversos parámetros a través de, principalmente, dos archivos de configuración.
- Archivo de configuración **web.xml**
  - ✓ Archivo de configuración estándar (J2EE) para la configuración de parámetros de la aplicación.
  - ✓ Permite configurar:
    - Nombre del módulo.
    - Archivos de bienvenida.
    - Filtros sobre las peticiones HTTP.
    - Otros Recursos de la aplicación web.
- Archivo de configuración **weblogic.xml**
  - ✓ Archivo de configuración propietario a **BEA WebLogic**, para configurar parámetros administrados por el servidor.
  - ✓ Permite configurar:
    - Nombre del contexto del módulo (*Context root*).
    - Librerías que requiere la aplicación para funcionar.
    - Otros recursos.

## Archivo web.xml

### Archivos de bienvenida

- Los archivos de bienvenida, son aquellos son entregados al usuario por defecto cuando éste no indica el recurso del módulo web que está solicitando.  
Ej: <http://localhost/HelloITWeb>.
- Se pueden listar muchos archivos. El algoritmo se define de la siguiente manera:
  - ✓ El servidor buscará el primero de la lista, de no encontrarlo pasará al siguiente en la lista, y así hasta retornar el que encuentre.
  - ✓ Si no encuentra ningún archivo de bienvenida, el servidor enviará una respuesta de error: (404: Not Found) como si el cliente hubiese solicitado el último archivo de la lista de archivos de bienvenida.

## Archivo web.xml

### Archivos de bienvenida (?)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>HelloITWeb</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

En este ejemplo hay un solo archivo de bienvenida.

## Archivo weblogic.xml

### Nombre de contexto del módulo web: Context-root

- El nombre de contexto de un módulo (o aplicación) web es aquel que se indica en la URL para acceder a algún recurso de ésta, por ejemplo: <http://www.movistar.cl/portal>.
- El nombre de contexto puede ser también simplemente "/". Ejemplo: <http://www.latercera.cl/>.
- En general es recomendable que siempre una aplicación tenga un nombre de contexto distinto a "/".
  - ✓ De esta manera no hay que preocuparse de si en el servidor donde se desplegará existe ya una aplicación con el mismo nombre de contexto (siendo "/" bastante frecuente).
  - ✓ Por otro lado, a través de una configuración de proxy se puede resolver el redireccionamiento de las peticiones, simulando que la aplicación se encuentra desplegada en "/".

## Archivo weblogic.xml

### Cambiando el nombre de contexto

```
<?xml version="1.0" encoding="UTF-8"?>
<wls:weblogic-web-app
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance"
  xmlns:wls="http://www.bea.com/ns/weblogic/90"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd
  http://www.bea.com/ns/weblogic/90
  http://www.bea.com/ns/weblogic/90/weblogic-web-
  app.xsd">
  <wls:context-root>GoodByeITWeb</wls:context-root>
  ...
</wls:weblogic-web-app>
```

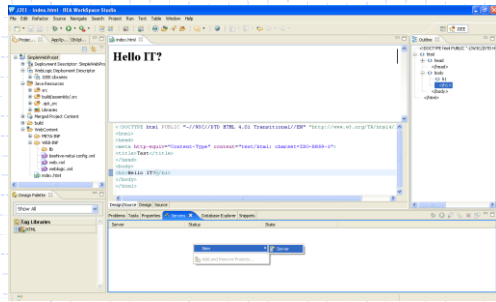
Este cambio implica re-  
desplegar la aplicación.

Desplegando fácilmente hacia el servidor.

## DESPLIEGUE VÍA WORKSHOP

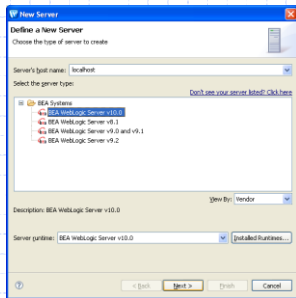
## El proyecto web

### Definición del servidor



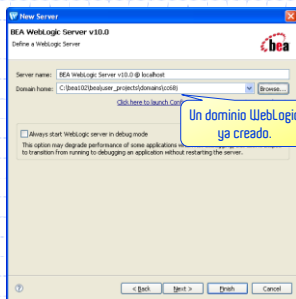
## El proyecto web

### Definición del servidor (2)



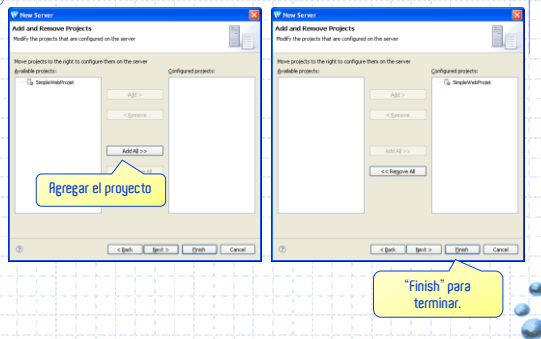
## El proyecto web

### Definición del servidor (3)



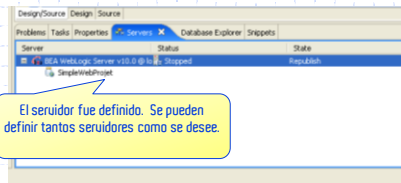
## El proyecto web

### Definición del servidor (4)



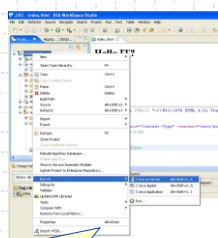
## El proyecto web

### Definición del servidor (5)



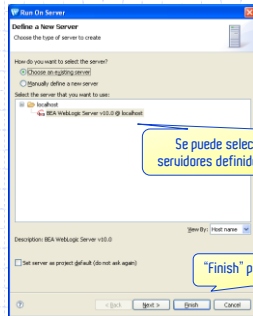
## El proyecto web

### Despliegue automático



- Para ejecutar la aplicación web, se ejecuta en el servidor con la opción "Run as" → "Run on Server".
- Para los amantes de vi: ALT-SHIFT-X, R.
- La aplicación web se despliega (deploy) automáticamente en la consola de administración del dominio asociado al servidor.
- Casi cualquier modificación se ve reflejada inmediatamente en la aplicación desplegada.

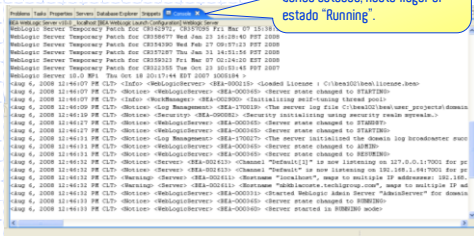
## El proyecto web Despliegue



Se puede seleccionar cualquiera de los servidores definidos, o bien definir uno nuevo.

"Finish" para terminar.

## El proyecto web Despliegue (2)



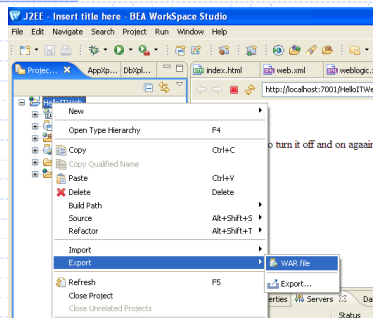
La pestaña "Console" muestra el log del servidor. WLS pasa por varios estados, hasta llegar al estado "Running".

Usando la consola de administración WebLogic

## DESPLIEGUE DESDE LA CONSOLA

## WAR de la aplicación

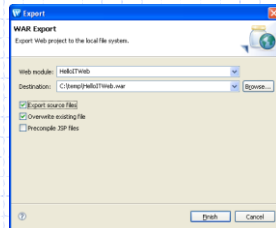
### Generación vía Workshop



## WAR de la aplicación

### Generación vía Workshop

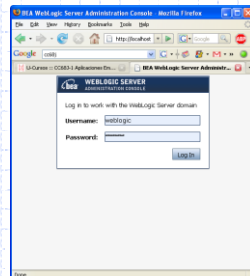
- Esta pantalla permite seleccionar un módulo web a exportar.
- Se indica también la ruta del archivo WAR que se generará.
- Las tres opciones son:
  - ✓ Incluir en el WAR los archivos fuentes (java).
  - ✓ Sobre-escribir el war si ya existe.
  - ✓ Precompilar los archivos JSP. Esta opción permite ahorrar el primer esfuerzo al servidor.



## Consola de Administración

### Login al servidor WLS

- Se lanza el servidor de administración del dominio CC&Bj creado en la clase anterior.
- Este servidor corre (normalmente) en la siguiente dirección:  
<http://localhost:7001/console>.
- El username normalmente es weblogic, y el password normalmente es "password" o "weblogic" (depende obviamente de los valores ingresados durante la creación del dominio).



## Consola de Administración

### La consola de administración

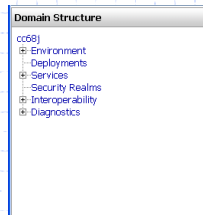
- **Centro de cambios.**
  - ✓ Permite tomar control sincronizado de la consola para realizar cambios.
  - ✓ Aplicar cambios.
- **Estructura del Dominio.**
  - ✓ Lista distintos elementos de administración del dominio, en forma de árbol.



## Consola de Administración

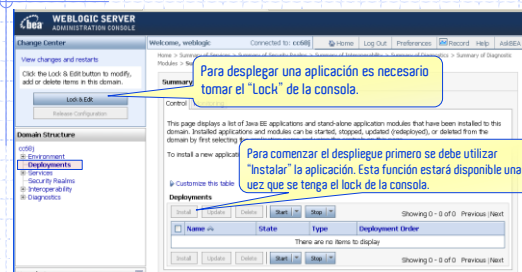
### La consola de administración

- **Environment:** Contiene información de:
  - ✓ Servidores.
  - ✓ Clusters.
  - ✓ Máquinas, etc.
- **Deployments:**
  - ✓ Lista los elementos desplegados.
  - ✓ Permite desplegar nuevas librerías y aplicaciones.
- **Servicios:** Permite configurar servicios del servidor tales como:
  - ✓ Mensajería
  - ✓ JDBC
  - ✓ JTA, etc.
- **Security Realms:** Es un contenedor para mecanismos de protección de los recursos administrados por el servidor.
- **Interoperabilidad:**
  - ✓ Permite configurar características de interoperabilidad entre WLS y Tuxedo.
- **Diagnostics:**
  - ✓ Proporciona herramientas de diagnóstico para analizar el desempeño de WebLogic.
  - ✓ Contiene acceso a los archivos de Log del servidor y dominio.



## Despliegue de la aplicación web

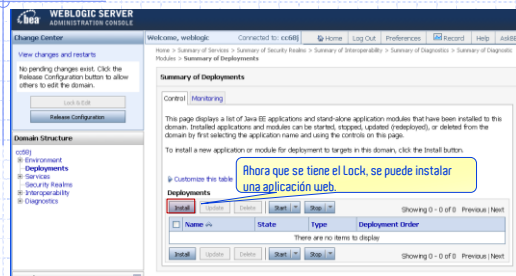
### Página de despliegue (Deployments)





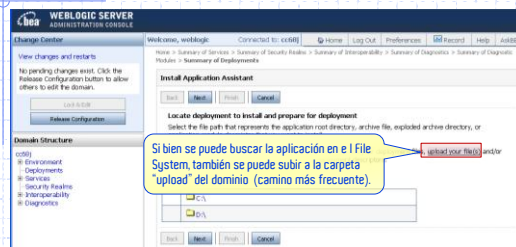
## Despliegue de la aplicación web

### Página de despliegue (Deployments) (2)



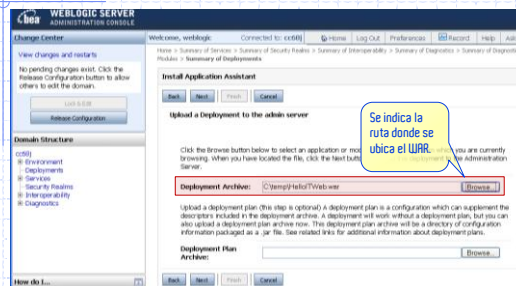
## Despliegue de la aplicación web

### Cargar el archivo WAR



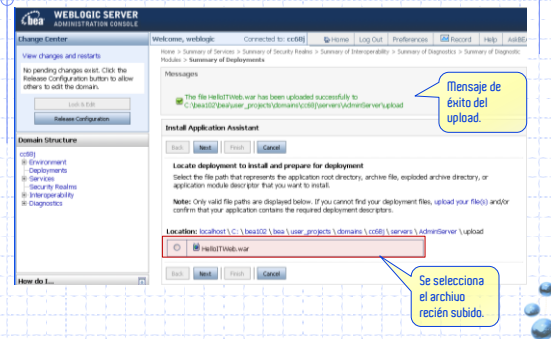
## Despliegue de la aplicación web

### Cargar el archivo WAR (2)



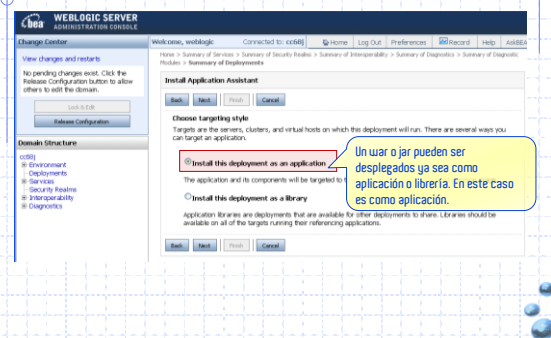
## Despliegue de la aplicación web

### Cargar el archivo WAR (2)



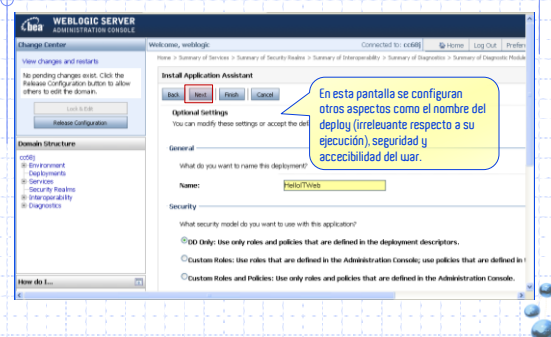
## Despliegue de la aplicación web

### Instalación del WAR



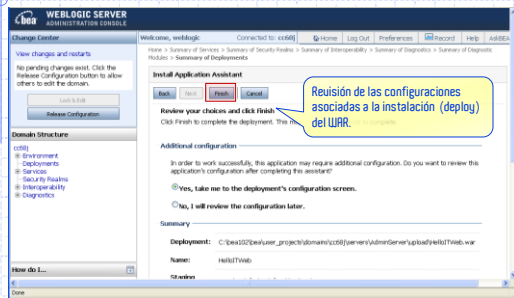
## Despliegue de la aplicación web

### Otras opciones



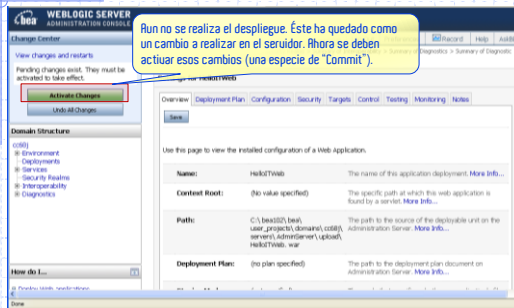
## Despliegue de la aplicación web

### Resumen del deploy



## Despliegue de la aplicación web

### Activación de los cambios



## Despliegue de la aplicación web

### Activación de los cambios (?)



## Activación de la aplicación

### Lista de despliegues

La aplicación aparece en la lista de aplicaciones desplegadas. Su estado es "Prepared" lo que significa que aun no está siendo ejecutada.

Para que comience a aceptar peticiones, la aplicación debe ser lanzada con "Start -> Servicing all request".

Name	Type	Deployment Order
HelloworldWeb	Web Application	100

## Activación de la aplicación

### Confirmación de activación

Pantalla de confirmación. También indica si la aplicación se encuentra en un estado desde el cual se puede realizar la operación solicitada (no se puede lanzar una aplicación ya activa).

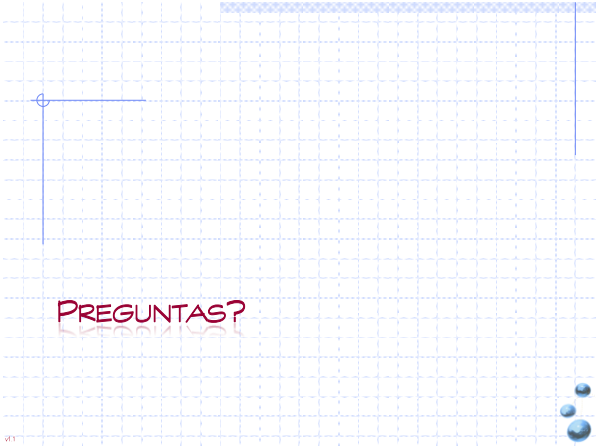
## Activación de la aplicación

### Confirmación de activación

Mensaje de Log. Todo resultó OK.

La aplicación se encuentra ahora en estado "Active".

Name	Type	Deployment Order
HelloworldWeb	Web Application	100



PREGUNTAS?

---

---

---

---

---

---

---