Componentes Web

Componentes dinámicos en la plataforma Java.

Introducción

"Hoy en día los desarrolladores reconocen la necesidad de aplicaciones portables, transaccionales y distribuidas que aprovechen la velocidad, seguridad y confiabilidad de las tecnologías del lado del servidor."

Las aplicaciones enterprise deben ser diseñadas, construidas y producidas

con: -dinero +velocidad -recursos

Modelo App EE

La plataforma Java EE utiliza un

"modelo de aplicación distribuida de varios niveles".

- La lógica de una aplicación se divide en componentes de acuerdo a su función.
- Estos componentes, que conforman una aplicación Java EE, se instalan en máquinas diferentes.

https://docs.oracle.com/javaee/7/firstcup/java-ee001.htm

Java Platform, Enterprise Edition: Your First Cup: An Introduction to the Java EE Platform

Previous Page Next Page

2.1 Overview of Enterprise Applications

This section describes enterprise applications and how they are designed and developed.

As stated above, the Java EE platform is designed to help developers create large-scale, multi-tiered, scalable, reliable, and secure network applications. A shorthand name for such applications is "enterprise applications," so called because these applications are designed to solve the problems encountered by large enterprises. Enterprise applications are not only useful for large corporations, agencies, and governments, however. The benefits of an enterprise application are helpful, even essential, for individual developers and small organizations in an increasingly networked world.

The features that make enterprise applications powerful, like security and reliability, often make these applications complex. The Java EE platform reduces the complexity of enterprise application development by providing a development model, API, and runtime environment that allow developers to concentrate on functionality.

2.1.1 Tiered Applications

In a multi-tiered application, the functionality of the application is separated into isolated functional areas, called tiers. Typically, multi-tiered applications have a client tier, a middle tier, and a data tier (often called the enterprise information systems tier). The client tier consists of a client program that makes requests to the middle tier. The middle tier is divided into a web tier and a business tier, which handle client requests and process application data, storing it in a permanent datastore in the data tier.

https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/14.1.2/jeemg/index.html

Java EE application development concentrates on the middle tier to make enterprise application management easier, more robust, and more secure

Modelo de App EE

Una aplicación web puede ser:

1. Orientada-Presentación: Genera páginas web interactivas, basada en requerimientos / respuestas.

2. Orientada-Servicios: La aplicación web implementa el endpoint del Servicio Web.

- SOAP XML WSDL UDDI -

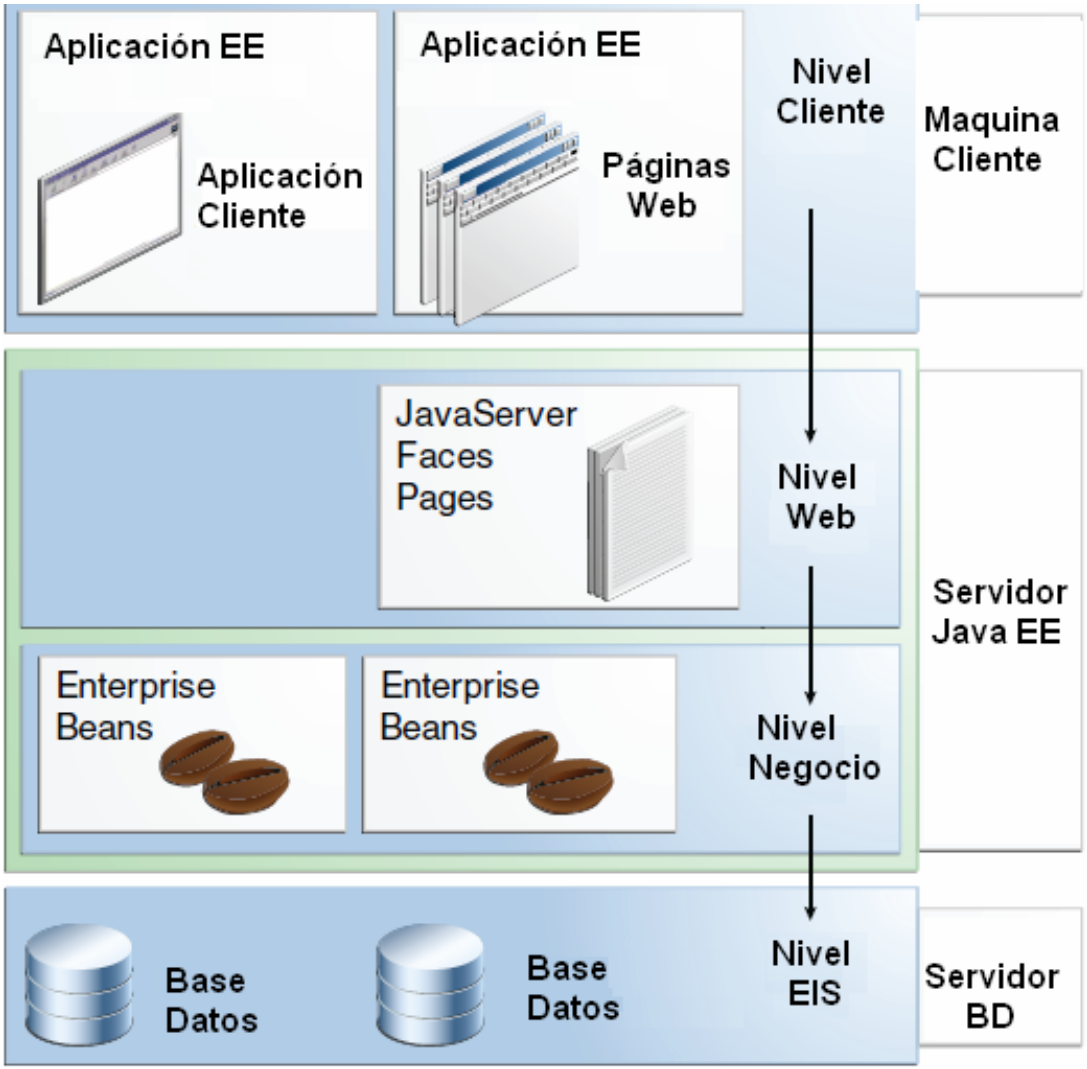
- REST FULL -

Modelo de App EE

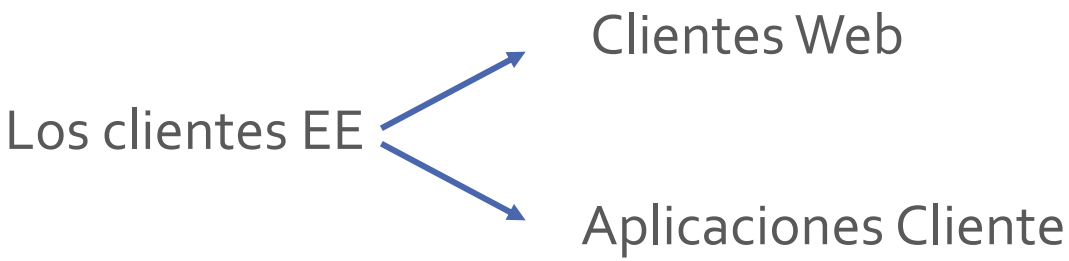
Las especificaciones EE define los siguientes componentes:

- Aplicaciones cliente y applet, corren sobre el *cliente*.
- Servlet, JSP y JSF son componentes *web* que corren sobre el servidor.
- JavaBeans son componentes de *negocio* que corren sobre el servidor.

Modelo de App EE



Nivel Cliente



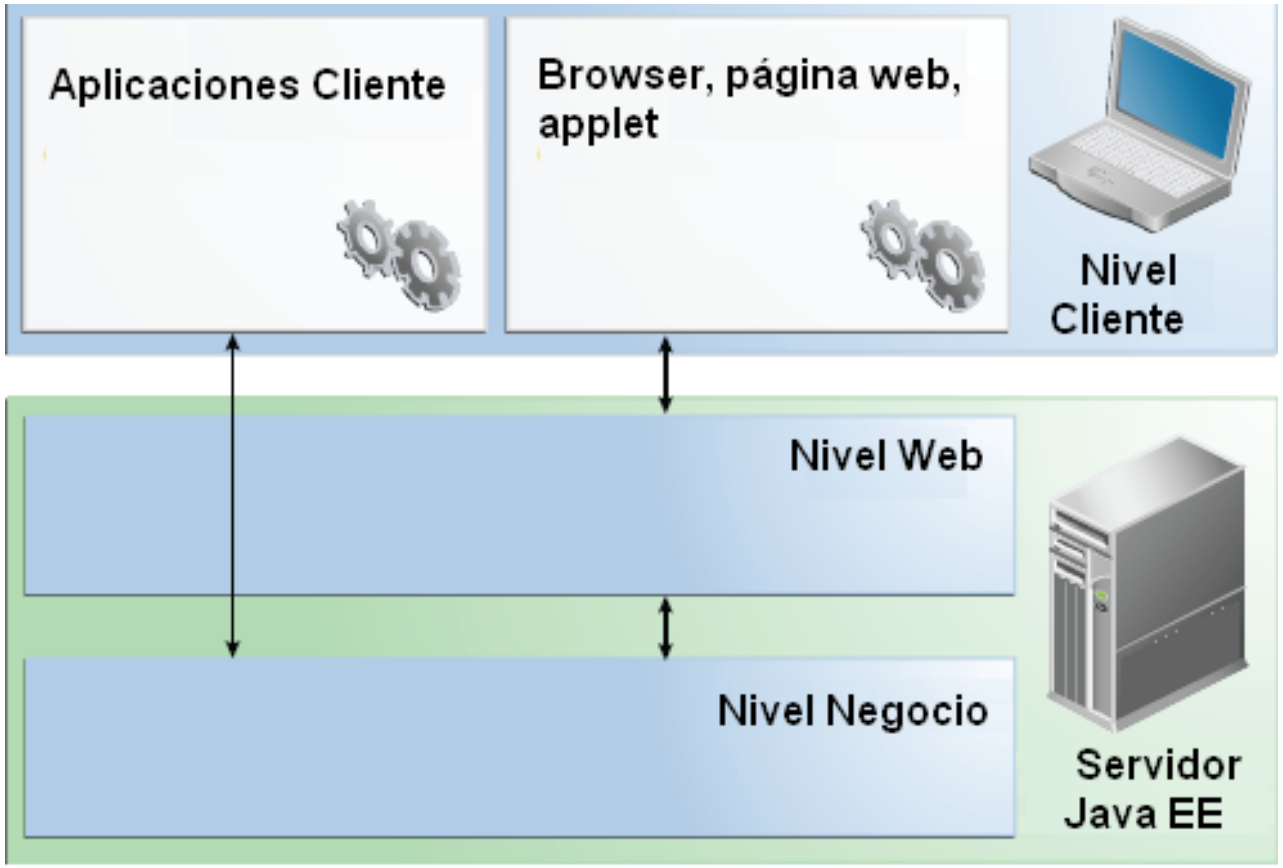
Clientes Web

- Páginas Dinámicas que contiene código en distintos lenguajes (HTML, XML), generado por un componente web.
- Browser web, que interpreta la página recibida del servidor.

Aplicación Cliente

- Provee la manera para que los usuarios manejen las tareas a través de una interfaz gráfica (también línea de comandos).
- Pueden acceder al Nivel de negocios siempre y cuando la aplicación lo garantice.

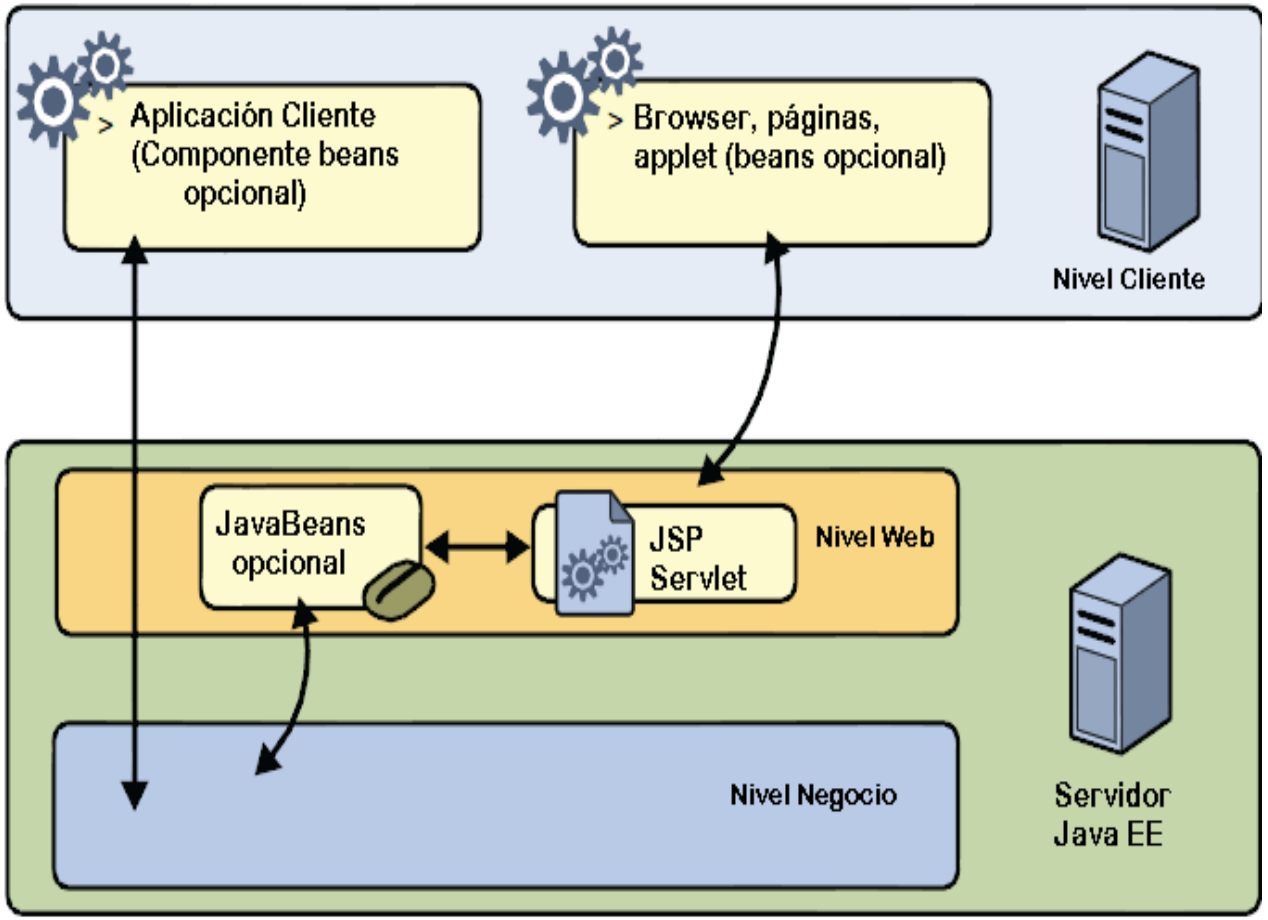
Comunicación con Servidor



Componentes Web

- Servlet: Son clases del lenguaje java, que procesan peticiones dinámicamente y construyen las respuesta.
- Java Server Page: Es un documento de texto que se ejecuta como un servlet, permite contenido estático.
- Java Server Face: Es la interface de usuario construida sobre servlet y JSP.

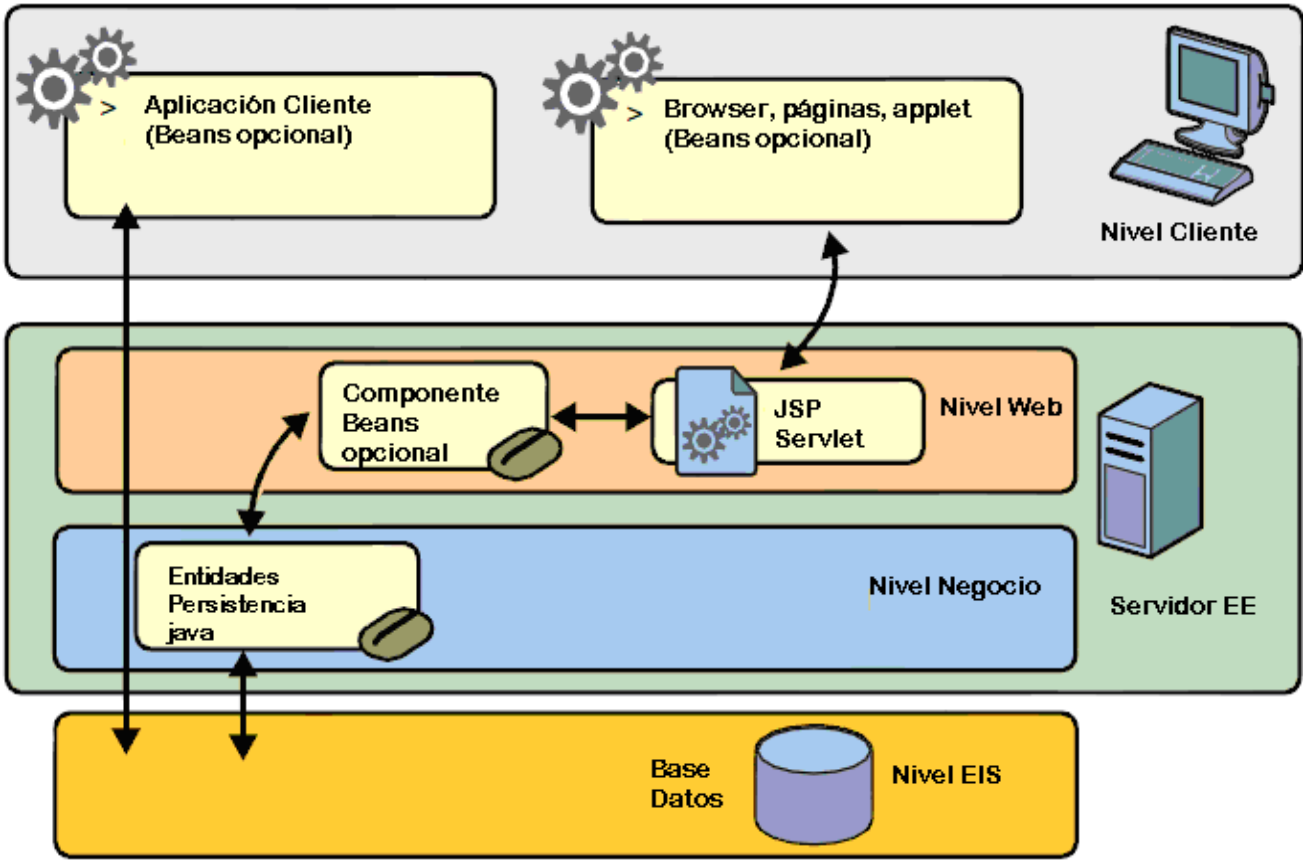
Componentes Web



Componentes Negocio

- Es el código (lógica) que resuelve y satisface las necesidades de un dominio particular de la empresa.
- Los Sistemas Información de la Empresa incluyen, procesamiento de transacciones, sistemas de base de datos, otros sistemas de información heredados.

Componentes Negocio



Componentes y Contenedores

- Los componentes web (unidades de software) están en la forma de Servlet o JSP (junto con JavaBean)
- Los componentes corren en un contenedor Web
  - Tomcat, GlassFish, Resin
  - Todos los servidores compatibles con app J2EE proveen contenedores web
- Los contenedores constituyen la interface entre los componentes y las funcionalidades específicas de bajo nivel.

Desarrollo y Despliegue de una aplicación

Un ejemplo paso a paso

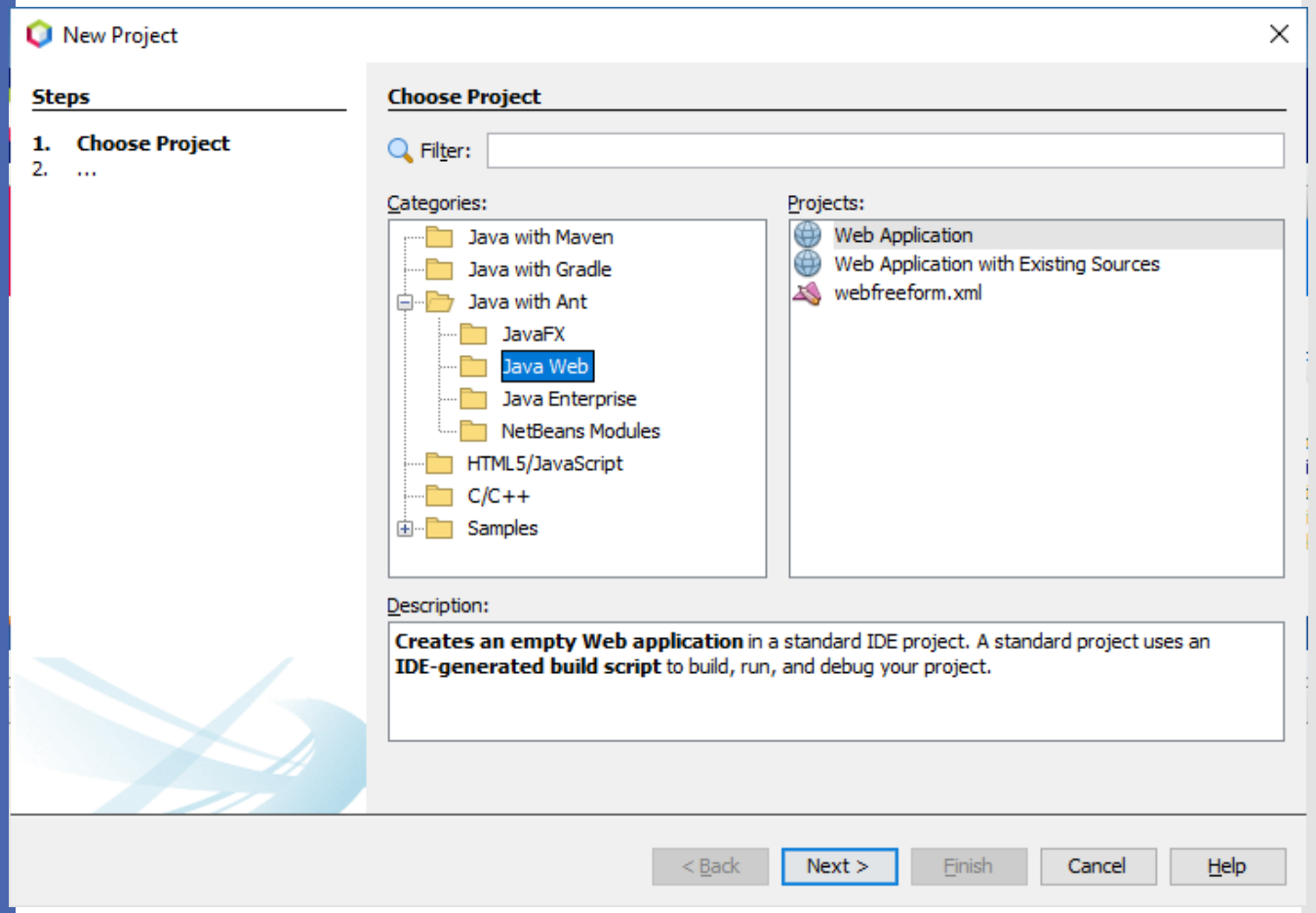
Desarrollo

1. Escribir y compilar el código (Servlet or JSP)
2. Crear un recurso estático (Por ej.: una imagen)
3. Crear el descriptor de despliegue (web.xml)
4. Construir la aplicación web (archivo *.war o un directorio listo para despliegue)
5. Desplegar la aplicación en un contenedor web
   - Los clientes web están listos para acceder via URL

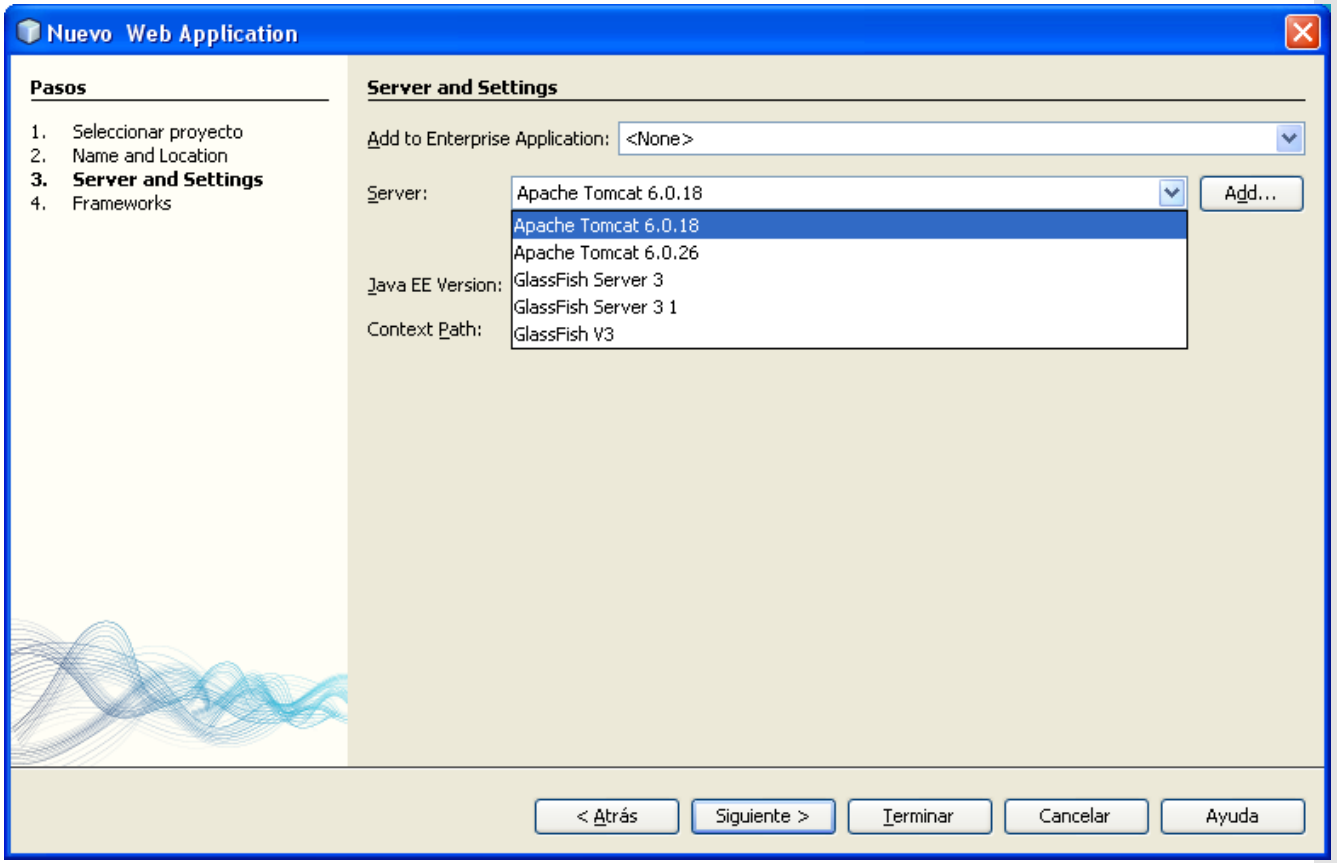
1. Escribir y compilar

- Crear una estructura de carpetas y subcarpetas, para conservar el código fuente separado del código compilado.
- Escribir el código servlet y/o la página JSP
- Crear build.xml
  Es un documento de texto en formato xml que incluye tareas especificas a realizar antes o después de iniciar el proyecto, la compilación, etc..
- El entorno NetBeans maneja estas tareas.

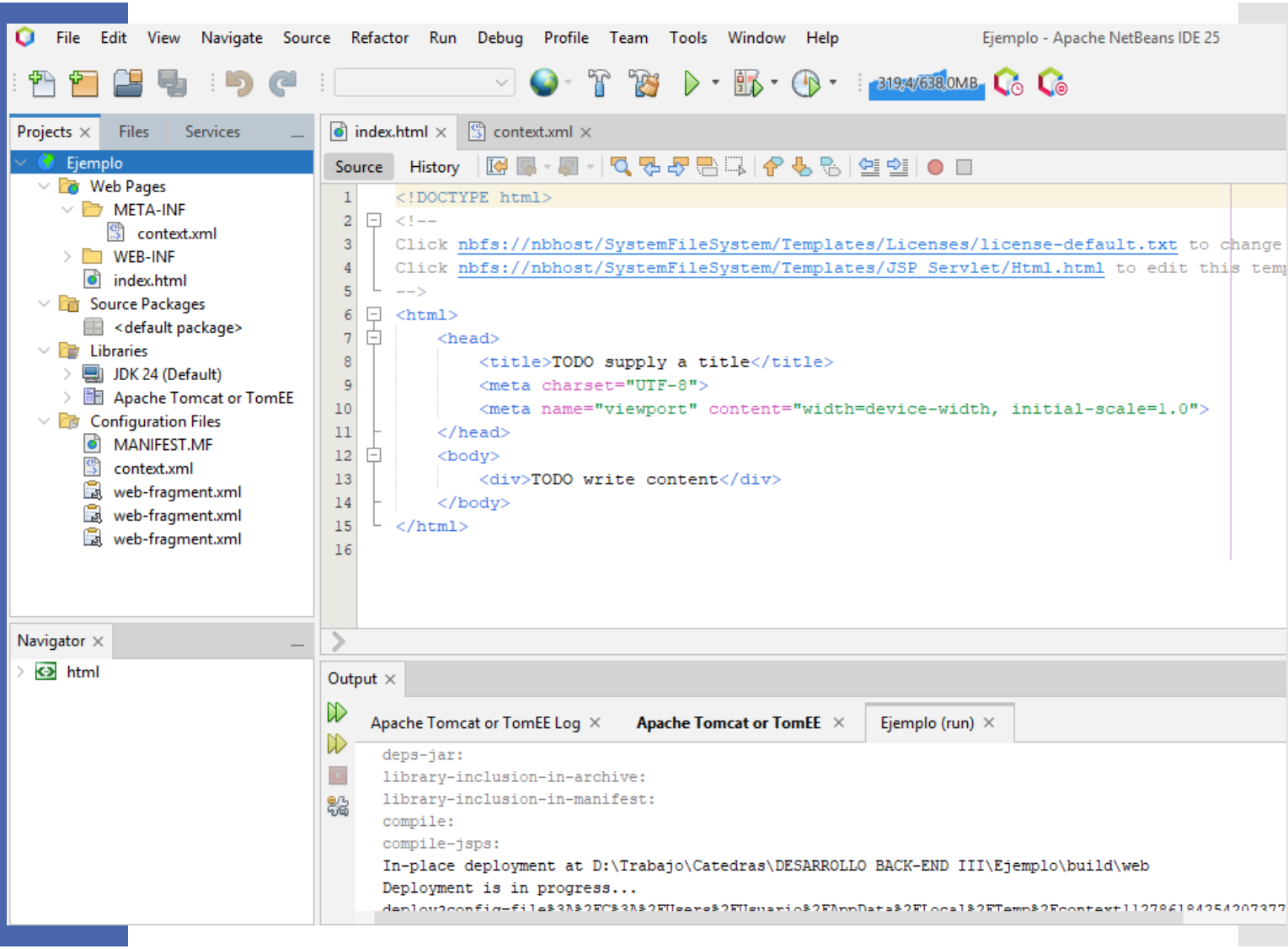
IDE
NetBeans



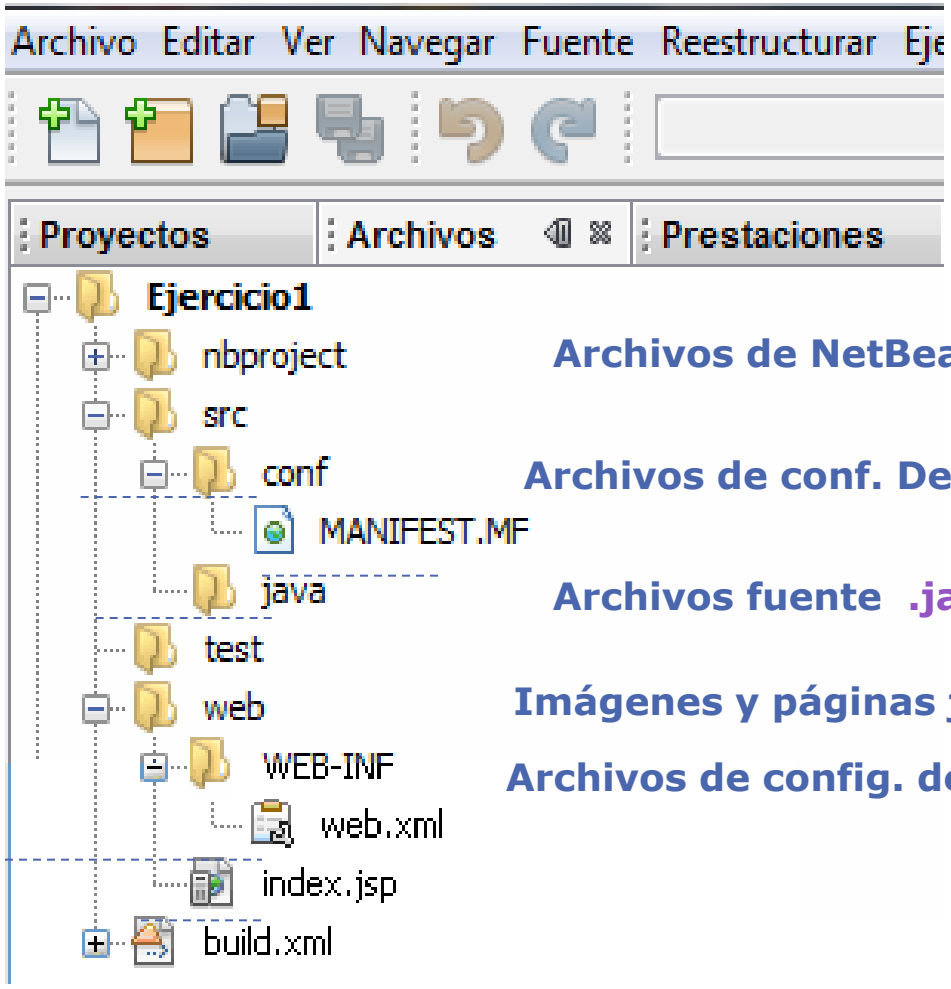
IDE NetBeans



Estructura creada



Estructura de carpetas



Archivos de NetBeans

Archivos de conf. Del módulo

Archivos fuente .java

Imágenes y páginas jsp y HTML

Archivos de config. de la apl. web

2. Crear recursos estáticos

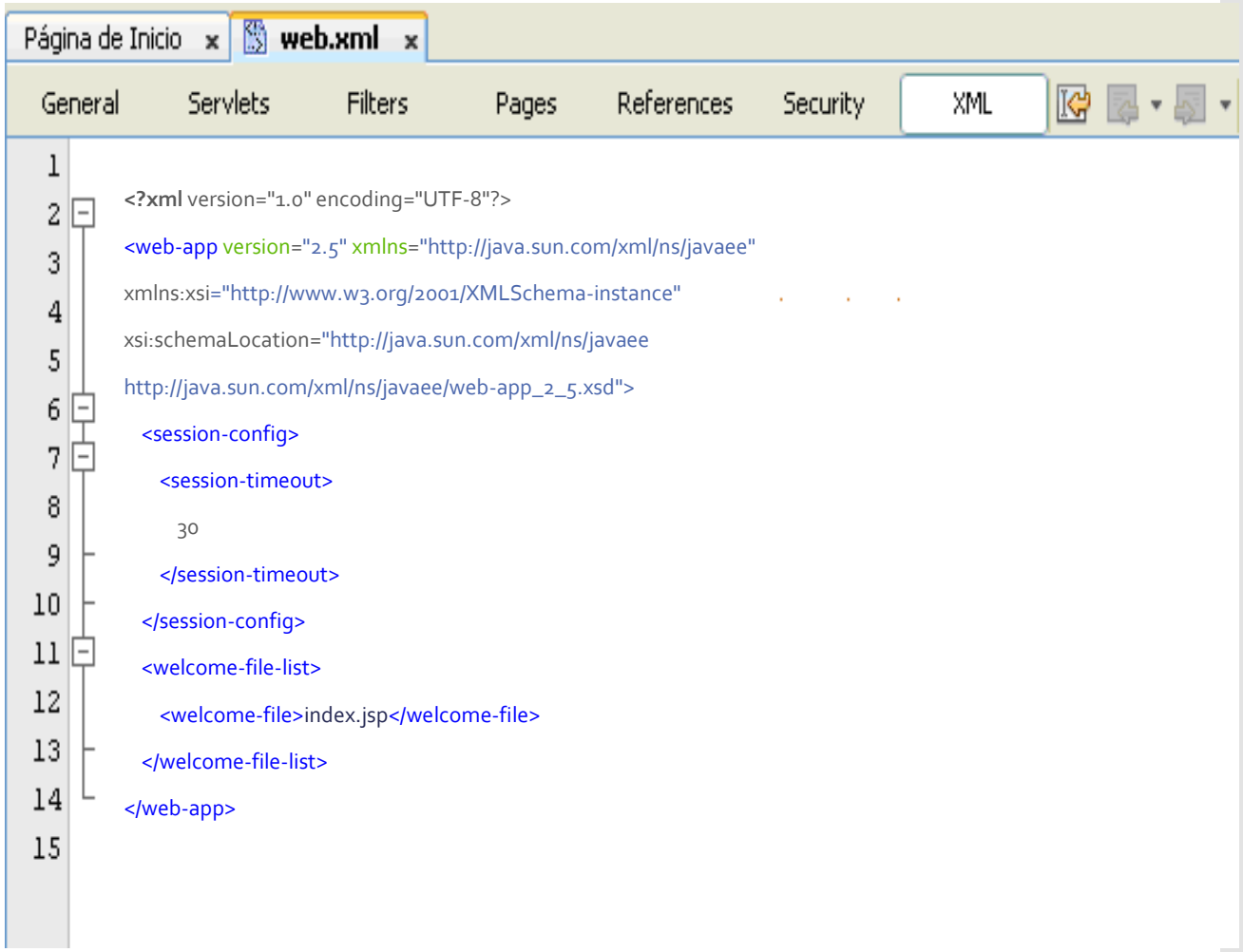
- Páginas HTML.
- Archivos imágenes que son usados por los documentos JSP y HTML

En la arquitectura EE son llamados **recursos web.**

3. Crear el descriptor de despliegue

- El descriptor web.xml contiene instrucciones de despliegue en tiempo de ejecución para el contenedor web.
  - URL que el cliente usa para acceder al componente web.

    …sun-web.xml es el descriptor para Glassfish…
- Si la aplicación solo tiene documentos html y archivos imágenes no necesita el descriptor.

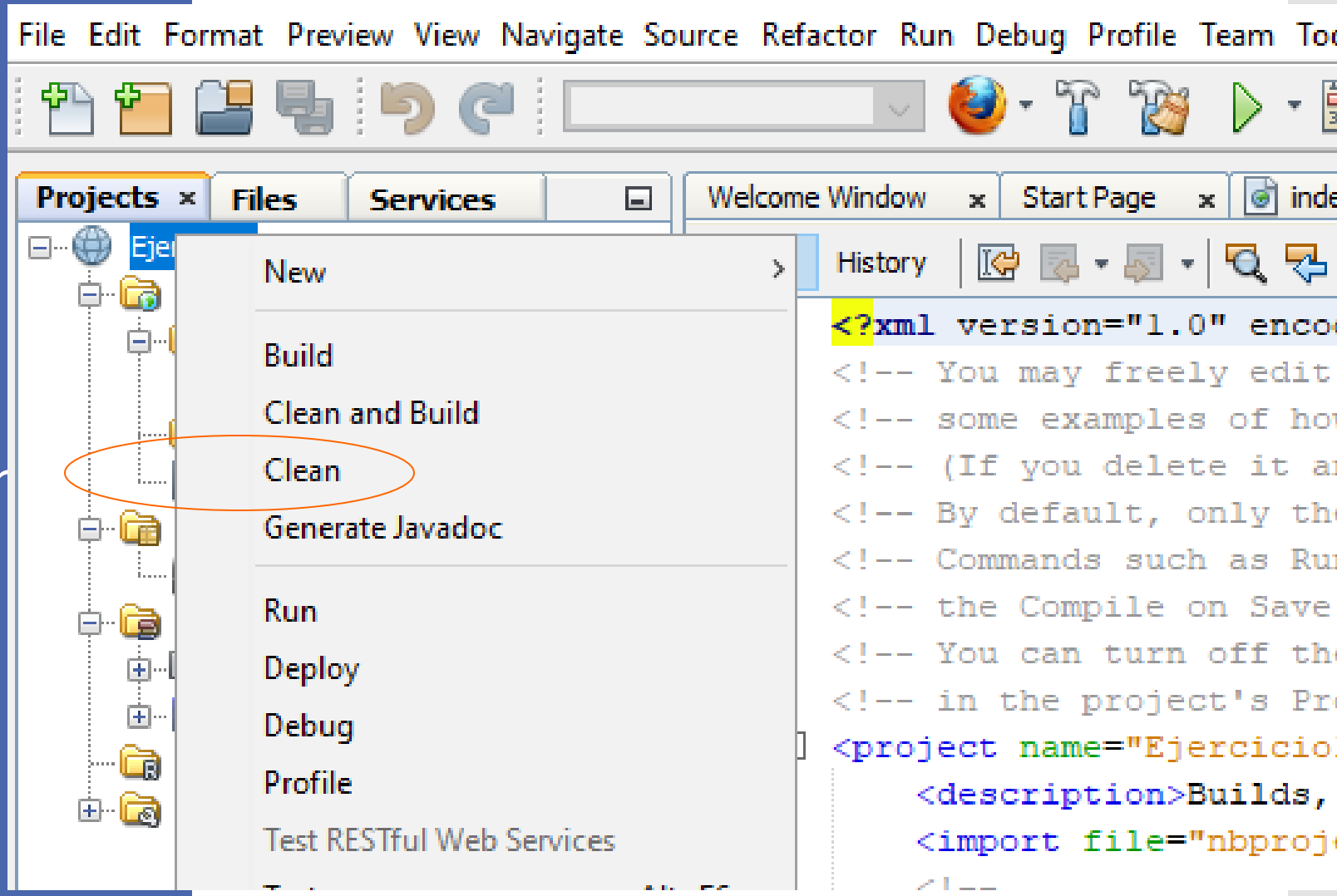

Página de Inicio
web.xml
General
Servlets
Filters
Pages
References
Security
XML
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
<session-config>
<session-timeout>
30
</session-timeout>
</session-config>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>

4. Construir la aplicación

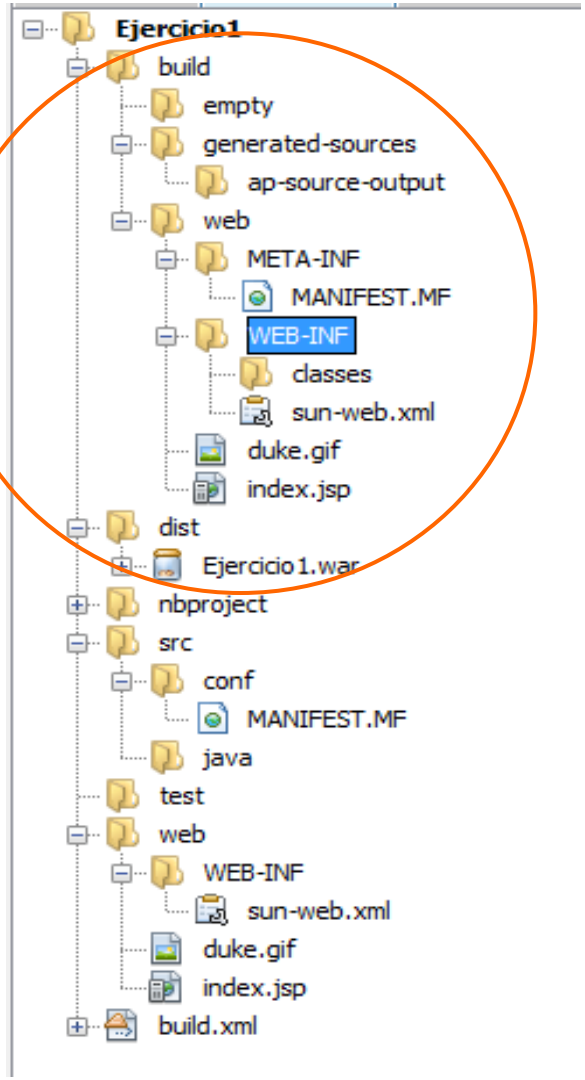
El proceso de **generar** o **build** la aplicación consta de:

- Crear la carpeta **build** y sus subcarpetas.
- Compilar los archivos .java y guardarlos en la carpeta **build/WEB-INF/classes**
- Crear la carpeta **dist** con el archivo .war

IDE
NetBear


File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team
Projects
Files
Services
Welcome Window
Start Page
New
Build
Clean and Build
Clean
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
History
<?xml version="1.0" enco
<!-- You may freely edit
<!-- some examples of ho
<!-- (If you delete it a
<!-- By default, only th
<!-- Commands such as Ru
<!-- the Compile on Save
<!-- You can turn off th
<!-- in the project's Pr
<project name="Ejercicio
<description>Builds,
<import file="nbproj

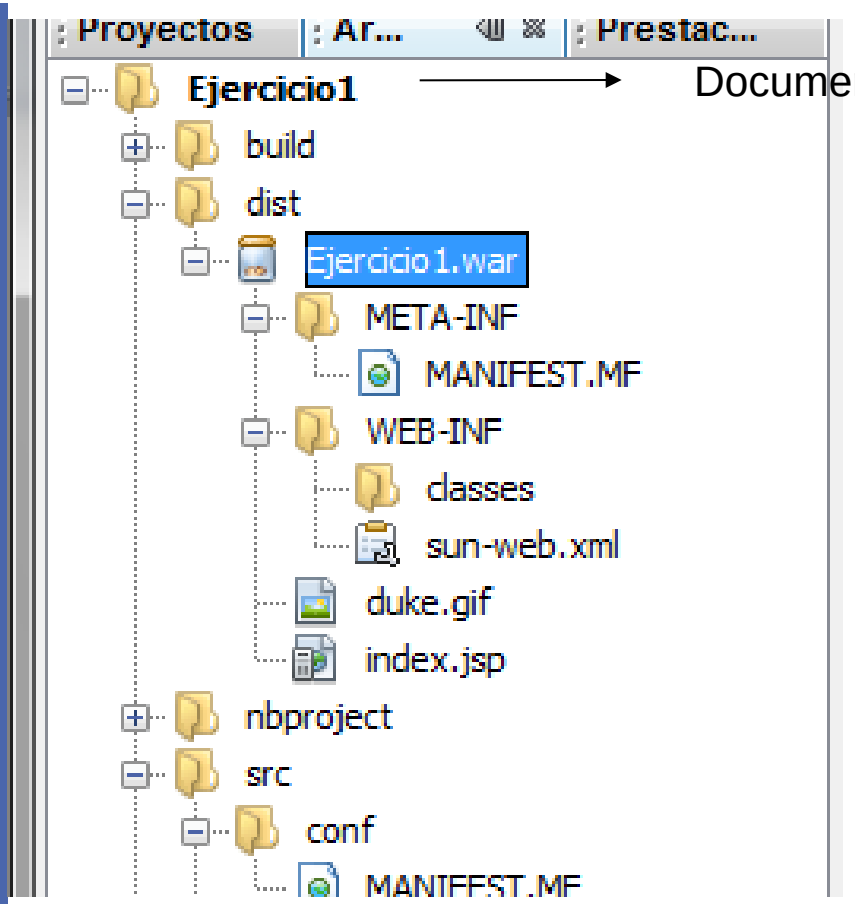
IDE NetBeans



Archivo .war

- Es un archivo compacto listo para ser desplegado sobre cualquier contenedor web.
  - Componentes Web (servlets or JSP's).
  - Clases del lado-Servidor
  - Contenido estático de la presentación Web
  - Clases del lado-cliente
- Refleja su contenido en la carpeta build

Archivo .war



Documento raiz - **contexto**

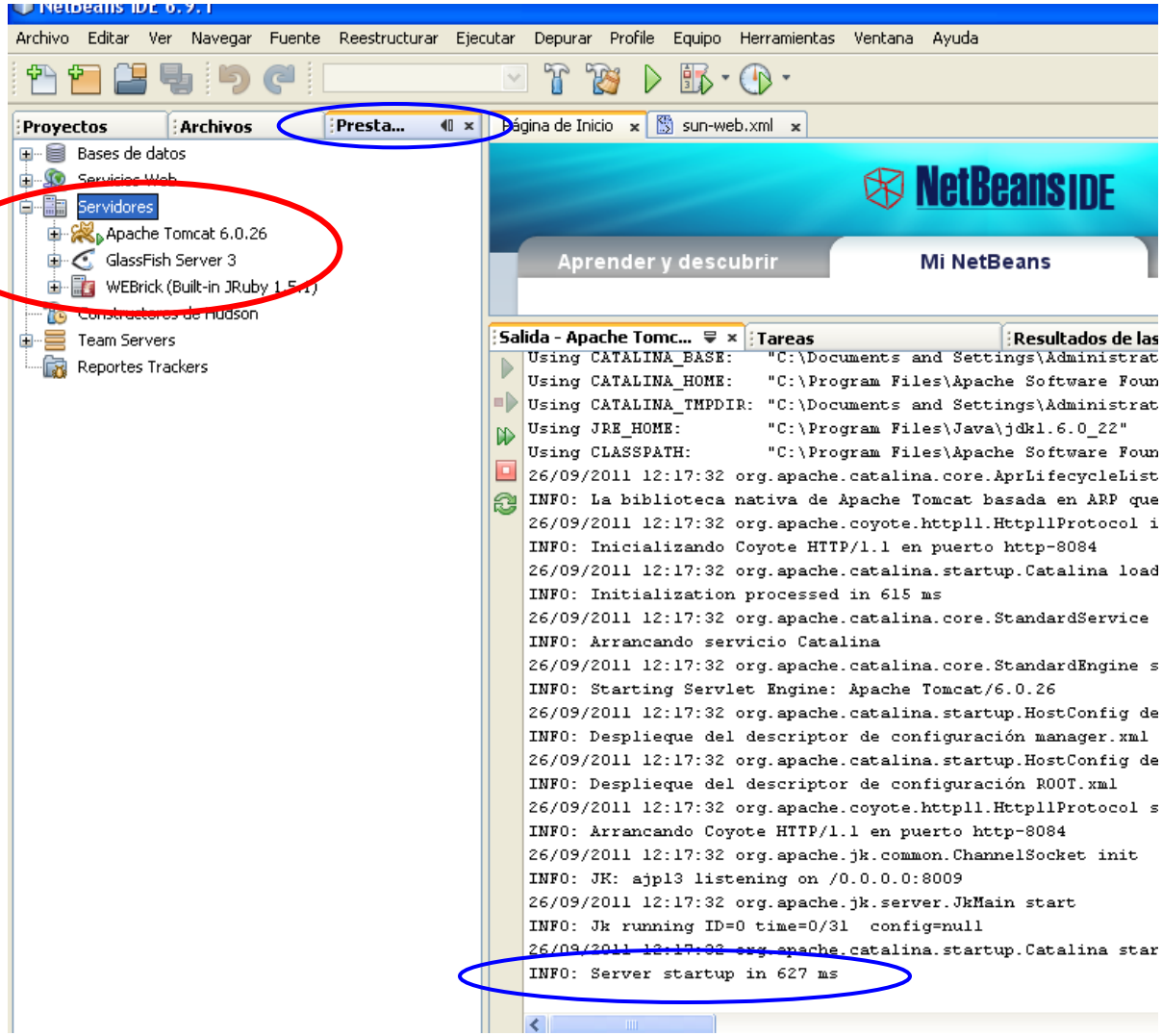
5. Desplegar la aplicación

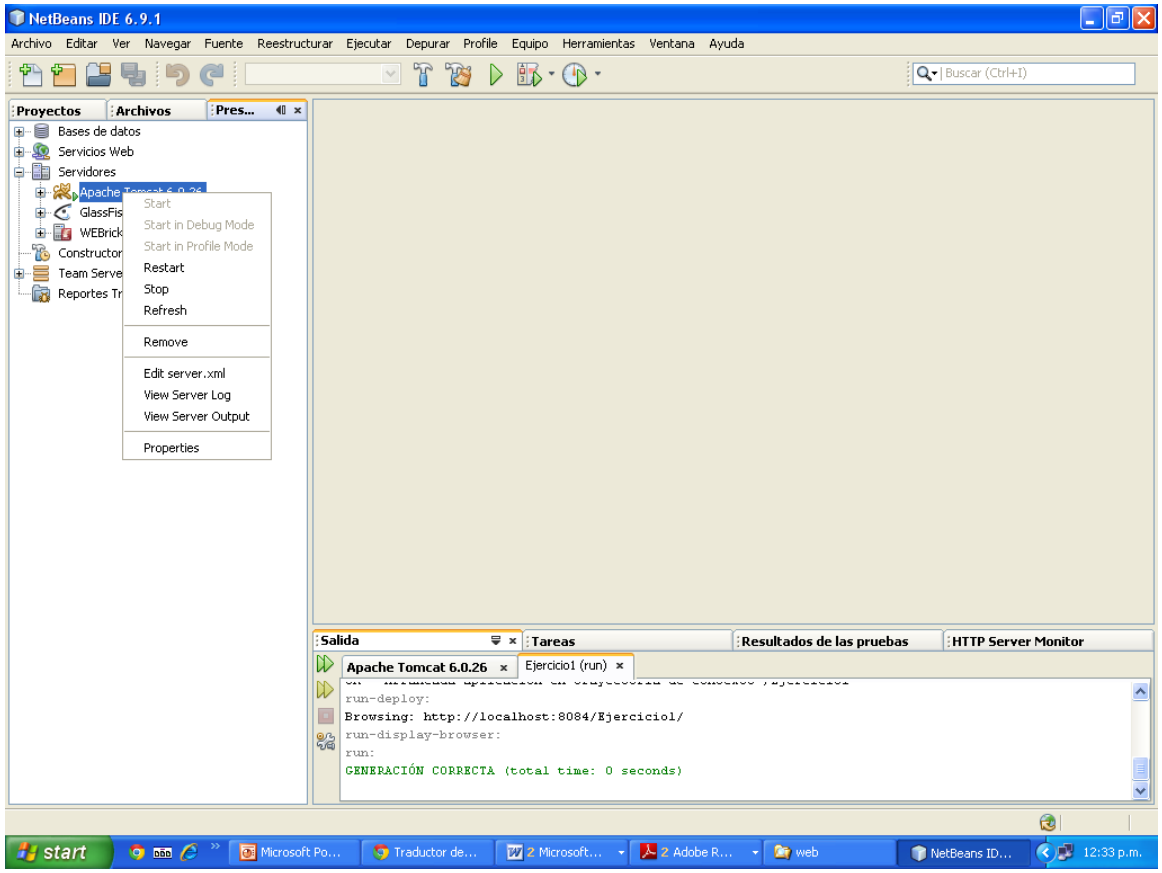
La aplicación puede ser desplegada de dos maneras diferentes:

- archivo *.war
- Carpeta descomprimida con el mismo formato del archivo *.war

Usar el archivo *.war cuando sea desplegado desde una máquina remota

Contenedor iniciado




NetBeans IDE 6.9.1
Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Depurar Profile Equipo Herramientas Ventana Ayuda
Buscar (Ctrl+I)
Proyectos
Archivos
Pres...
Bases de datos
Servicios Web
Servidores
Apache Tomcat 6.0.26
GlassFis
WEBrick
Constructor
Team Serve
Reportes Tr
Start
Start in Debug Mode
Start in Profile Mode
Restart
Stop
Refresh
Remove
Edit server.xml
View Server Log
View Server Output
Properties
Salida
Tareas
Resultados de las pruebas
HTTP Server Monitor
Apache Tomcat 6.0.26
Ejercicio1 (run)
run-deploy:
Browsing: http://localhost:8084/Ejercicio1/
run-display-browser:
run:
GENERACIÓN CORRECTA (total time: 0 seconds)
start
Microsoft Po...
Traductor de...
2 Microsoft...
2 Adobe R...
web
NetBeans ID...
12:33 p.m.

URL de la petición HTTP y del componente

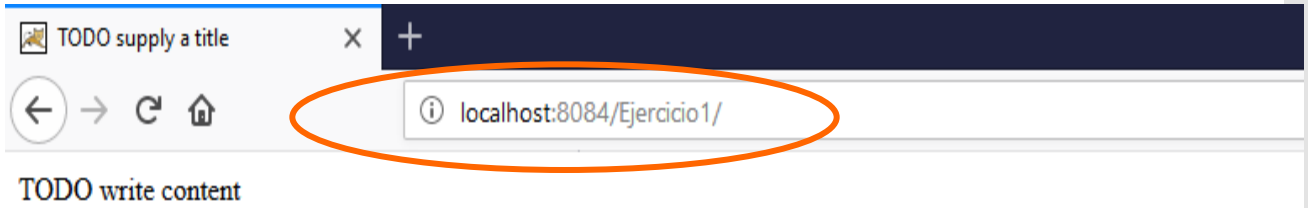
Punto de acceso al recurso web:

http://[host]:[puerto]/[contexto]/[componente]

Ej.:

http://localhost:8084/ejercicio1

IDE NetBeans



Recursos

- Tutorial JAVA
- JDK24 API
- Documentación API EE7
- API documents JEE6
- Tutorial JEE7

... manos
a la obra ...

1. Preparar plataforma.
2. Crear un proyecto web.
3. Construir un formulario que permita crear una cuenta con nombre, correo y contraseña.
4. Validar:
   - Nombre: Ver patrones.
   - Correo: ver type.
   - Contraseña: ver password.
5. Compilar y desplegar.

NOTA: Obtener una solución desde Html5.

1. Java Standard Development Kit (JDK™).

   (Es conveniente agregar la variable de entorno:

   JAVA_HOME : C:\Program Files\Java\jdk-24 y agregar a **PATH:** %JAVA_HOME%\bin)

2. **NetBeans**. Como mínimo se necesita la tecnología **Java web y EE**, y **Glassfish** y/o **Apache.**

Ayuda:

https://projects.apache.org/project.html?tomee

https://tomee.apache.org/download.html