



# Programación Orientada a Servicios

## Introducción a las Aplicaciones Web

Programa de  
**Ingeniería en Computación**  
UAM – Azcapotzalco

A cargo de:  
Dra. Maricela Claudia Bravo Contreras  
[mcbc@correo.azc.uam.mx](mailto:mcbc@correo.azc.uam.mx)

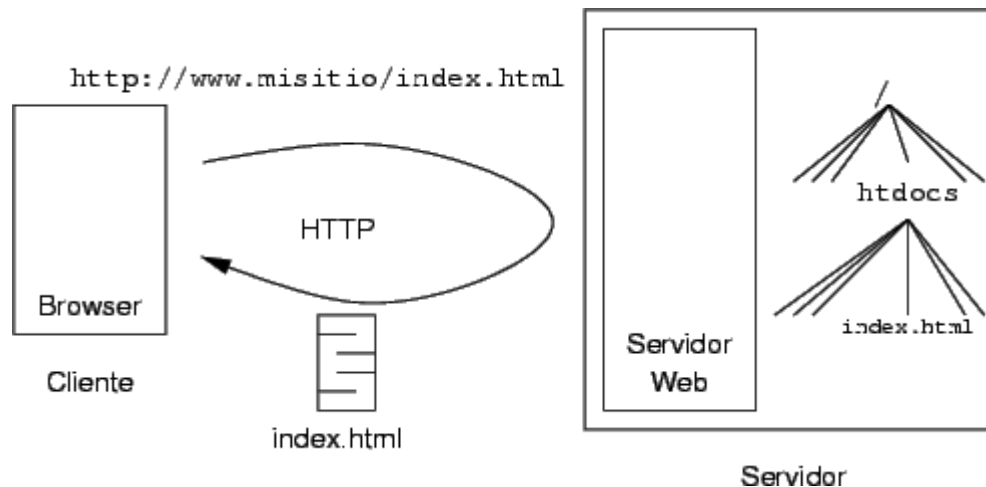
# ¿Qué es una aplicación Web?

- Una aplicación Web genera páginas Web interactivas que contienen varios tipos de lenguajes de marcas (HTML, XML) y contenido dinámico.
- Una aplicación Web basada en Java normalmente contiene componentes Web:
  - Java Server Pages (JSPs)
  - Servlets
  - Java Beans

# Arquitectura de las aplicaciones Web

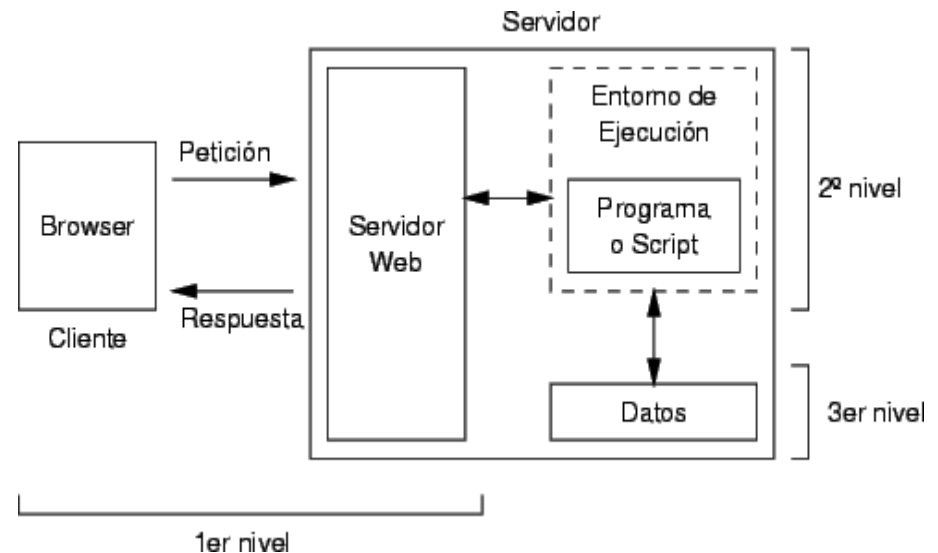
## Arquitectura de dos niveles:

Es la más simple, se tiene el nivel del “Cliente” y el nivel del “Servidor”.



# Arquitectura Web de tres niveles

- El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado.
- El segundo nivel está referido habitualmente a algún tipo de programa o *script*.
- El tercer nivel proporciona al segundo los datos necesarios para su ejecución.



# Evolución en el desarrollo de aplicaciones Web

- Inicialmente, únicamente información estática.
  - HTML estático
  - Aplicaciones “pobres”
- Posteriormente, ejecución de código en el lado del cliente:
  - HTML dinámico: JavaScript
  - Principalmente mejoras de interfaz

# Evolución en el desarrollo de aplicaciones Web

- **Programación en el lado del servidor:**
  - CGIs
  - Páginas ASP
  - Páginas JSP, servlets
  - PHP
  - J2EE, .NET
  - Zope, Ruby on Rails
  - ...

# Lenguajes de programación del lado del cliente

- Los programas del lado del cliente están incluidos dentro de la página HTML, se descargan del servidor junto con este.
- Los programas se ejecutan dentro del ámbito del browser.

# Tecnologías y lenguajes del lado del cliente

1. Navegadores para Web.
2. HTML.
3. Javascript y Vbscript.
4. Applets en Java.
5. Flash (lenguaje ActionScript).
6. XML.
7. PDF.
8. AJAX, acrónimo de **A**synchronous **J**ava**S**cript **A**nd **X**ML



# Algunos navegadores Web

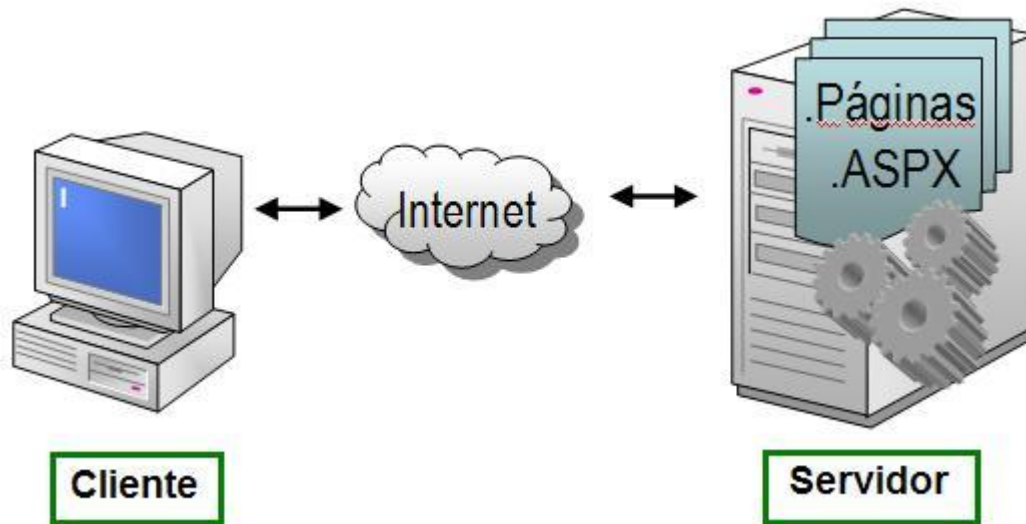
1. Amaya
2. Epiphany
3. Galeon
4. Internet Explorer
5. Konqueror
6. Lynx
7. Mozilla suite navigator
8. Mozilla Firefox
9. Netscape Navigator
10. Opera
11. Safari
12. Shiira
13. Maik Navigator

# Tecnologías y lenguajes del lado del cliente

- Algunos de estos lenguajes y tecnologías requieren de un programa especial (plug-in) instalado en la computadora del usuario. Ejemplo: Adobe Flash Player.
- Un **complemento** (o **plug-in** en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

# Lenguajes de programación del lado del servidor

- Se ejecutan en el servidor de Web y son dependientes de la plataforma del servidor.

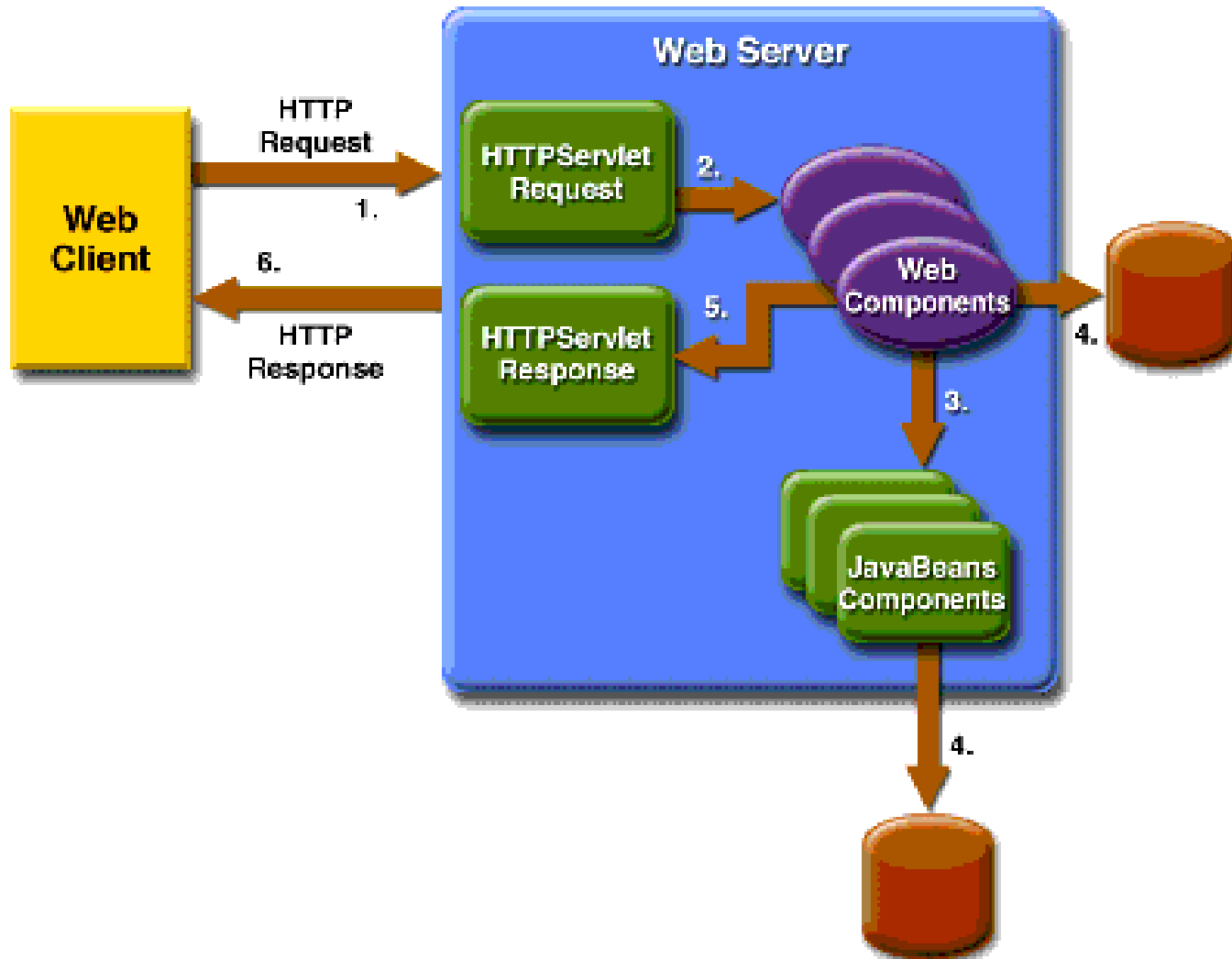


- Se usan para acceder a recursos del servidor, como bases de datos y generación de contenido dinámico para las páginas.

# Lenguajes de programación del lado del servidor

Algunos ejemplos de lenguajes del lado del servidor:

- ASP,ASP.NET (son tecnologías, soportan diferentes lenguajes como VB, C#, C++, etc.).
- PHP.
- JSP.
- Perl.
- Ruby.
- Python.
- XML.

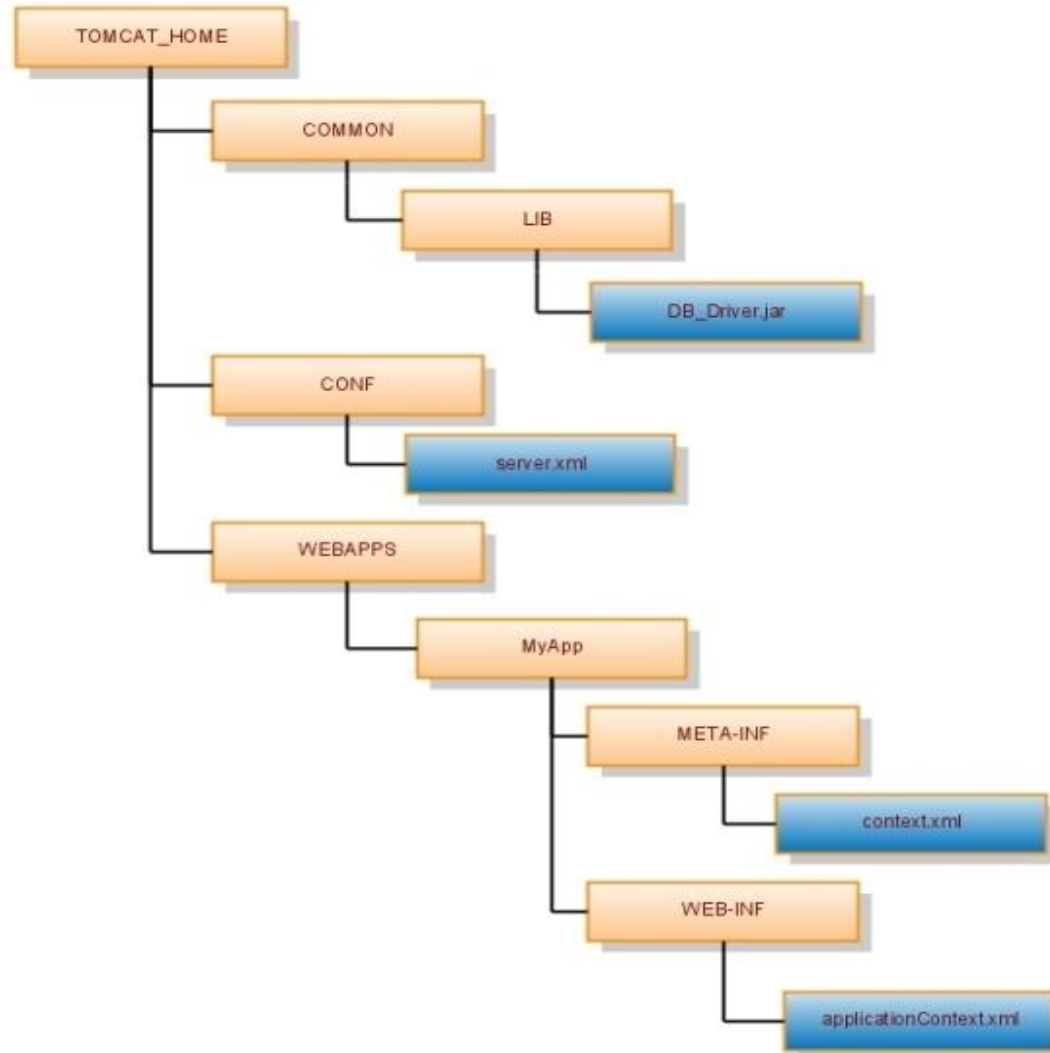


Estructura y configuración



# APACHE TOMCAT

# Directorio de Tomcat





**J2EE**

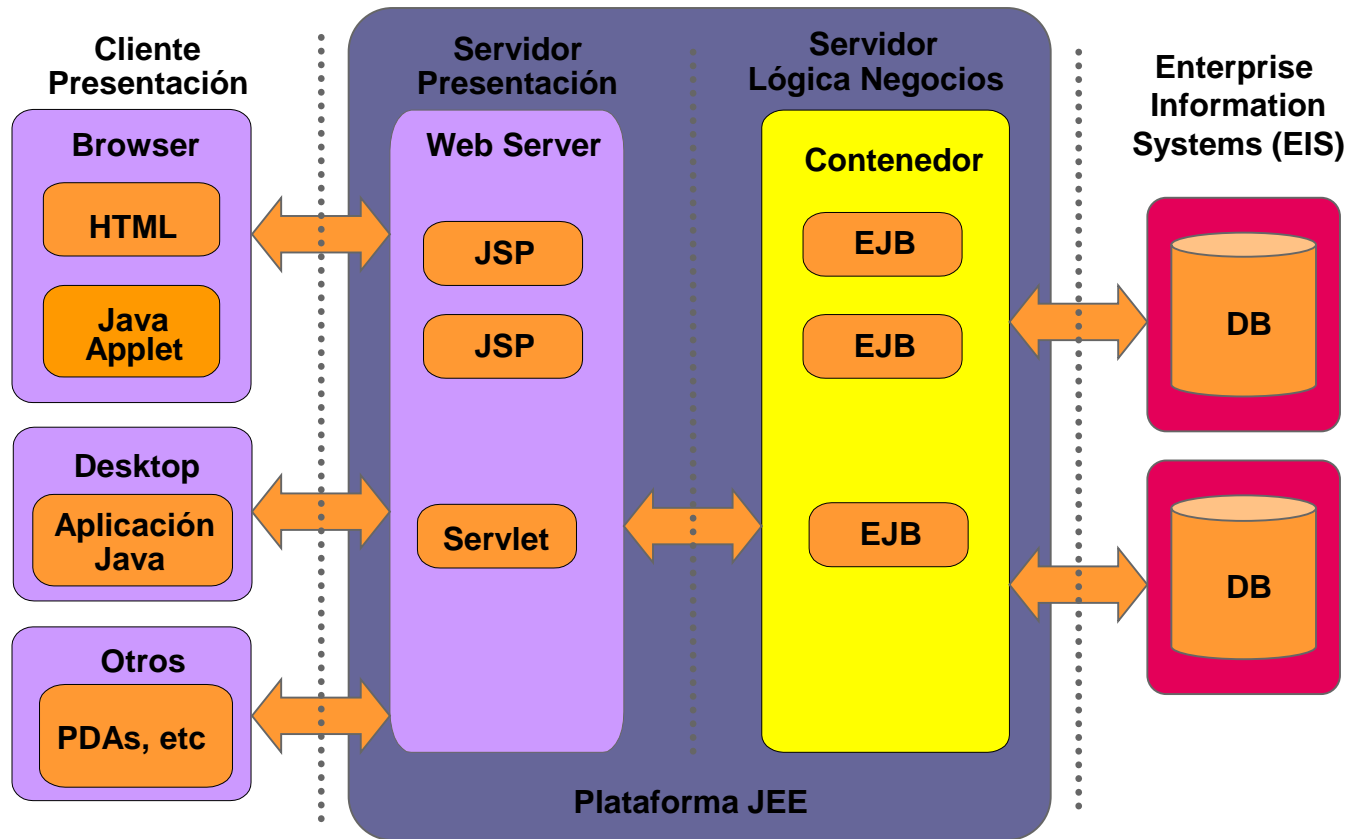


# Java Enterprise Edition

- Es una plataforma de desarrollo ampliamente utilizada que ofrece un conjunto de tecnologías que permiten reducir el costo y la complejidad del desarrollo, despliegue y manejo de aplicaciones de múltiples capas.
- Entre los componentes fundamentales de Java EE se encuentran:
  - Enterprise Java Beans (EJB)
  - Java Persistence API (JPA)

# JEE Arquitectura Multi-capas

- Presentación, lógica de negocios y datos bien separados



# Qué es JEE?

- JEE es un conjunto de tecnologías para el desarrollo de aplicaciones multi-capas
  - Es una especificación que se implementa en Servidores de Aplicación.
  - Las aplicaciones desarrolladas en JEE pueden ejecutarse en diferentes Servidores de Aplicación, sin modificar el código.
  - Define la interacción y API entre diferentes servicios de base como Directorios, Transacciones, Bases de Datos, Mensajería y otros sistemas empresariales.
  - Elimina la necesidad de codificar los servicios de base dentro de la aplicación:
    - Facilita el desarrollo
    - Reduce el tiempo de desarrollo
    - Evita la introducción de errores por estos servicios

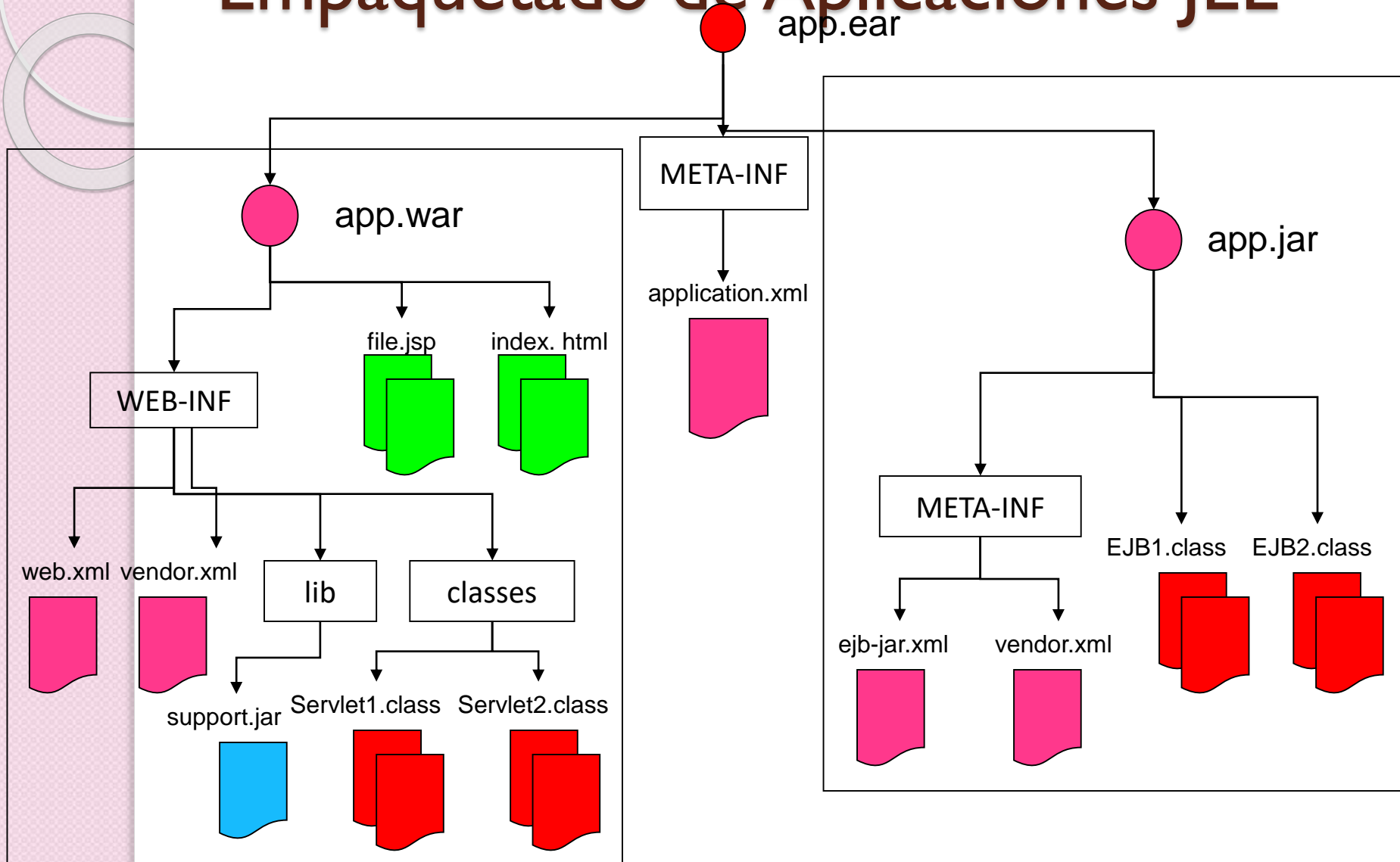
# Objetivos de JEE

- Define una plataforma para construir aplicaciones Servidor portables
  - Las aplicaciones pueden ser instaladas en diferentes Servidores de Aplicación sin cambiar el código.
  - Todos los Servicios comunes se acceden de una forma estándar:
    - Transacciones
    - Seguridad
    - Directorio
    - Mensajería
    - Persistencia
    - Concurrencia
    - Ciclo de Vida
  - Antes, las aplicaciones tenían que manejar esto en su propio código o usar mecanismos propietarios.
- Estandariza la forma de empaquetar aplicaciones e instalarlas
  - Uso de Java Archive (JAR),
  - Web Archive (WAR),
  - Enterprise Application Archive (EAR) y otros.
- Favorece la configuración externa por sobre el código
  - Descriptores en XML favorecen el cambio sin recompilación de código
  - Anotaciones en código fuente

# Tecnologías usadas

- EJB
- Servlets, JSP, tags, JSF
- JMS
- Java Transaction
- Java Mail
- JAX (SAX, DOM, XSLT)
- Web Services (SOAP, WSDL)
- JDBC, JCA
- Java Persistence
- JNDI
- Otras ...

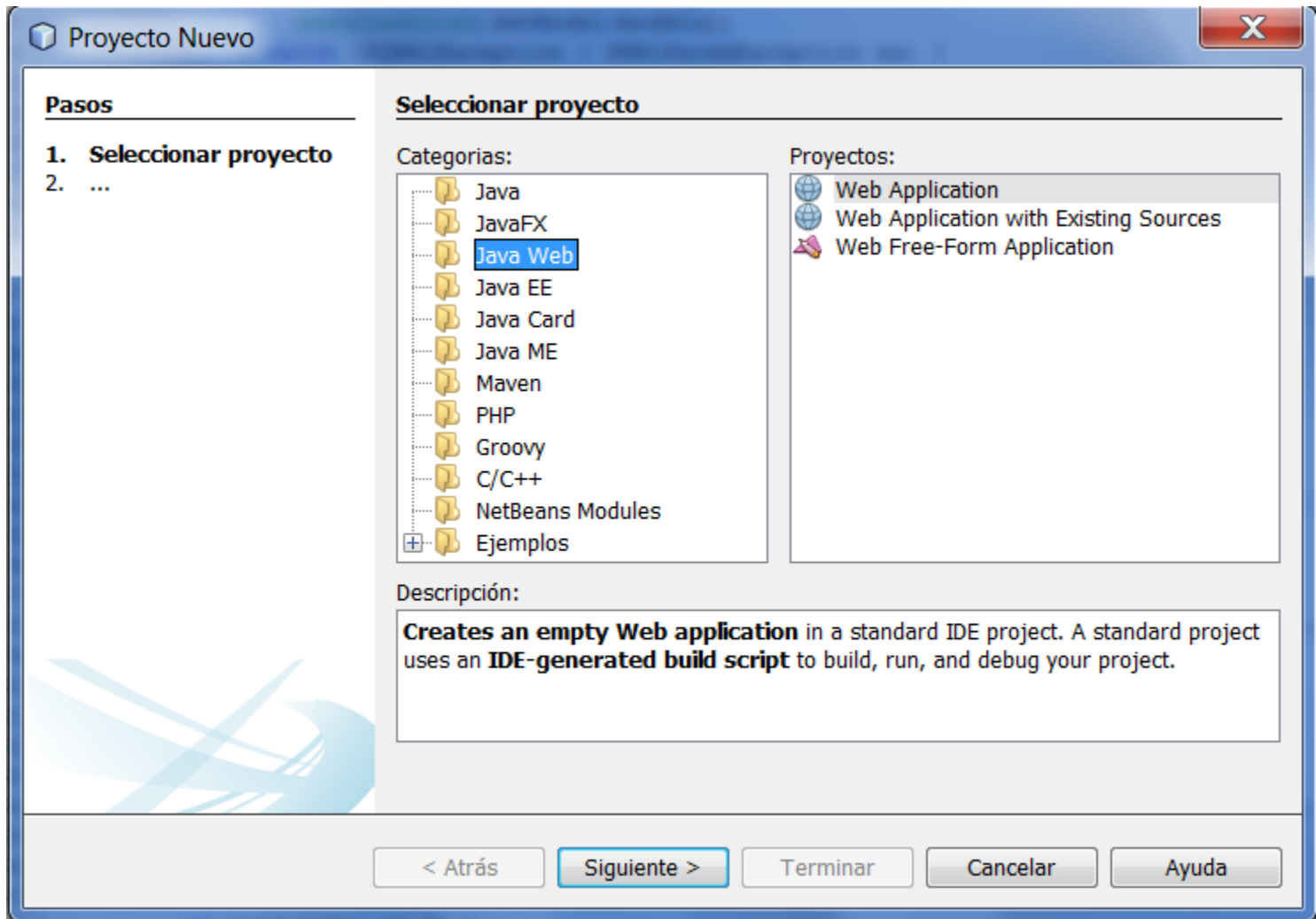
# Empaquetado de Aplicaciones JEE



# Configuración del servidor Web en NetBeans

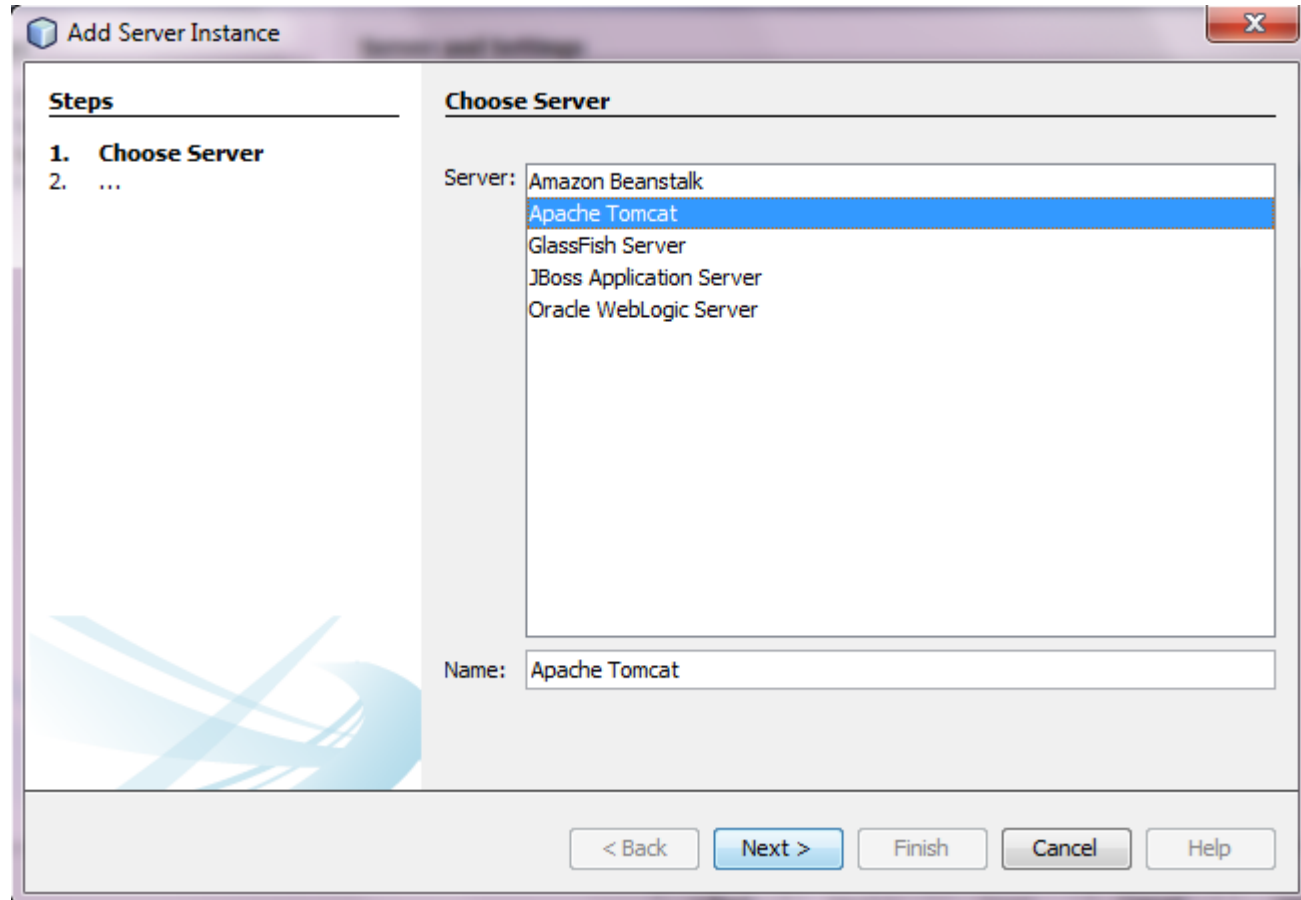
- Requerimientos:
  1. Tener instalado el JDK
  2. Tener instalada una versión de Tomcat
  3. Tener instalados los plugins de Java Web en NetBeans

# Seleccionar el tipo de proyecto: Java Web

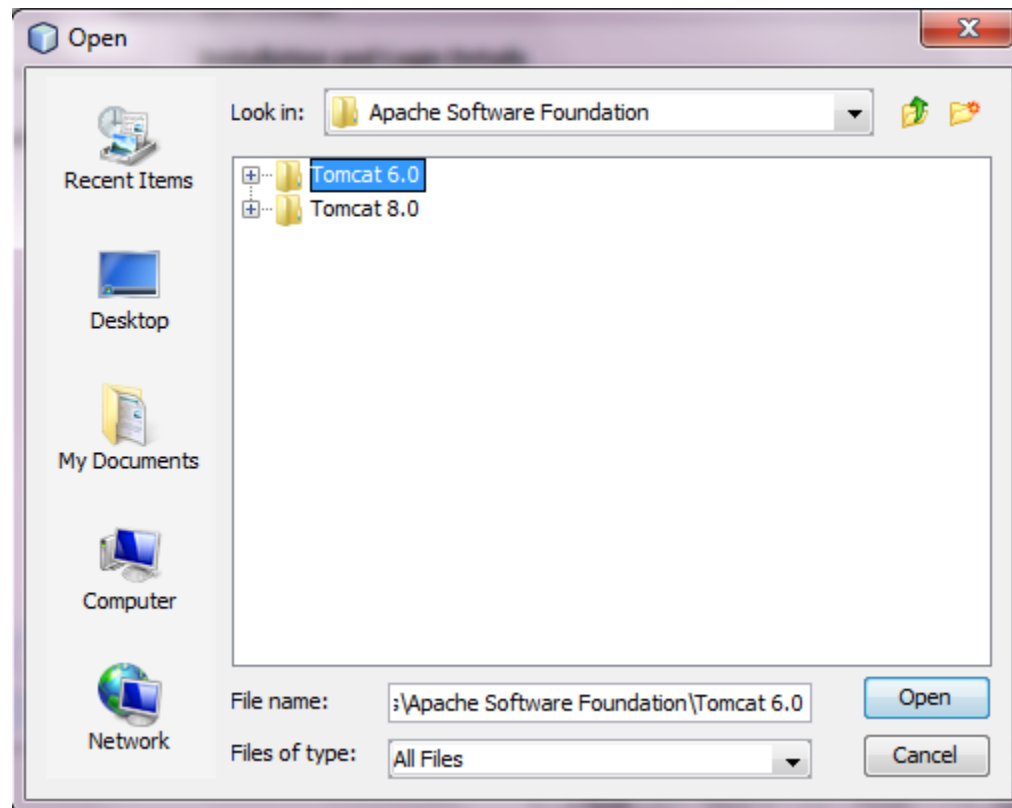




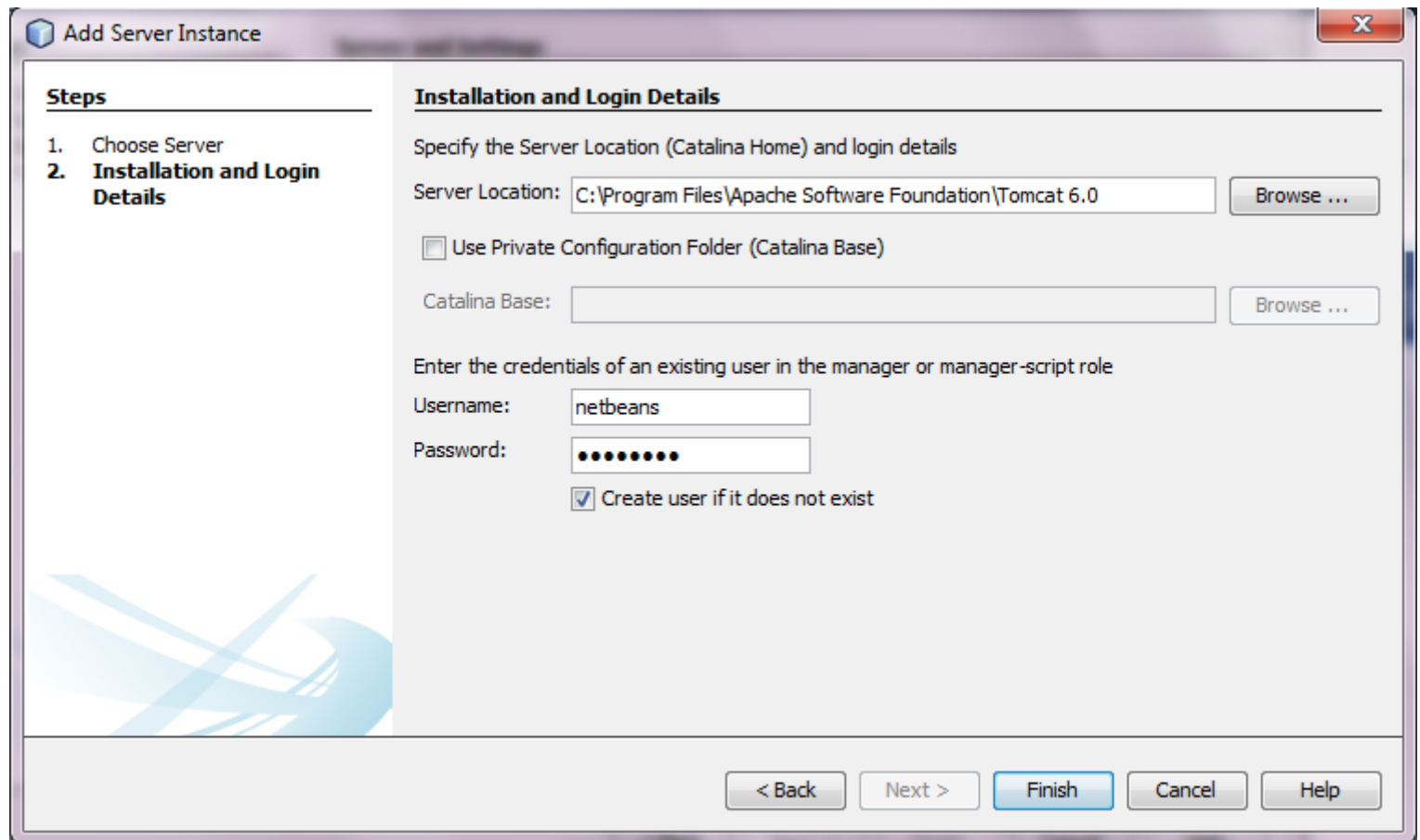
# Seleccionar Apache Tomcat como servidor Web



# Seleccionar el directorio de instalación de Tomcat



# Especificando los detalles del servidor Web Tomcat



The screenshot shows the 'Add Server Instance' dialog box in NetBeans IDE. The dialog is titled 'Add Server Instance' and has a close button (X) in the top right corner. It is divided into two main sections: 'Steps' and 'Installation and Login Details'.

**Steps:**

1. Choose Server
2. **Installation and Login Details**

**Installation and Login Details:**

Specify the Server Location (Catalina Home) and login details

Server Location:

Use Private Configuration Folder (Catalina Base)

Catalina Base:

Enter the credentials of an existing user in the manager or manager-script role

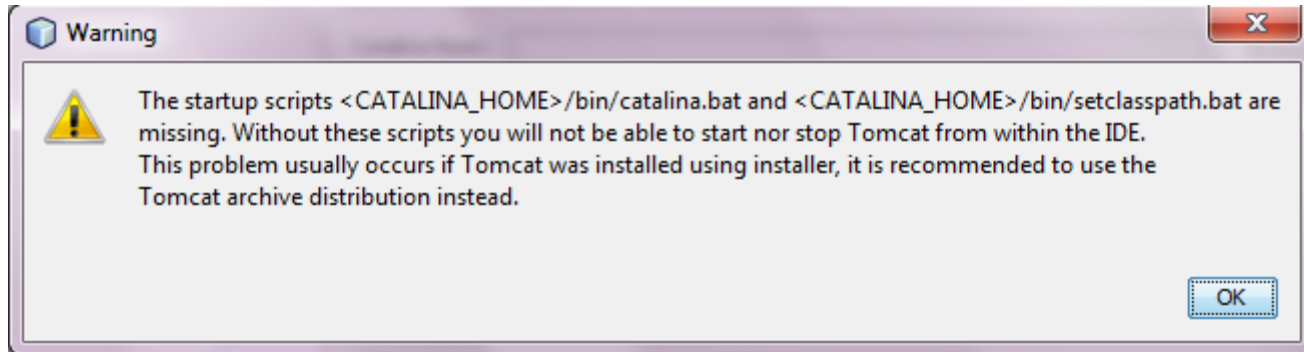
Username:

Password:

Create user if it does not exist

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted in blue.

# Error en la configuración del servidor



# Configuración del Servidor Tomcat Exitosa!

**Nuevo Web Application**

**Pasos**

1. Seleccionar proyecto
2. Name and Location
- 3. Server and Settings**
4. Frameworks

**Server and Settings**

Add to Enterprise Application: <None>

Server: Apache Tomcat (1) **Add...**

Java EE Version: Java EE 6 Web  Enable Contexts and Dependency Injection

Context Path: /WebApplication1

< Atrás    Siguiete >    Terminar    Cancelar    Ayuda

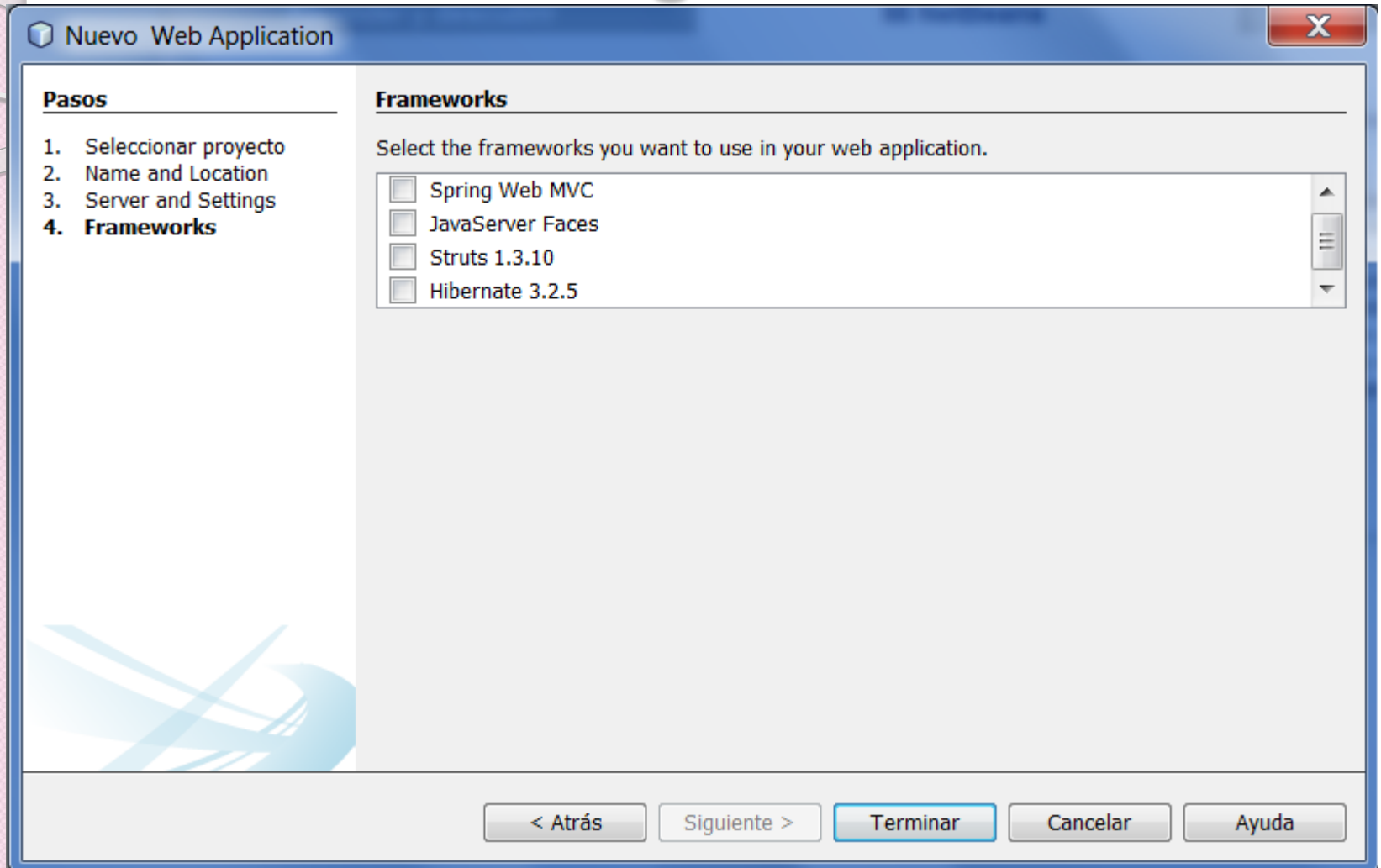
# ¿Qué es un framework?

- Es una estructura conceptual y tecnológica de soporte definido.
- Cuenta con artefactos o módulos de *software* concretos, que puede servir de base para la organización y desarrollo de *software*.
- Incluye soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

# ¿Qué es un framework?

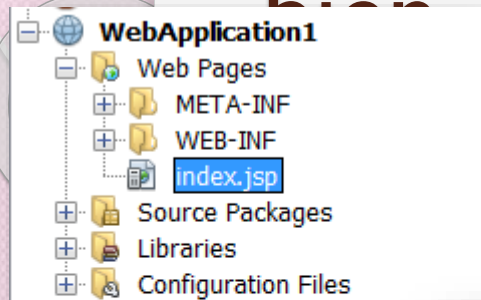
- Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

# Seleccionar algún Framework





# Si la configuración del servidor salió bien...



A screenshot of a web browser window displaying the source code of a JSP page. The browser's address bar shows 'Página de Inicio' and the page title is 'index.jsp'. The source code is displayed in a monospaced font with syntax highlighting. The code includes a JSP page directive, a DOCTYPE declaration, and an HTML document structure with a head section containing meta and title tags, and a body section containing a single h1 element with the text 'Hello World!'.

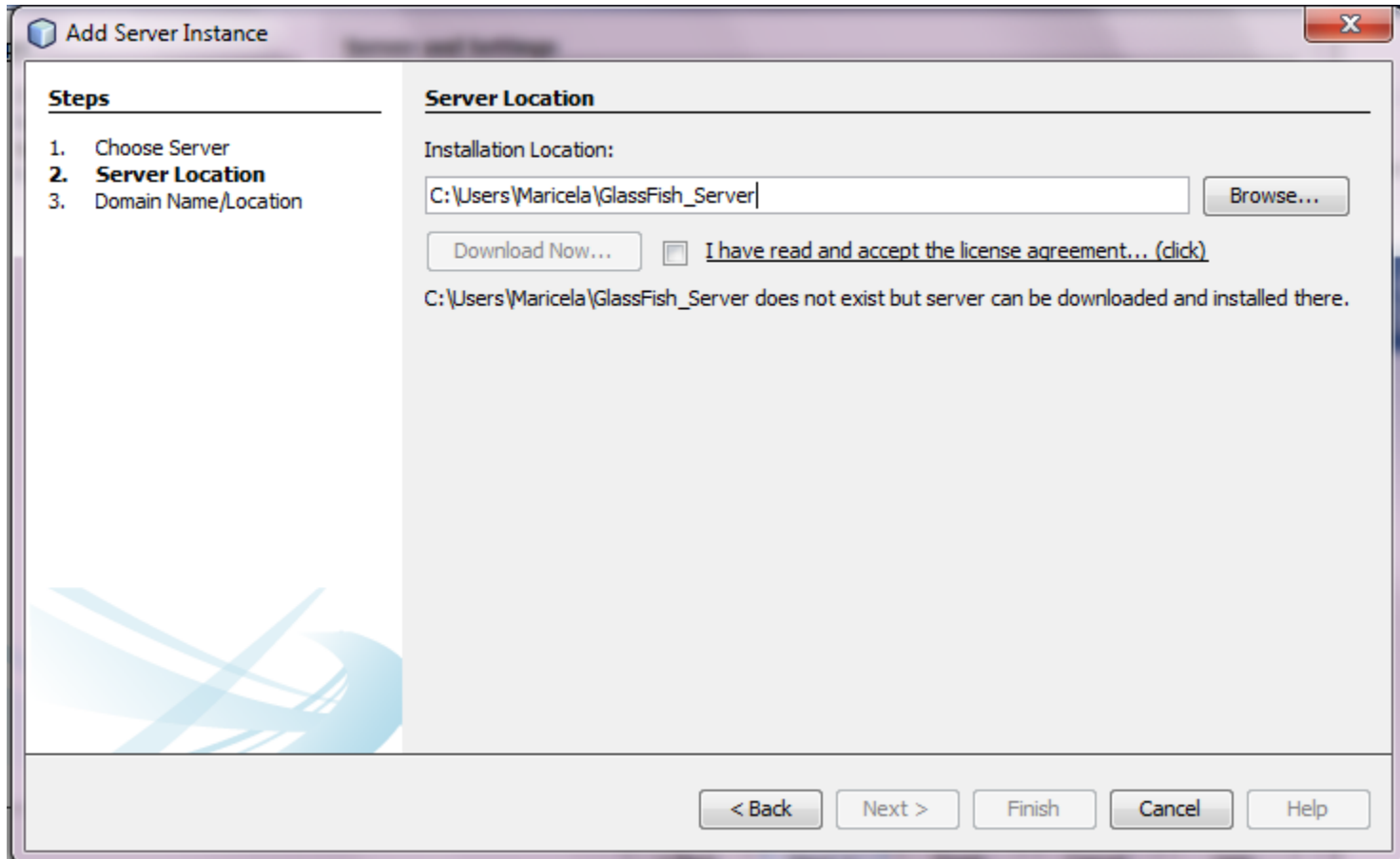
```
1 <!--
2 Document : index
3 Created on : 02-sep-2013, 0:28:19
4 Author : user
5 -->
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <h1>Hello World!</h1>
16 </body>
17 </html>
18
```



Cambia el texto que debe desplegar el JSP por este:

**Mi primer ejemplo!**

# Alternativamente...

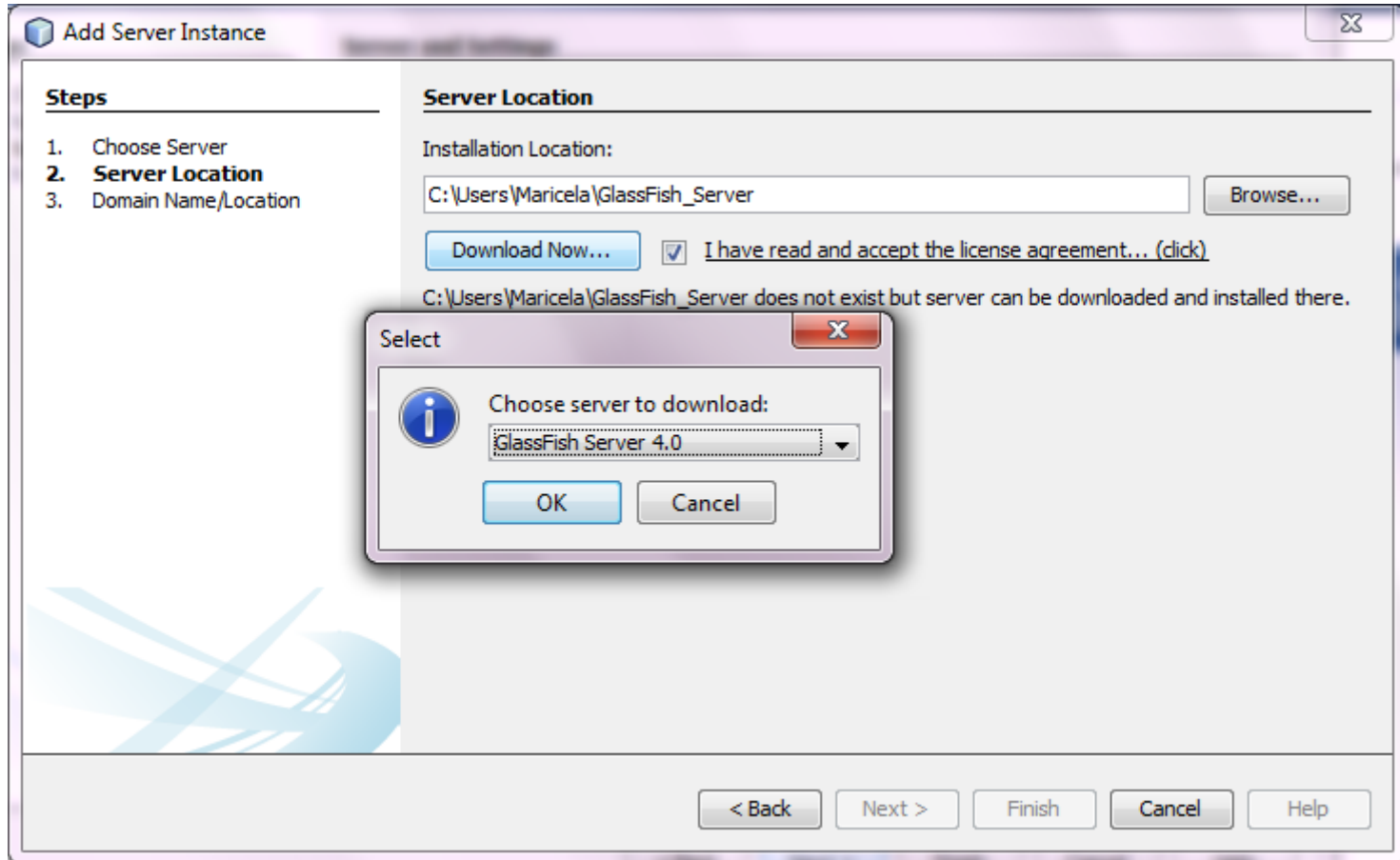


The screenshot shows a Windows-style dialog box titled "Add Server Instance". The window is divided into two main sections. On the left, a "Steps" pane lists three steps: "1. Choose Server", "2. **Server Location**", and "3. Domain Name/Location". The "Server Location" section on the right contains the following elements:

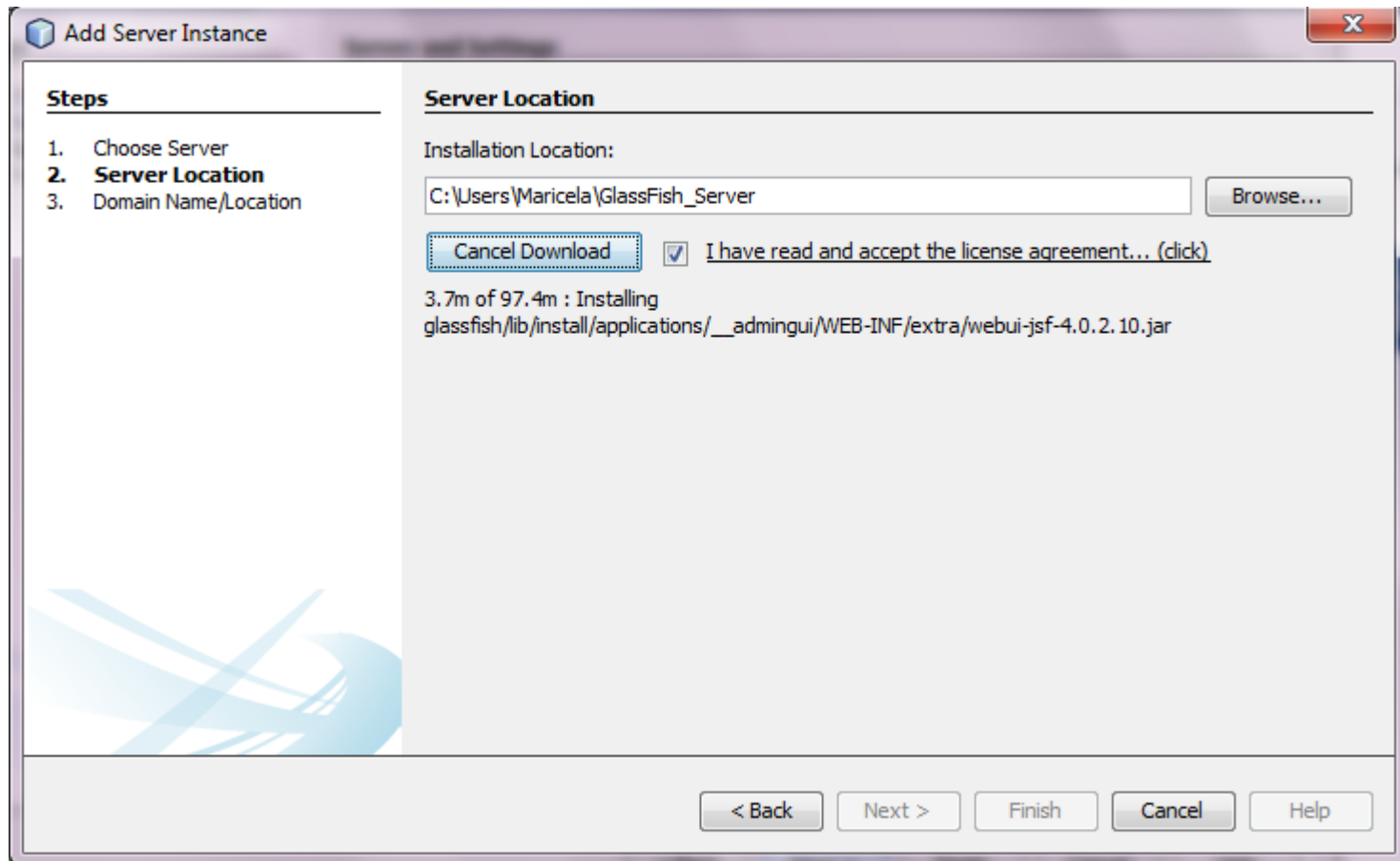
- Server Location** (Section Header)
- Installation Location: A text box containing "C:\Users\Maricela\GlassFish\_Server" and a "Browse..." button.
- A "Download Now..." button.
- A checkbox labeled "I have read and accept the license agreement... (click)".
- A message: "C:\Users\Maricela\GlassFish\_Server does not exist but server can be downloaded and installed there."

At the bottom of the window, there are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".

# Instalamos GlassFish Server



# Instalando GlassFish Server



**Add Server Instance**

**Steps**

1. Choose Server
2. Server Location
3. **Domain Name/Location**

**Domain Location**

Register Local Domain

Domain:

Use default ports (disabled due to port conflicts)

Register Remote Domain

Host Name:

Admin Port:

Target:

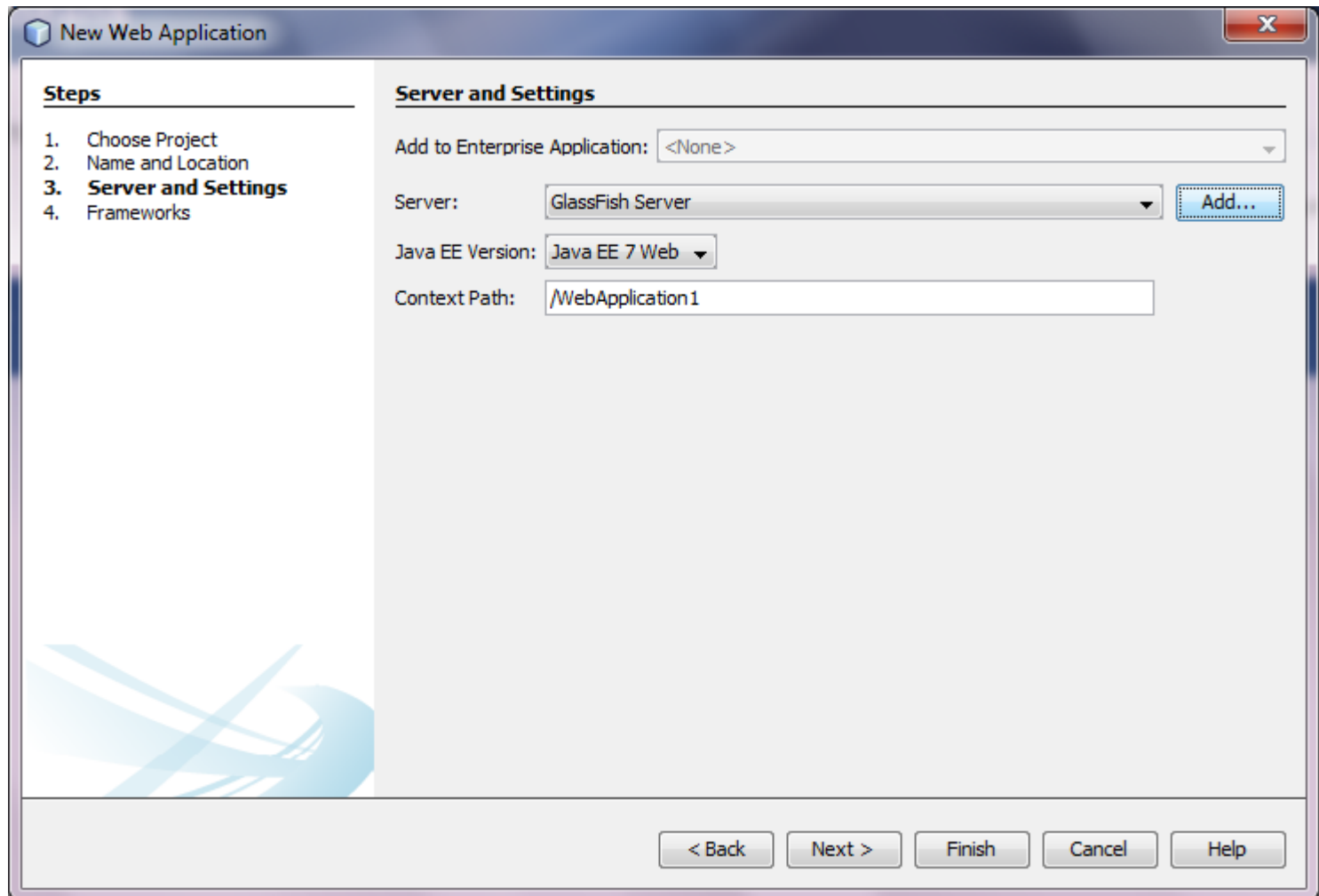
User Name:

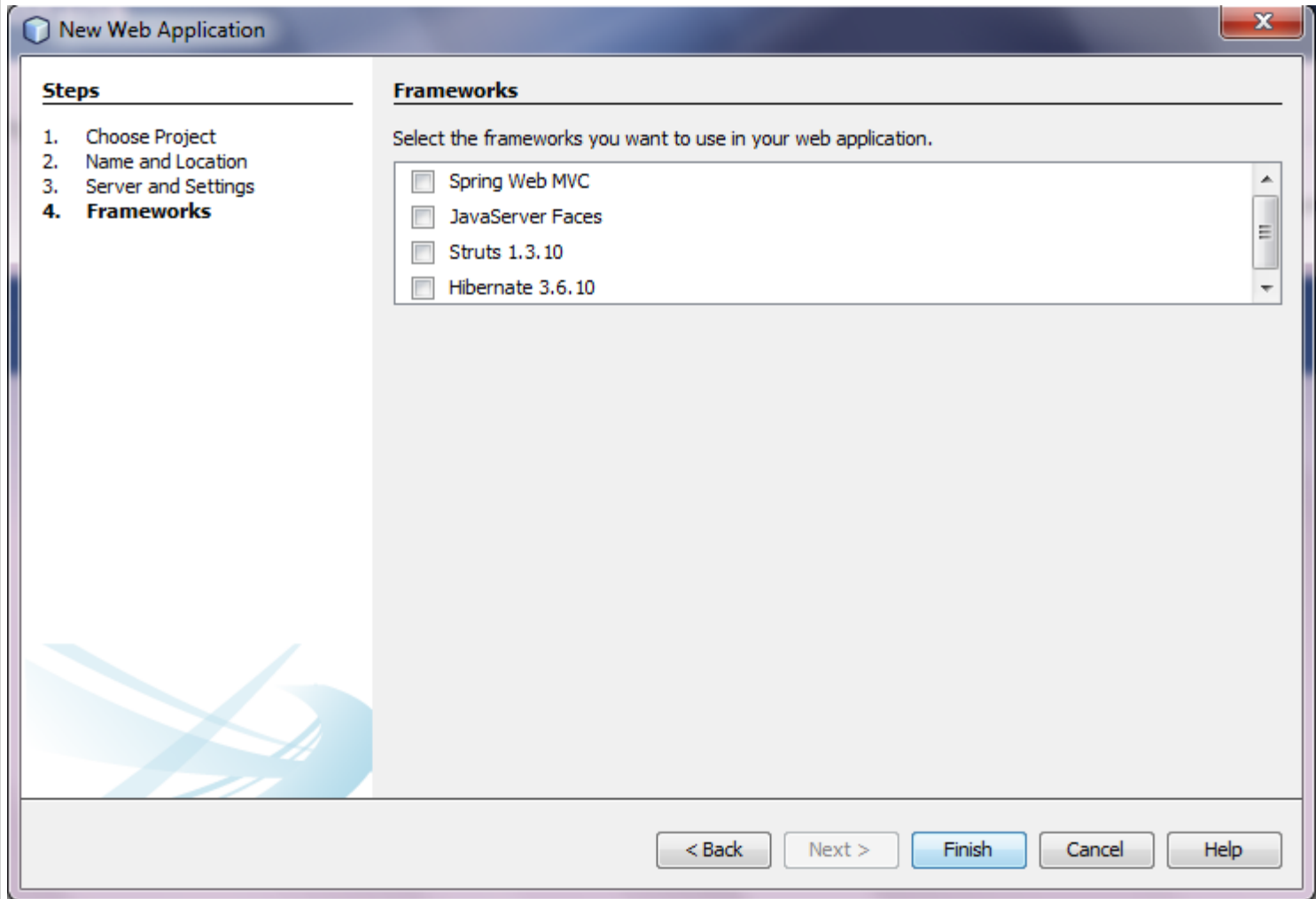
Password:

**i** Register existing embedded domain: domain1

< Back   Next >   **Finish**   Cancel   Help

# Configuración del Servidor GlassFish Exitosa!









# TAREA

Crear una nueva aplicación web en Tomcat y levantarla sin la ayuda del entorno de desarrollo.