
J2EE LA PROPUESTA DE SUN PARA ARQUITECTURA DISTRIBUIDA

Eduard Lara

INDICE

1. Introducción
2. Partes de J2EE
3. Arquitectura J2EE
4. Tecnologías J2EE
5. Montando el puzzle
6. Proveedores

1. Introducción

Actualmente existen dos alternativas:

- DNA (Microsoft)
- J2EE (Sun).

Internet cambia el entorno:

- Más requerimientos.
- Menos tiempo para desarrollar.
- Mayor capacidad de adaptación.
- Mayor portabilidad.

1. Introducción

Solución propuesta:

- Framework estándar: ayuda a construir e implantar sistemas.
 - J2EE
- Best Practices: ayudan a utilizar el framework correctamente.
 - Sun Blueprints Design Guidelines for J2EE

1. Introducción

Complejidades resueltas:

- Gestión de transacciones
- Gestión del ciclo de vida
- Pooling de recursos

Conclusión:

- Desarrollo de aplicaciones de manera más rápida y fáciles de adaptar.

2. Partes de J2EE

Aplicaciones J2EE se dividen en tres partes fundamentales:

- Componentes (*beans*)
- Contenedores
- Conectores

2. Componentes

- Los desarrolladores no se deben preocupar más que de la lógica del sistema.
- Permite un desarrollo organizado en capas de funcionalidad.
- Desarrollo basado en componentes se plantea como un ensamblaje de piezas, que se pueden reutilizar.

2. Contenedores

- Puente entre un cliente y un componente. Ofrece servicios a ambos.
- El uso de contenedores permite especificar el comportamiento de los componentes en tiempo de *deploy* (configuración, puesta en servicio).

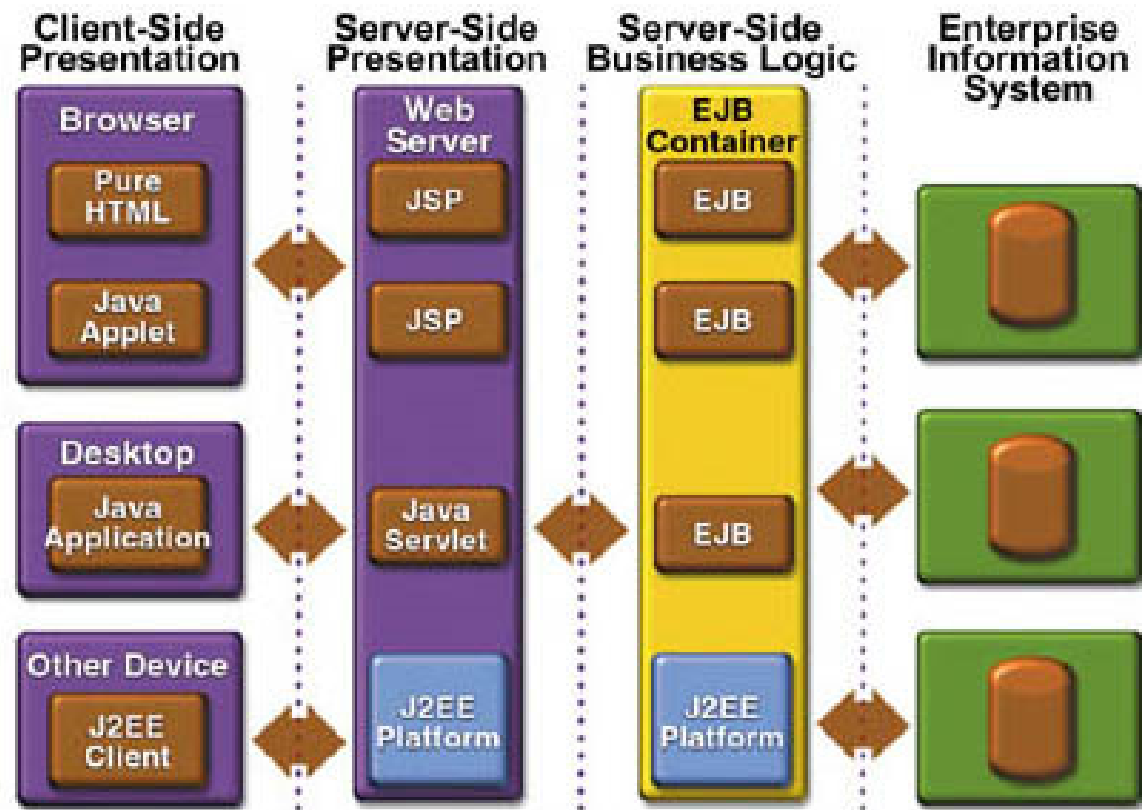
2. Conectores

- Mecanismo para interconectar EIS (Enterprise Information Systems) con los contenedores de J2EE.
- Estándar Connector Service Provider Interface.

3. Arquitectura J2EE

- La arquitectura propuesta por J2EE se compone de cuatro capas:
 - Cliente
 - Web
 - Negocio
 - Sistema de Información

2. Arquitectura J2EE



3. Arquitectura: Cliente

Existen dos alternativas en las aplicaciones J2EE:

- Basadas en Web.
 - Las aplicaciones funcionan sobre el navegador y la capa Web realiza el trabajo de presentación.
- No basadas en Web
 - Las aplicaciones conectan directamente con la capa de negocio.

3. Arquitectura: Web

- En el caso de aplicaciones basadas en Web, esta capa se encarga de la lógica de la presentación.
- Los componentes web corren en contenedores capaces de manejar el protocolo HTTP.

3. Arquitectura: Negocio

- Contiene la lógica para procesar la información:
 - Recibe información desde la capa cliente y la procesa, para luego enviarla a la capa EIS.
 - Recupera información de la capa EIS para enviársela a la capa cliente.
- Los componentes de negocio corren en contenedores que se encargan de las transacciones, multithreading, etc.

3. Arquitectura:

Enterprise Information System

- Plantea la integración de J2EE con sistemas ERP, sistemas de base de datos, y otros sistemas de información propietarios.
- Connector Architecture.

4. Tecnologías J2EE

La tecnología en la que se basa J2EE es Java.

Tecnologías por capas:

- HTML, JavaScript
- JSP, Servlets
- EJB

4. Tecnologías: Cliente

- Aplicaciones basadas en Web
 - HTML.
 - JavaScript, permite dotar de mayor dinamismo a la presentación.
 - Java, para incorporar applets (Java Plugin).
- Aplicaciones no basadas en Web
 - Java, u otros lenguajes que permitan acceder a través de CORBA a la capa de negocio.

4. Tecnologías: Web

- JSP (Java Server Pages)
 - Utilizadas en la presentación dinámica de la información al usuario.
- Servlets
 - Recomendable para el desarrollo de funciones de más bajo nivel y que no van a requerir ser modificadas frecuentemente.

4. Tecnologías: Negocio

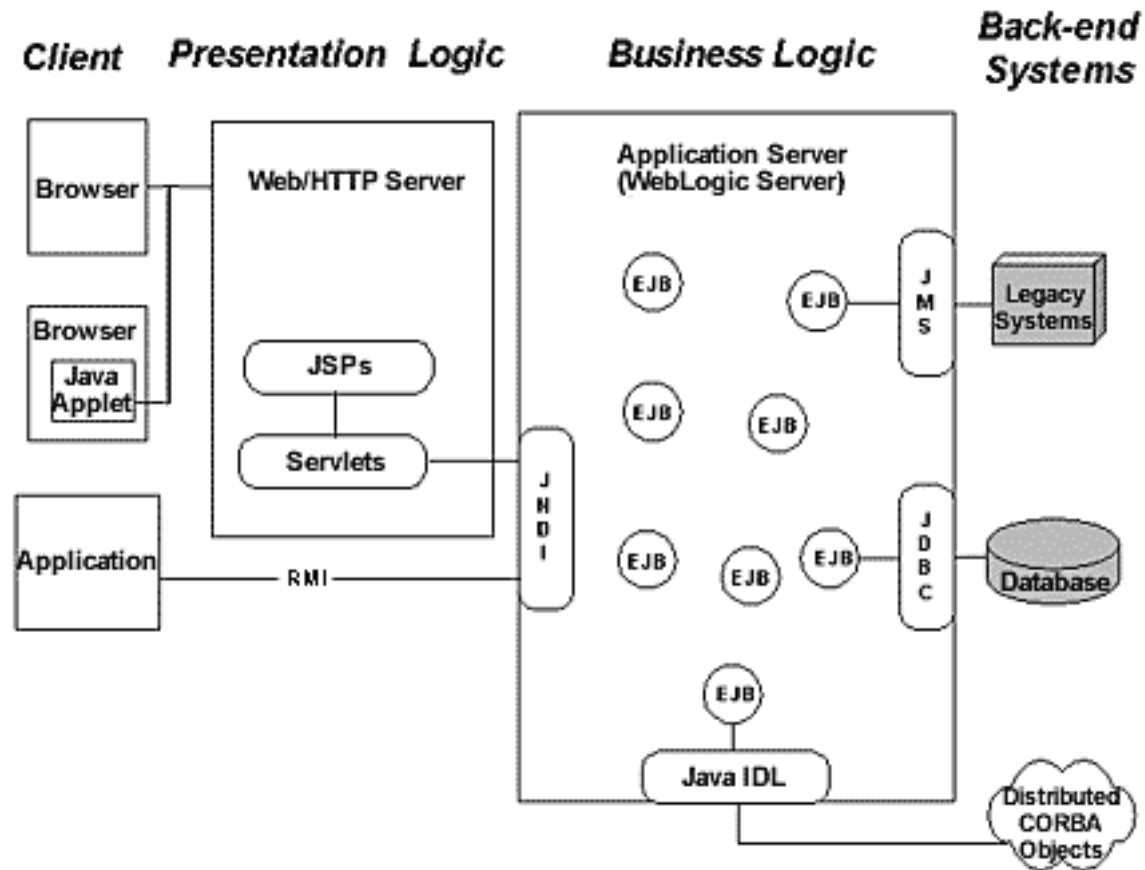
- EJB (Enterprise JavaBean)
 - Session beans
 - Stateless, no mantiene información relativa al cliente. Ejemplo: ejb que aplique una regla de negocio.
 - Stateful, mantiene información relativa al cliente. Ejemplo: carrito de la compra de una tienda virtual.
 - Entity beans. Representa datos de la base de datos y proporciona los métodos para manipularlos.

5. Montando el puzzle

Interconexión de la arquitectura:

- Cliente - Web
- Web - Negocio
- Cliente - Negocio
- Negocio - EIS

5. Montando el puzzle



5. Montando el puzzle:

Cliente - Web

- El cliente utiliza un navegador para acceder / ejecutar la aplicación, por lo tanto se conecta con un servidor web: utilización del protocolo HTTP para las comunicaciones.

5. Montando el puzzle:

Web - Negocio

- La comunicación se lleva a cabo entre los servlets y el servidor de aplicaciones: JNDI (Java Naming Directory Interface), que permite acceder de forma remota a objetos a través del nombre con el que se publican.

5. Montando el puzzle:

Cliente - Negocio

- El cliente se basa en una aplicación que se conecta directamente con el servidor de aplicaciones.
Existen dos posibilidades:
- Java, utilizando el protocolo RMI de acceso remoto.
- Otros, a través de un interfaz CORBA.

5. Montando el puzzle:

Negocio - EIS

- EIS = base de datos, esto quiere decir que en espera de la Connector Architecture, el mecanismo de conexión utilizado es JDBC.

5. Montando el puzzle:

Otros mecanismos

- JMS (Java Message Service) permite a la capa de negocio ofrecer servicios de envío / recepción de mensajes.
- JTS (Java Transaction Service, monitor de transacciones) provee de un Transaction Manager al servidor de aplicaciones.
- JavaMail es una API que permite acceder a los servidores de correo.
- XML (eXtensible Markup Language) es independiente de Java, pero en conjunción con Java permite conseguir un sistema independiente de la plataforma.

6. Proveedores

Servidores Web (JSP / Servlets):

Tomcat, Iplanet (Netscape) ...

Servidores de Aplicaciones (EJB):

WebLogic (BEA), Websphere (IBM), IAS (Inprise)

...

JDBC:

Oracle, SQLServer ...

CORBA:

Visibroker (Inprise) ...