DESARROLLO WEB

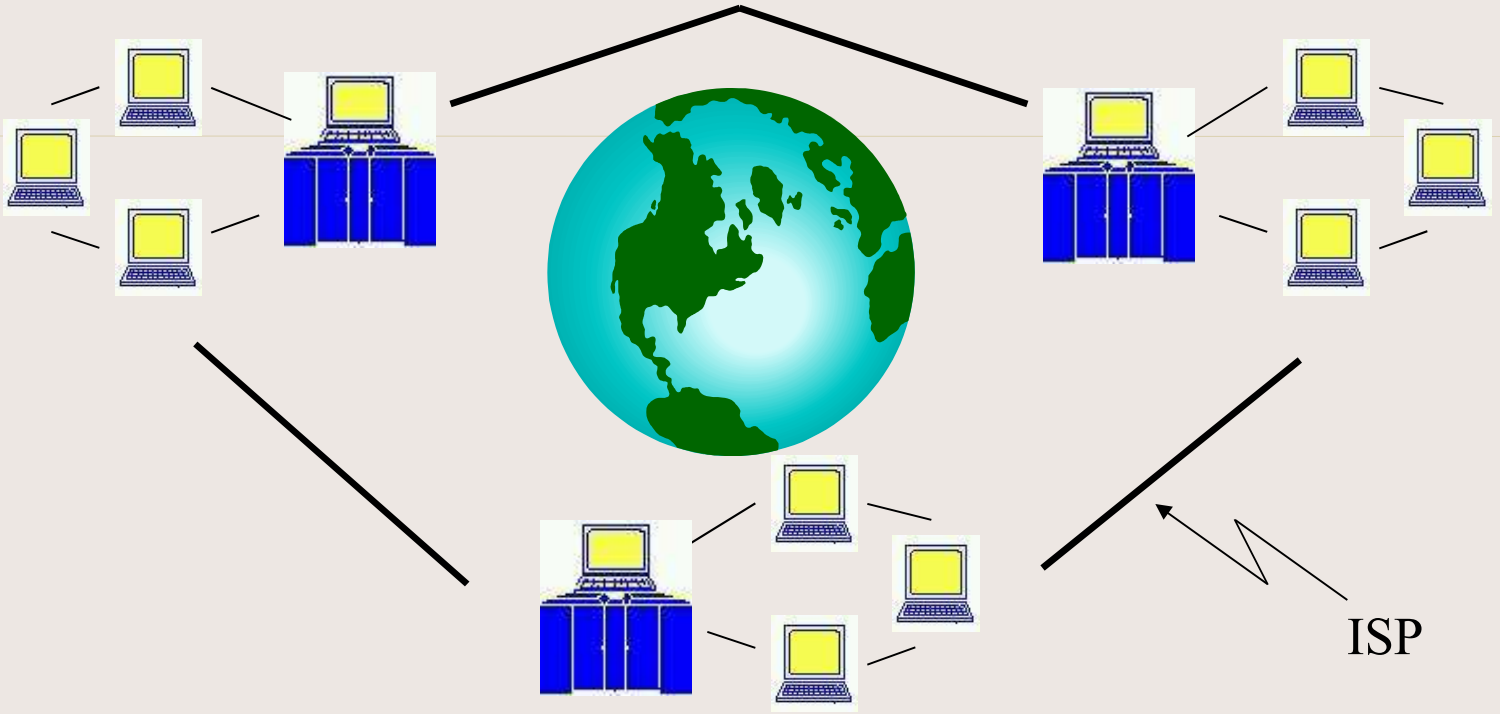
Yanin Castelazo Luna
María de la Luz Colín García

Ingeniería de Software. Febrero, 2005

DESARROLLO WEB

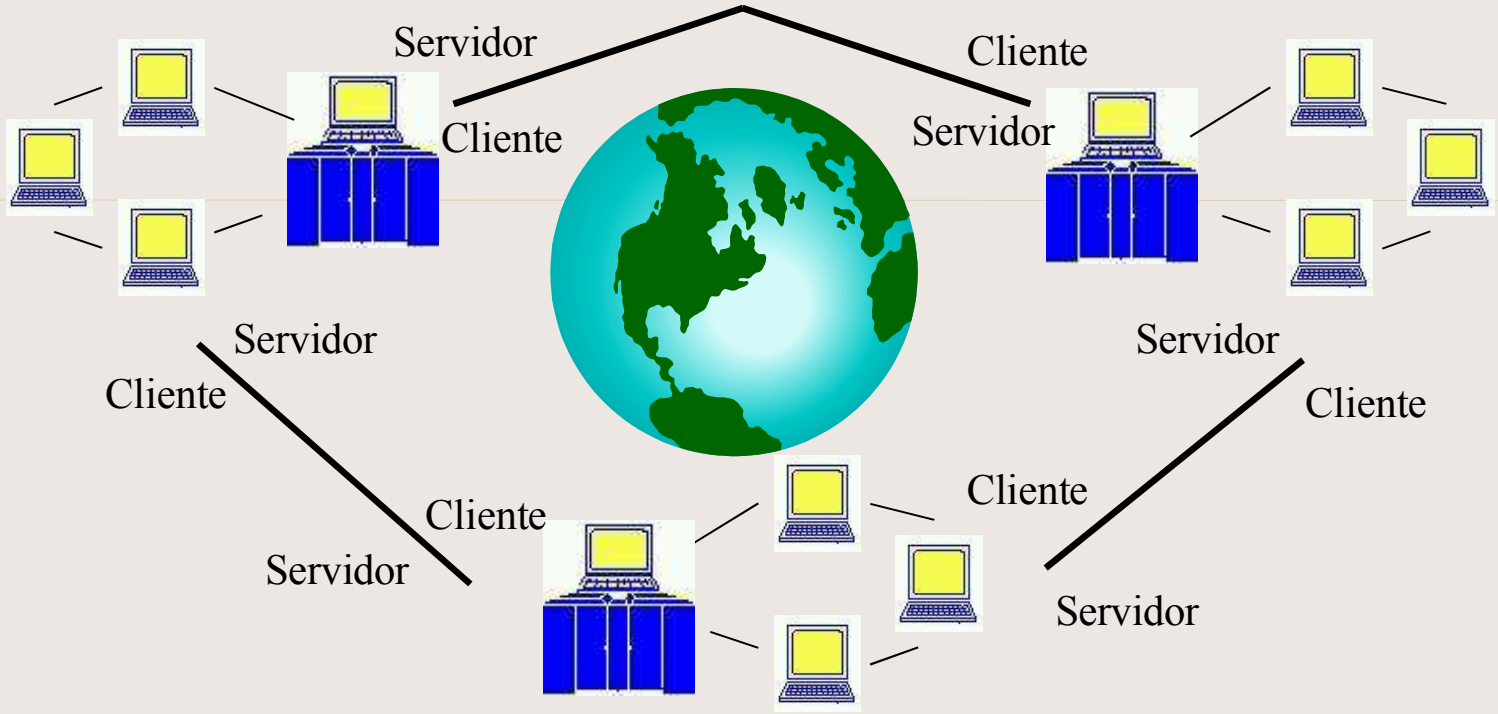
- Marco conceptual
- Particularidades
- Soluciones

* Surge: 50´s



Internet *

Colección de *redes de computadoras*, las cuales están interconectadas entre sí alrededor del mundo.



World Wide Web*

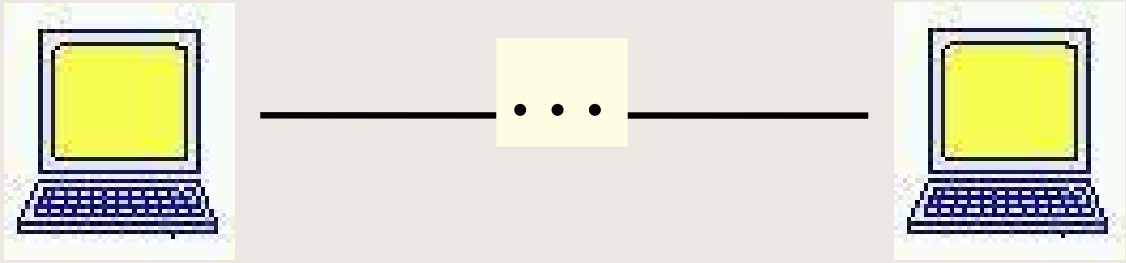
Tecnología *Cliente / Servidor* sobre la *Internet*, utilizada para acceder a una gran variedad de *información digital*.

* Surge en 1989, CERN Geneva, Switzerland

Software

Cliente
(WEB Browser)

Servidor
(Información digital)



- *Mosaic*
- Microsoft *Internet Explorer*
- *Netscape*

- *Internet Information Service* (IIS)

Información Digital

Información residente en el Servidor, ya sea en una base de datos o en archivos digitales. Esta información es a la que el Cliente del WWW desea *acceder* y quiere *desplegar*.

Puede ser de cualquiera de los siguientes formatos:

- Texto
- Gráficos
- Sonido

Protocolo

Sucesión de pasos/métodos estandarizados, llevados a cabo por los interlocutores en un sistema de comunicación.

Ejemplos:

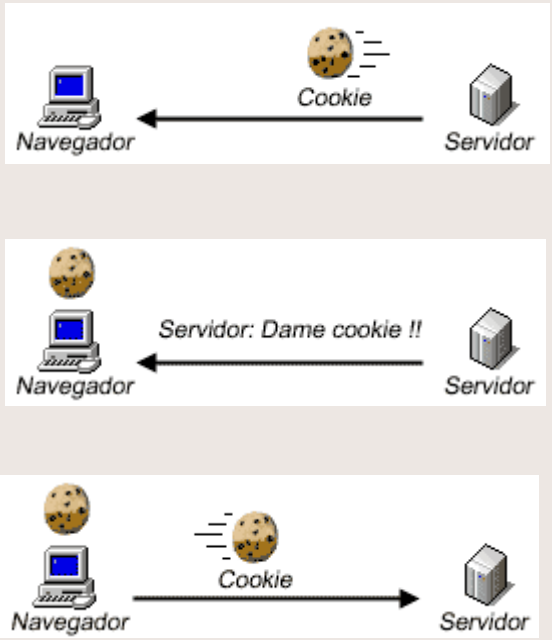
- TCP/IP
- FTP
- Telnet
- ***HTTP***

HTTP

- Protocolo de nivel de aplicación.
- Protocolo de petición/respuesta.
- Protocolo sin conexión (connectionless).
- Genérico, sin manejo de estados.

HTTP

Mecanismo alternativo para manejo de estado en HTTP: Cookies.

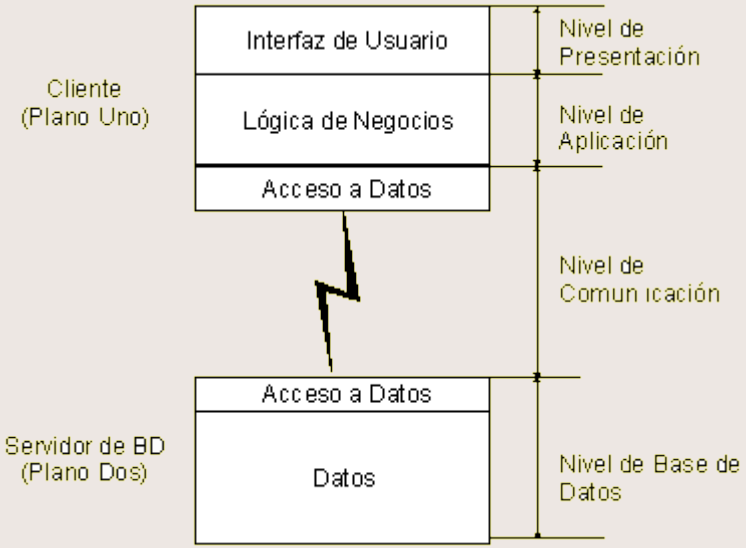


ESTRUCTURA GENERAL DE UNA APLICACIÓN WEB

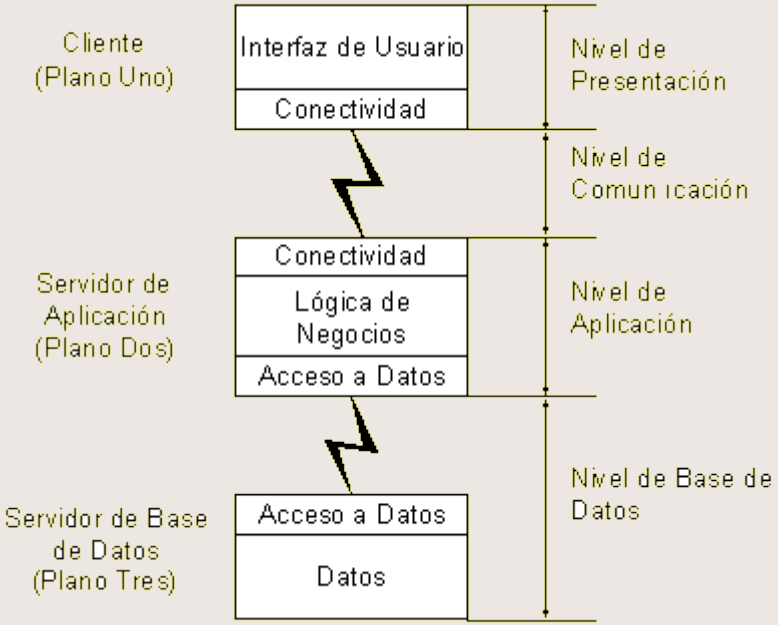
Básicamente, se pueden identificar tres partes:

- Página de contenido inicial.
- Mecanismo que permita determinar en todo momento cuál es el estatus del usuario.
- Medio de almacenamiento persistente.

Arquitectura Dos capas (Two Tier)



Arquitectura tres tercios
(Three Tier)

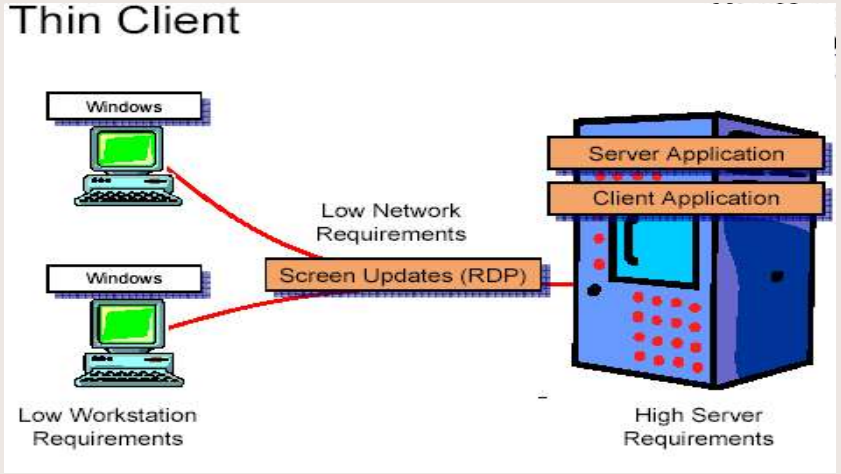


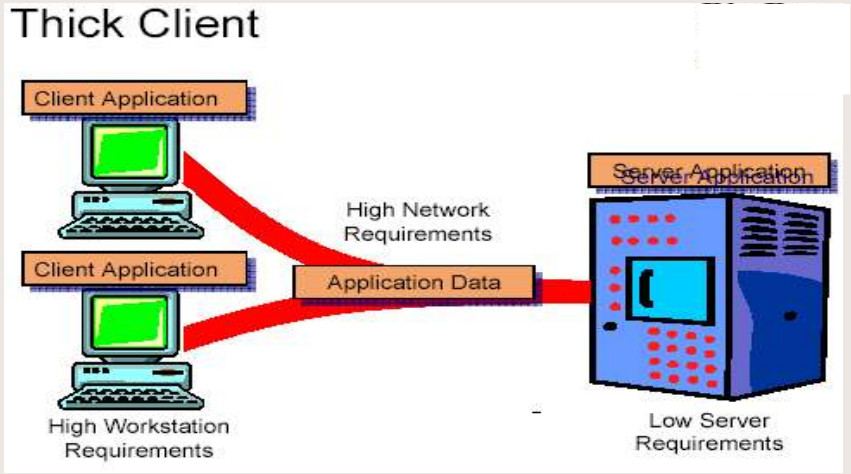
Arquitectura tres tercios
(Three Tier)

Pros	Contras
•Maximiza la reutilización de código y minimiza la duplicación del código. •Modificación de alguna capa sin afectar la las demás	•Necesidad de adquirir BD. •Necesidad de DBA •El mapeo entre objetos y BD relacionales es difícil

Clientes Web

- Cliente Ligero (Thin Client).
- Cliente Pesado (Thick Client).





Web Dinámica

- Soluciones de la primera generación.

  CGI

- Soluciones de la primera generación.

  plug-ins y a APIs

Web Dinámica

Dos tecnologías:

- Código lado servidor
  - PHP y JSP
- Código lado cliente
  - JavaScript o Applets

JSP

- Desarrollada por Sun Microsystems.

Características:

- Independientes de la plataforma .
- Velocidad y Escalabilidad.
- Etiquetas Extensibles.
- Libertad de Elección.

PHP

- Acrónimo de ***Hypertext Preprocessor.***
- ***1994 por Rasmus Lerdorf***
- Macros para ayudar en el mantenimiento de páginas web.

Características:

- Gratuito.
- Independiente de plataforma.
- Librería de funciones.
- Sencillo, sintaxis cómoda, similar a C.
- Rápido, multiplataforma, multitud de librerías reutilizables.
- OOP.
- Se puede instalar un servidor gratuito (Apache) en Linux.

PHP

- Compatibilidad con las bases de datos:

Adabas D	Ingres	Oracle (OCI7 y OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unixdbm

- MySQL y PHP
  - BD de libre distribución.
  - Creación de páginas web, de forma rápida, fácil y barata.

JAVASCRIPT

- Applets
- LiveScript
- Netscape 2.0
- versión 3.0. JAVA Script

JAVASCRIPT

- Simple
- Orientado a objetos
- Dinámico

J2EE - Java 2 Enterprise Edition

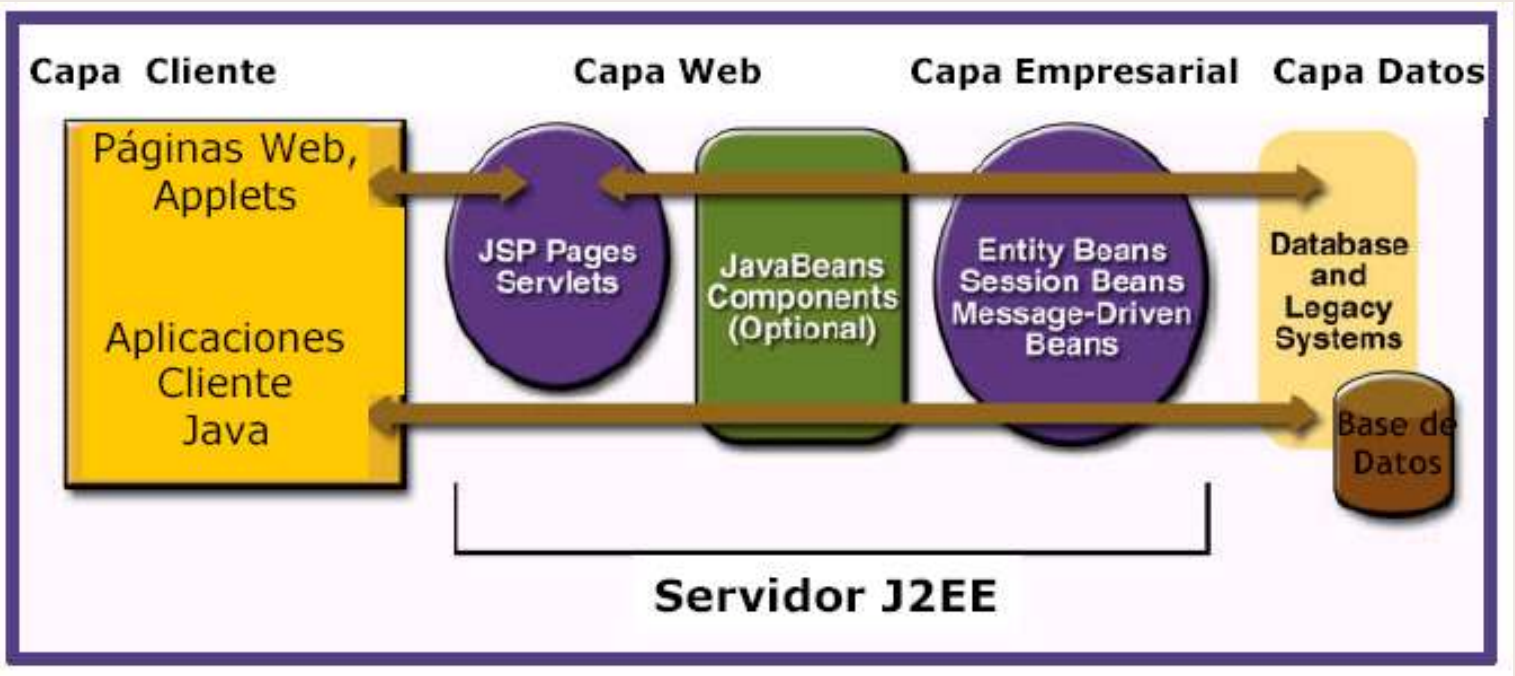
Objetivo: Crear un modelo simple de desarrollo para aplicaciones empresariales utilizando componentes basados en el modelo de aplicación.

J2EE - Java 2 Enterprise Edition

Características:

- Arquitectura multicapa (Multi-tier).
- Soporta gran variedad de tipos de aplicaciones.
- Los componentes utilizan servicios proporcionados por el contenedor.

Arquitectura J2EE-Componentes

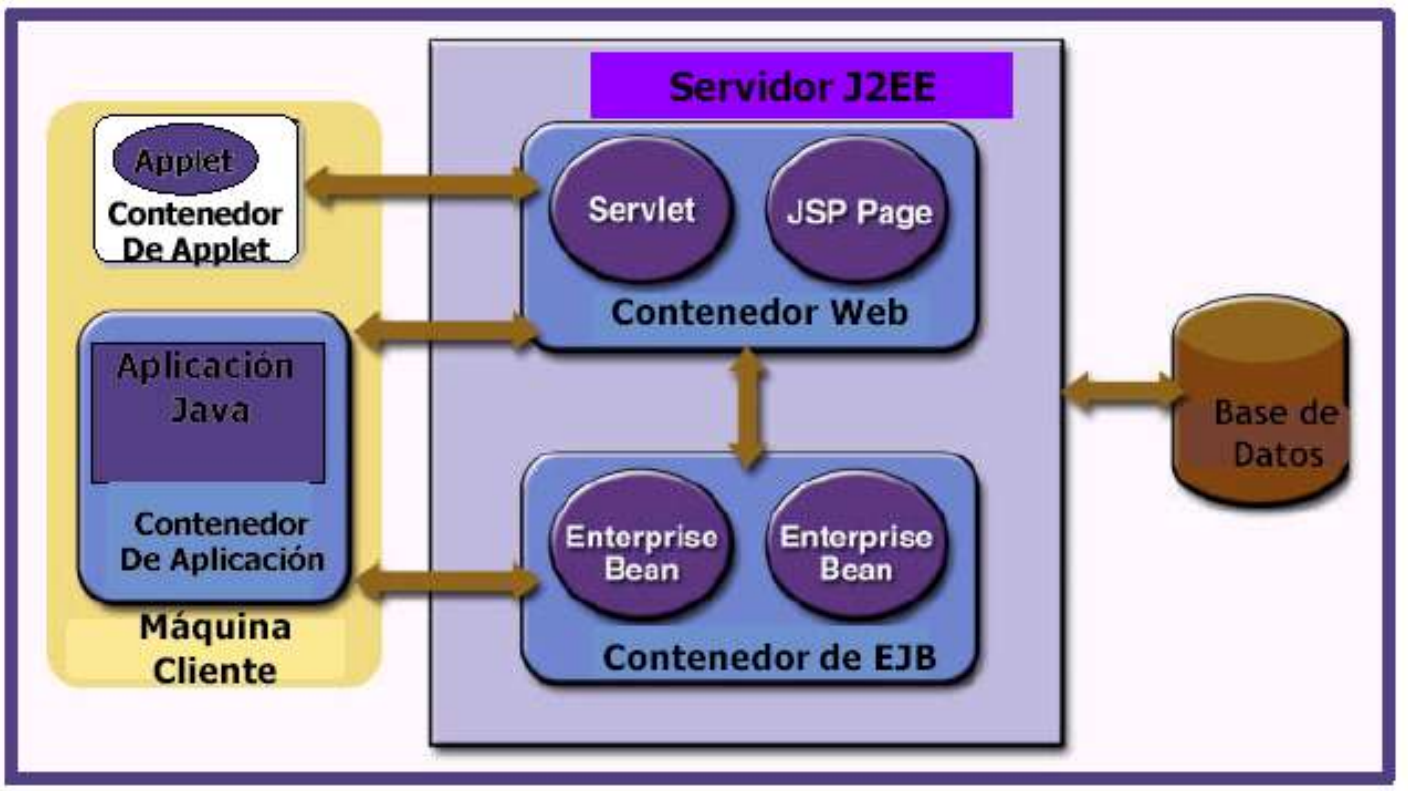


Arquitectura J2EE-Componentes

La especificación J2EE define los siguientes componentes:

- Componentes cliente: Páginas HTML, Applets y Aplicaciones Cliente Java.
- Componentes Web: Servlets y JSP (JavaBean, Custom Tags).
- Componentes empresariales: Enterprise JavaBeans.

Arquitectura J2EE-Contenedores



¿Qué tipo de servicios provee el contenedor J2EE?

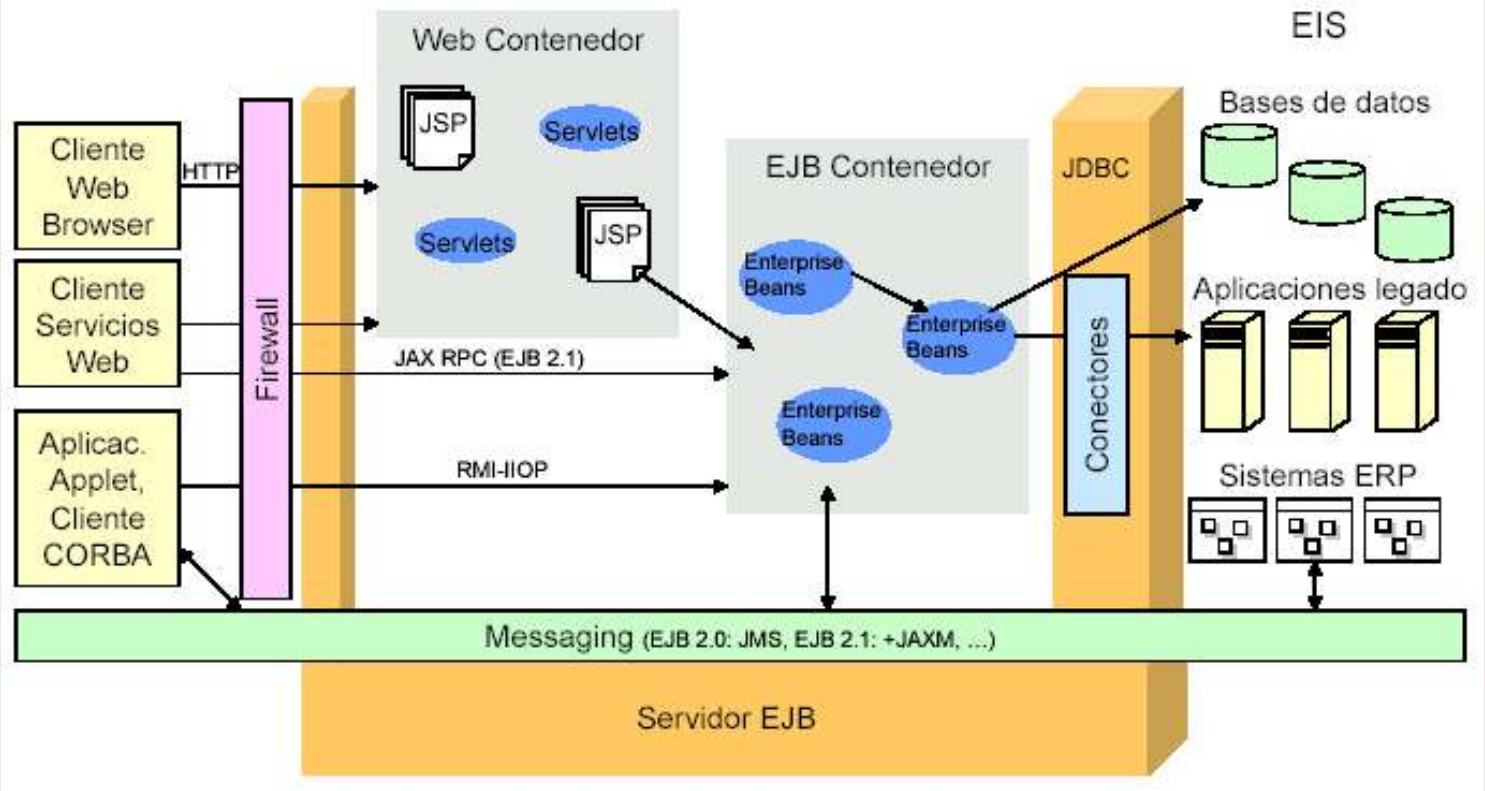
Servicios configurables:

- Seguridad.
- Transaccionabilidad.
- Servicio de nombres.
- Conectividad remota.
- Balance de carga

¿Qué tipo de servicios provee el contenedor J2EE?

- **Servicios no configurables:**
- Gerenciamiento del ciclo de vida de los componentes.
- Pooling de conexiones a la base de datos.
- Pooling de objetos.
- Persistencia de datos.

Instalación típica J2EE



Enterprise JavaBeans (EJB) Características Generales

- Componente escrito en Java, *server-side*.
- Tecnología central de J2EE para la creación de lógica y datos de negocio.
- Une los distintos APIs de J2EE en una arquitectura integradora.

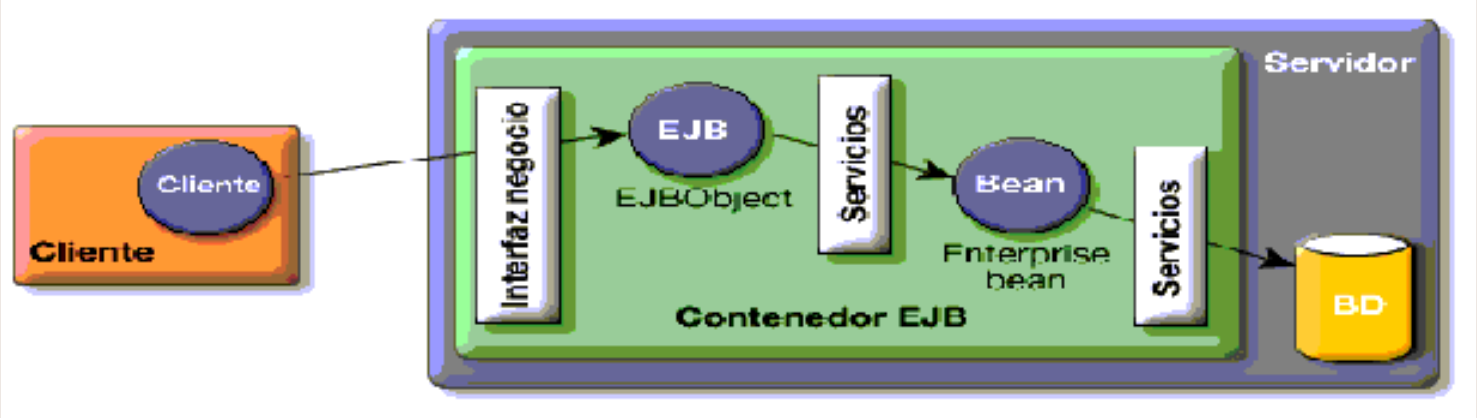
Enterprise JavaBeans (EJB) Características Particulares

- Usados como parte de aplicaciones corporativas distribuidas.
- Cada beans encapsula parte de la lógica de negocio de la aplicación.
- Se comunica con gestores de recursos, y con otros EJB's.

Enterprise JavaBeans (EJB) Características Particulares

- Accedido por distintos tipos de clientes: EJB's, servlets, clientes de aplicación, etc.
- En tiempo de ejecución, reside en un contenedor EJB: servicios de seguridad, transacción, instalación (deployment), concurrencia y gestión del ciclo de vida.
- Una aplicación puede tener uno o varios EJB's en uno o varios contenedores EJB.

Arquitectura EJB



Tipos de EJB

- Session.
- Entity.
- Message Driven.

Ventajas de los EJB's (para el desarrollador de aplicaciones)

- Simplicidad.
- Portabilidad de la aplicación.
- Reusabilidad de los componentes.
- Posibilidad de construcción de aplicaciones complejas.
- Separación de la lógica de presentación de la lógica de negocio.

Ventajas de los EJB's (para el desarrollador de aplicaciones)

- Despliegue de muchos entornos operativos.
- Despliegue distribuido.
- Interoperabilidad entre aplicaciones.
- Integración con sistemas no-Java.
- Recursos educativos y herramientas de desarrollo.

Ventajas de los EJB's (para los clientes)

- Elección del servidor.
- Gestión de las aplicaciones.
- Integración con aplicaciones y datos ya existentes.
- Seguridad.

Desventajas

- Tiempo de desarrollo.
- Conocimiento exhaustivo de Java.

El futuro...

Servicios Web (Web Services)