



especialista universitario en
Tecnologías Java Enterprise
para Aplicaciones Web

Introducción a las tecnologías J2EE

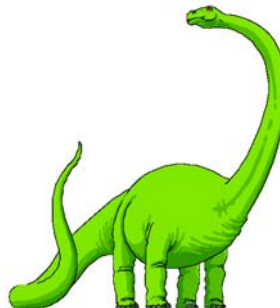
Sistemas distribuidos



- Los sistemas distribuidos dividen una aplicación en distintos módulos independientes
- El fallo de un módulo tiene menos impacto en el sistema completo, lo que los hace más:
 - disponibles
 - escalables
 - mantenibles



Sistemas distribuidos:
pequeños, ágiles y adaptables



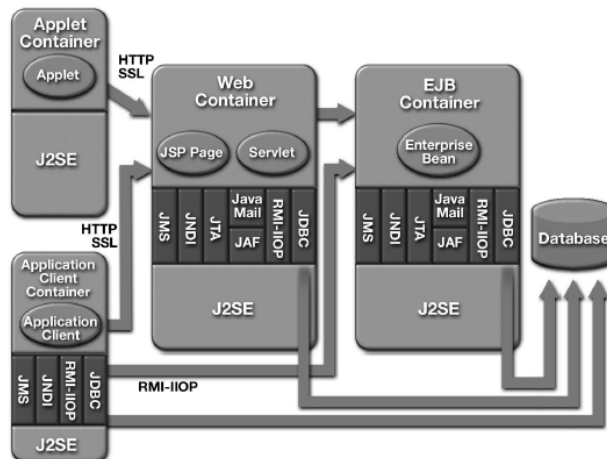
Sistemas
monolíticos:
grandes,
Inflexibles e
inadaptables

El enfoque Java 2 Enterprise Edition (J2EE)



- **J2EE facilita el desarrollo de sistemas distribuidos, basados en componentes**
- **Las aplicaciones desplegadas con la tecnología J2EE:**
 - están estandarizadas
 - siguen una guías de especificación estrictas
 - están escritas en Java
 - se pueden desplegar en cualquier servidor de aplicaciones

Componentes y contenedores



Tecnologías de servicio



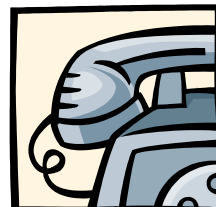
- **JDBC**
- **Java Transaction API (JTA)**
- **Java Naming and Directory Interface (JNDI)**
- **J2EE Conector Architecture**
- **Java API for XML Processing (JAXP)**



Tecnologías de comunicación



- **Protocolos de Internet**
 - TCP/IP, HTTP, SSL
- **Protocolos de invocación remóta de métodos**
 - RMI
- **Protocolos OMG**
 - CORBA, Java IDL, RMI-IIOP
- **Tecnologías de mensaje**
 - Java Message Service (JMS), JavaMail



Java

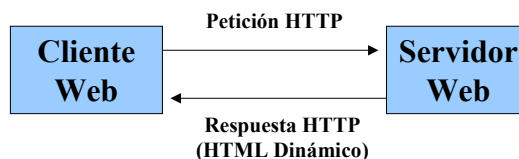


- **Programación con J2SE**
- **Base para realizar una aplicación J2EE**
 - Los componentes de J2EE son código Java
- **Características básicas del lenguaje**
 - Hilos, excepciones, E/S, red, reflection, tipos de datos, etc
- **Característica de la plataforma J2SE:**
 - Seguridad, extensiones, JAR.
- **Programación en el lado del cliente**
 - Aplicaciones
 - Applets

Servlets



- **Los servlets son clases Java que proporcionan un servicio de petición – respuesta basado en un determinado protocolo**
- **Los servlets HTTP son la forma más común de servlet, y se emplean generalmente para producir páginas con contenido dinámico:**



JSP – JavaServer Pages



- Código Java en páginas HTML

```
<%@page import="java.util.Date" %>
<html>
<head> <title> Ejemplo de JSP </title> </head>
<body>
  Hoy es <%= (new Date()).toString() %>
</body>
</html>
```

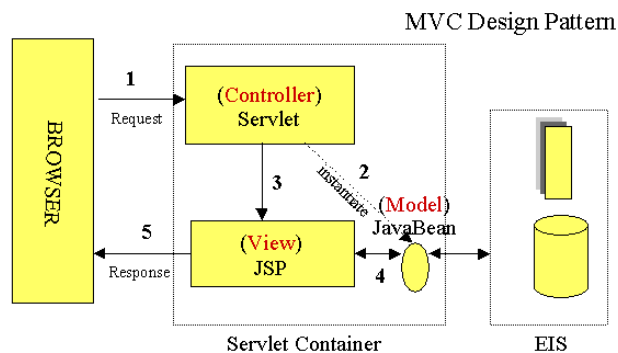
- Acceso directo al API de Servlets

JSP en la plataforma J2EE



- Localización: contenedor web (junto con servlets)

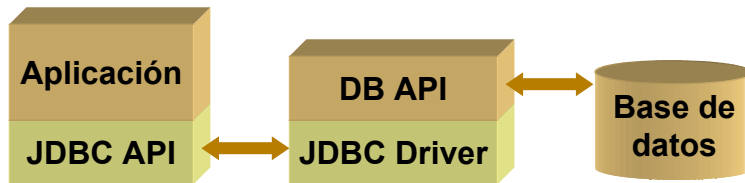
- Uso: capa de presentación de datos (MVC)



Java DataBase Connectivity: JDBC



- **JDBC es:**
 - un interfaz java estándar para el acceso a bases de datos heterogéneas



XML



- **Estándar para definir lenguajes de marcado propios**
- **Ejemplo: periódico electrónico**

```
<noticia>
  <titulo> Baja el precio del crudo </titulo>
  <autor> Pedro Márquez </autor>
  <resumen> Los países de la OPEP...</resumen>
  <contenido> Ante las circunstancias que rodean ...
    </contenido>
</noticia>
```

XML en la plataforma J2EE



- **Formato portable de almacenamiento e intercambio de datos (ficheros de configuración, descripción de servicios web, etc.)**

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
  <Service name="Tomcat-Standalone">
    <Connector className="HttpConnector"port="8080"/>
    ...
    (Fichero de configuración del servidor Tomcat)
```

- **Lenguaje de descripción de las páginas web**

Servidores de aplicaciones



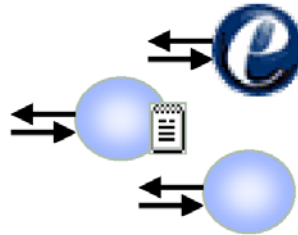
- **Un servidor de aplicaciones proporciona servicios que soportan la ejecución de las aplicaciones.**
- **Algunas características del servidor de aplicaciones BEA WebLogic:**
 - **Balaneo de carga**
 - **Servicios de clustering**
 - **Seguridad**
 - **Servidor de Administración**



Enterprise JavaBeans (EJBs)



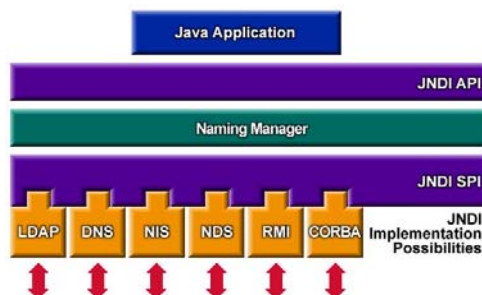
- Los EJBs proporcionan interfaces para construir componentes distribuidos en el lado del servidor
- El contenedor de EJB proporciona servicios de infraestructura
- Los EJB:
 - tienen interfaces bien definidas
 - diferentes niveles de estado
 - son reusables entre servidores de aplicaciones



JNDI (Naming and Directory Interface)



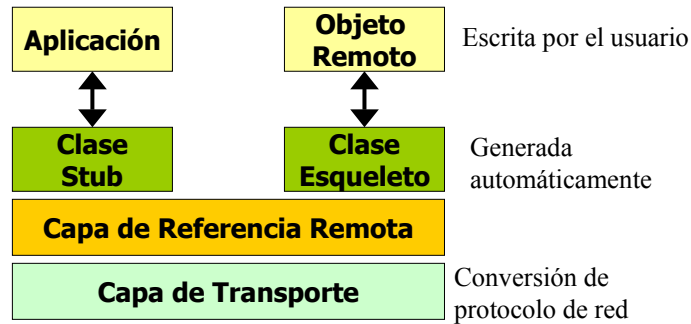
- Acceso a diversos servicios de nombrado/directorio, independientemente de su implementación.
- Ciclo: Conexión, autenticación, búsquedas, operaciones (buscar, añadir, etc) y desconexión.



RMI



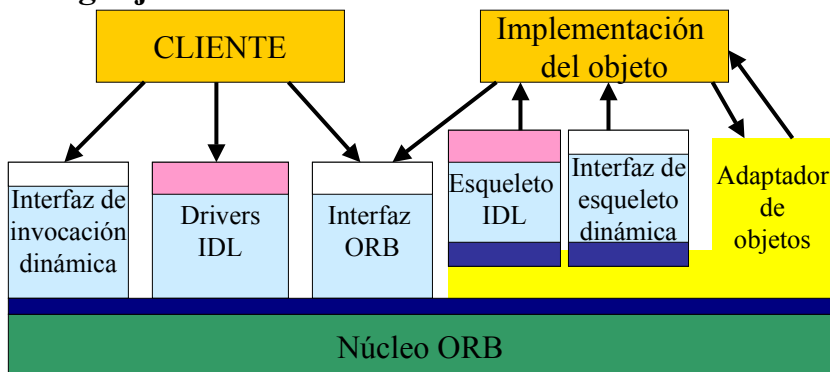
- RMI es un protocolo que permite invocar métodos de objetos remotos escritos en Java
- RMI permite pasar datos por valor entre dos objetos mediante *"serialización"*



CORBA



- CORBA es un protocolo que permite invocar métodos de objetos remotos escritos en cualquier lenguaje



Seguridad en Java



- JCA (Java Cryptography Architecture)
- JCE (Java Cryptography Extension)
- Encriptado simétrico y asimétrico.
- Firma y certificado digital.
- Core Security Model y Applets
- Seguridad en EJBs, SSL/RMI y BDs.
- Seguridad en aplicaciones completas.
- Implementar un proveedor propio.

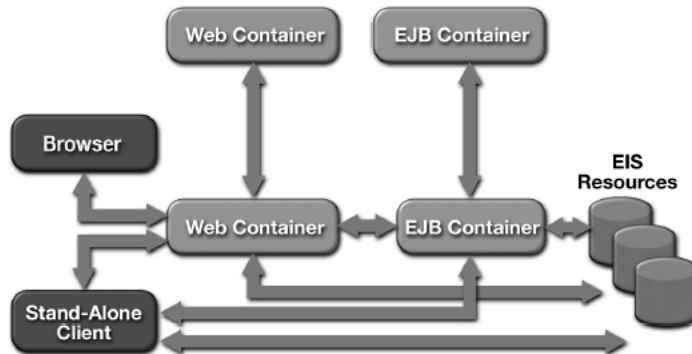


Servicios Web



- Similar a RPC, a través de Internet
- Los SW permiten llevar toda la presentación al lado del cliente
 - Dispositivos heterogéneos se conectan a Internet
 - Pueden ser utilizados por clientes o por otros servidores
- Se invocan mediante protocolo HTTP
- La llamada y la respuesta se codifican en XML
- Lenguajes derivados de XML
 - SOAP: Llamada y respuesta de un servicio
 - UDDI: Localización de servicios

Escenarios de aplicación



Diseño de aplicaciones con J2EE



▪ Objetivos

- Aplicaciones robustas, escalables, reusables y fáciles de mantener

▪ Arquitectura “MVC”

- Modelo = EJB
- Vista = JSP
- Controlador = Servlet

▪ Patrones de diseño

- Detectan situaciones comunes y definen una solución software para la situación

