

Escuela Politécnica del Ejército

Sede Latacunga

Facultad de Sistemas e Informática

- **Galarza Maira**
- **Tapia Cevallos Paulina**

DESARROLLO DE APLICACIONES DISTRIBUIDAS
UTILIZANDO PATRONES DE DISEÑO MODELO/VISTA
/CONTROLADOR (MVC) BAJO J2EE.

APLICACIONES DISTRIBUIDAS

- Las aplicaciones distribuidas son un conjunto de programas o sistemas que se comunican entre sí para realizar una tarea común. Son ejecutados en procesos diferentes, en la misma máquina o bien en máquinas distintas, en el mismo sitio o en sitios diferentes e interconectadas a través de una red de comunicación

APLICACIONES DISTRIBUIDAS

CARACTERÍSTICAS

- Seguridad.
- Confiabilidad
- Escalabilidad.
- Tolerancia a Fallos.
- Compartimiento de recursos.
- Y ser un sistema abierto.

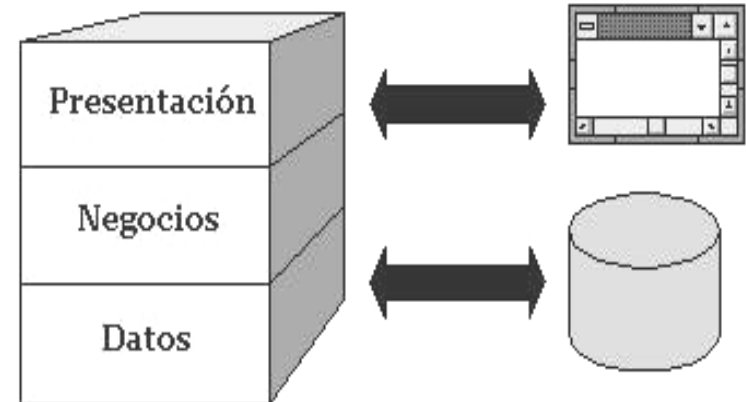
APLICACIONES DISTRIBUIDAS

EVOLUCIÓN.

La arquitectura de los sistemas de software desarrollados en los últimos años ha evolucionado de la mano con la tecnología informática. Las empresas hoy en día están aprovechando estas tecnologías que permiten a sus usuarios el tener acceso sencillo a sus aplicaciones corporativas sin necesidad de realizar gastos innecesarios de mantenimiento que conlleva el modelo cliente servidor. A continuación se detalla el avance y desarrollo de las aplicaciones, así como sus características.

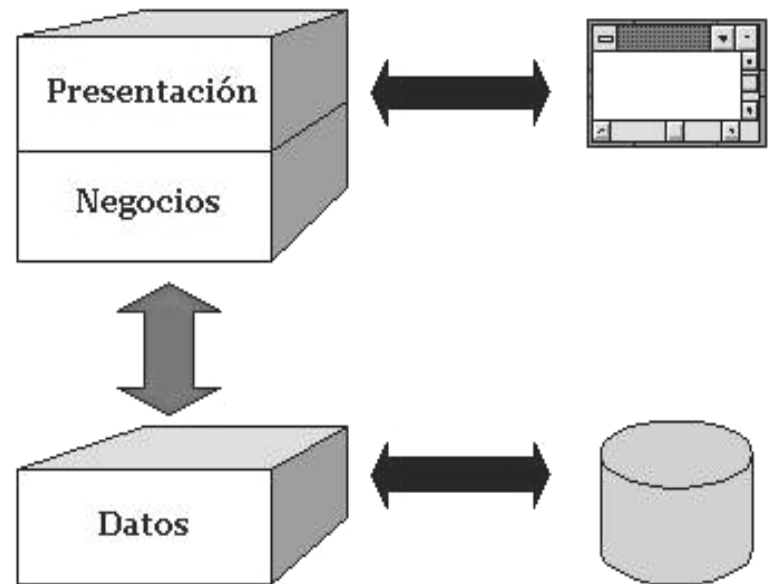
APLICACIONES MONOLÍTICAS

- Las aplicaciones monolíticas están desarrolladas de modo que las tres capas residen en una única aplicación



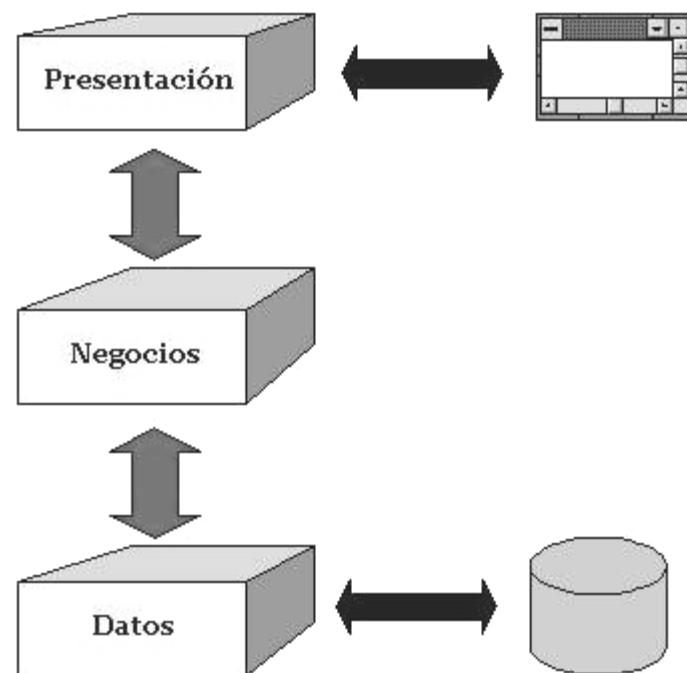
APLICACIONES EN DOS CAPAS (CLIENTE-SERVIDOR)

- Los sistemas típicos cliente / servidor pertenecen a la categoría de las aplicaciones de **dos** capas. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor.



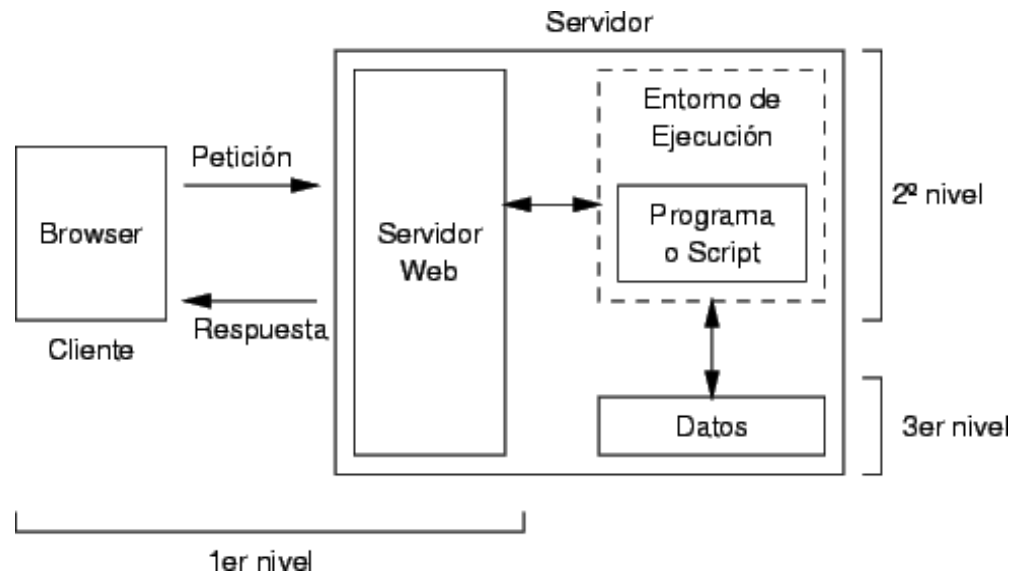
APLICACIONES EN TRES O MAS CAPAS

- Las aplicaciones en 'n' capas surgieron por primera vez como una forma de resolver algunos de los problemas asociados a las aplicaciones cliente / servidor tradicionales, pero con la llegada del internet esta arquitectura ha llegado a dominar el nuevo desarrollo. Además se han convertido en el modelo para el desarrollo de software actual , una aplicación en 'n' niveles es algo dividido en distintas partes lógicas. La opción más habitual está formada por una división en tres partes: presentación, lógica de negocio y datos,



ARQUITECTURA N CAPAS

El modelo n-capas de informática distribuida ha emergido como la arquitectura para la construcción de aplicaciones multiplataforma. se están convirtiendo en la nueva base para el desarrollo de aplicaciones de misión crítica y ofrecen la única arquitectura funcional para la siguiente generación de soluciones informáticas distribuidas basadas en Internet, presenta un esquema de 3 niveles. El primer nivel consiste en la capa de presentación que incluye el navegador y el servidor web. El segundo nivel esta referido a algún programa o scripts. El tercer nivel proporciona al segundo datos necesarios para su ejecución.



Java 2 Enterprise Edition

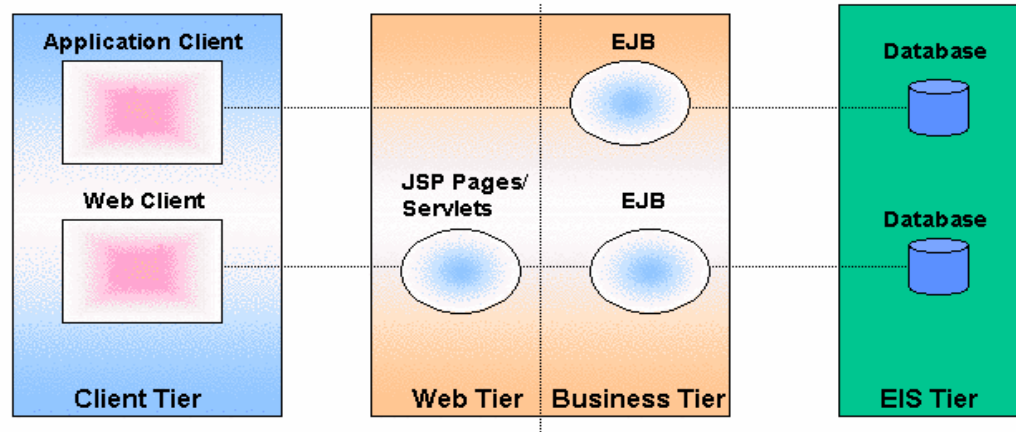
J2EE

- La meta de J2EE es definir un estándar que ayude a suplir los retos tecnológicos en esta nueva era.
- J2EE soporta aplicaciones distribuidas que toma ventajas de las tecnologías existentes y en desarrollo simplificando el proceso a través de un modelo de aplicaciones basados en componentes.

Características de J2EE

- Separación de tareas en la plataforma Mult. – capa.
- Portable.
- Escalable.
- Altamente Soportada.
- Segura.

Componentes de J2EE

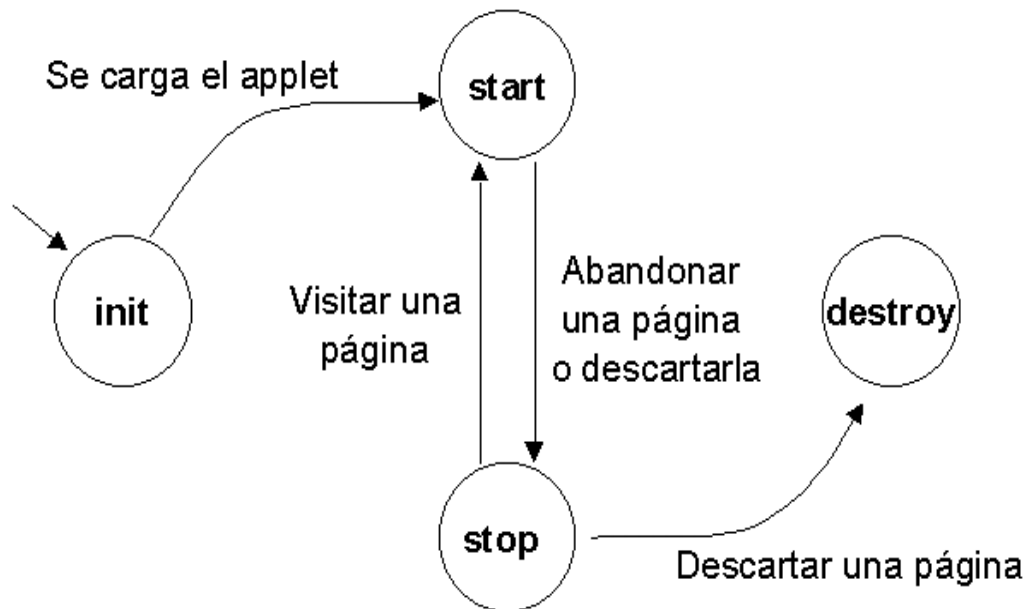


Componentes Cliente

- Applets.

Características.

Ciclo de vida de un applet.



Componentes Web

Servlet

- Un servlet es un programa que extiende la funcionalidad de un servidor Web.
- Reciben un requerimiento de un cliente, dinámicamente generan la respuesta y luego la mandan un documento HTML o XML al cliente.
- Contenedor de servlets.
- Estructura.
 - `javax.servlet.`
 - `java.servlet.http_`

- . Ciclo de vida de un servlet.
 - Inicializar un servlet.
 - Interactuar con clientes.
 - Destruir un servlet.

- Clases principales.
 - Servlet. Define la funcionalidad básica de un servlet.
 - HttpServlet . Agrega la funcionalidad para procesar los variados tipos de request http .
 - HttpServletRequest. Proporciona la información del request del cliente.
 - HttpServletResponse . Asiste a un servlet para enviar una respuesta al cliente .

Componentes Web

Java Server Pages

- Es un documento texto que describe como procesar un requerimiento para crear una respuesta
- Elementos de un JSP.
 - Comentarios.
 - Expresiones.
 - Scriptets.
 - Declaraciones.
 - Directivas.
 - Page.- Importar clases.
 - Include.-Incluir ficheros.
 - Variables Predeterminadas.

Componentes Empresariales

Enterprise Java Beans (EJB)

- La arquitectura EJB es una tecnología del lado del servidor para desarrollo y ejecución de componentes para lógica de negocio de una aplicación empresarial.
- Existen tres tipos de componentes
 - Session Bean
 - Entity Bean
 - Message Driven Bean

Arquitectura de un EJB

- *Bean Class*: Clase común que implementa un conjunto de operaciones bien definidas. Implementa: operaciones para interactuar con el contenedor y los servicios que el componente exporta
- *Remote Interface*: Interfaz que exporta todos los métodos de negocio que un componente exporta y que pueden ser invocados desde fuera
- *Home Interface*: Permite manejar la creación, destrucción y búsqueda (solo “entities”) de instancias del componente. También permite que la ubicación de los componentes se maneje transparentemente

Tipos de EJBs

Session Beans

- Es creado para proveer algún servicio detrás de un cliente y usualmente durante una única sesión
- Ejecuta operaciones como cálculos o accesos a DB para el cliente.
- No es recuperable si el contenedor cae
- Existen 2 tipos:
 - Stateful Session Beans: El estado conversacional se mantiene mientras la sesión con el cliente esté viva.
 - Stateless Session Beans: El estado conversacional se mantiene por el tiempo que dura la invocación a un método. Estos normalmente deben recibir del cliente toda la información necesaria para realizar su tarea u obtenerla de una fuente externa como una base de datos

Entity Beans

- Los beans de entidad normalmente representan datos persistentes. Estos datos están mantenidos directamente en una base de datos.
- Existen 2 tipos de EB :
 - Bean Managed Persistent (BMP): Es en donde el desarrollador se encarga de implementar la comunicación con la base de datos. Para esto es necesario tener acceso a ella a través de JDBC,
 - Container Managed Persistent (CMP): el contenedor es responsable de manejar la persistencia.

Message Driven Beans

- Los MDB son un tipo especial de EJB que puede recibir mensajes
- Los MDB están suscritos a una cola o tópico
 - Cuando llega un mensaje el contenedor despierta al MDB para que procese el mensaje

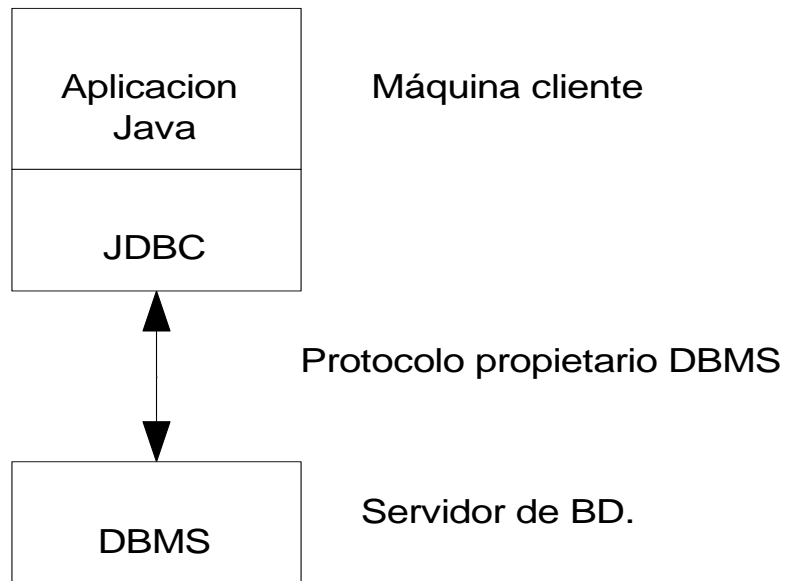
ACCESO A BASE DE DATOS

JDBC (Java Data Base Connectivity)

- Provee conectividad independiente a bases de datos .
- La combinación de Java y JDBC permite al programador escribir una sola vez y ejecutarla en cualquier entorno.
- JDBC , hace posible tres cosas:
Establecer una conexión a la base de datos.
Envía una sentencia SQL:
Procesa resultados.

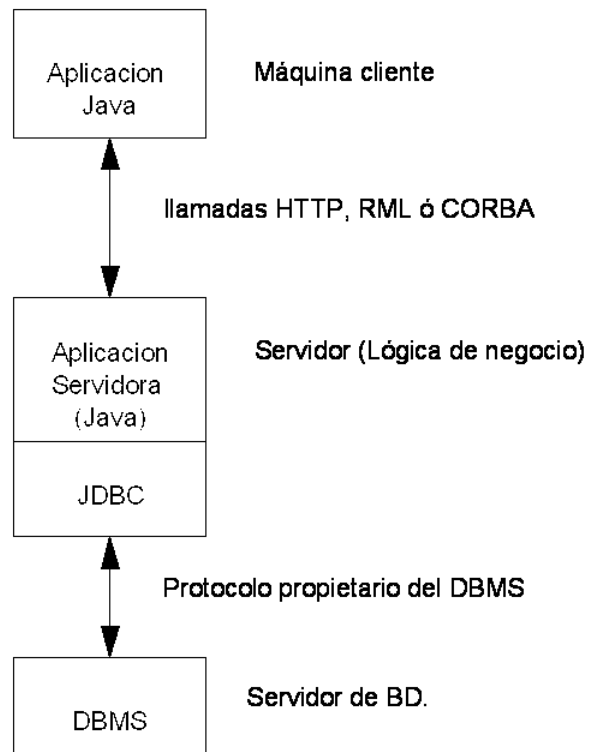
MODELOS DE ACCESO

Modelo dos pisos.



MODELOS DE ACCESO

Modelo de tres pisos

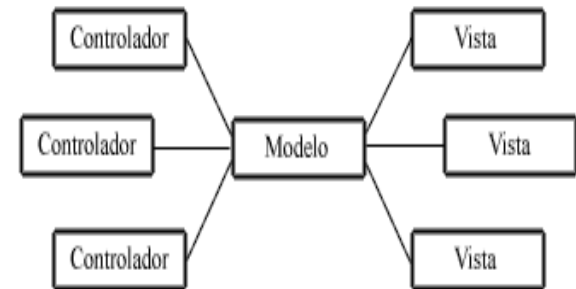


MODELO VISTA CONTROLADOR (MVC)

- Las aplicaciones en el navegador hoy en la actualidad para su desarrollo son mas costosos que las aplicaciones tradicionales.
- Modelo Vista Controlador. Cuyo objetivo es separar la presentación de la lógica de negocio.

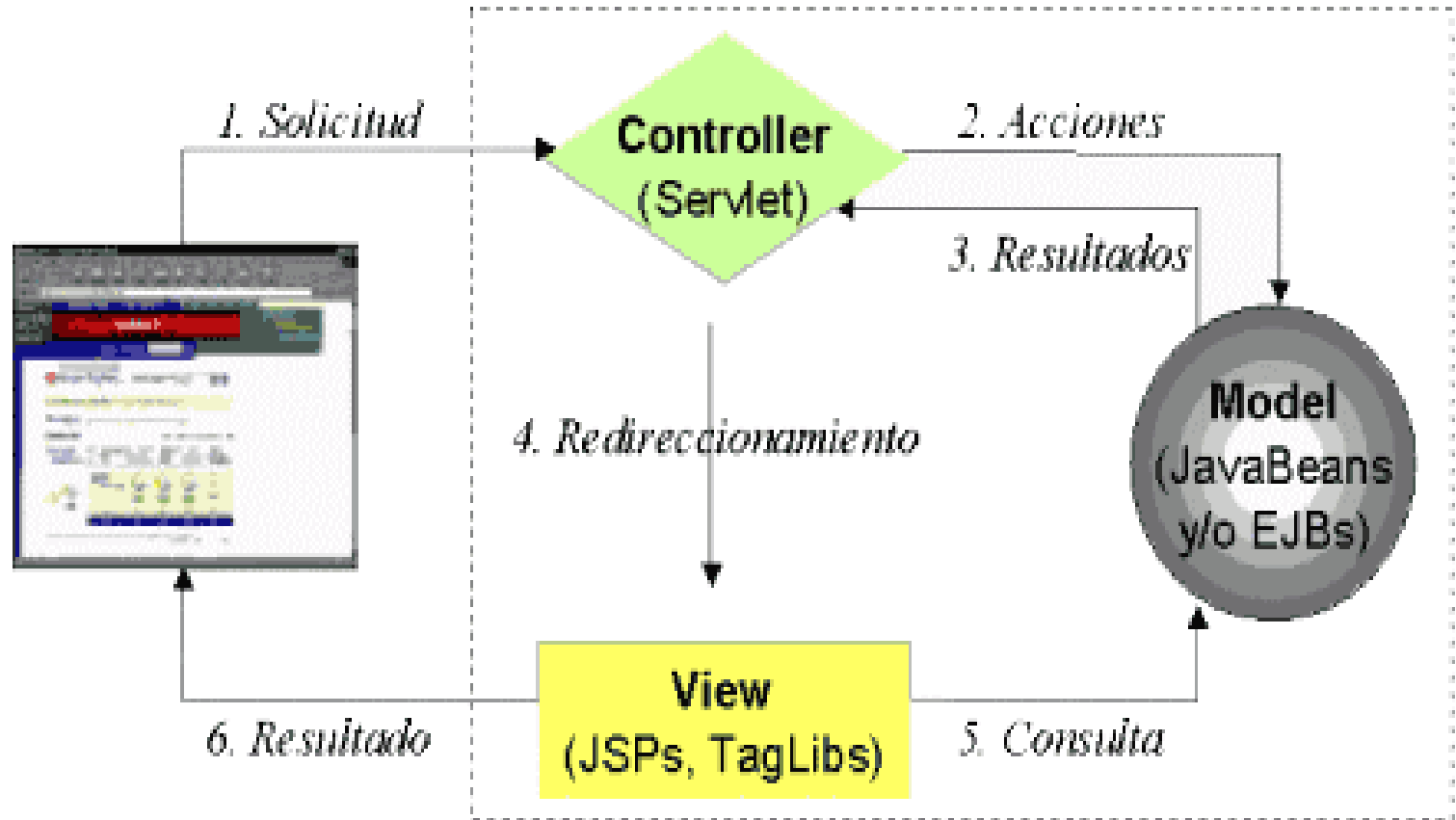
LA ARQUITECTURA MODELO / VISTA / CONTROLADOR

- La arquitectura MVC (Modelo/Vista/Controlador) fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos.



Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

DESCRIPCIÓN DEL MODELO MVC



STRUTS

Es un framework para aplicaciones web java que implementa el modelo MVC. Realmente lo que provee es un conjunto de clases y librerías que conforman el Controlador, la integración con el Modelo (o lógica de negocio) y facilitan la construcción de vistas.

El Modelo

- Representa la lógica de negocio de una aplicación; maneja los datos y controla todas sus transformaciones.

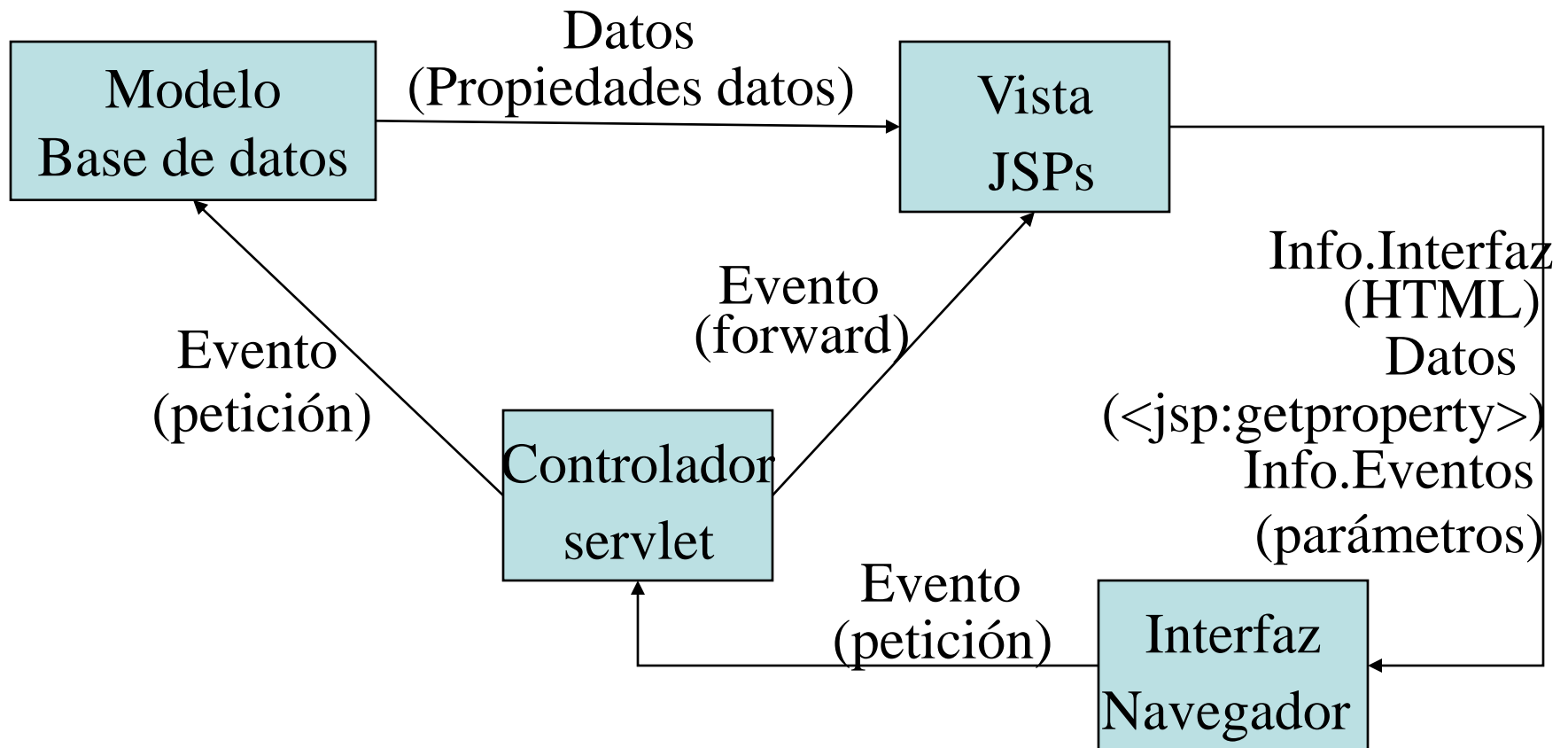
La vista

- Constituye la lógica de presentación de una aplicación.
- Los componentes de la vista obtienen el estado actual del sistema del modelo y proporcionan el interfaz de usuario para el protocolo involucrado (en nuestro caso HTTP de los navegadores web).
- Como parte de la generación del interfaz de usuario la vista presenta los eventos que el usuario puede activar en cada momento.
- **JSP** es la elección natural para implementar la vista.

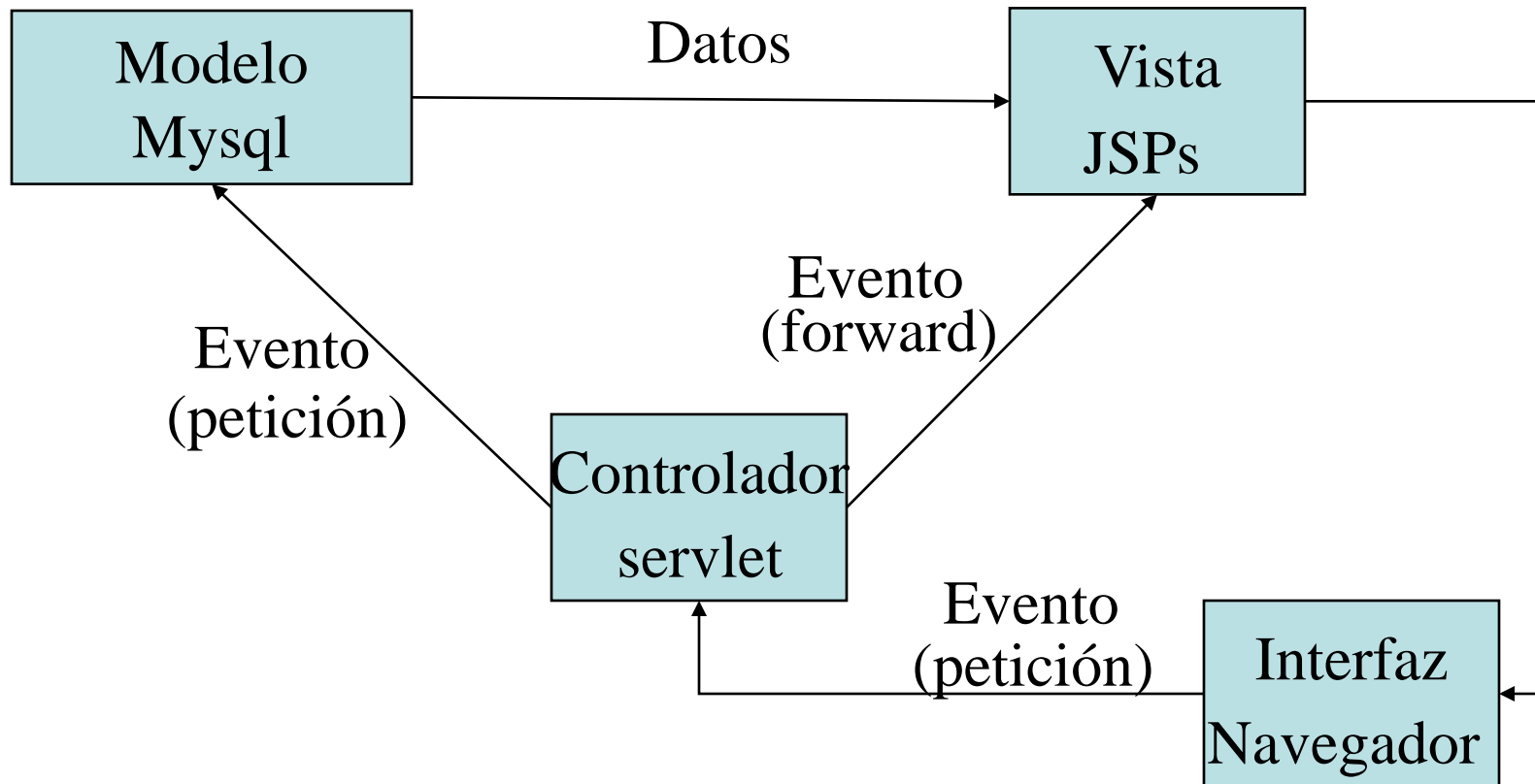
El controlador

- Proporciona unión a toda la arquitectura.
- Responsable de:
 - recibir eventos,
 - determinar cual es el manejador apropiado,
 - invocar este manejador y
 - determinar la generación de la respuesta apropiada.
- Los `Servlet` son la elección ideal para la tecnología del controlador.

Componentes arquitectura MVC



Ejemplo de MVC



VENTAJAS DEL MVC

La estructuración de aplicaciones distribuidas utilizando el modelo MVC (Modelo-Vista-Controlador) nos ofrece las siguientes ventajas :

- Al incorporar el modelo de arquitectura MVC el diseño de las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución.
- La separación de capas en presentación, lógica de negocio y acceso a datos .
- El costo de mantenimiento del sistema disminuye ya que una modificación en una capa no debe afectar a las demás. Con esto tenemos menores tiempos de desarrollo y por lo tanto menos costo.
- Son fácilmente escalables, esto es, pueden desde unas cuantas docenas de usuarios hasta decenas de miles de usuarios con solo unas cuantas modificaciones a una sola capa, por lo que la inversión en software se puede mantener mientras el sistema crece.
- En cualquier momento en que los datos del modelo cambien, el modelo le notifica a la vista que depende de él.

CONCLUSIONES Y RECOMENDACIONES

- CONCLUSIONES.
 - Las aplicaciones distribuidas permiten incrementar la lógica de programación del lado del servidor siendo esto transparente para el usuario.
 - De acuerdo a la investigación efectuada podemos decir que actualmente J2EE (Java2 Enterprise Edition) se está convirtiendo en la plataforma mas usada por las empresas que buscan construir aplicaciones que demuestren eficiencia, estabilidad, seguridad, escalabilidad y flexibilidad, porque fomenta la separación en 3 o mas capas y el desarrollo especializado de cada componente .

- Para el desarrollo del sistema se utilizó el patrón de diseño MVC (Modelo Vista Controlador), es decir, se realizó la separación de los componentes del sistema, lo que permitió reducir el esfuerzo de programación porque se trataron como entidades separadas, para luego unirlos en tiempo de ejecución.
- El uso de software libre utilizado en esta investigación como Java y mysql, fue muy importante a nivel económico y legal.
Económico porque los costos en cuanto a software y hardware fueron menores que si hubiésemos utilizado software propietario, ya que con una sola licencia de software, pudimos instalar en las máquinas necesarias, y sin requerimientos excesivos de hardware para su funcionamiento.
Legal ya que el software libre es siempre legal, salvo cuando se le imponen restricciones.

- La aplicación desarrollada “Editor de Cursos en Línea para Internet” es una herramienta que permitirá crear una nueva modalidad de enseñanza aprendizaje, lo cual resulta de gran utilidad para la comunidad politécnica.

Recomendaciones.

- El diseño de la interfaz web debe ser desarrollada en un ambiente amigable y fácil de usar para el usuario.
- Se imparta cursos o seminarios con temas de actualidad, motivando de esta manera a los alumnos a desarrollar proyectos innovadores.
- En el desarrollo de aplicaciones Web se sugiere la utilización de J2EE (Java 2 Enterprise Edition), My SQL, Servidor Tomcat, debido a su libre distribución, como también por no requerir de licencias, y por su estabilidad.

- El trabajo investigativo se basa en el modelo MVC, y para la comprobación de su funcionamiento se inicio en el desarrollo de una aplicación Web como es el Editor de Cursos en Línea para internet, se recomienda realizar la ampliación y fortalecimiento, es decir, debería complementarse con la creación de cuestionarios y exámenes que permitan evaluar el aprendizaje del alumno que tome un determinado curso. Y que permita constituirse en una herramienta práctica en beneficio de nuestra institución.

Editor de Cursos en Línea

- Es una herramienta que servirá como una nueva modalidad de estudio; el cual permite que el estudiante desarrolle habilidades como el aprendizaje, de organización y responsabilidad personal, independencia, capacidad para entender y seguir instrucciones, disciplina, actitud crítica, etc.

