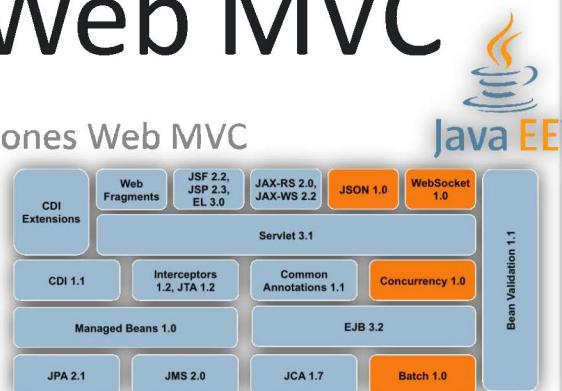


Tema 3: Diseño y Desarrollo de Aplicaciones Web MVC

Tema 3: Diseño y Desarrollo de Aplicaciones Web MVC

Desarrollo Web en Entorno Servidor



Desarrollo Web en Entorno Servidor

Alejandro Cardo Grau

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Servlets

¿Qué es un servlet?

- Son componentes de una aplicación web que se ejecutan en el servidor
 - Permiten extender la funcionalidad del servidor (tanto de servidores HTTP como de otro tipo de servidores como por ejemplo ftp)
 - Cada **petición** se ejecuta en un **hilo**. Los Servlets quedan residentes en memoria cuando la petición termina
- Un servlet es un pequeño código Java que el servidor Web carga para manejar peticiones del cliente
 - Estas clases Java utilizan **el API Servlet**
 - Se cargan y ejecutan dentro de un servicio de red (ej. un Servidor Web)
 - Implementan determinados interfaces que le permiten:
 - Recibir una petición HTTP
 - Generar una respuesta

Alejandro Cardo Grau

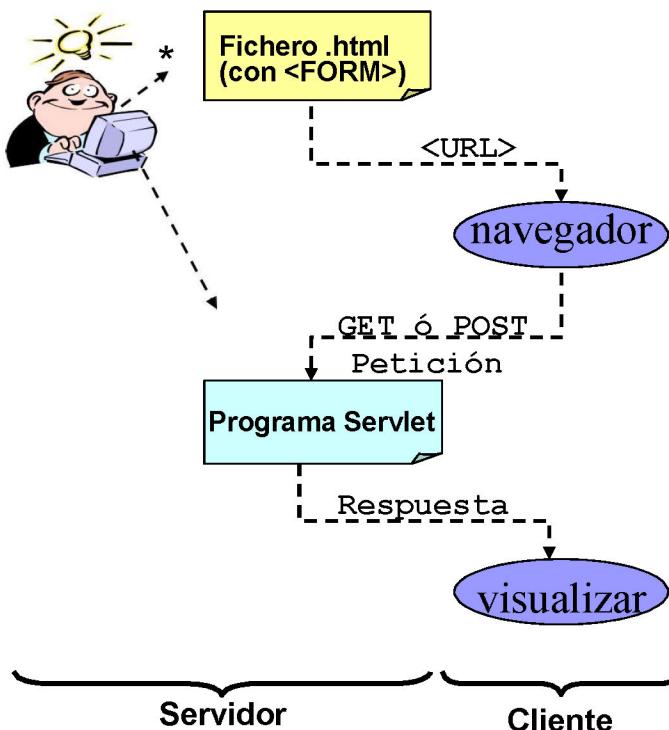
Desarrollo Web en Entorno Servidor

¿Para qué sirve un servlet?

- Extiende funcionalidad del servidor
- Sus usos
 - Procesamiento de formularios en la parte servidor
 - Colaboración entre personas
 - Ej. Conferencias en-línea, chats, mensajería, etc.
 - Debido a la concurrencia y a la sincronización de peticiones
 - Reenvío de peticiones a otros servidores y servlets

Documentación e implementaciones

- **Documentación**
 - <http://java.sun.com/products/servlet/docs.html>
 - <http://docs.oracle.com/javaee/6/tutorial/doc/bnafi.html>
- **JavaEE SDK**
 - Implementación de JavaEE provisto por Oracle
- **Apache Jakarta Tomcat**
 - versión 8: implementación contenedor web para servlets 3.0

Servlets**¿Cómo funcionan?**

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

1. Se parte de un **Fichero.html que tenga un formulario (<FORM>)** accesible a través de Internet
 - El <FORM> deberá especificar las diversas formas de paso de parámetros
2. El cliente accederá al fichero html a través de un **Navegador**, rellenará el formulario y dará a la tecla de aceptar (Submit). Empaquetando la petición.
3. La petición de cliente es atendida por un **Programa Servlet** (escrito en Java) quien da la debida respuesta tras procesarse la petición.
4. El cliente **Visualizará** la respuesta

Servlets**¿Cómo funcionan?**

- Usando **etiquetas HTML**
 - Usando la etiqueta A:


```
<a href="direccion_del_servlet?param1=value">Texto</a>
```
 - Usando la etiqueta IMG:


```

```
 - Otras (javascript, CSS, etc.)

**SÓLO OPERARÁN
POR EL MÉTODO
GET**

- Usando un **formulario** (forma más habitual):


```
<form action="direccion_del_servlet" method="?">
        <!-- Elementos del formulario -->
      </form>
```

Tareas de un servlet

1. Leer datos enviados por el usuario.

Procedentes de: formulario HTML, un applet o una aplicación cliente HTTP, etc.

2. Recuperar información del usuario incluida en petición HTTP

Ej: Nombre de la máquina cliente, tipo de navegador, cookies, etc.

3. Generar resultados

Para ello puede necesitar acceder a BD, o llamar otros programas locales o remotos

4. Formatear los resultados en un documento

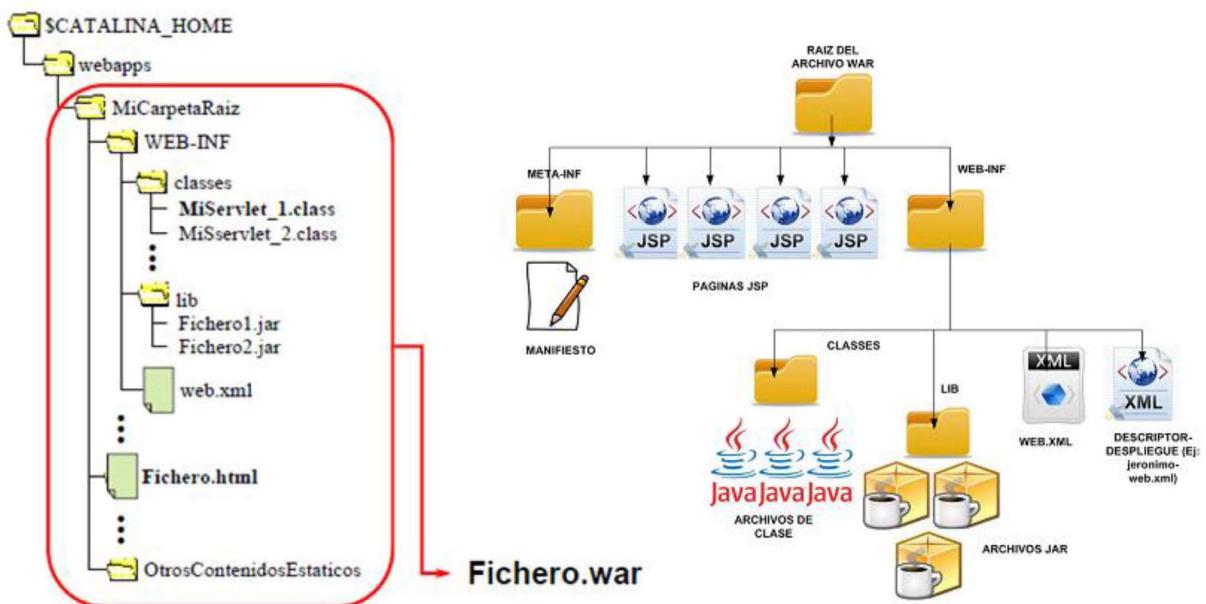
Típicamente una página web (contentType: html), aunque puede haber otro tipo MIME

5. Poner parámetros de respuesta HTTP

Content-type, cookies, etc.

6. Devolver el documento al cliente

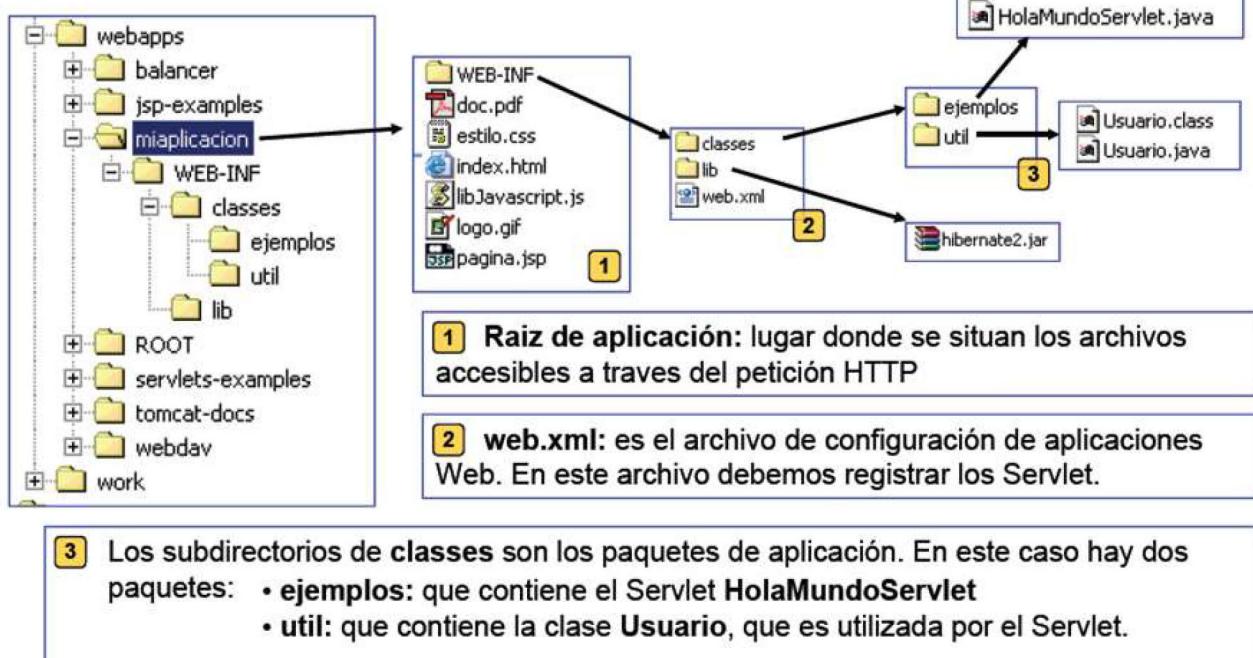
Documento HTML, zip, gif, etc.



Estructura de directorios del contenedor web

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Despliegue de un servlet



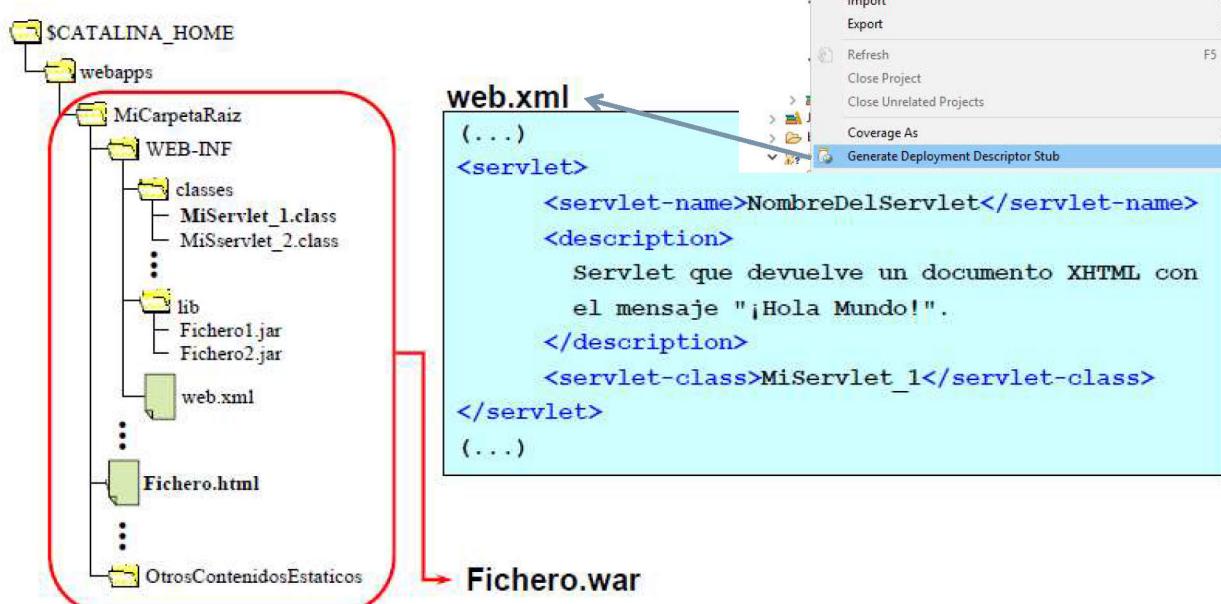
Estructura de directorios del contenedor web

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Despliegue de un servlet



http://localhost:8080/MiCarpetaRaíz/servlet/MiServlet_1

Estructura de directorios del contenedor web:
Generación del descriptor de despliegue: `web.xml`

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Despliegue de un servlet

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://
  xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_
  _3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>My Webapp for DWEBS</display-name>
  <context-param>
    <param-name>idbcURL</param-name>
    <param-value>jdbc:mysql://localhost:3306/javaweb_crud</param-value>
  >
  </context-param>
  <context-param>
    <param-name>idbcUsername</param-name>
    <param-value>root</param-value>
  </context-param>
  <context-param>
    <param-name>idbcPassword</param-name>
    <param-value></param-value>
  </context-param>
</web-app>
```

Estructura de directorios del contenedor web: Parámetros de inicialización para BBDD en web.xml

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

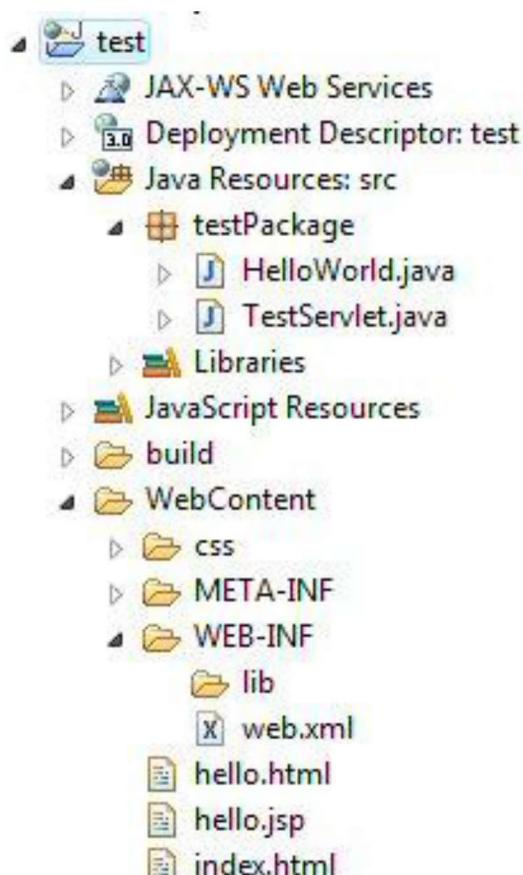
T3: Diseño y Desarrollo de Aplicaciones Web MVC

Dynamic Web Project

IDE ECLIPSE

Directarios del Dynamic Web Project

- **WebContent:**
 - Ficheros web normales (HTML, JavaScript, CSS, JSP, imágenes, etc.)
- **WebContent/subdirectorio**
 - Ficheros web en el subdirectorio.
- **WebContent/WEB-INF**
 - **web.xml:** Este descriptor de despliegue se utiliza para el mapeo de servlets y otras tareas. Actualmente este fichero es opcional, porque el mapeo de servlets se puede hacer mediante anotaciones @WebServlet annotation en el código fuente Java.
- **WebContent/WEB-INF/lib**
 - Ficheros JAR files específicos de la aplicación.
- **src/paquete**
 - Código Java en el paquete.



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Esqueleto básico

Todo Servlet tiene que extender la clase `HttpServlet` y sobrescribir el método `doGet` ó `doPost`, dependiendo de si los datos se envían con GET o con POST

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MiServlet extends HttpServlet {
    public void doGet (HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Se usa "request" para leer cabeceras HTTP entrantes (e.j.
cookies)
        // y datos de formularios HTML (e.j. Datos que el usuario
proporciona en el "submit")
        // Se usa "response" para especificar la línea de respuesta HTTP y
las cabeceras
        // (e.j. Especificar el tipo de contenido, poner las cookies).
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        // Se usa "out" para enviar el contenido al navegador del cliente
        out.println("hola mundo");
    }
}
```

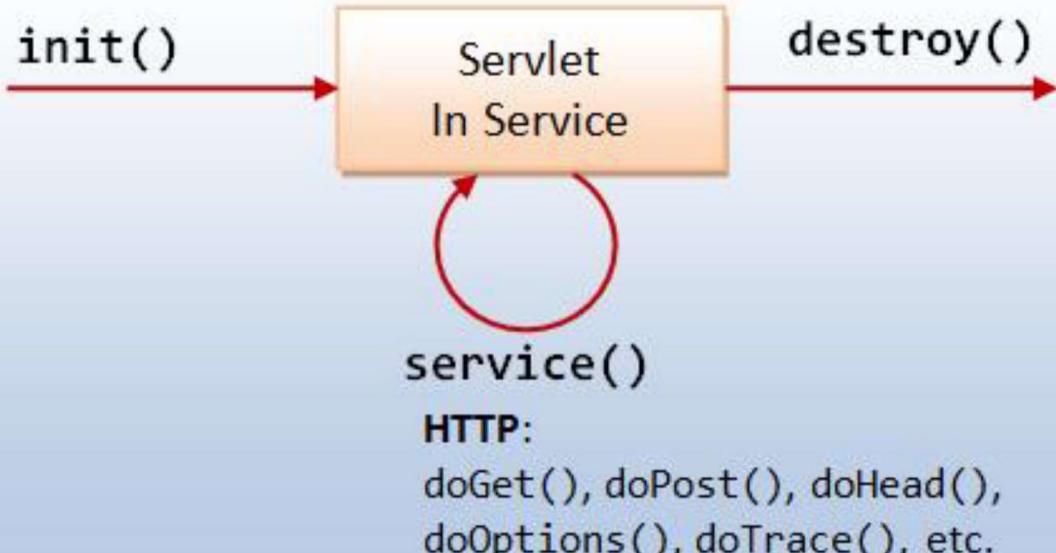
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Ciclo de vida de un servlet

1. Instanciación e inicialización del servlet. Si aún no existe ninguna instancia del servlet (es la primera llamada), el contenedor web:
 - D Primero **carga la clase** del servlet,
 - D luego **crea una instancia** y la inicializa invocando su método **init**
2. Entonces, el servidor puede servir continuamente peticiones. Para cada llamada:
 - D El contenedor **crea un nuevo thread** (hilo)
 - D Se invoca al método **service** de dicho thread.
 - D Los servlets residen típicamente en servidores multithread pero el programador es el responsable de sincronizar el acceso a los recursos compartidos.
 - D También se puede restringir el acceso a un único thread
3. El método **service** **determina tipo de petición** que ha llegado y llama al método correspondiente **doGet**, **doPost**, etc.
4. Un servlet se desactiva cuando recibe una llamada al método **destroy** o cuando su servidor se muere

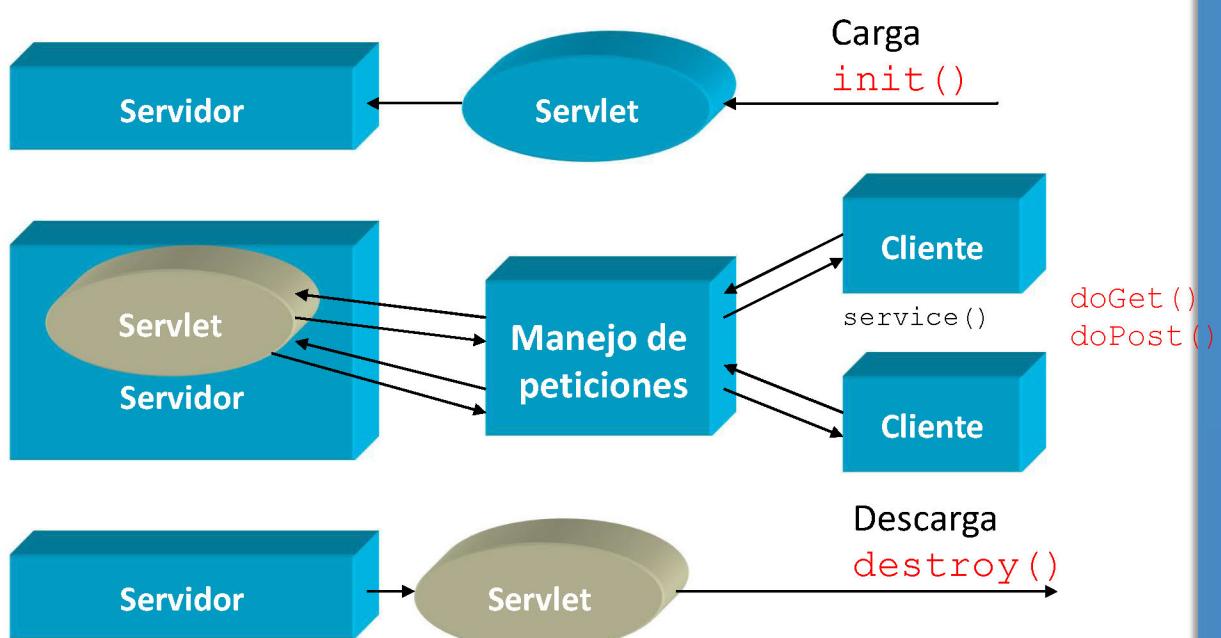
Servlet Container



Ciclo de Vida del Servlet: Métodos asociados

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

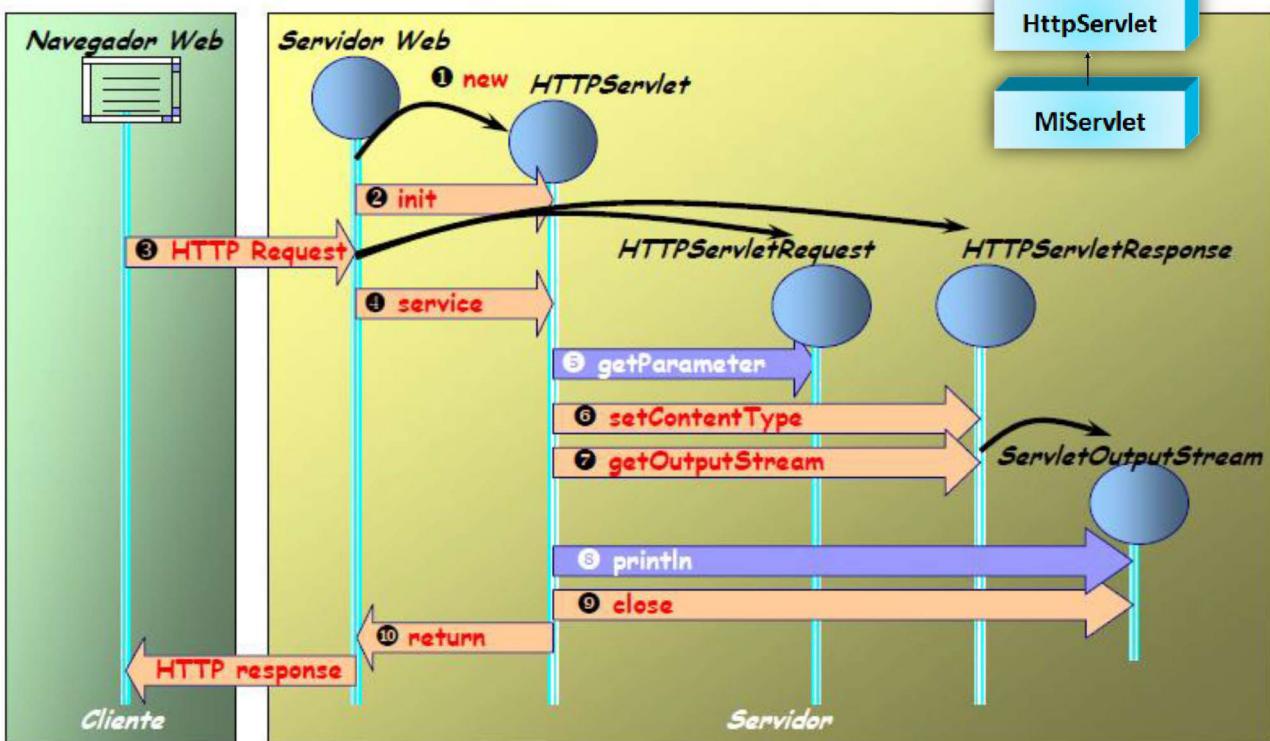


Ciclo de Vida del Servlet: Métodos asociados

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Servlets



Alejandro Cardo Grau

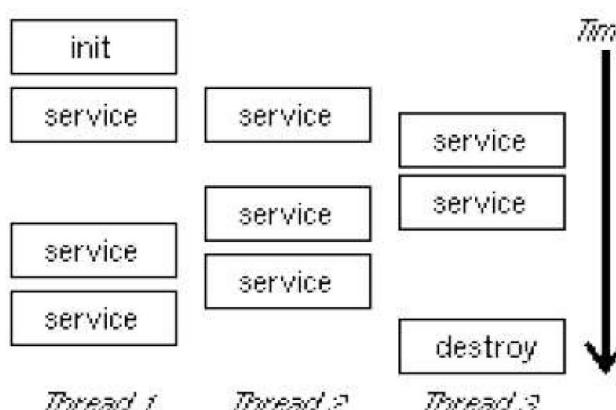
Desarrollo Web en Entorno Servidor

Tareas de sincronización en un servlet

• Sincronización en servlets

- Un contenedor de servlets puede invocar al método service() (o doGet, doPost,...) de forma concurrente a través de varios threads.
- Por tanto todos hilos generados y servlets deben sincronizar adecuadamente el acceso a objetos o recursos compartidos.
- Opciones para la carga de recursos de ficheros del WEB-INF:

```
URL resource =
    Thread.currentThread().getContextClassLoader().getResource("/img/i.jpg");
```

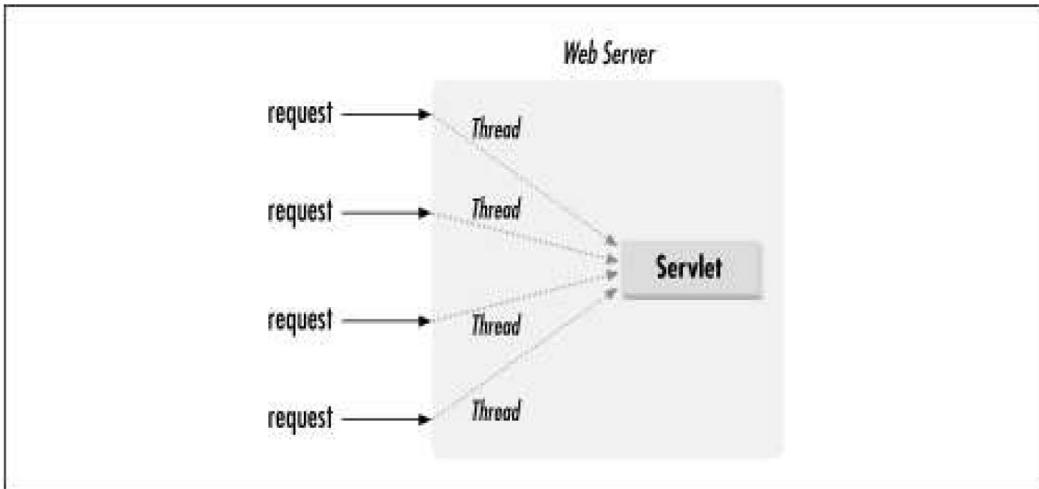


Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Tareas de sincronización en un servlet

- **Sincronización en servlets**



```

count++           // Thread 1
count++           // Thread 2
out.println      // Thread 1
out.println      // Thread 2

```

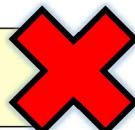
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Tareas de sincronización en un servlet

- **Sincronización en servlets**

```
public synchronized void doGet(HttpServletRequest req,
HttpServletResponse res)
```



```

PrintWriter out = res.getWriter();
synchronized(this) {
    count++;
    out.println("este servlet ha accedido " + count + " veces.");
}

```

1

```

PrintWriter out = res.getWriter();
int local_count;
synchronized(this) {
    local_count = count++;
}
out.println("este servlet ha accedido " + count + " veces.");

```

2

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Contexto del Servlet**• Contexto del Servlet**

- **ServletContext** define la vista que un servlet tiene de la aplicación web en la que está contenido.
- Un servlet puede utilizar el **ServletContext** para:
 - Realizar logs.
 - Obtener referencias URL a recursos del WEB-INF.
 - Fijar y almacenar atributos a los que otros servlets de la misma aplicación pueden acceder.
- Existe una instancia de **ServletContext** por cada aplicación web.



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Despliegue de un servlet

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://
  xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_
  _3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>My Webapp for DWEBS</display-name>
  <context-param>
    <param-name>jdbcURL</param-name>
    <param-value>jdbc:mysql://localhost:3306/javaweb_crud</param-value>
  </context-param>
  <context-param>
    <param-name>jdbcUsername</param-name>
    <param-value>root</param-value>
  </context-param>
  <context-param>
    <param-name>jdbcPassword</param-name>
    <param-value></param-value>
  </context-param>
</web-app>
  
```

```

String jdbcURL = getServletContext().getInitParameter("jdbcURL");
String jdbcUsername = getServletContext().getInitParameter("jdbcUsername");
String jdbcPassword = getServletContext().getInitParameter("jdbcPassword");
  
```

**Estructura de directorios del contenedor web:
Parámetros de inicialización para BBDD en web.xml**

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

Contexto del Servlet

- **Atributos del contexto**

Atributos de contexto

Permiten compartir información entre servlets que forman parte de la misma aplicación web.

public java.lang.Object getAttribute(java.lang.String name)

Devuelve el objeto que se ha asociado con el nombre *name*.

public java.util.Enumeration getAttributeNames()

Devuelve una enumeración con los nombres de los atributos.

public void removeAttribute(java.lang.String name)

Elimina el atributo de nombre *name*.

public void setAttribute(java.lang.String name, java.lang.Object object)

Fija el valor del atributo de nombre *name* al objeto *object*.

Contexto del Servlet

- **Acceso a recursos del contexto**

Recursos

La interfaz ServletContext ofrece acceso directo a la jerarquía de contenidos estáticos que son parte de la aplicación web como HTML's, GIF's, etc.

public java.net.URL getResource(java.lang.String path)

throws MalformedURLException

Nos permite construir la URL de un recurso local al sistema de ficheros

public java.io.InputStream getResourceAsStream(java.lang.String path)

Devuelve el recurso que se encuentra en el camino *path* como un InputStream.

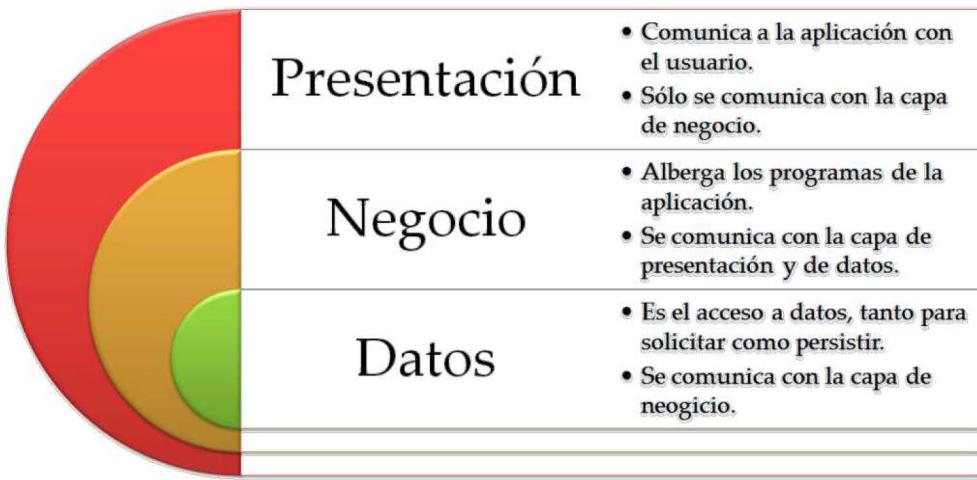
T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Patrones de diseño en la arquitectura del software:**

- **Desarrollo en tres capas**

- El desarrollo en capas nace de la necesidad de separar la lógica de la aplicación del diseño, separando a su vez los datos de la presentación al usuario.
- El desarrollo por capas facilita la estructura del software y aporta la posibilidad de **interoperar con otros sistemas**.



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

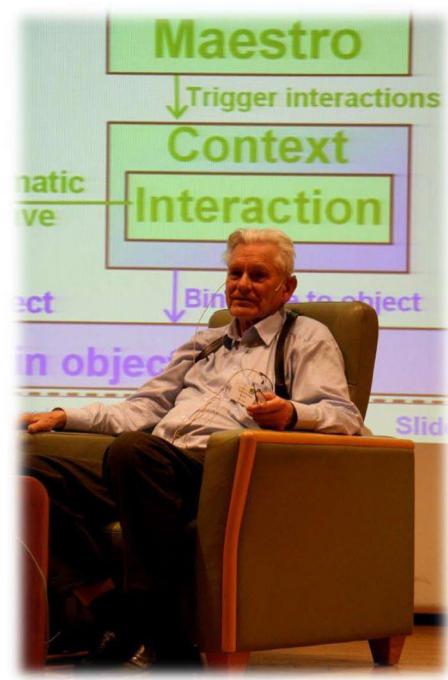
T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Patrones de diseño en la arquitectura del software:**

- **Modelo Vista Controlador**

- El **Modelo Vista Controlador** (Model View Controller - MVC) es un patrón de diseño que permite separar en capas nuestra aplicación para lograr un menor acoplamiento entre el código.
- Es una solución que sirve en todo tipo de aplicaciones, no solo de tipo web.
- El estilo fue descrito por primera vez en 1979 por Trygve Reenskaug, entonces trabajando en Smalltalk en laboratorios de investigación de Xerox.
- La implementación original está descrita en el documento “**Programación de Aplicaciones en Smalltalk-80: Como utilizar Modelo Vista Controlador**”.



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

- Patrones de diseño en la arquitectura del software:

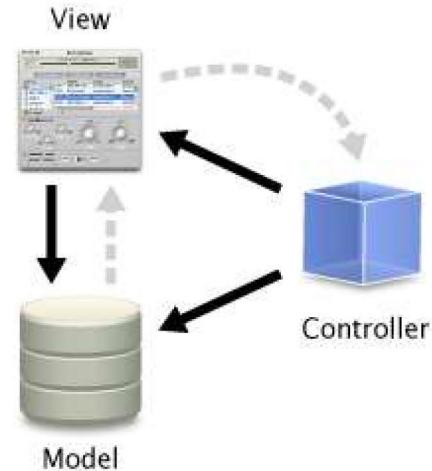
- **Modelo-Vista-Controlador (MVC)**

- Aporta distribución de responsabilidades e interoperar entre otros sistemas (su origen nace del modelo de tres capas).
- Usado para separar el modelo de lógica del negocio, capa de datos y presentación de la aplicación.
- Muy usado en cualquier desarrollo de aplicación web (*front-end/back-end*), multiplataforma, videojuegos, de acceso a datos, etc.

- Otros patrones de diseño que combinan métodos:

- DAO
- Singleton
- Fachada
- ...

Alejandro Cardo Grau



Desarrollo Web en Entorno Servidor

- Patrones de diseño en la arquitectura del software:

- **Modelo Vista Controlador (Implementaciones)**

- Existen varias implementaciones del patrón MVC basados en frameworks con diferentes tecnologías que nos permiten realizar diseños de programación basados en este patrón, los principales son:
 - Para PHP: **Symfony**, **Laravel**, **CakePHP**
 - Para Ruby: **RubyOnRails**
 - Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby
 - Para Java: **Struts**, **Spring**
 - Struts es una framework para el desarrollo de Java EE y es actualmente un proyecto independiente conocido como Apache Struts.
 - Para .net: **MonoRail**
 - Para Python: **Django**
 - Framework de desarrollo de aplicaciones web, escrito en Python, fué desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005.

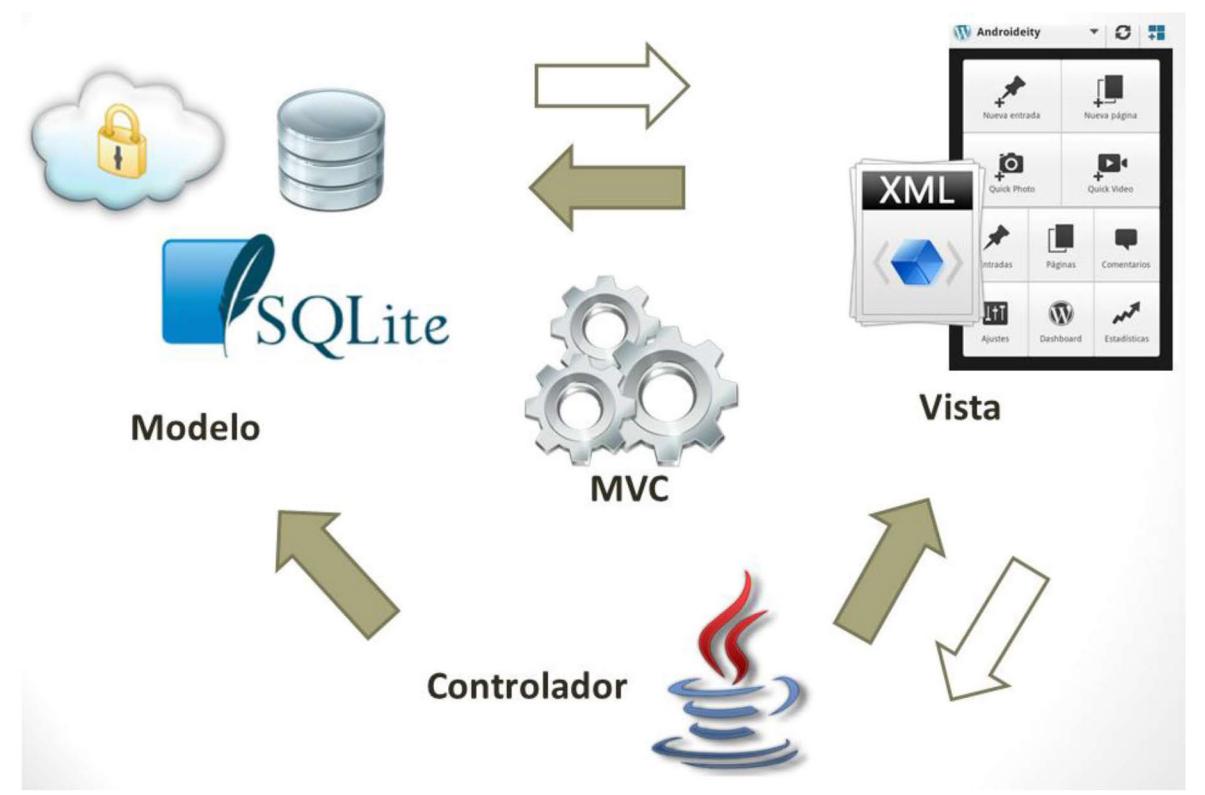
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



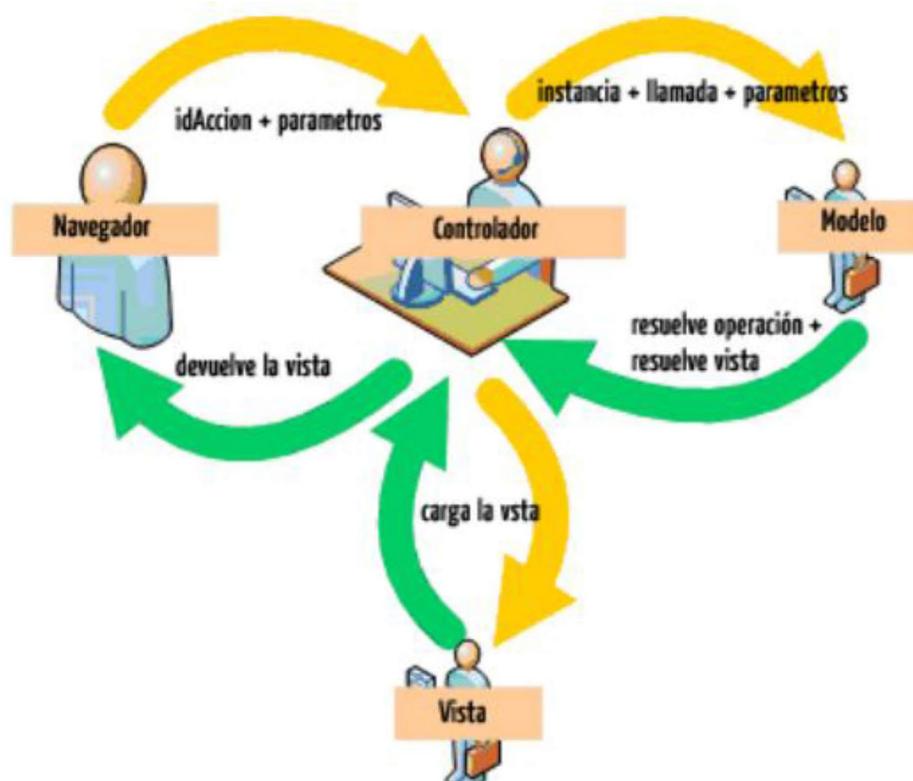
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



- Representación de los datos del dominio
- Lógica de negocio
- Mecanismos de persistencia



- Interfaz de usuario
- Elementos de interacción



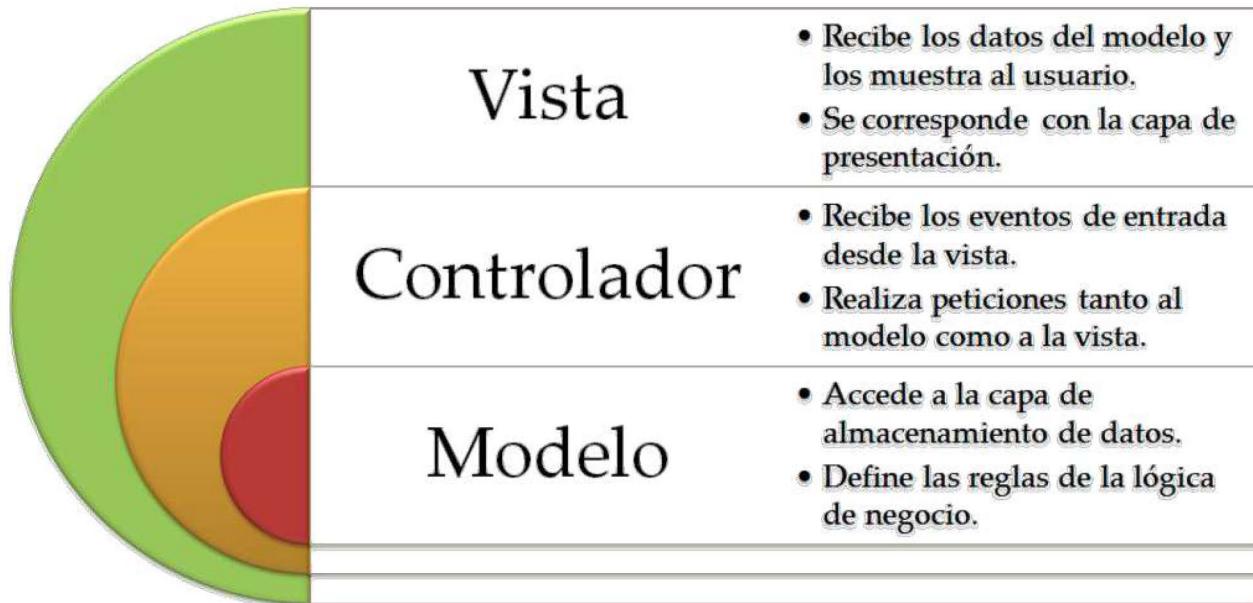
- Intermediario entre Modelo y Vista
- Mapea acciones de usuario → acciones del Modelo
- Selecciona las vistas y les suministra información

Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC
 Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**

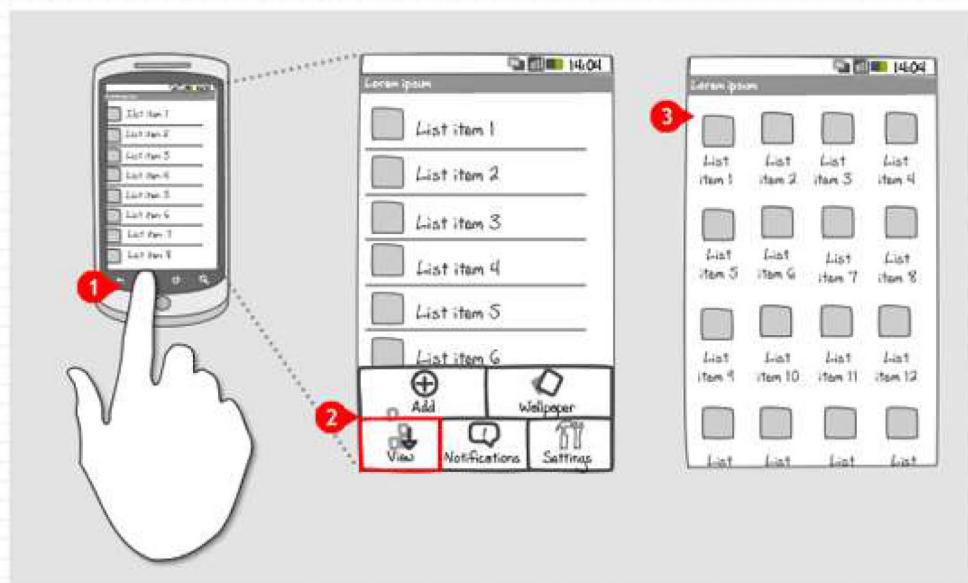


Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC
 Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



1. By pressing the hard key for the Options menu...
2. ... the Options icon menu slides in from the bottom. Touching the item to change the view...
3. ... changes the way the data is viewed

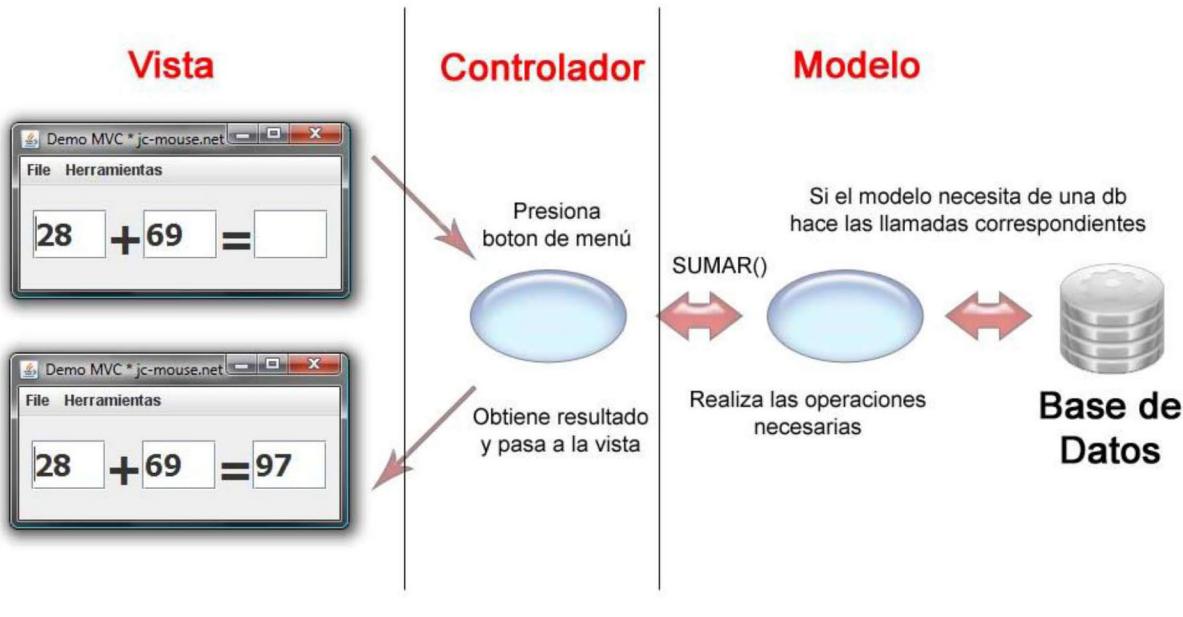
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



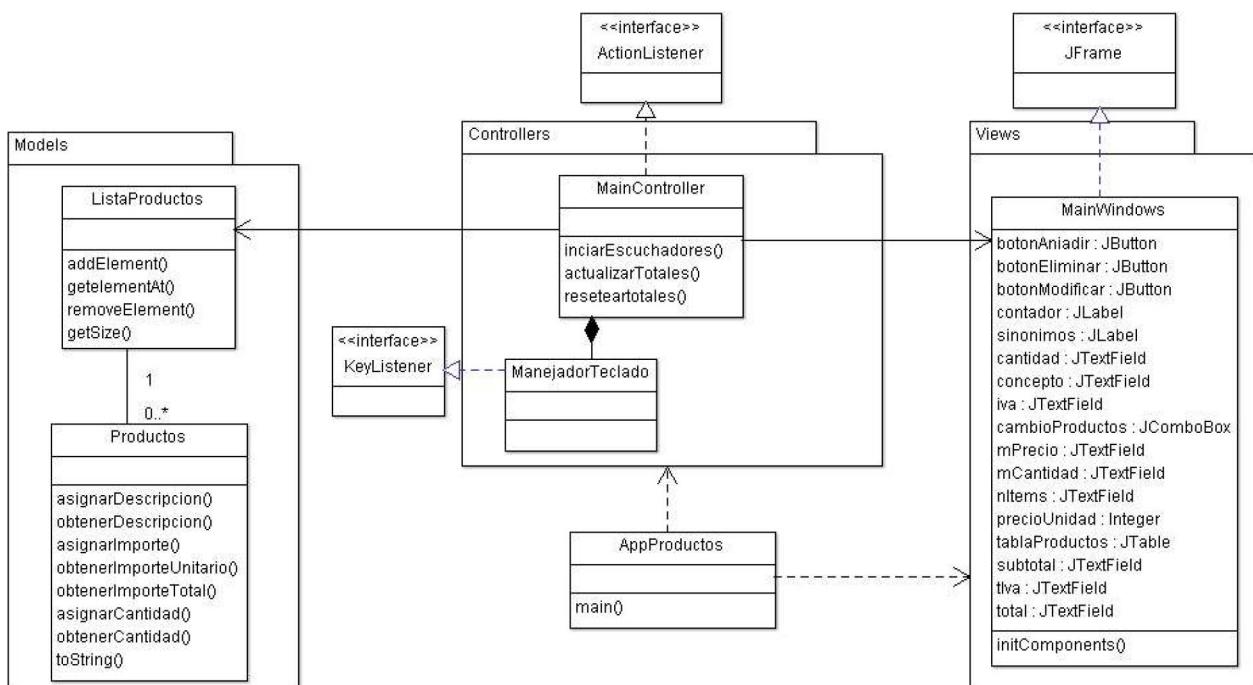
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



Modelo del Diagrama de Clases aplicando MVC

Alejandro Cardo Grau

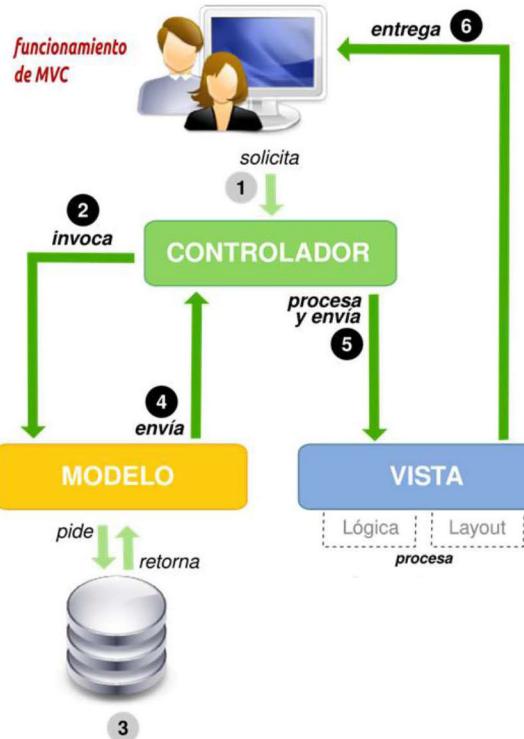
Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software



- Representación de los datos del dominio
- Lógica de negocio
- Mecanismos de persistencia



- Interfaz de usuario
- Elementos de interacción



- Intermediario entre Modelo y Vista
- Mapea acciones de usuario → acciones del Modelo
- Selecciona las vistas y les suministra información

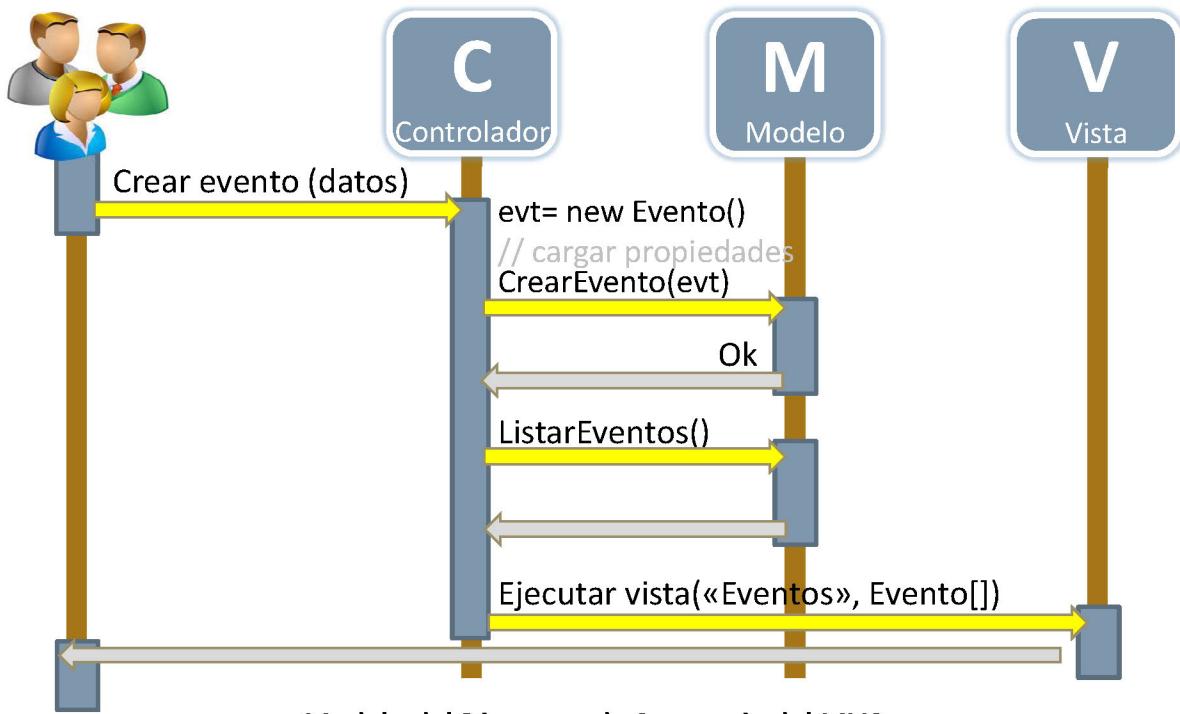
Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor

T3: Diseño y Desarrollo de Aplicaciones Web MVC

Patrones de diseño en la arquitectura del software

- **Modelo-Vista-Controlador**



Alejandro Cardo Grau

Desarrollo Web en Entorno Servidor