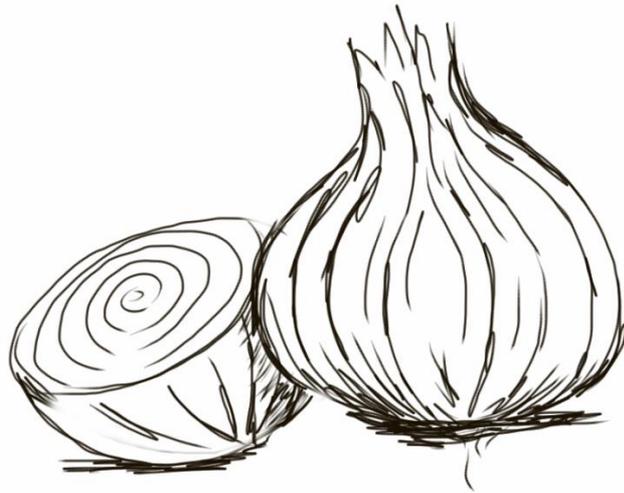


# Proyecto Final de Carrera

- Diseño e implementación de un Framework de presentación para aplicaciones J2EE



**Alumno: Xavier García Soto**  
**Consultor: Oscar Escudero Sánchez**

Enero 2015

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 1. Descripción Proyecto
- 2. Patrones de diseño J2ee
- 3. Estudio Frameworks Java Actuales
- 4. Framework presentación SignartFw
- 5. Aplicación web SignartApp
- 6. Conclusiones

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 1.Descripción del proyecto:
  - 1.1 Objetivos del proyecto
  - 1.2 Ranking Frameworks J2EE en el mercado
  - 1.3 Planificación del proyecto

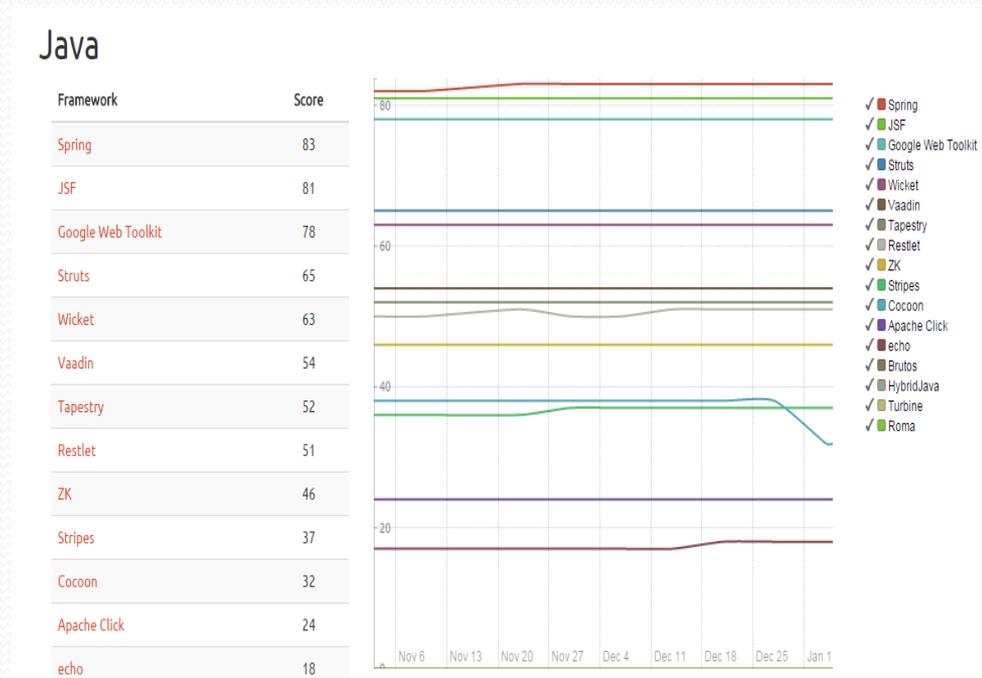
# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 1.1 Objetivos del proyecto
  - Analizar los Frameworks de la capa de presentación J2EE.
  - Comprender y aplicar los patrones J2EE de la capa de presentación.
  - Estudio del modelo MVC (Model Vista Controlador).
  - Diseño de un Framework J2EE para la capa de presentación.
  - Realizar una aplicación web utilizando el Framework diseñado.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

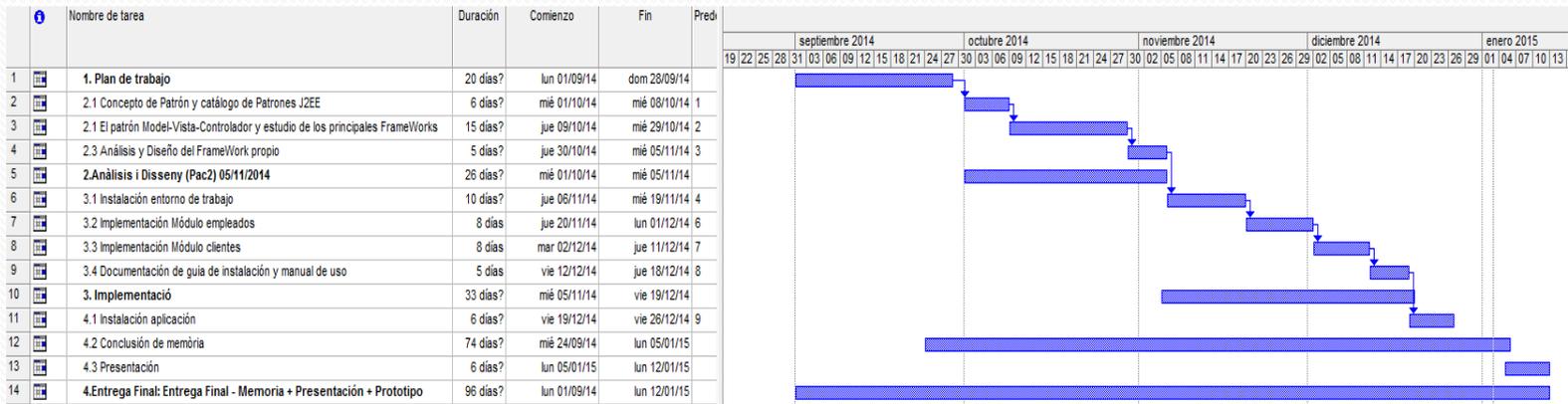
## 1.2 Ranking Frameworks J2EE en el mercado

- **Spring 83**
- **JSF 81**
- **Struts 65**
- **Google web Toolkit 78**  
(plataforma basada en Ajax)



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 1.3 Planificación del proyecto
  - Plan de trabajo (Pac1) 01/10/2014
  - Análisis y Diseño (Pac2) 01/10/2014 - 05/11/2014
  - Implementación Framework + AppWeb (Pac3) 05/11/2014 - 19/12/2014
  - Memoria + Presentación (PFC) 19/12/2014 - 12/01/2015



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 2. Patrones de diseños J2EE Capa Presentación
  - **Intercepting Filter:** Añade filtros para verificar, codificar y transformar la información antes de enviarla o recibirla.
  - **Front Controller:** Centraliza todas la solicitudes de entrada y salida en una única clase java controladora.
  - **View Helper:** Aglutina cierto código para presentar la información situada en estructuras de datos complejas y abstractas en la capa de presentación.
  - **Composite View:** Nos facilita la opción de incluir elementos repetitivos en los jsp. (footer, header,...)

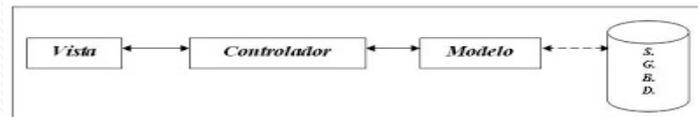
# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 2. Patrones de diseños J2EE Capa Presentación
  - Dispatcher view: Centraliza la gestión del control de acceso, recuperación de contenido y gestión de la vistas en la propia vista del usuario.
  - Service to Worker: Combina y agrupa los patrones Front Controller, View Helper y Dispatcher View en un solo patrón, pero no en la propia vista.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 3. Estudio de los Frameworks J2EE actuales

- 3.1 El modelo MVC



- 3.2 Java Server Faces



- 3.2 Struts

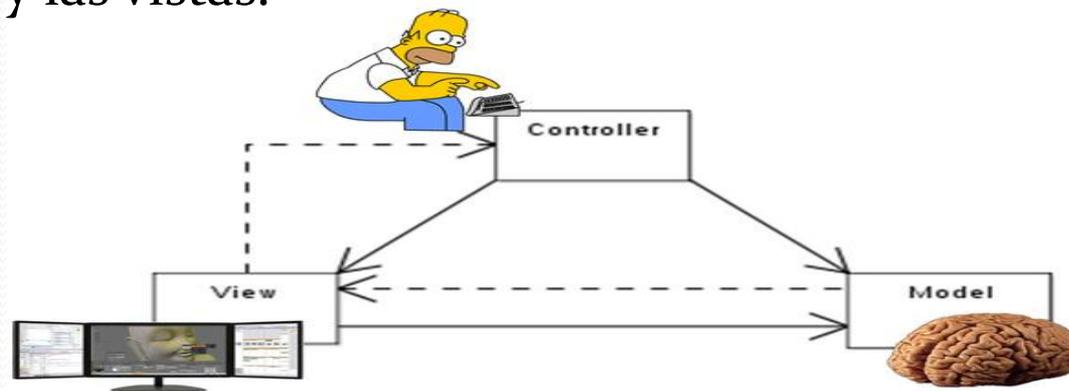


- 3.2 Spring



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 3.1 El Modelo MVC - “Divide y vencerás”: Separa las responsabilidades de aplicación en 3 capas.
  - **Modelo:** Cerebro o Núcleo. Implementa la lógica de la aplicación.
  - **Vista:** Visualización. Presenta la información al usuario.
  - **Controlador:** Director de Orquesta. Dirige las peticiones entre el modelo y las vistas.



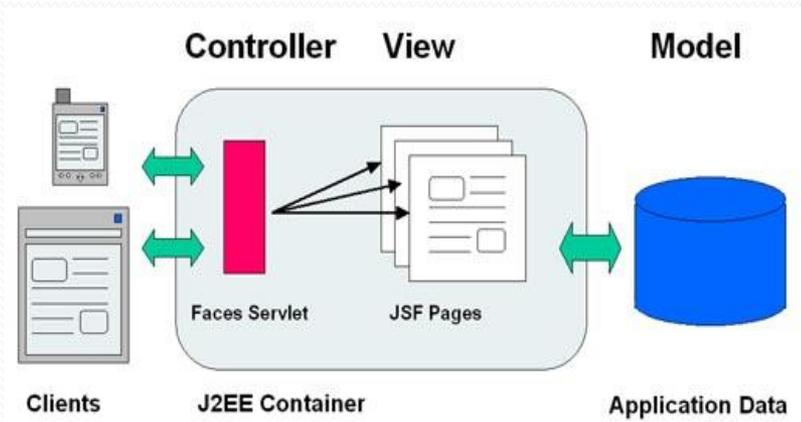
# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.2 JSF

- Desarrollado por SUN Microsystems a partir de 2004

## Características:

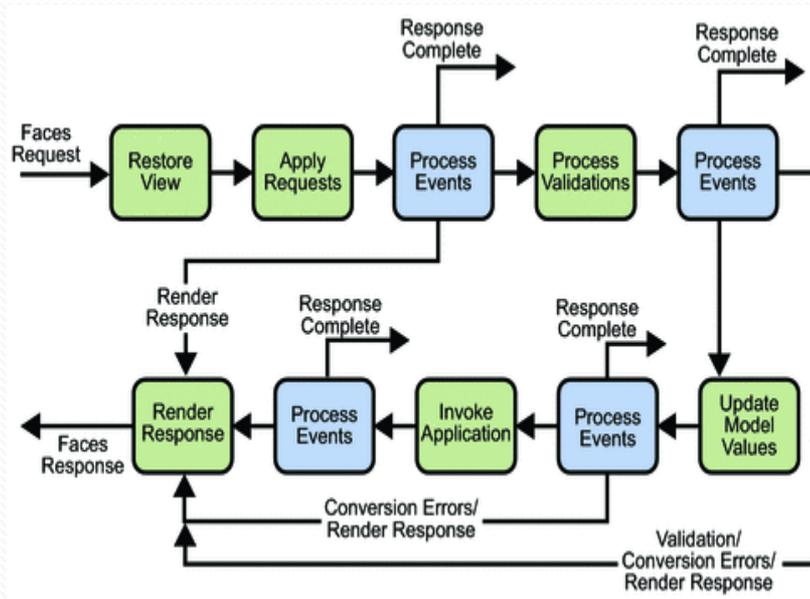
- Forma parte del estándar J2EE y aplica arquitectura MVC.
- Dispone de herramientas visuales para su desarrollo.
- Introduce el concepto de código reutilizable.
- Utiliza JSP para generar vistas.
- Los controladores se implementan con POJOS.



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.2 JSF (Ciclo de vida)

## Compuesto por 6 fases



- Crea o restaura la vista
- Aplica los valores del request
- Ejecuta las validaciones
- Actualiza el modelo
- Invoca a la lógica de la aplicación
- Devuelve respuesta

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.3 Struts

- Se implemento a partir del año 2000 por Apache Software Foundation.
- Struts 1 ya no disponible. Este FW evoluciono a Struts 2.



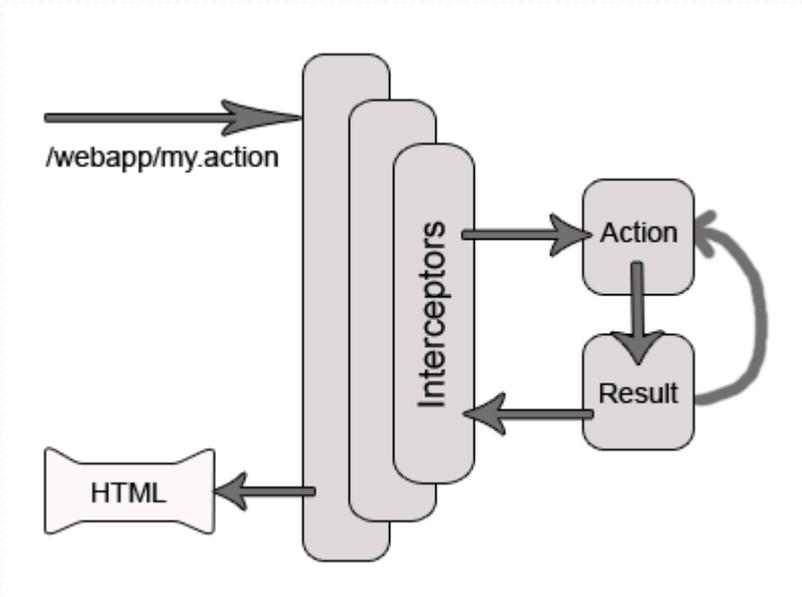
## Características

- Forma parte del estándar J2EE y aplica arquitectura MVC.
- Las vistas se implementan en JSP
- La navegación entre páginas se realiza en archivos .xml
- Todas las clases se basan en interfaces.
- Acciones se basa en POJO.
- Utiliza el patrón Front Controller que llama directamente a las interfaces

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.3 Struts (ciclo de vida)

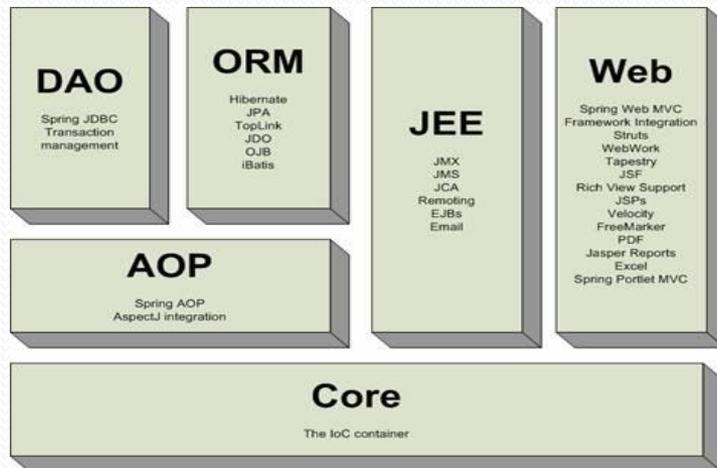
- Usuario envía petición a través de un formulario ya indica el acción debe realizar.
- Los Interceptors hacen de controladores. Según el formulario y acción recibidos ejecutan una o otra acción.
- El modelo realiza la acción y devuelve el resultado a los interceptors.
- Estos finalmente entregan los datos a la vista indicada.



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.4 Spring

- Su primera versión fue lanzada por Apache en 2004



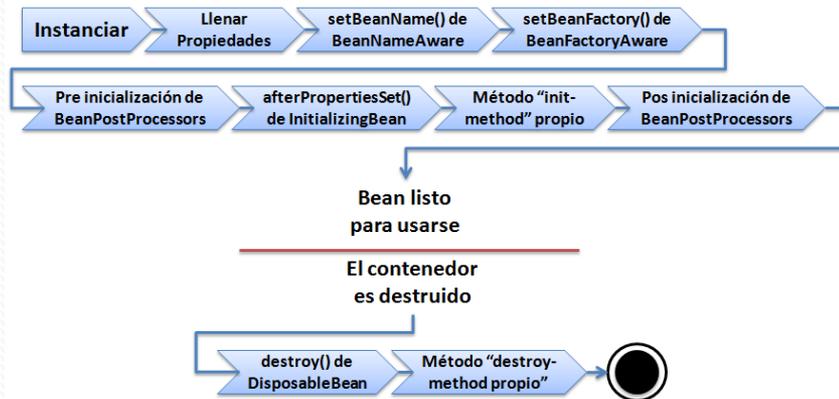
## Características

- Forma parte del estándar J2EE y aplica arquitectura MVC.
- Código es más sencillo, más natural y más reducido.
- Programación orientada a aspectos.
- Inyección Orientada a Aspectos.
- Código más fácil de depurar y más fácil de mantener.
- Código más reusable y independiente entre sí.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 3.4 Spring (ciclo de vida)

- Instancia el bean.
- Inyecta las propiedades según se hayan definido en los ficheros de configuración.
- Establece el nombre del bean (atributo id) y de la factoría.
- Si hay algún BeanPostProcessor, se llamará al método `postProcessBeforeInitializacion()`.
- Se llama al método `afterPropertiesSet()` provocando que si el bean tiene declarado un método de inicialización personalizado se ejecute este método.
- Si hay algún BeanPostProcessor, se llama al método `postProcessAfterInitializacion()`.



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- **4. Framework presentación SignartFw**
  - 4.1 Características SignartFw
  - 4.2 Arquitectura
  - 4.3 Diseño externo
  - 4.4 Diseño interno

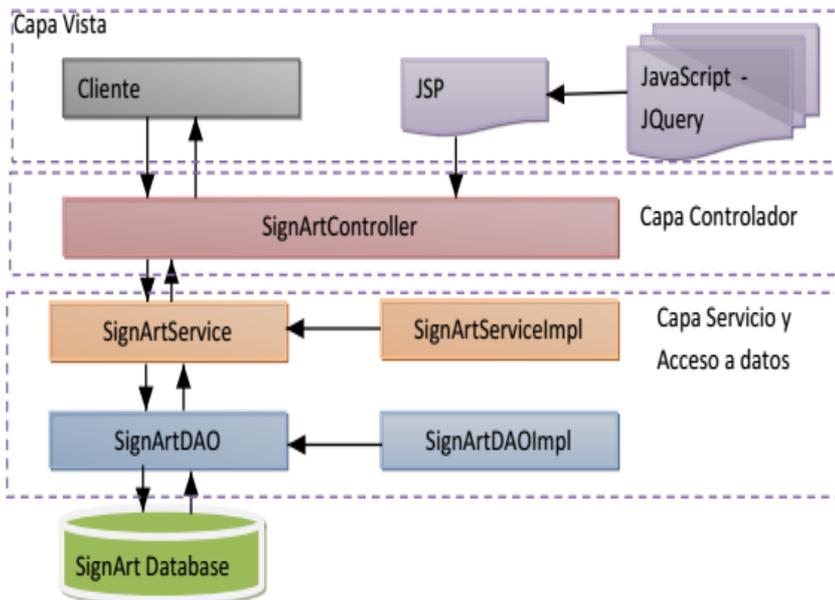


# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 4.1 Características **SignartFw**
  - Framework basado en MVC y que utiliza la combinación de diferentes patrones J2EE.
  - Internacionalización: Las aplicaciones web realizadas con Signartw se podrán escribir en diferentes idiomas de manera sencilla.
  - Validación de campos: Los campos en SignartFw se validaran mediante JQuery.
  - Estilo propio: Se han implementado hojas de estilos propias para la capa de presentación en css.
  - Uso de Jsp y taglibs en las pantallas del clientes.
  - Configuración a través de ficheros XML.
  - Cabecera y Footer propios
  - Conexión a base de datos a través del módulo Hibernate.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

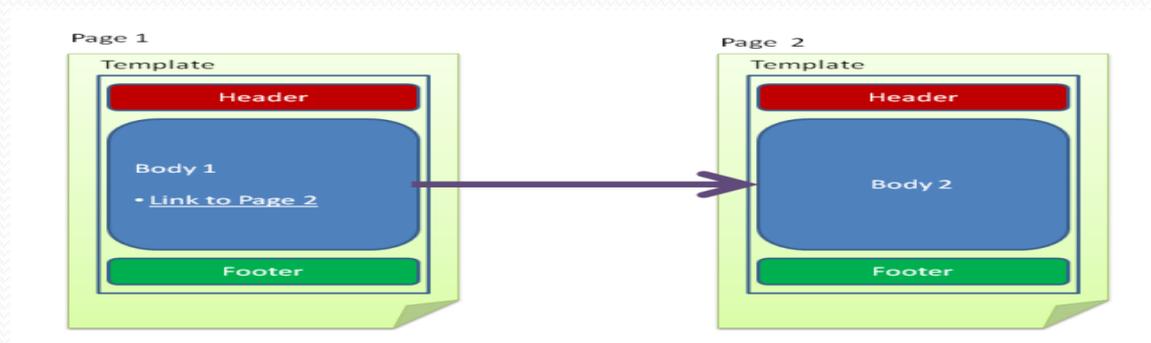
## 4.2 Arquitectura SignartFw



- Arquitectura distribuida en varias capas.
- Combinaremos diferentes patrones para la capa de presentación: Front Controller, Intercepting filter, Composite View
- Existirán tantos controladores como módulos implementemos.
- En la capa de servicio los métodos se encontrarán referenciados en interfaces e implementados en clases.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 4.3 Diseño externo SignartFw
  - Las vistas se implementaran siguiendo el patrón Composite View.
  - Tendrán un header, body, footer, independientes.
  - Los tres elementos se integraran en el mismo Jsp
  - El Header y el footer se reutilizarán.

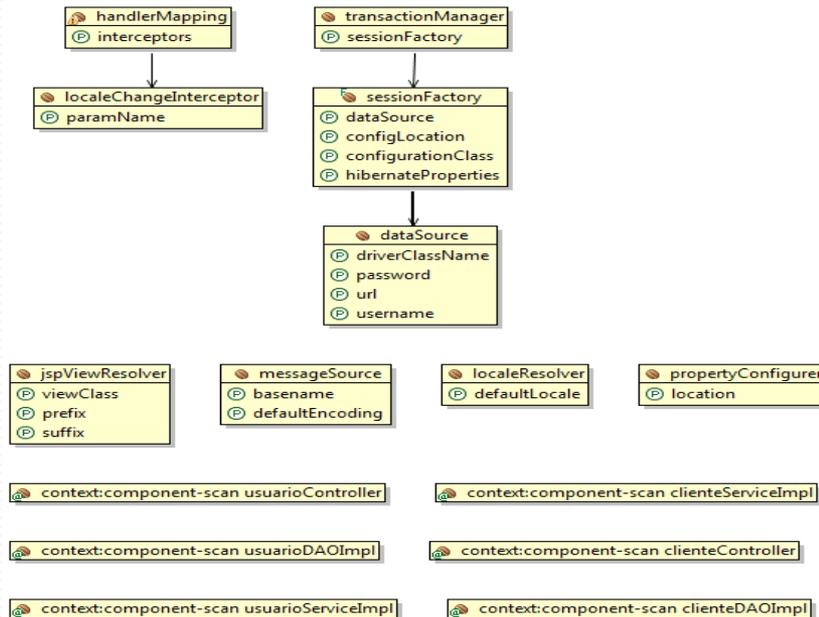


# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 4.4 Diseño interno SignartFw

### Características :

- Estructura de capas, organizadas en packages.
- Archivos de configuración XML (signartFw-servlet.xml)
- Diseño Filtros a través del patrón Intercepting Filter mediante Jquery
- Diseño de taglibs a través del patrón View Helper

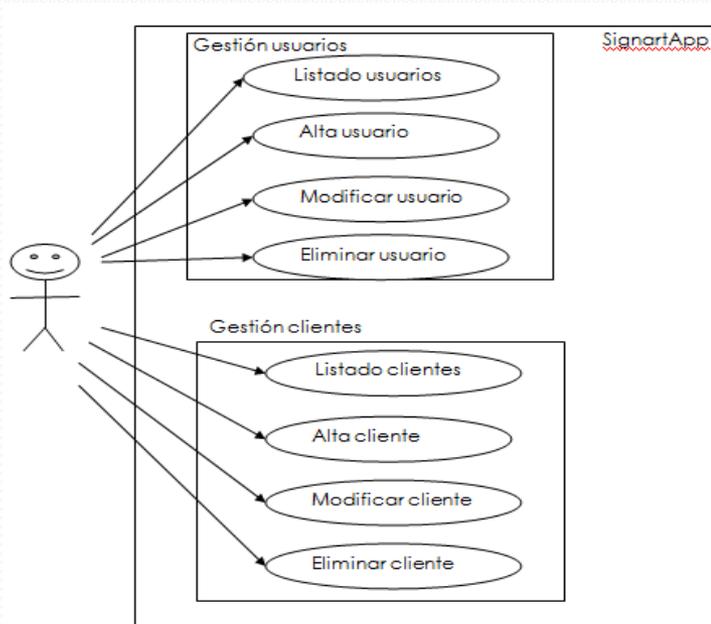


# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5. Aplicación web SignartApp
  - 5.1 Diagrama de caso de Uso
  - 5.2 Página inicial
  - 5.3 Gestión Usuarios
  - 5.4 Gestión Clientes

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

## 5.1 Diagrama de caso de uso SigartApp

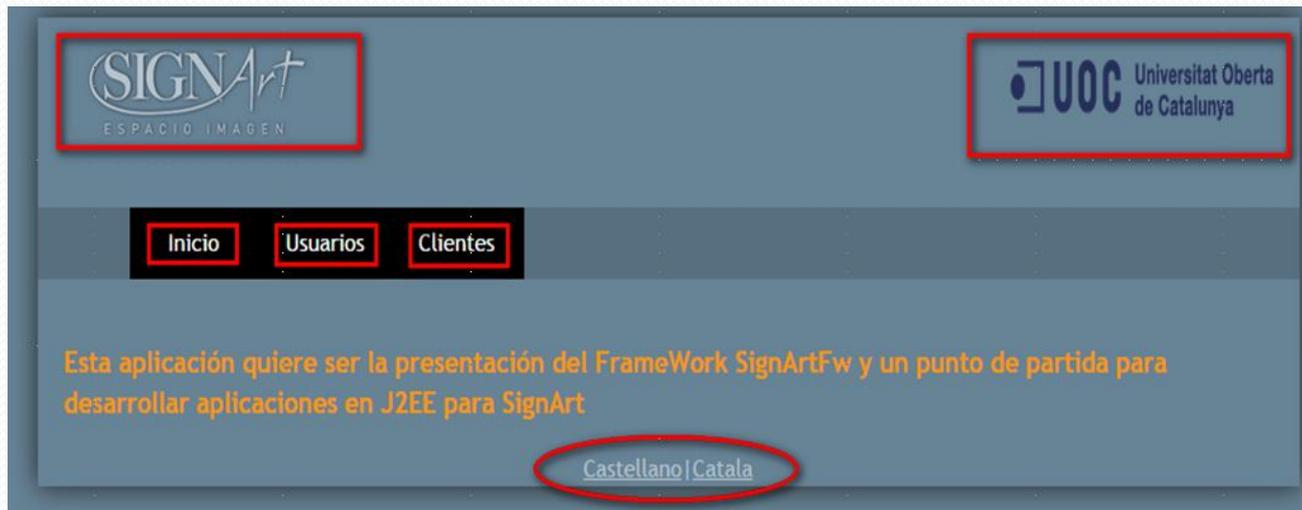


## Funcionalidades básicas

- Página bienvenida
- Listado de usuarios
- Alta de usuarios
- Modificación usuarios
- Eliminar usuarios
- Listado clientes
- Alta de clientes
- Modificar clientes
- Eliminar clientes

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5.1 Página inicial SignartApp
  - Header: Logos y Botonera
  - Body: Mensaje de bienvenida
  - Footer: Cambio de idioma de la aplicación



# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5.3 Gestión de Usuarios
  - Listado de usuarios (Se podrán crear/modificar/eliminar)




Inicio
Usuarios
Clientes

### Listado de Usuarios

Nombre	Primer Apellido	Segundo Apellido	Email	Telefono	Perfil
<a href="#">Xavier</a>	<a href="#">Garcia</a>	<a href="#">Soto</a>	xaviergarso@uoc.edu	933401101	1 <a href="#">Eliminar</a>
<a href="#">Oscar</a>	<a href="#">Escudero</a>	<a href="#">Sanchez</a>	oscar.escudero@uoc.edu	651520581	2 <a href="#">Eliminar</a>
<a href="#">Raquel</a>	<a href="#">Perez</a>	<a href="#">De Amo</a>	raque.perez@gmail.com	933401102	2 <a href="#">Eliminar</a>
<a href="#">Julia</a>	<a href="#">Casacuberta</a>	<a href="#">Martí</a>	julia.casacuberta@gmail.com	916545258	1 <a href="#">Eliminar</a>

Crear nuevo usuario
Castellano | [Catala](#)

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5.3 Gestión de Usuarios
  - Alta/Modificación de usuarios

The screenshot shows a web application interface for user management. At the top left is the logo for SIGNArt (ESPACIO IMAGEN). At the top right is the UOC (Universitat Oberta de Catalunya) logo. Below the logos is a navigation menu with three items: 'Inicio', 'Usuarios', and 'Clientes'. The 'Usuarios' item is highlighted. The main content area is titled 'Modificar usuario' and contains a form with the following fields:

Nombre	<input type="text" value="Raquel"/>
Primer Apellido	<input type="text" value="Perez"/>
Segundo Apellido	<input type="text" value="De Amo"/>
Telefono	<input type="text" value="933401102"/>
Email	<input type="text" value="raque.perez@gmail.com"/>
Perfil	<input type="text" value="2"/>

Below the form is a button labeled 'Modificar usuario'. At the bottom right of the page, the text 'Castellano | Catala' is displayed.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5.4 Gestión de Clientes
  - Listado de clientes (Se podrán crear/modificar/eliminar)

The screenshot shows a web application interface for client management. At the top left is the SIGNArt logo (ESPACIO IMAGEN) and at the top right is the UOC logo (Universitat Oberta de Catalunya). Below the logos is a navigation menu with three items: 'Inicio', 'Usuarios', and 'Clientes'. The 'Clientes' item is highlighted. The main content area is titled 'Listado de clientes' and contains a table with the following data:

Nombre	Alias	Nif	Email	Teléfono 1	Mas datos
<a href="#">Restaurante Creixell</a>	<a href="#">Creixell</a>	<a href="#">43434781E</a>	<a href="#">creixell@gmail.com</a>	9335869852	 <a href="#">Eliminar</a>
<a href="#">Universitat Oberta Catalunya</a>	<a href="#">UOC</a>	<a href="#">57424595G</a>	<a href="#">recursos@uoc.edu</a>	987654321	 <a href="#">Eliminar</a>

Below the table is a button labeled 'Añadir nuevo cliente'. At the bottom of the page, there is a language selector: 'Castellano | Catala'.

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 5.4 Gestión de clientes
  - Alta/Modificación de clientes

**SIGNArt**  
ESPACIO IMAGEN

UOC Universitat Oberta de Catalunya

**Inicio Usuarios Clientes**

**Modificar cliente de Signart**

Nombre	Universitat Oberta Catalunya
Alias	UOC
Descripción	Universitat On-lines
Nif	57424595G
Cif	
Email	recursos@uoc.edu
Teléfono 1	987654321
Teléfono 2	
Dirección	Avinguda Tibidabo 37
Población	Barcelona
Código Postal	08025

Fecha creación: 2015-01-01 21:18:41.0

Modificar cliente de Signart

Castellano | Catala

# Diseño e implementación de un Framework de presentación para aplicaciones J2EE

- 6. Conclusiones
  - Ha sido un reto elaborar un Framework J2EE desde cero.
  - Aprendizaje teórico y practico sobre los patrones de diseño Java y como se pueden aplicar a nivel de aplicación.
  - Aprendizaje de tecnologías  
Maven/Hiberate/JQuery/Taglibs
  - Creo que la realización de este proyecto supone una mejora de mi carrera académica/profesional

**Gracias por vuestra  
atención**



**Alumno: Xavier García Soto**  
**Consultor: Oscar Escudero Sánchez**

Enero 2015