



Frameworks

(Marco de Trabajo)

PATRONES Y ARQUITECTURA DE SOFTWARE

MSC. NIXON DUARTE A.

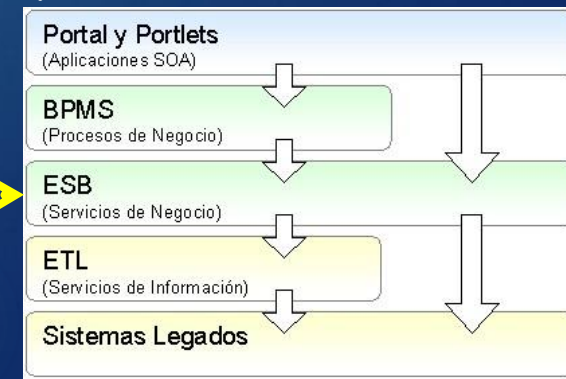
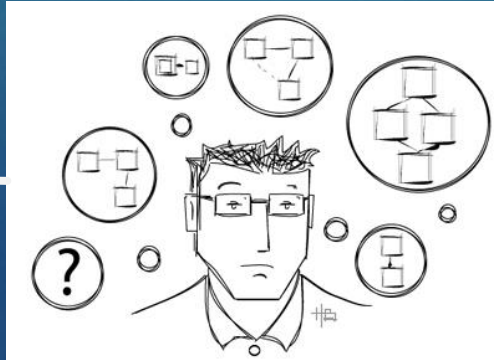
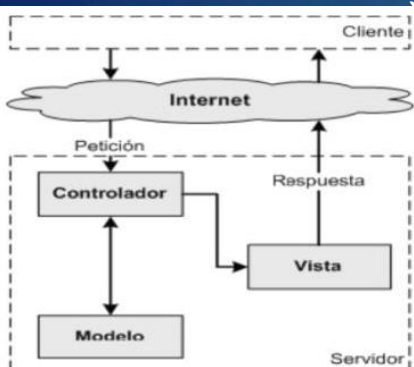
2015-10

Introducción

- ▶ El diseño de **arquitecturas de software** se basa en la toma de decisiones que involucran la selección de conceptos con el fin de **satisfacer** los **requerimientos** que influyen sobre la **arquitectura**. Dentro de estos conceptos se encuentran los **patrones de diseño**, las **técnicas** y los **frameworks**. Uno de los retos importantes del diseño es la **selección** oportuna y adecuada de dichos elementos, en particular porque existen **cientos** de **patrones de diseño** y una gran variedad de frameworks. Además de **asegurarnos** de que los conceptos de diseño seleccionados **satisfagan** los **requerimientos** que influyen sobre la arquitectura, debemos tomar en cuenta el **impacto negativo** que pudieran tener en la satisfacción de los **atributos de calidad**. Un ejemplo de ello es la introducción de un **framework de persistencia** que facilita el **mapeo** orientado objetos a relacional, y por lo tanto promueve la **modificabilidad**, pero posiblemente influye **negativamente** sobre el **desempeño**.

Introducción

- ▶ Los **patrones** son **fundamentales** en la creación de arquitecturas de software, pero estos conceptos, resultan abstractos y no es muy claro como conectarlos con la tecnología que se utiliza para el desarrollo
- ▶ Definición de **Framework**:
 - ▶ Es un **patrón arquitectónico** que proporciona una **plantilla** extensible para aplicaciones dentro de un dominio.
 - ▶ Son **elementos reutilizables** de software que proveen funcionalidades genéricas enfocadas a **resolver** distintos aspectos de una aplicación.



Introducción

Java Frameworks



Rich Internet Applications: Frameworks

Open Source:



Proprietary:

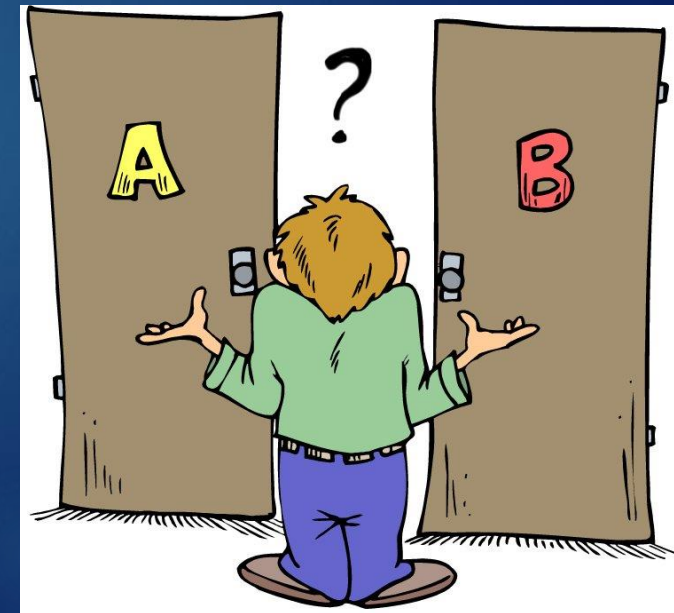


Microsoft
.net



Seleccionando un Frameworks

- ▶ Que framework utilizar ?
- ▶ Seleccionar un framework utilizado por un amigo ?
- ▶ Estudiar objetivamente la oferta de frameworks?
- ▶ Cual escoger dentro del abanico de opciones?



Frameworks:

Verticales y Horizontales

- ▶ De acuerdo con el alcance se definen dos tipos:
 - ▶ **Verticales:**
 - ▶ Permite realizar una (y solo una) tarea. Un claro ejemplo es Hibernate, para el mapeo de BDs relacionales y objetos.
 - ▶ **Horizontales:**
 - ▶ Permiten realizar una aplicación completa. Obligan a adaptar la forma de trabajo a la definida por el framework. Un claro ejemplo es Spring.



Frameworks:

Tareas Comunes

- ▶ Las tareas comunes y que facilitan los frameworks son:
 - ▶ **Ajax**: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).
 - ▶ **MVC**: es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
 - ▶ **Internacionalización**: para el manejo de diferentes idiomas.
 - ▶ **Object-Relational Mapping**: para olvidarnos de SQL.

Frameworks:

Tareas Comunes

- ▶ Las tareas comunes y que facilitan los frameworks son:
 - ▶ **Testing:** permitiendo probar de forma unitaria los objetos que forman la aplicación.
 - ▶ **Templates engine:** motor de plantillas que permita el desarrollo rápido de la capa de presentación.
 - ▶ **Form validation:** la validación de formularios es una tarea dura y muy común.
 - ▶ **Security:** La seguridad entendida como la autenticación y la autorizaciones obligatoria para el 90% de aplicaciones Web.
 - ▶ **Caching:** guardar durante un tiempo las páginas generadas, total o parcialmente, permite ahorrar mucho tiempo de computación. Una buena práctica muy recomendable.

Frameworks:

Ventajas y Desventajas

▶ Ventajas:

- ▶ Facilitan el desarrollo de aspectos muy comunes si el framework objeto ya se conoce.

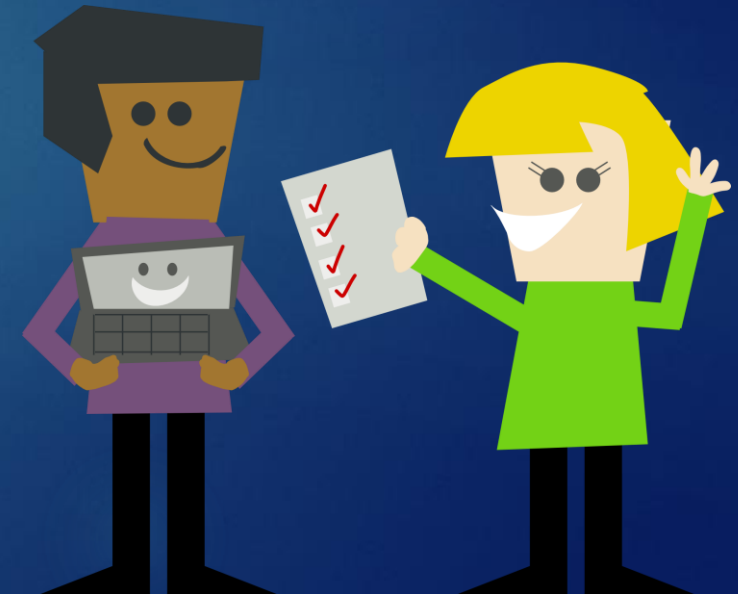
▶ Desventajas:

- ▶ **Curva** de aprendizaje.
- ▶ **Sobrepasar** el alcance de nuestro proyecto, ofreciendo **más posibilidades** que las utilizadas.
- ▶ Nuestro proyecto **dependerá** de terceros **proveedores**, que en caso de discontinuar el framework obligarán a sustituirlo (obligando, en ocasiones, a **reescribir** el código de nuestro proyecto)
- ▶ **Sobrecargan** la aplicación con librerías y dependencias. Ocupando **memoria** y en consecuencia reduciendo el rendimiento general del servidor que hospeda nuestra Web.

Frameworks:

Justificación

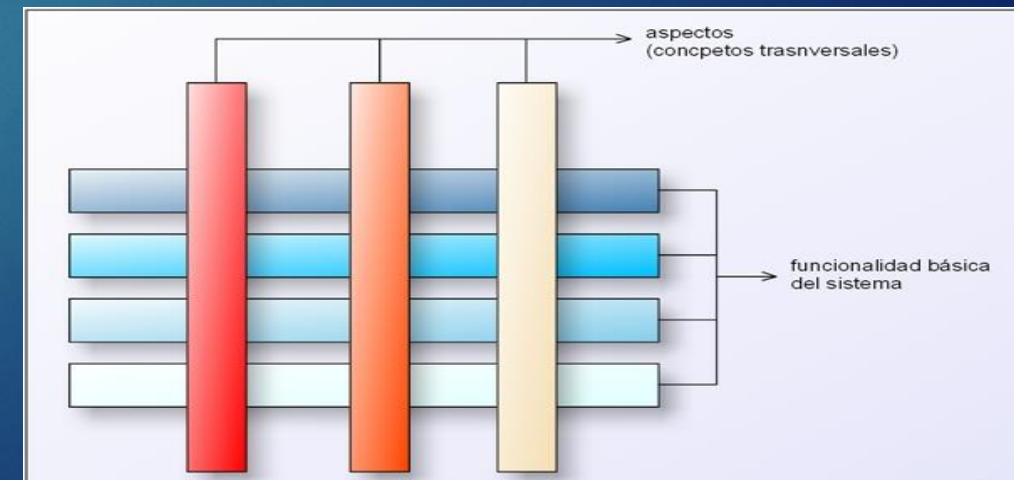
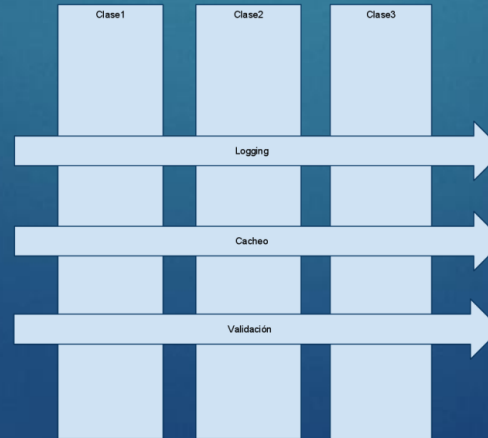
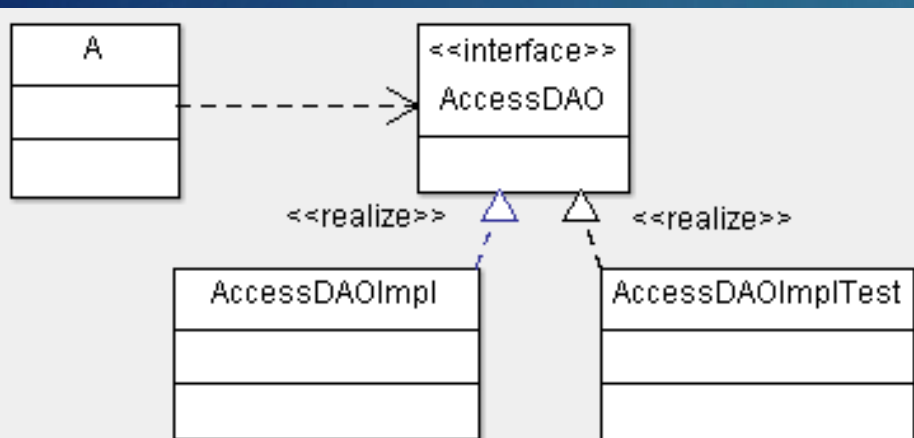
- ▶ Se justifica el uso de un framework cuando:
 - ▶ El equipo de desarrollo está formado en él y tiene experiencia en su uso.
 - ▶ El proyecto se adapta perfectamente al framework pues este implementa (y solo implementa) funcionalidades necesarias.



Frameworks Horizontales:

Spring

- ▶ Es un framework ligero para construir aplicaciones empresariales. Aunque Spring se encuentra dividido en distintos módulos, cada uno de los cuales se encarga de partes diferentes de nuestra aplicación.
- ▶ **Es liviano y no intrusivo**: generalmente los objetos que programamos no tienen dependencias en clases específicas de Spring. Sus características principales son **inyección** de dependencias y programación orientada a **aspectos**.



Frameworks Horizontales:

Spring: Módulos

Módulos de Spring Framework

Spring AOP

Soporte para la Programación Orientada a Aspectos. Incluye clases de soporte para el manejo transaccional, seguridad, etc.

Spring ORM

Soporte para Hibernate, iBATIS y JDO

Spring Web

Soporte a diferentes Frameworks Web, tales como JSF, Struts, Tapestry, etc

Spring MVC

Solución MVC de Spring, además incluye soporte para Vistas Web JSP, Velocity, Freemarker, PDF, Excel, XML/XSL

Spring DAO

Soporte JDBC
Manejo Excepciones SQL
Soporte para DAOs

Spring Context

ApplicationContext
Soporte UI
Soporte JNDI, EJB, Remoting, Mail

Spring Core

Utilerias de Soporte Supporting Utilities
Contenedor IoC / Fábrica de Beans

Frameworks Horizontales:

Java Server Faces (JSF)

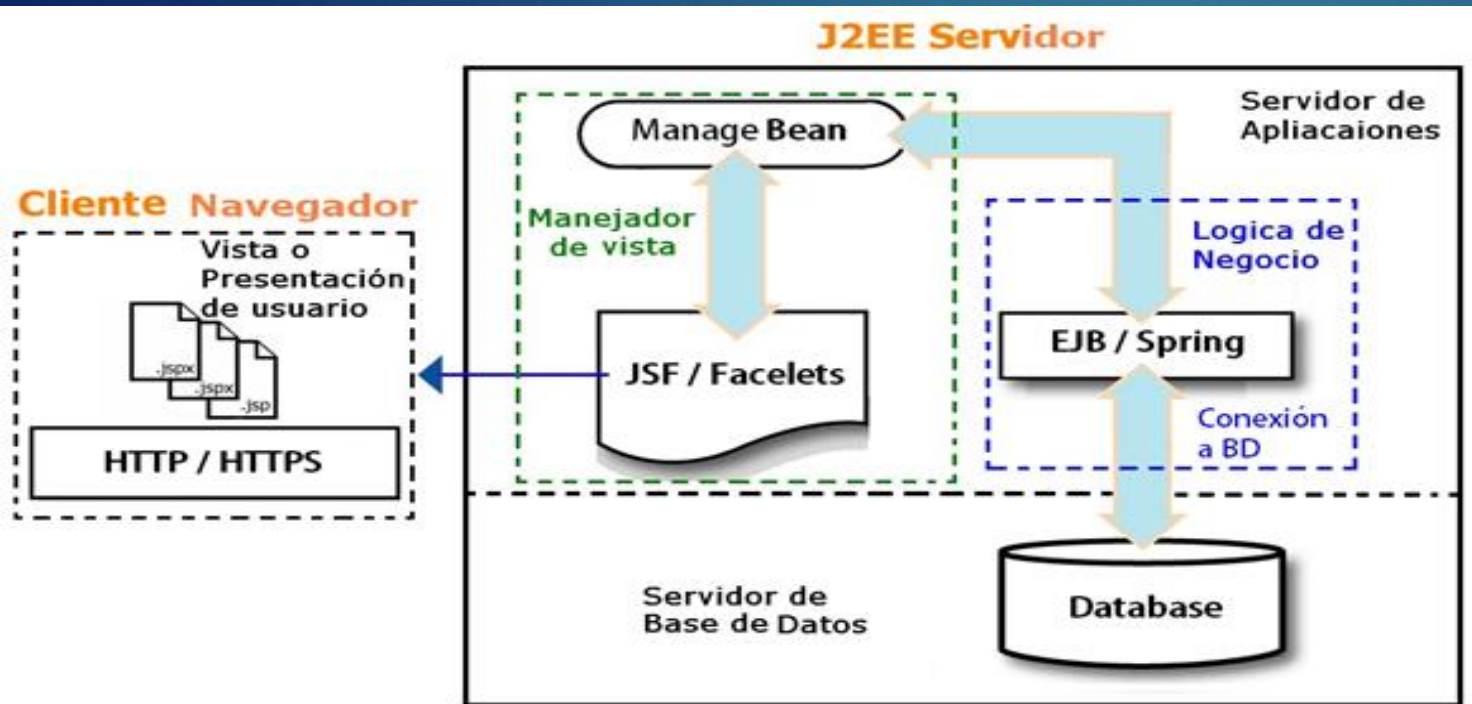
- ▶ JSF es una **especificación** de desarrollo para aplicaciones Web liberado en marzo del 2004 por Sun Microsystems y que está basada en el Modelo Vista Controlador (MVC). Esta orientado a mejorar la interfaz gráfica del usuario y mantiene el concepto básico del patrón **MVC** (separar el comportamiento de la presentación).
- ▶ **Cuenta** con muchos **frameworks** que facilitan el desarrollo de las aplicaciones **web** ya que utiliza un conjunto de componentes enriquecidos. Entre los principales Frameworks JSF se encuentran: **PrimeFaces**, **IceFaces**, **RichFaces**, **MyFaces** Trinidad y MyFaces.



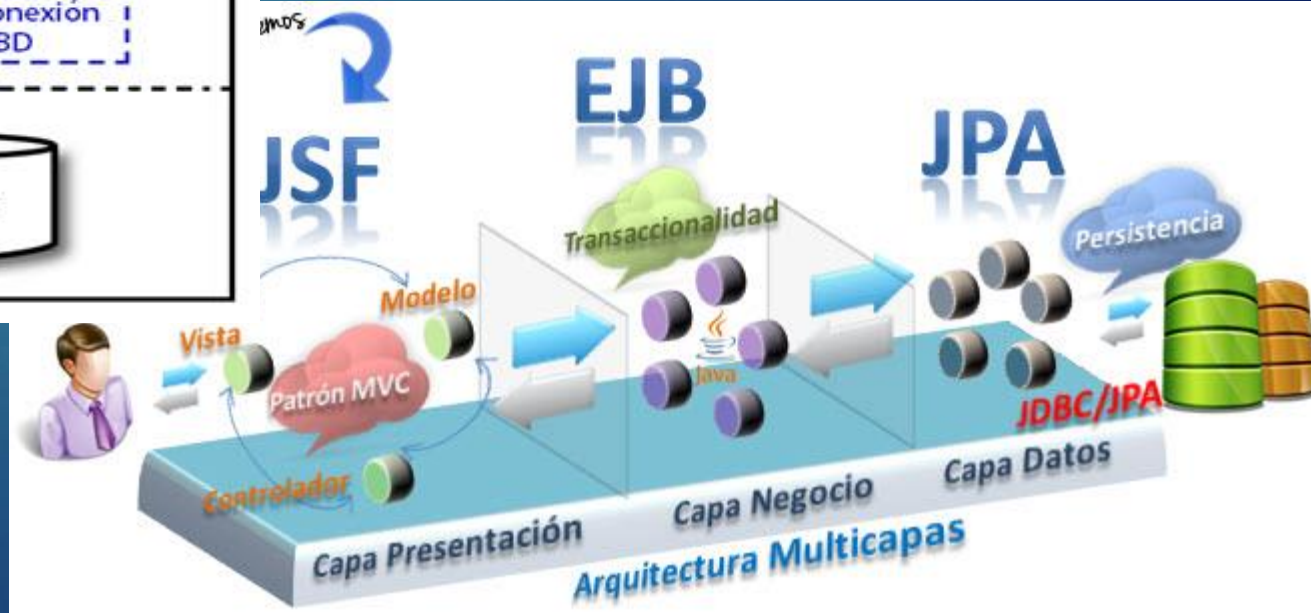
Frameworks Horizontales:

Java Server Faces (JSF)

ARQUITECTURA



ARQUITECTURA



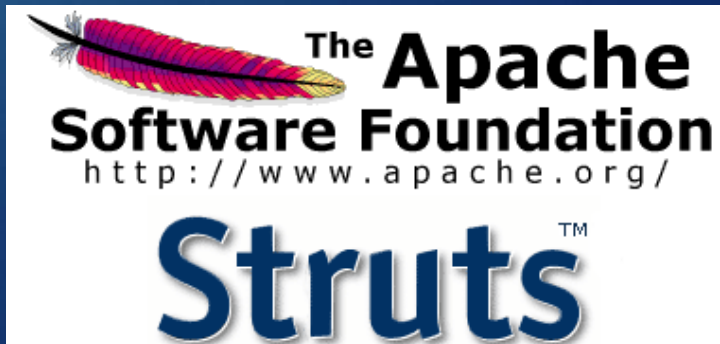
Frameworks Horizontales:

Struts

- Hay dos "generaciones" de Struts: la 1.x y la 2. El cambio de la generación 1 a la 2 no significa únicamente que se hayan añadido nuevas características. Es un cambio completo, tanto a nivel interno como externo. Cuando surgió **Struts 1** era el único **framework** existente de este tipo y en su diseño original había **deficiencias** en cuanto a **flexibilidad** y **simplicidad** de uso. Todo esto se ha solucionado en la versión 2. No obstante, y de manera paradójica, **Struts 2**, aun siendo mucho más potente, flexible y fácil de usar, **no ha tenido** ni de lejos tanta difusión como la versión 1, simplemente porque ahora ya tiene la **competencia** de otros frameworks como **Spring** o **JSF**.

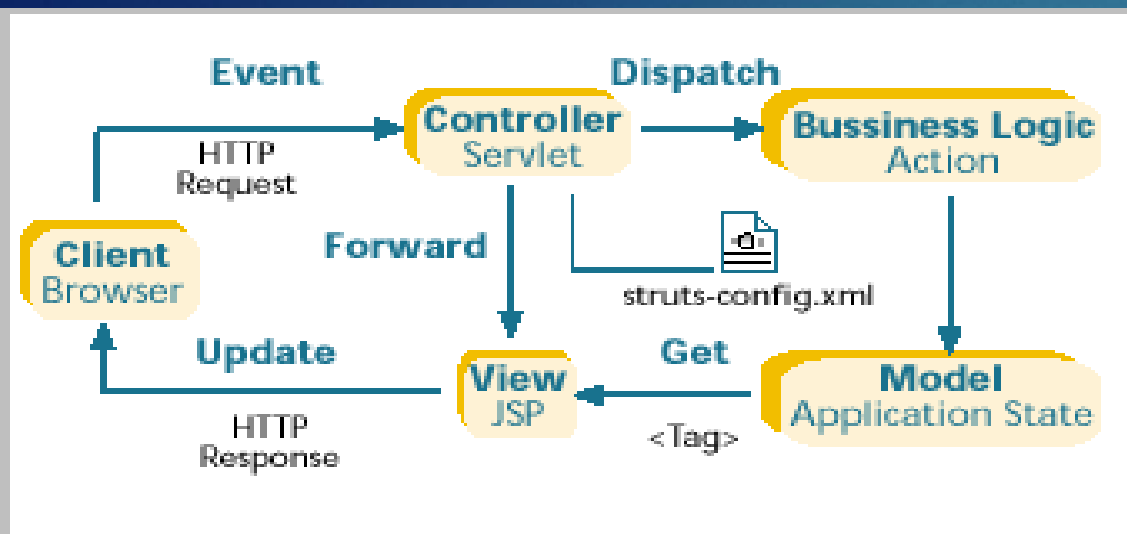


Struts²



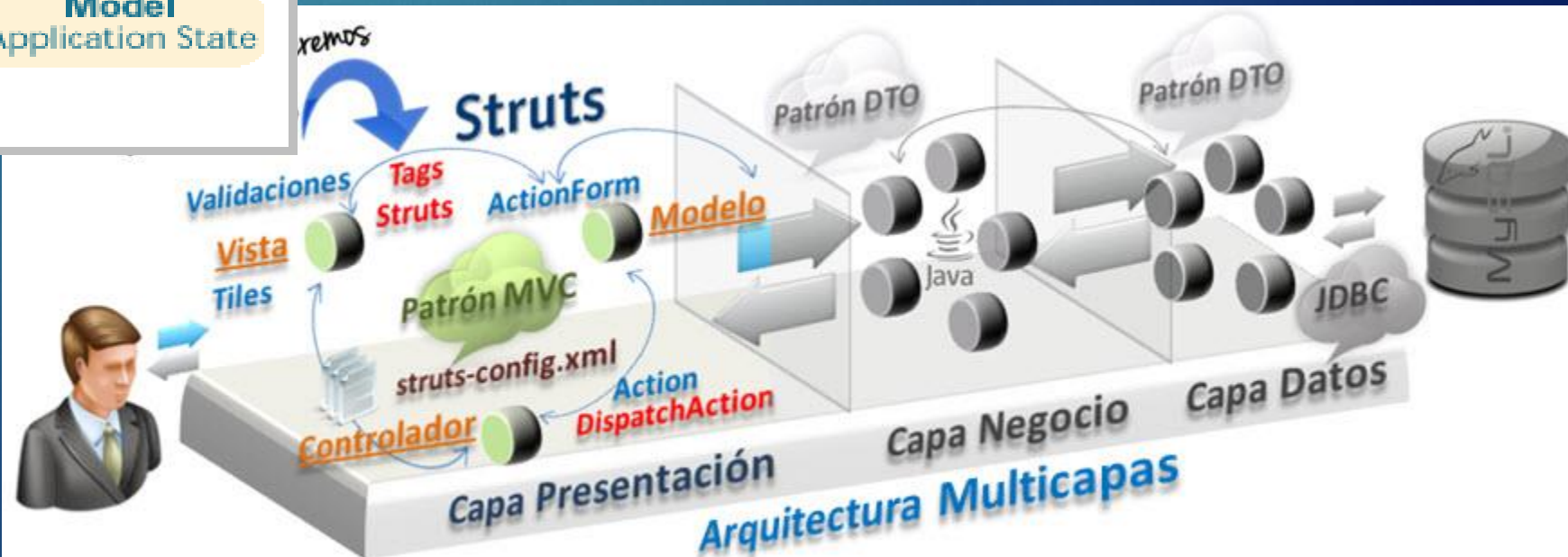
Frameworks Horizontales:

Struts



ARQUITECTURA

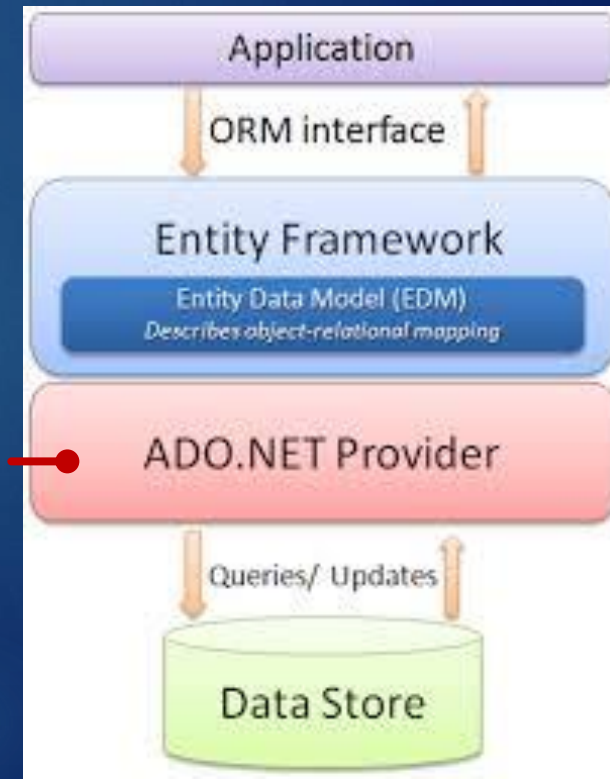
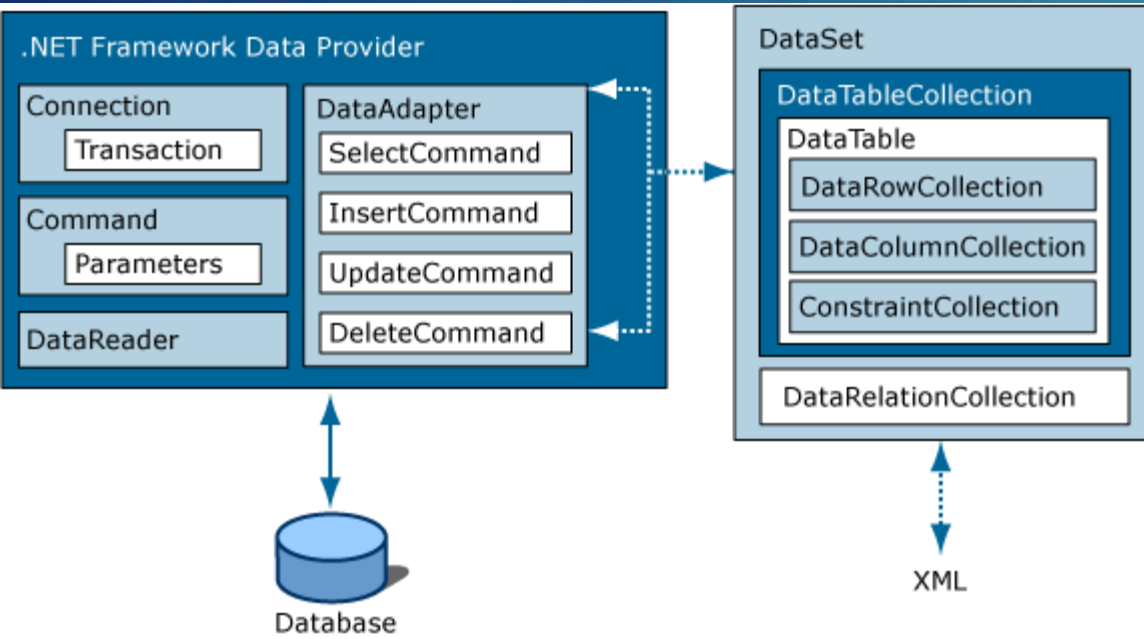
ARQUITECTURA



Frameworks Horizontales:

.Net

- es un framework de **Microsoft** que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita **un rápido desarrollo** de aplicaciones



Frameworks Verticales:

Hibernate

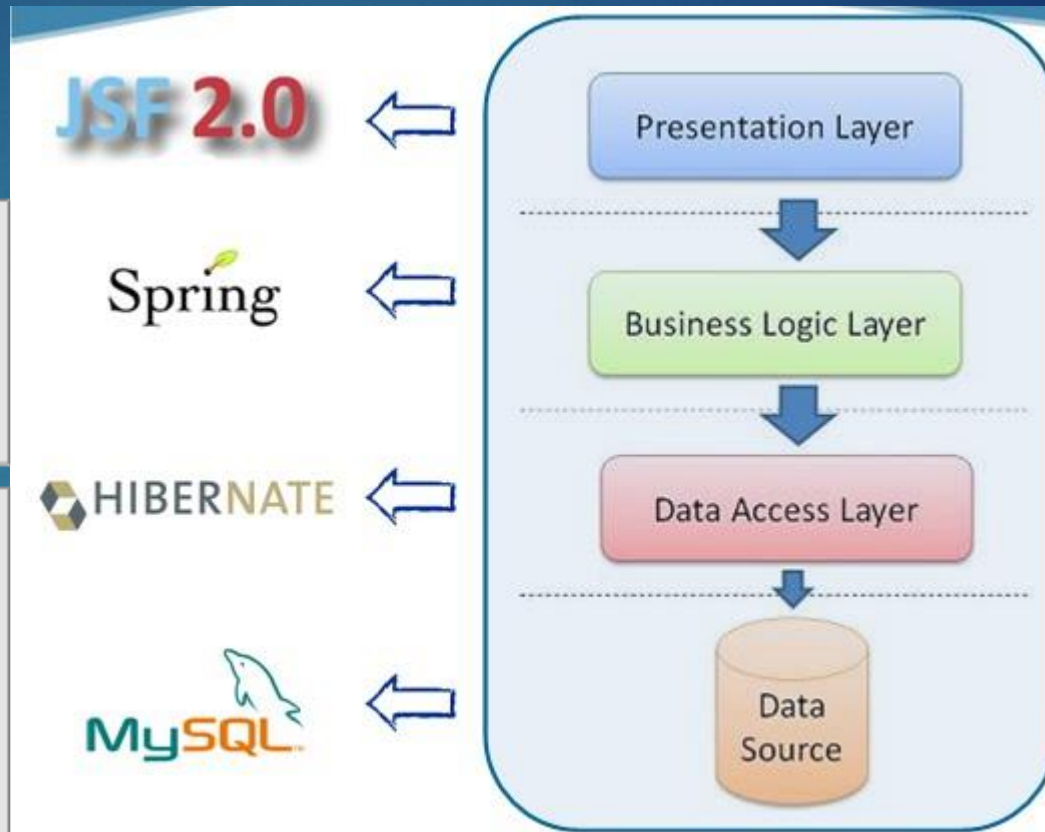
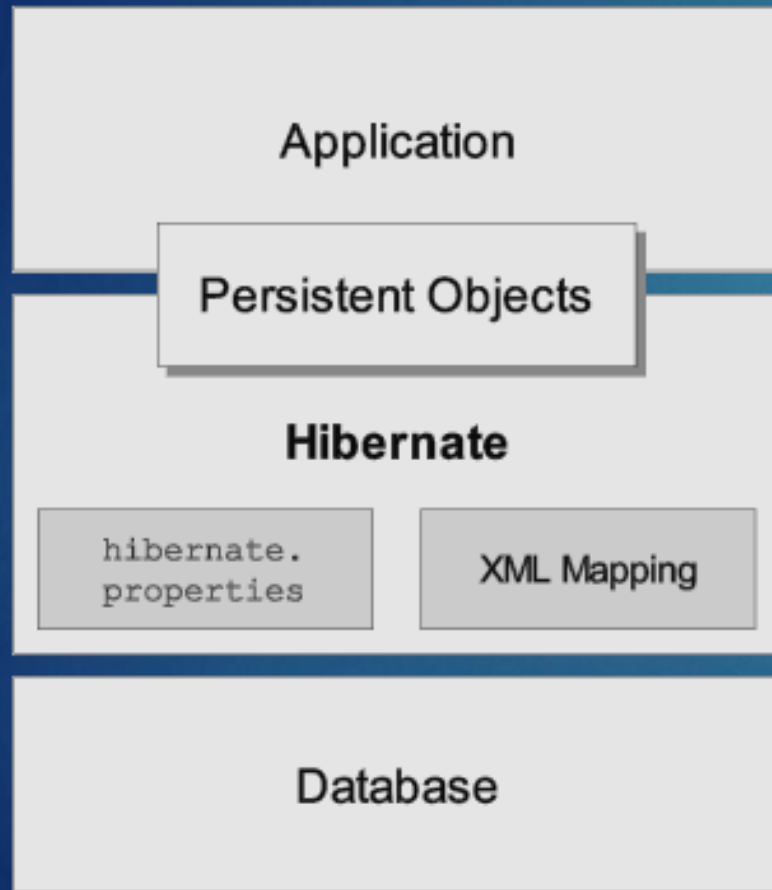
- ▶ Es una herramienta de **Mapeo** objeto-relacional (ORM) para la plataforma **Java** (y disponible también para **.Net** con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, **mediante archivos declarativos (XML)** o **anotaciones** en los **beans** de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la **licencia GNU LGPL**.



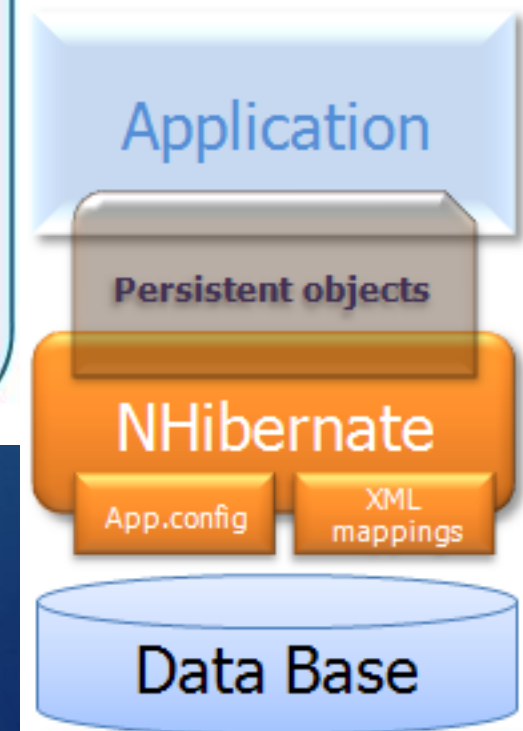
HIBERNATE

Frameworks Verticales:

Hibernate



ARQUITECTURA



Frameworks Verticales:

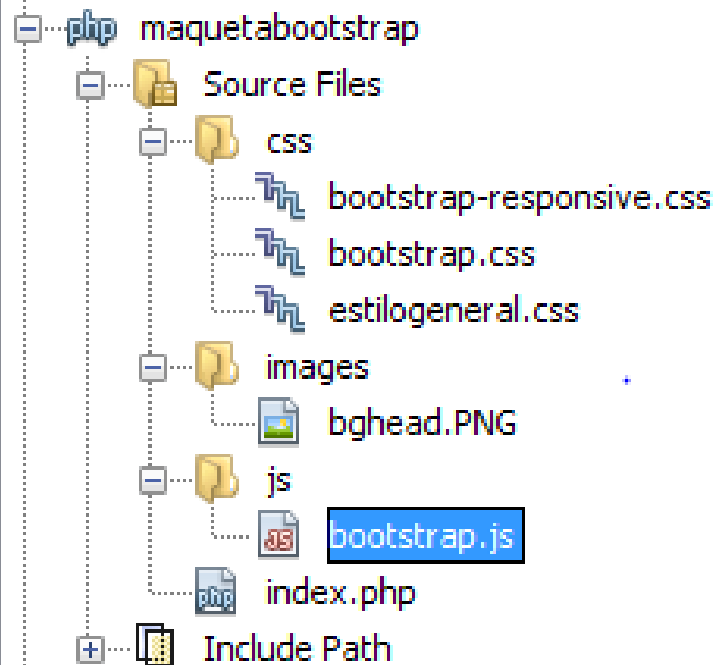
Bootstrap

- ▶ Es un framework diseñado para simplificar el proceso de creación de diseños web. Para ello nos ofrece una serie de plantillas CSS y de ficheros JavaScript, los cuales nos permiten conseguir:
 - ▶ **Interfaces** que funcionen de manera **brillante** en los navegadores actuales, y correcta en los no tan actuales.
 - ▶ Un **diseño** que pueda ser **visualizado** de forma correcta en **distintos dispositivos** y a distintas escalas y **resoluciones**.
 - ▶ Una **mejor integración** con las librerías usadas habitualmente, como por ejemplo **jQuery**.
 - ▶ Un diseño sólido basado en herramientas actuales y potentes como **LESS** o estándares como **CSS3/HTML5**



Frameworks Verticales:

Bootstrap

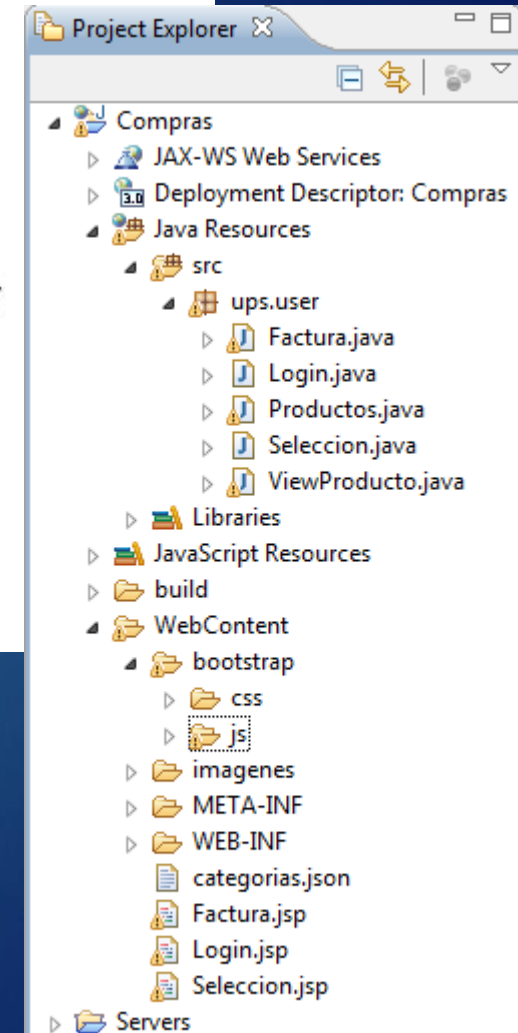


Organizando los assets web

```
proyecto/  
├─ app/  
├─ src/  
├─ vendor/  
└─ web/  
    ├─ css/  
    │   ├─ bootstrap.min.css  
    │   └─ app.css  
    └─ js/  
        ├─ jquery.min.js  
        └─ app.js
```

```
proyecto/  
├─ app/  
│   └─ Resources/  
│       └─ assets/  
│           └─ scss/  
│               └─ bootstrap/  
│                   └─ app.scss  
│                       └─ js/  
├─ src/  
├─ vendor/  
└─ web/  
    ├─ css/app.css  
    └─ js/app.js
```

Organización del Proyecto



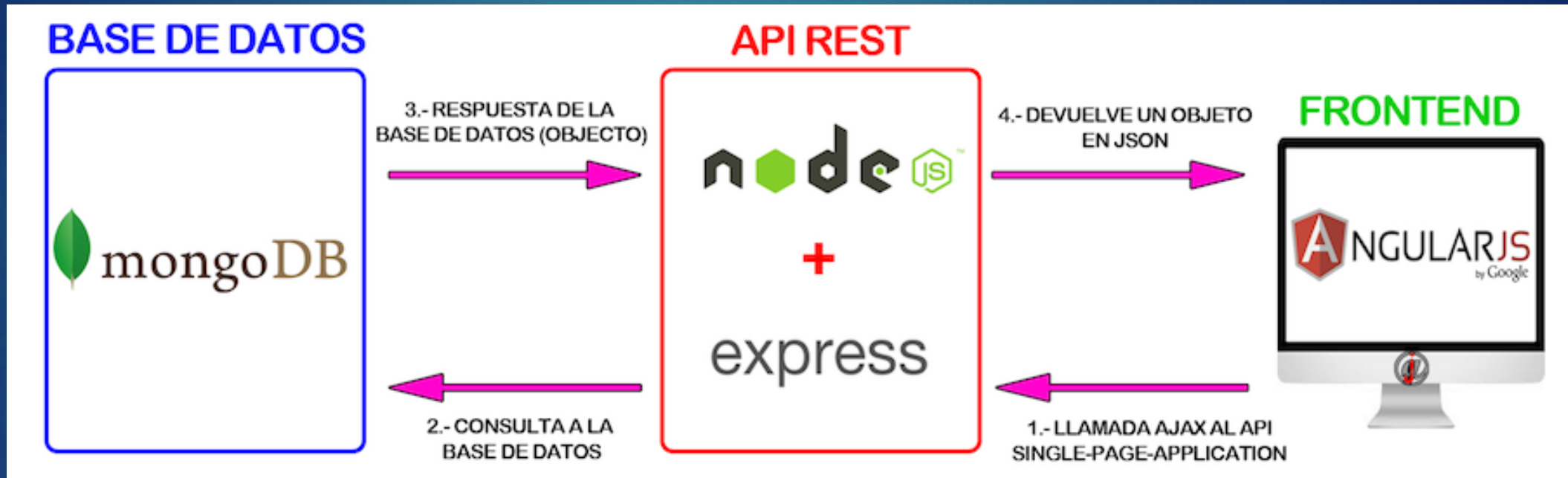
Frameworks Verticales:

Node.js y Express.js

- ▶ **Node.js** es un entorno de programación en la capa del servidor basado en el lenguaje de programación Javascript.
- ▶ Express.js, según sus creadores, es un framework de desarrollo de aplicaciones web minimalista y flexible para Node.js". Está inspirado en Sinatra, además es robusto, rápido, flexible y muy simple. Entre otras características, ofrece Router de URL (Get, Post, Put ...), facilidades para motores de plantillas (Jade, EJS, JinJS ...).

Frameworks Verticales:

Node.js y Express.js



Preguntas

