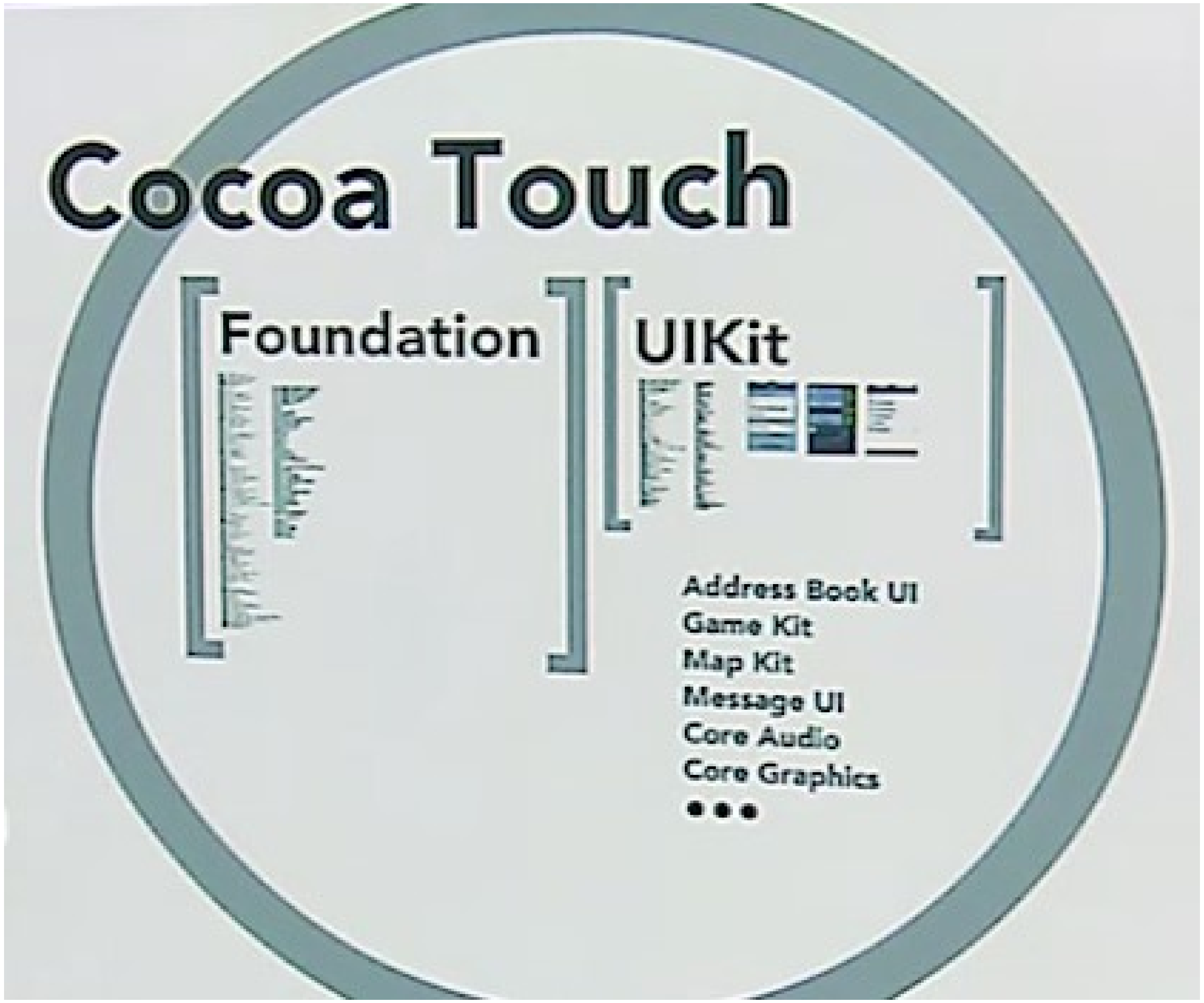
# Modularizando aplicaciones web: Patrón de diseño MVC

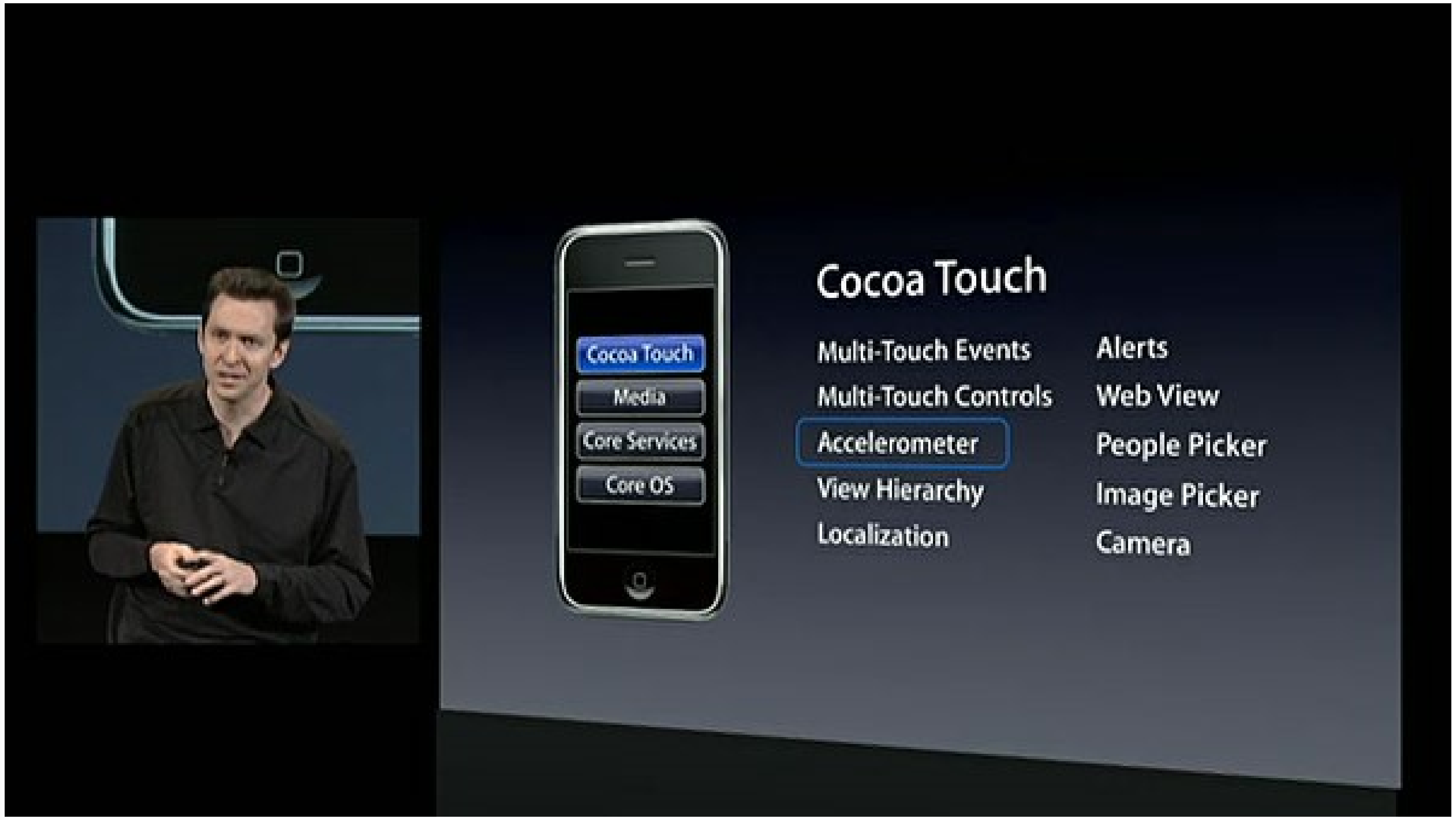


# Cocoa Touch

Foundation

UIKit

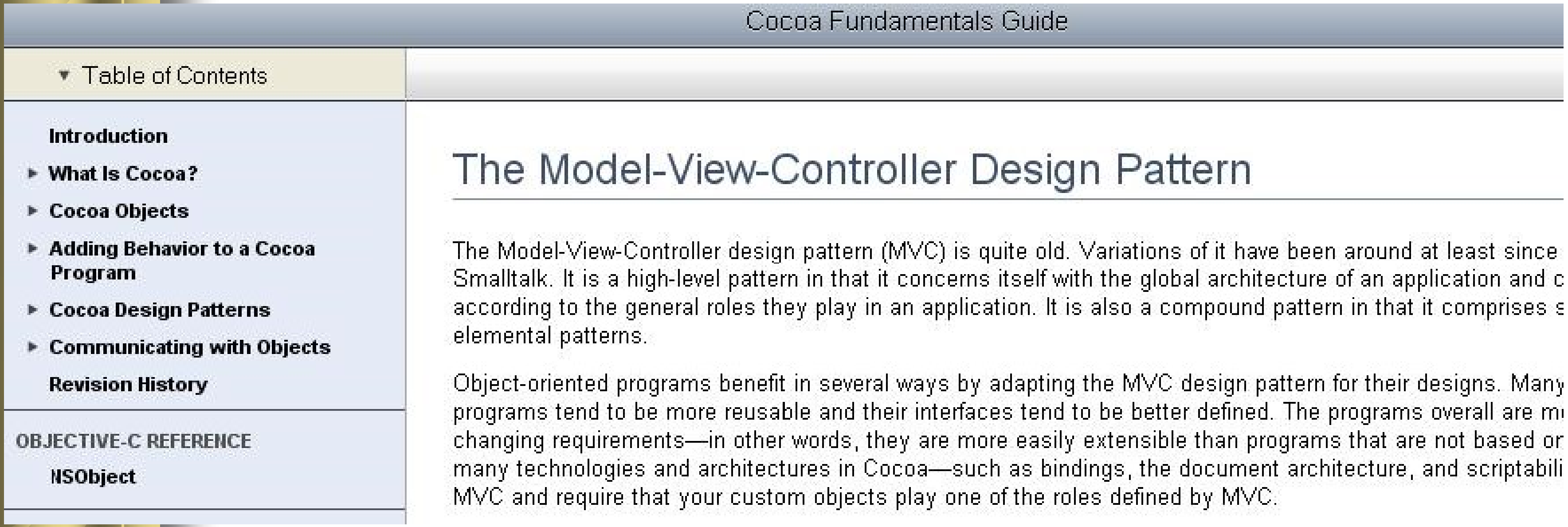
Address Book UI
Game Kit
Map Kit
Message UI
Core Audio
Core Graphics
•••



Cocoa Touch

Multi-Touch Events
Multi-Touch Controls
Accelerometer
View Hierarchy
Localization

Alerts
Web View
People Picker
Image Picker
Camera


Cocoa Fundamentals Guide

▼ Table of Contents

## The Model-View-Controller Design Pattern

The Model-View-Controller design pattern (MVC) is quite old. Variations of it have been around at least since Smalltalk. It is a high-level pattern in that it concerns itself with the global architecture of an application and c according to the general roles they play in an application. It is also a compound pattern in that it comprises s elemental patterns.

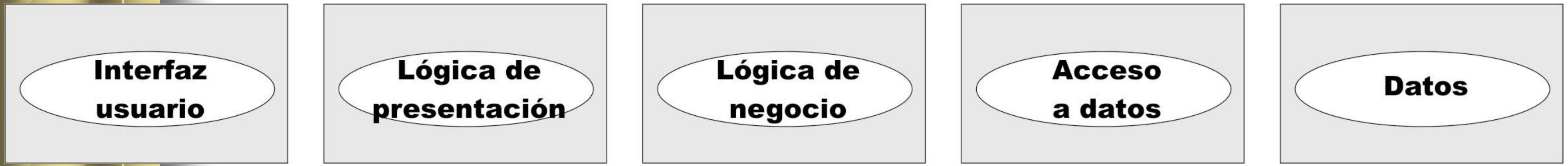
Object-oriented programs benefit in several ways by adapting the MVC design pattern for their designs. Many programs tend to be more reusable and their interfaces tend to be better defined. The programs overall are m changing requirements—in other words, they are more easily extensible than programs that are not based or many technologies and architectures in Cocoa—such as bindings, the document architecture, and scriptabili MVC and require that your custom objects play one of the roles defined by MVC.

## MVC in AngularJS

AngularJS designs the applications in MVC style. MVC is an important concept of this technology, hence it is imperative to have familiarity with it. MVC stands for Model View Controller. Let's find out what is it: Model - A...

## **Problema:** ¿Implementación de aplicaciones web sin diseño?

## Componentes de una aplicación web



## Roles de los Servlet/JSP

- Interfaz de presentación
- Lógica de presentación
- Lógica de negocio
- Acceso a datos

## ¿Problemas?

- Implementación con Servlets: Código java y marcas HTML en servlets
- Implementación con JSPs: Marcas HTML y scriptlets en JSP
- Con qué tecnología implementamos cada componente?
- Como distribuimos cada componente en Clases?

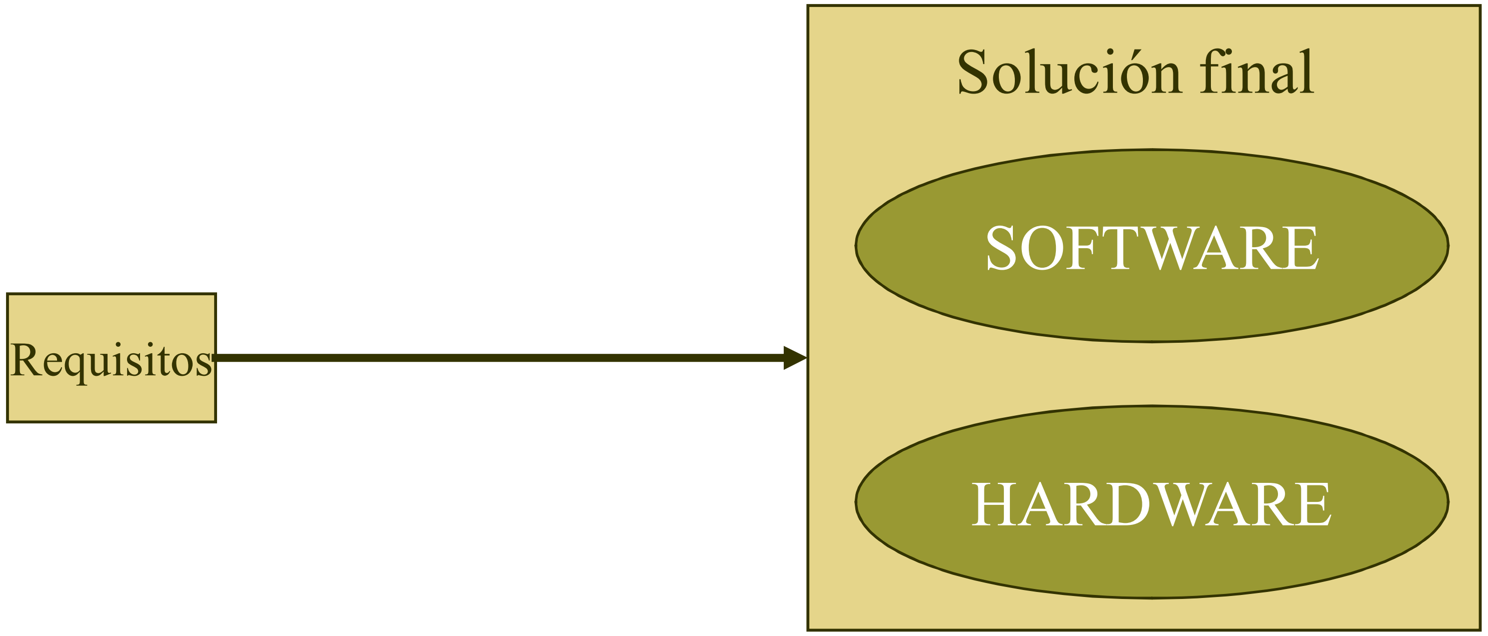
## Solución: diseño de aplicaciones web

- **Introducción de una fase de diseño** en el proceso de creación de una aplicación web

- Implementación: utilización de tecnologías web

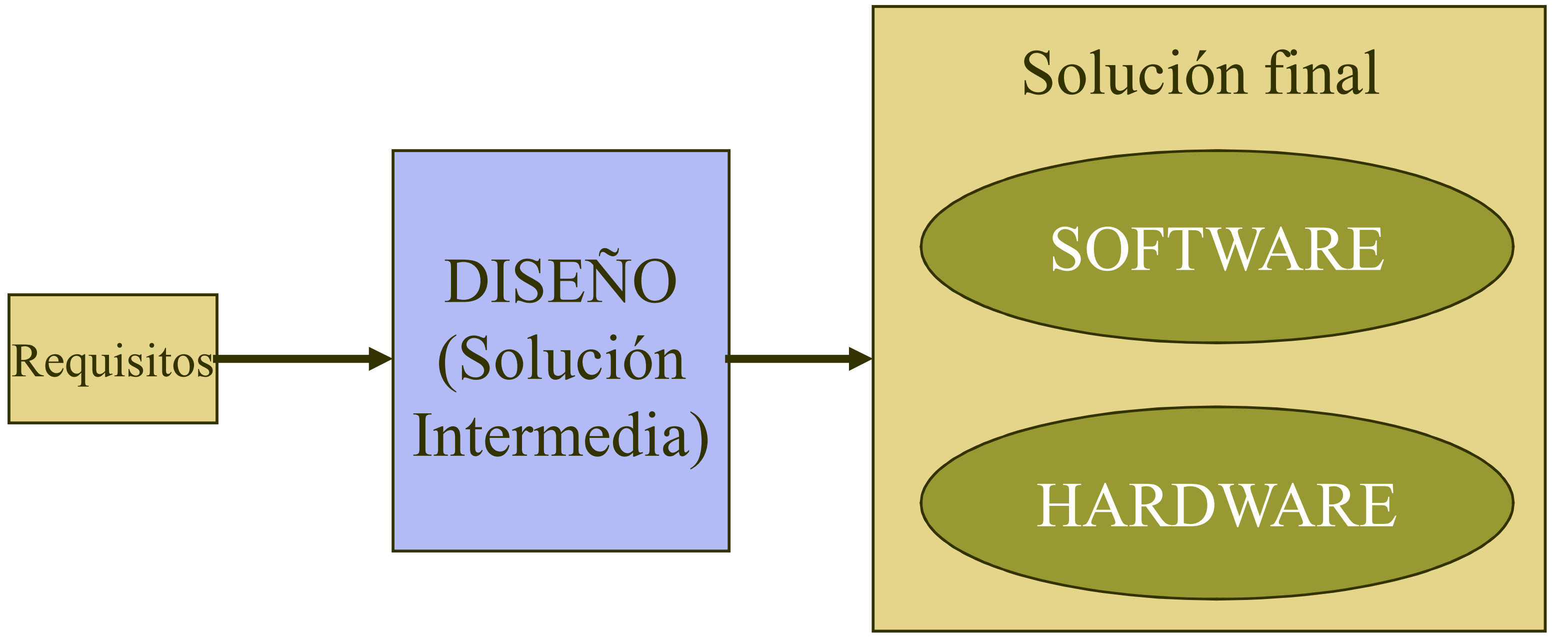
## ¿Por qué la fase de diseño?

- Opción inicial: analizar e implementar
- Problema: Muchos temas a resolver en un sólo paso.



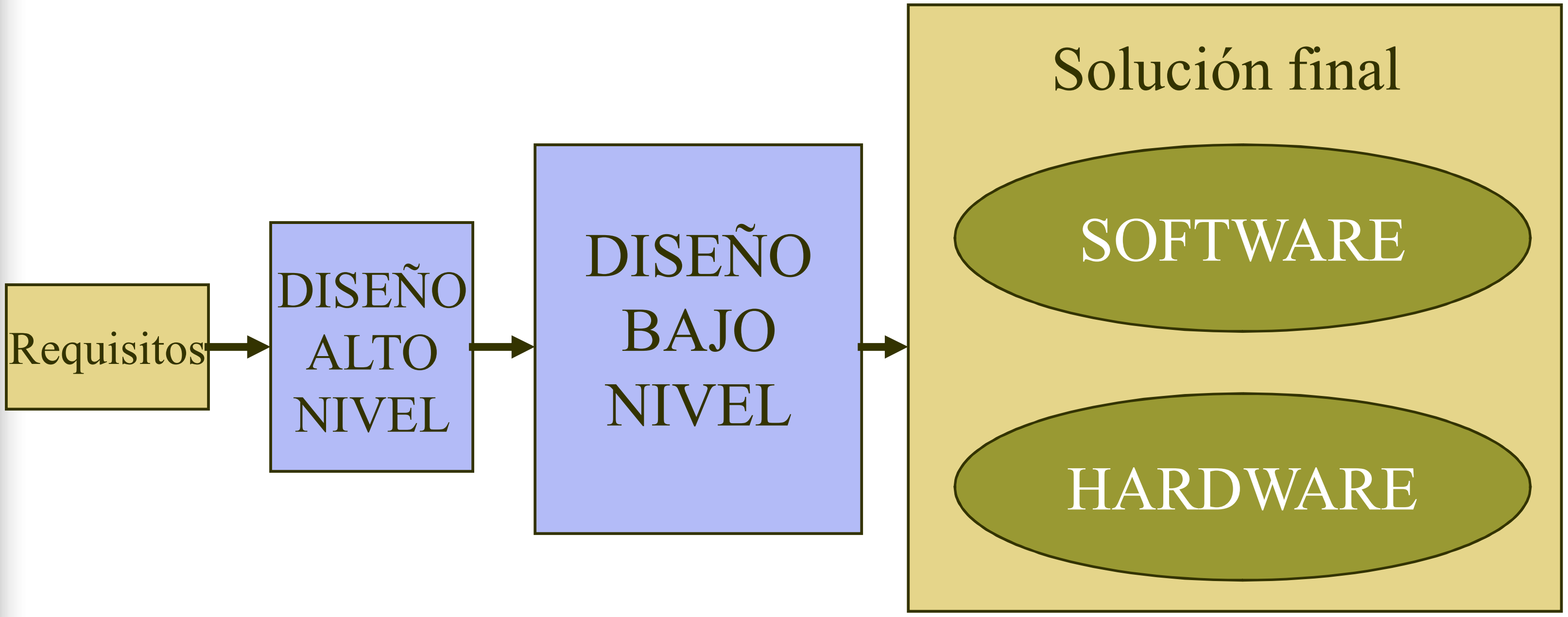
## ¿Por qué la fase de diseño?

- Solución: divide y vencerás



## ¿Por qué la fase de diseño?

- Y podemos dividir más



## Ventajas de incorporar una fase de diseño

- Permite analizar el problema y  pensar cómo se va a solucionar.
- Permite compartir y discutir la solución con el equipo de trabajo
- Permite explicar la solución a los clientes
- Reduce la complejidad del salto de los requisitos a la implementación.

## Ventajas de incorporar una fase de diseño

- Permite probar/chequear una idea antes de construirla
- Permite comprobar que la idea del sistema satisface a los clientes
- Ahorra costes: es más barato hacer dibujos que rehacer el código

## ¿Cómo se realiza el diseño?

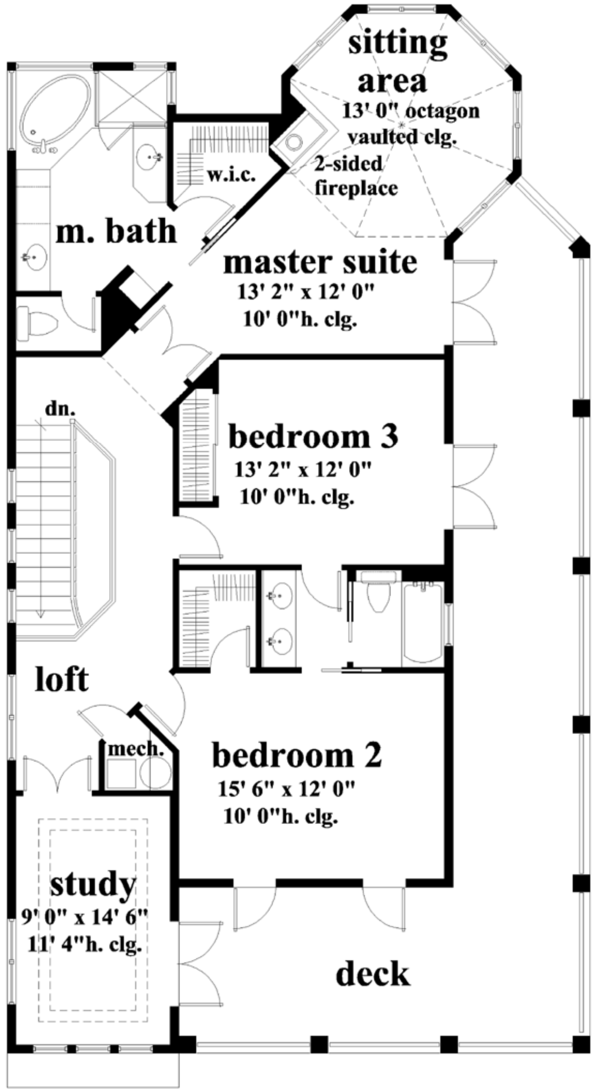
- Creando MODELOS del sistema a construir
- Veamos algunos ejemplos

## Analogía: arquitectura

- ¿Es esto una casa?

## Analogía: arquitectura

- ¿Es esto una casa?



## ¿Qué son los modelos?

- Son *abstracciones* de la realidad que nos facilitan entender y poder predecir la realidad.

- Modelos:

  + aspectos *fundamentales* del problema

  - aspectos *irrelevantes* del problema

- Los modelos pueden ser matemáticos, simbólicos, gráficos, etc.

## ¿Cómo construimos modelos?: patrones de diseño

- Un patrón representa el diseño de una parte de un sistema.
- Basado en la experiencia de diseños previos.
- Suelen ser buenas soluciones a problemas concretos.
- Implica no tener que empezar de cero en la fase de diseño

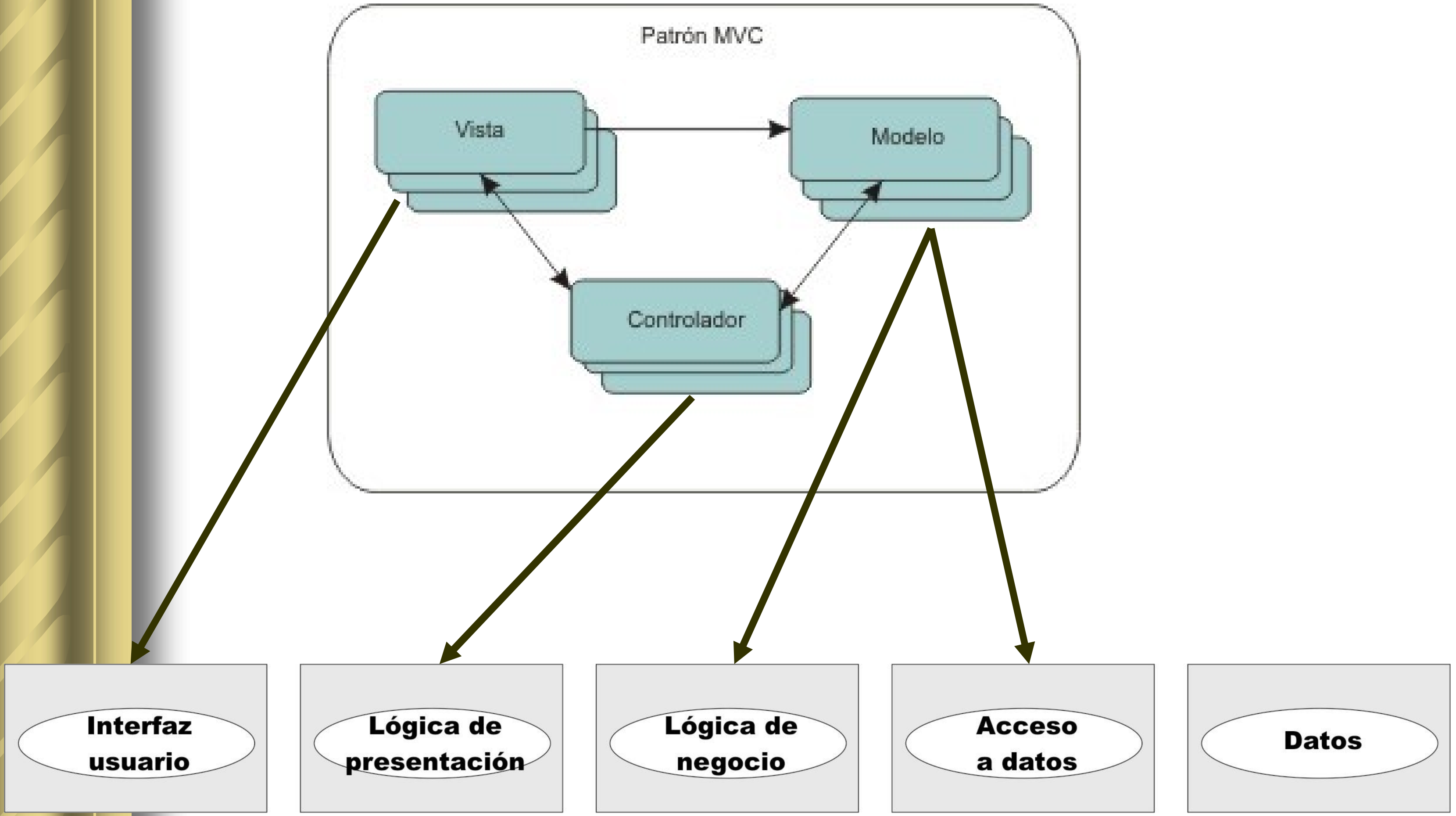
## Nuestros “planos”: Diagramas UML

- Diagrama de actividades
- Diagrama de clases
- Diagrama de secuencia
- Diagrama de paquetes

**Problema:**

¿Cuáles son los patrones de diseño más relevantes para una aplicación web?

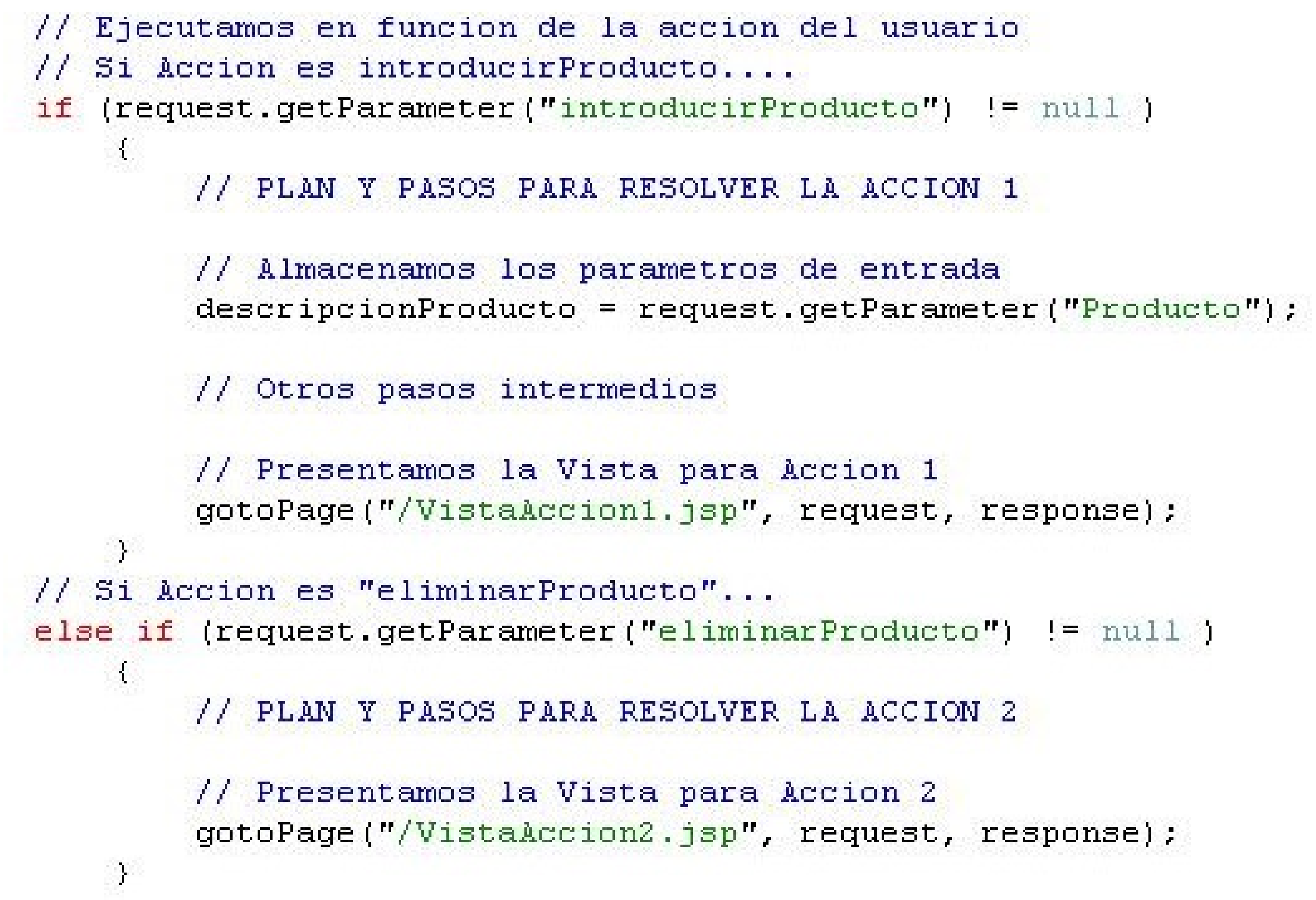
## Patrón MVC: el patrón estructural



## Implementación MVC2

- ¿Vista? -> JSP, EL y JSTL
- ¿Controlador? -> Servlets
- ¿Modelo? -> Java Beans y clases Java

# Organización del Controlador: Acciones y Planes



```
// Ejecutamos en funcion de la accion del usuario
// Si Accion es introducirProducto....
if (request.getParameter("introducirProducto") != null )
   {
     // PLAN Y PASOS PARA RESOLVER LA ACCION 1

     // Almacenamos los parametros de entrada
     descripcionProducto = request.getParameter("Producto");

     // Otros pasos intermedios

     // Presentamos la Vista para Accion 1
     gotoPage("/VistaAccion1.jsp", request, response);
   }
// Si Accion es "eliminarProducto"...
else if (request.getParameter("eliminarProducto") != null )
   {
     // PLAN Y PASOS PARA RESOLVER LA ACCION 2

     // Presentamos la Vista para Accion 2
     gotoPage("/VistaAccion2.jsp", request, response);
   }
```

## **Problema:** ¿Controlador muy complejo?

## Patrón "Service to Worker"

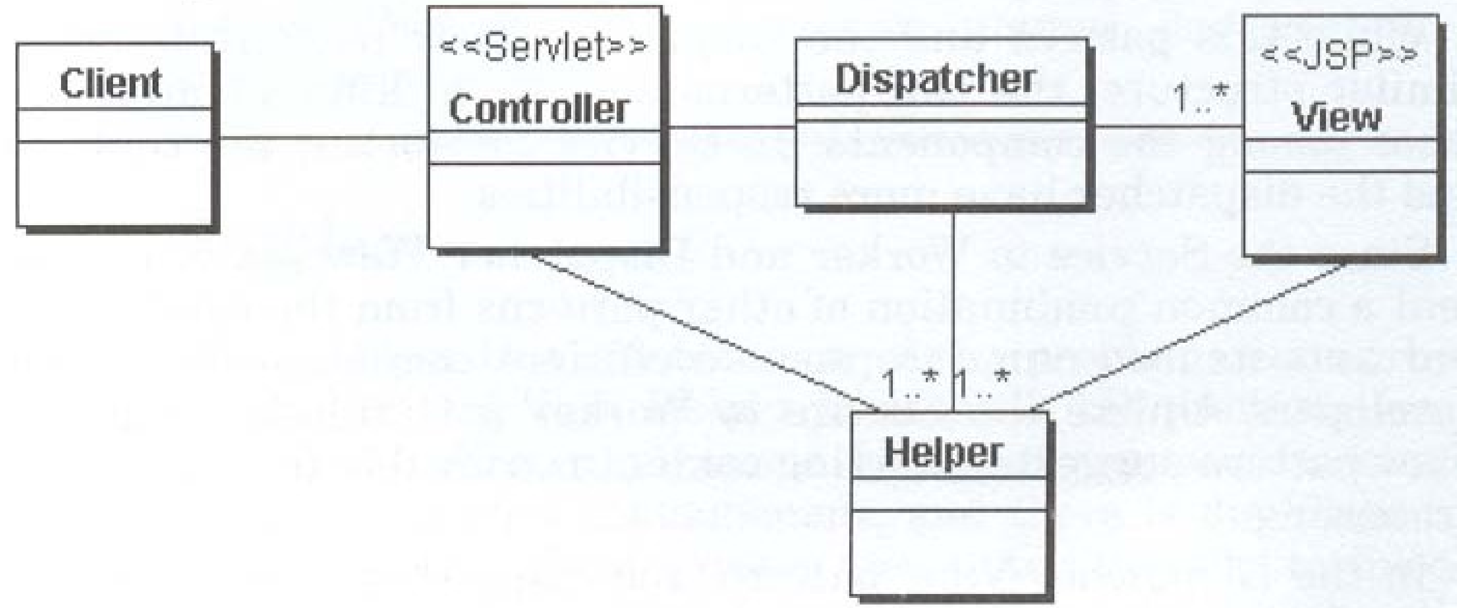
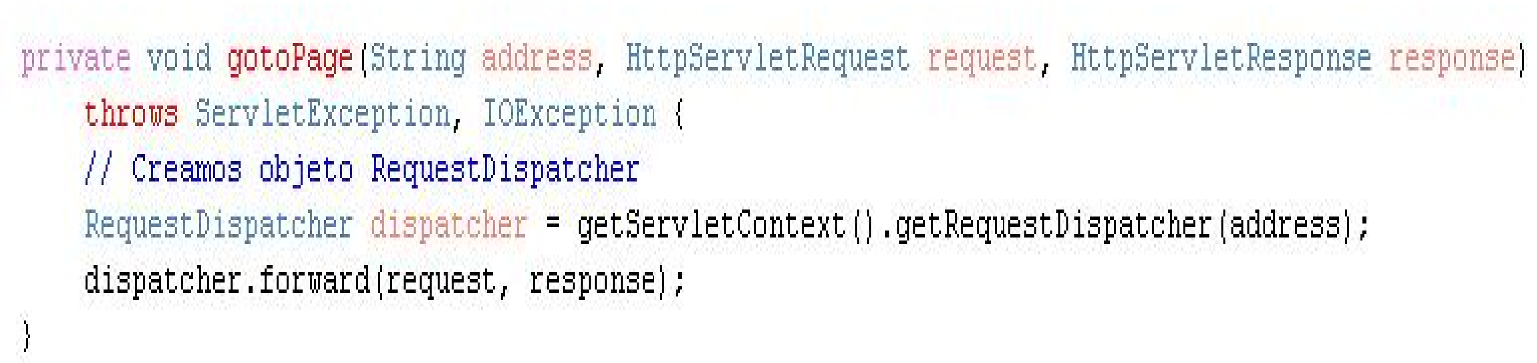


***Figure 7.20*** Service to Worker class diagram

## Clases del Patrón: Helper y Dispatcher

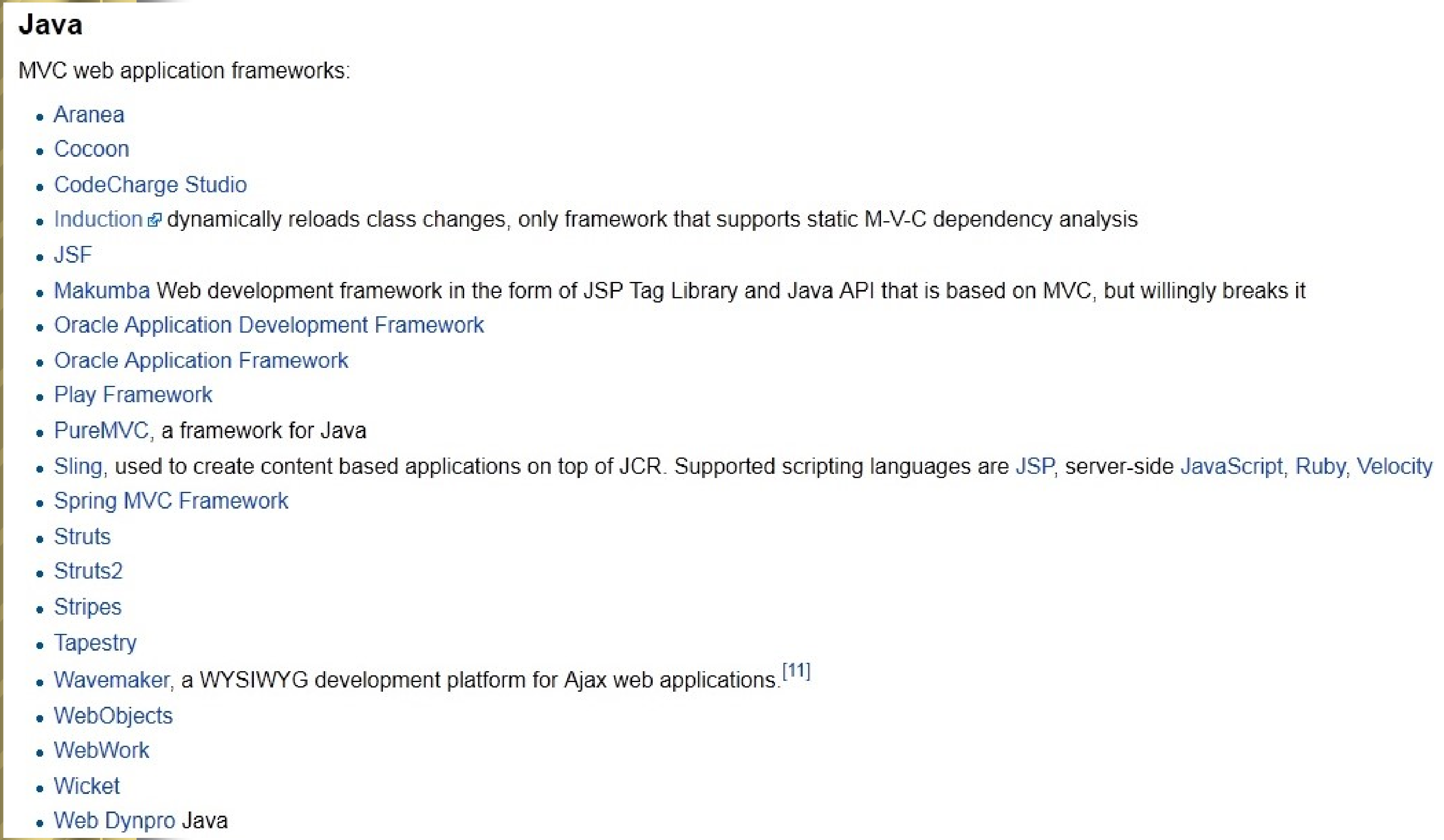
1. Helper (= Asistente): Clase para ejecutar los planes asociados a las acciones requeridas por el controlador

2. Dispatcher: Clase con una función que reenvía la petición y la respuesta a la vista correspondiente



```
private void gotoPage(String address, HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Creamos objeto RequestDispatcher
    RequestDispatcher dispatcher = getServletContext().getRequestDispatcher(address);
    dispatcher.forward(request, response);
}
```

# Frameworks MVC

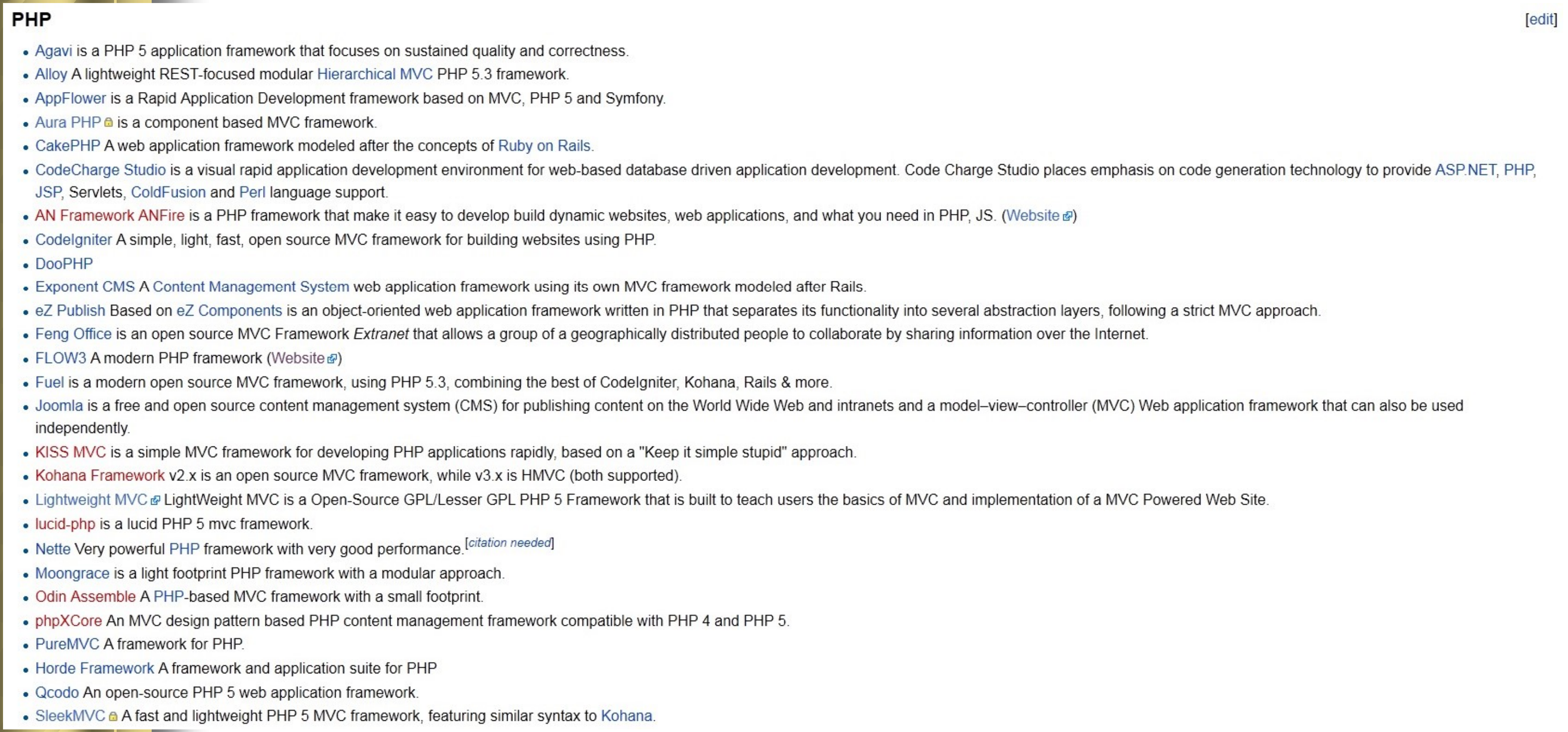


**Java**

MVC web application frameworks:

- Aranea
- Cocoon
- CodeCharge Studio
- Induction dynamically reloads class changes, only framework that supports static M-V-C dependency analysis
- JSF
- Makumba Web development framework in the form of JSP Tag Library and Java API that is based on MVC, but willingly breaks it
- Oracle Application Development Framework
- Oracle Application Framework
- Play Framework
- PureMVC, a framework for Java
- Sling, used to create content based applications on top of JCR. Supported scripting languages are JSP, server-side JavaScript, Ruby, Velocity
- Spring MVC Framework
- Struts
- Struts2
- Stripes
- Tapestry
- Wavemaker, a WYSIWYG development platform for Ajax web applications.[11]
- WebObjects
- WebWork
- Wicket
- Web Dynpro Java

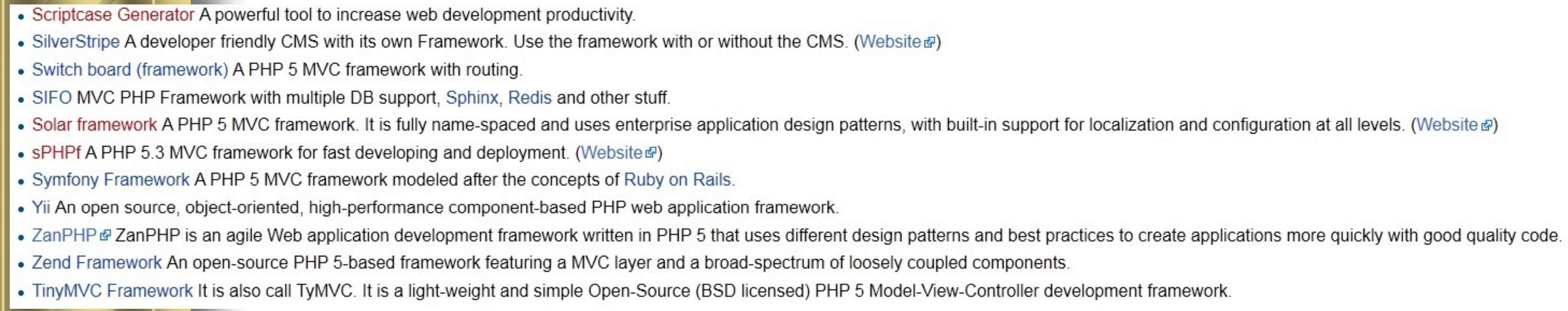
# Frameworks MVC



**PHP** [edit]

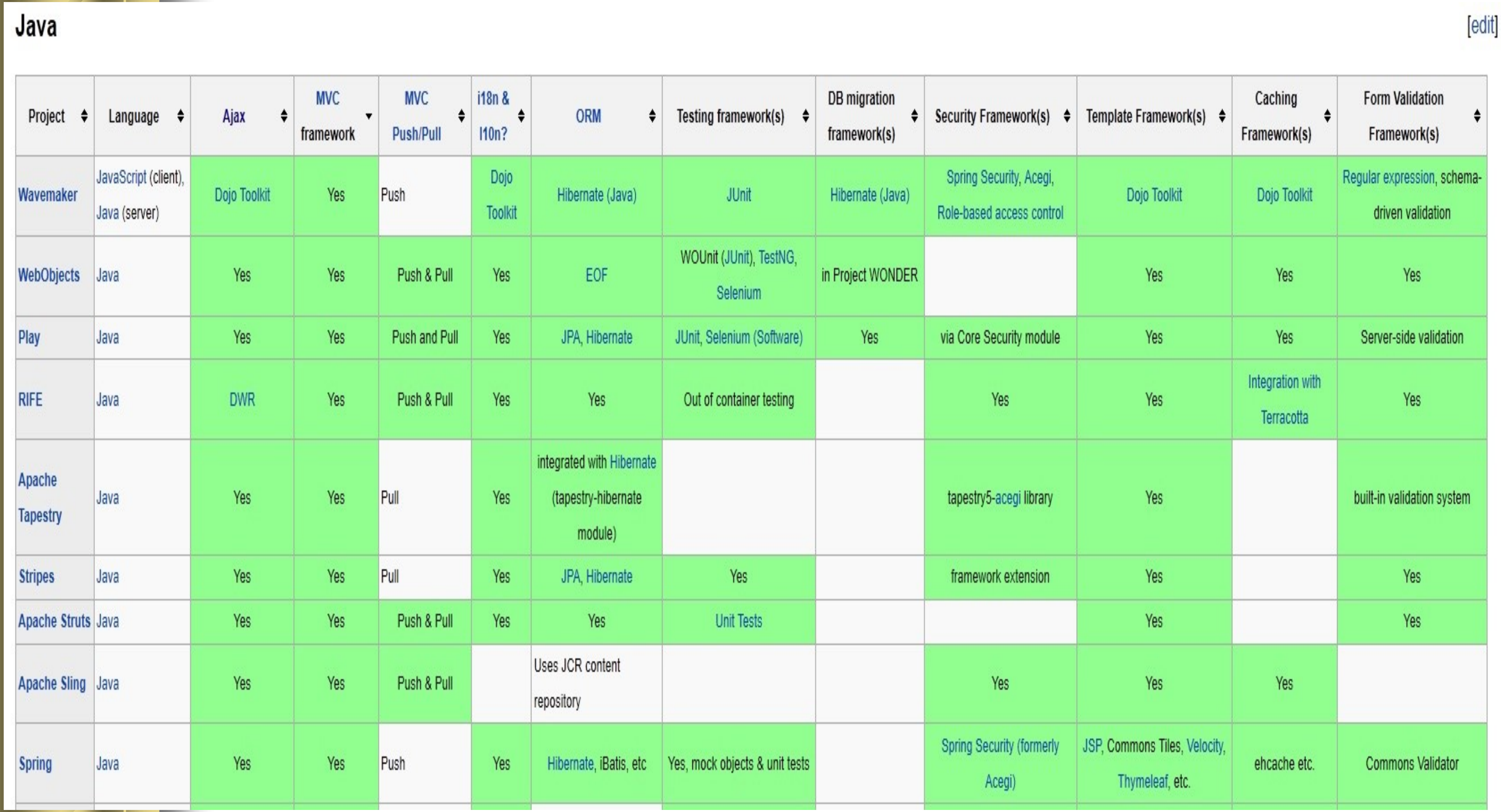
- Agavi is a PHP 5 application framework that focuses on sustained quality and correctness.
- Alloy A lightweight REST-focused modular Hierarchical MVC PHP 5.3 framework.
- AppFlower is a Rapid Application Development framework based on MVC, PHP 5 and Symfony.
- Aura PHP is a component based MVC framework.
- CakePHP A web application framework modeled after the concepts of Ruby on Rails.
- CodeCharge Studio is a visual rapid application development environment for web-based database driven application development. Code Charge Studio places emphasis on code generation technology to provide ASP.NET, PHP, JSP, Servlets, ColdFusion and Perl language support.
- AN Framework ANFire is a PHP framework that make it easy to develop build dynamic websites, web applications, and what you need in PHP, JS. (Website)
- CodeIgniter A simple, light, fast, open source MVC framework for building websites using PHP.
- DooPHP
- Exponent CMS A Content Management System web application framework using its own MVC framework modeled after Rails.
- eZ Publish Based on eZ Components is an object-oriented web application framework written in PHP that separates its functionality into several abstraction layers, following a strict MVC approach.
- Feng Office is an open source MVC Framework *Extranet* that allows a group of a geographically distributed people to collaborate by sharing information over the Internet.
- FLOW3 A modern PHP framework (Website)
- Fuel is a modern open source MVC framework, using PHP 5.3, combining the best of CodeIgniter, Kohana, Rails & more.
- Joomla is a free and open source content management system (CMS) for publishing content on the World Wide Web and intranets and a model–view–controller (MVC) Web application framework that can also be used independently.
- KISS MVC is a simple MVC framework for developing PHP applications rapidly, based on a "Keep it simple stupid" approach.
- Kohana Framework v2.x is an open source MVC framework, while v3.x is HMVC (both supported).
- Lightweight MVC LightWeight MVC is a Open-Source GPL/Lesser GPL PHP 5 Framework that is built to teach users the basics of MVC and implementation of a MVC Powered Web Site.
- lucid-php is a lucid PHP 5 mvc framework.
- Nette Very powerful PHP framework with very good performance.[*citation needed*]
- Moongrace is a light footprint PHP framework with a modular approach.
- Odin Assemble A PHP-based MVC framework with a small footprint.
- phpXCore An MVC design pattern based PHP content management framework compatible with PHP 4 and PHP 5.
- PureMVC A framework for PHP.
- Horde Framework A framework and application suite for PHP
- Qcodo An open-source PHP 5 web application framework.
- SleekMVC A fast and lightweight PHP 5 MVC framework, featuring similar syntax to Kohana.

## Frameworks MVC



- Scriptcase Generator A powerful tool to increase web development productivity.
- SilverStripe A developer friendly CMS with its own Framework. Use the framework with or without the CMS. (Website)
- Switch board (framework) A PHP 5 MVC framework with routing.
- SIFO MVC PHP Framework with multiple DB support, Sphinx, Redis and other stuff.
- Solar framework A PHP 5 MVC framework. It is fully name-spaced and uses enterprise application design patterns, with built-in support for localization and configuration at all levels. (Website)
- sPHPf A PHP 5.3 MVC framework for fast developing and deployment. (Website)
- Symfony Framework A PHP 5 MVC framework modeled after the concepts of Ruby on Rails.
- Yii An open source, object-oriented, high-performance component-based PHP web application framework.
- ZanPHP ZanPHP is an agile Web application development framework written in PHP 5 that uses different design patterns and best practices to create applications more quickly with good quality code.
- Zend Framework An open-source PHP 5-based framework featuring a MVC layer and a broad-spectrum of loosely coupled components.
- TinyMVC Framework It is also call TyMVC. It is a light-weight and simple Open-Source (BSD licensed) PHP 5 Model-View-Controller development framework.

# Frameworks MVC



## Java

[edit]

| Project | Language | Ajax | MVC framework | MVC Push/Pull | i18n & l10n? | ORM | Testing framework(s) | DB migration framework(s) | Security Framework(s) | Template Framework(s) | Caching Framework(s) | Form Validation Framework(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Wavemaker | JavaScript (client), Java (server) | Dojo Toolkit | Yes | Push | Dojo Toolkit | Hibernate (Java) | JUnit | Hibernate (Java) | Spring Security, Acegi, Role-based access control | Dojo Toolkit | Dojo Toolkit | Regular expression, schema-driven validation |
| WebObjects | Java | Yes | Yes | Push & Pull | Yes | EOF | WOUnit (JUnit), TestNG, Selenium | in Project WONDER | | Yes | Yes | Yes |
| Play | Java | Yes | Yes | Push and Pull | Yes | JPA, Hibernate | JUnit, Selenium (Software) | Yes | via Core Security module | Yes | Yes | Server-side validation |
| RIFE | Java | DWR | Yes | Push & Pull | Yes | Yes | Out of container testing | | Yes | Yes | Integration with Terracotta | Yes |
| Apache Tapestry | Java | Yes | Yes | Pull | Yes | integrated with Hibernate (tapestry-hibernate module) | | | tapestry5-acegi library | Yes | | built-in validation system |
| Stripes | Java | Yes | Yes | Pull | Yes | JPA, Hibernate | Yes | | framework extension | Yes | | Yes |
| Apache Struts | Java | Yes | Yes | Push & Pull | Yes | Yes | Unit Tests | | | Yes | | Yes |
| Apache Sling | Java | Yes | Yes | Push & Pull | | Uses JCR content repository | | | Yes | Yes | Yes | |
| Spring | Java | Yes | Yes | Push | Yes | Hibernate, iBatis, etc | Yes, mock objects & unit tests | | Spring Security (formerly Acegi) | JSP, Commons Tiles, Velocity, Thymeleaf, etc. | ehcache etc. | Commons Validator |

## Ejercicio:

Diseño MVC para una aplicación
de gestión de una agencia de viajes