

Un framework para el desarrollo de aplicaciones complejas basadas en Web

**Alejandro Siri
Mariano Montone**

Contenido

Introducción

Los problemas de las Aplicaciones Web

Estado del Arte

Ejemplo motivador

Diseño del Framework

Demostración

Implementación en PHP

Conclusiones

Trabajo Futuro

Contenido

Introducción

Los problemas de las Aplicaciones Web

Estado del Arte

Ejemplo motivador

Diseño del Framework

Demostración

Implementación en PHP

Conclusiones

Trabajo Futuro



Introducción

El desarrollador web enfrenta:

- Problemas clásicos:
 - Persistencia, Concurrencia, Interacción
- Nuevos Problemas:
 - Navegación, Arquitectura cliente-servidor, manejo de estado, AJAX, HTML, CSS, Javascript
- Existen Frameworks y librerías, pero son incompletos.



Hipótesis

¿Es posible desarrollar un Framework que resuelva todos estos problemas de forma integrada?



Desarrollo de Aplicaciones Web

Estándar de facto: MVC

Requerimientos en todas las capas:

- Modelo: Persistencia, Consultas, Transacciones, Concurrencia
- Controlador: Manejo de estado, Transacciones, Reuso
- Vista: Forma de interacción, Integración de diseño y estilos (templates), Reuso

Estado del Arte

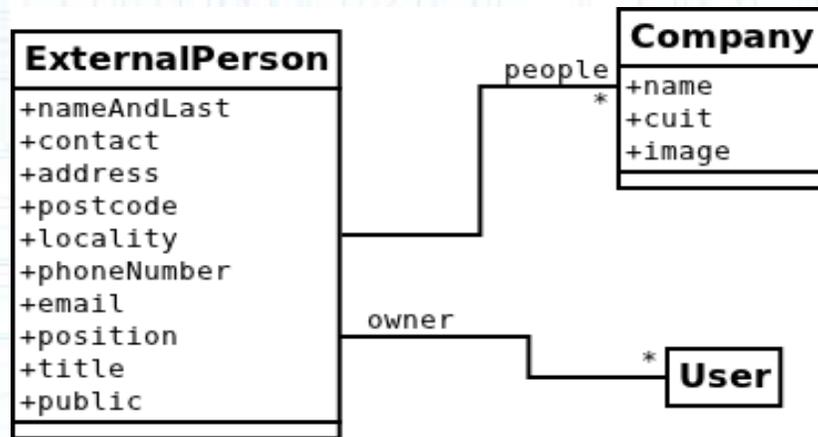


Estado del Arte - Comparación

		 	 	 	 
M	Persistencia (mapeo)	●	●	●	●
	Transacciones	●	●	●	●
	Consultas	●	●	●	●
	Concurrencia	●	●	●	●
C	Reuso	●	●	●	●
	Transacciones Largas	●	●	●	●
	Manejo de Estado	●	●	●	●
V	Integración de diseño	●	●	●	●
	Forma de interacción (Ajax)	●	●	●	●
	Reuso	●	●	●	●

Ejemplo motivador

Ejemplo: Agenda de Contactos



Un usuario puede agregar contactos

Los contactos son públicos o privados

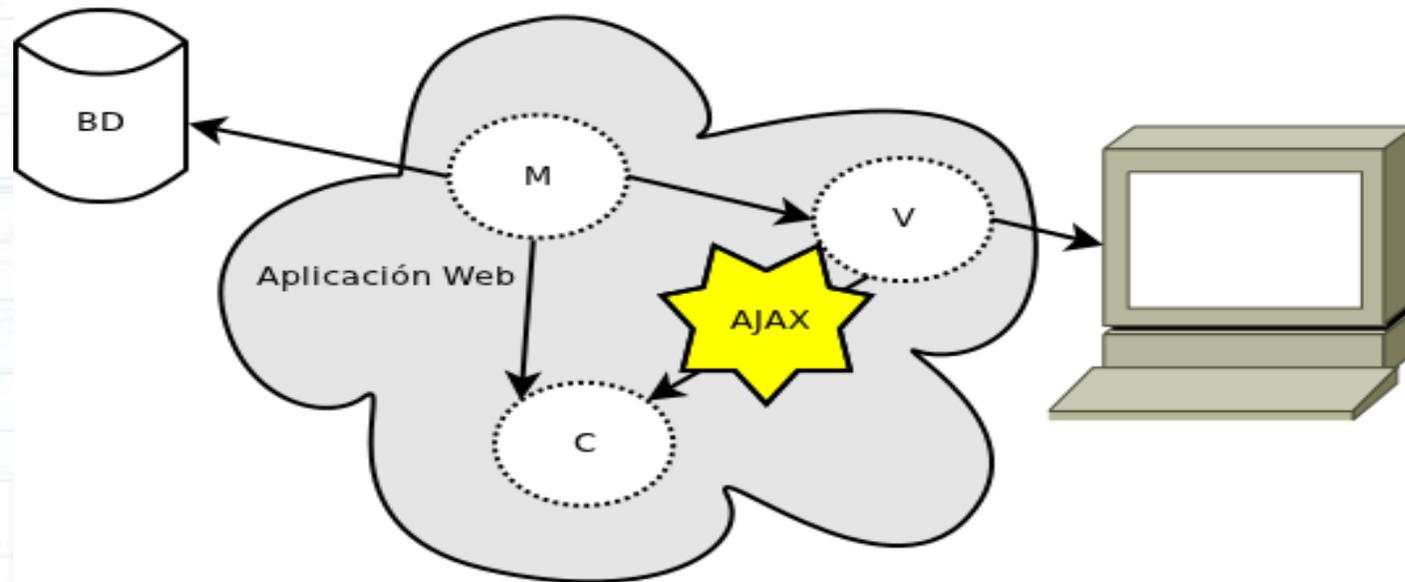
Un usuario puede ver sus contactos y los públicos

Los contactos son de empresas

Ejemplo en otros Frameworks

Caso: Chequear si el email está duplicado.

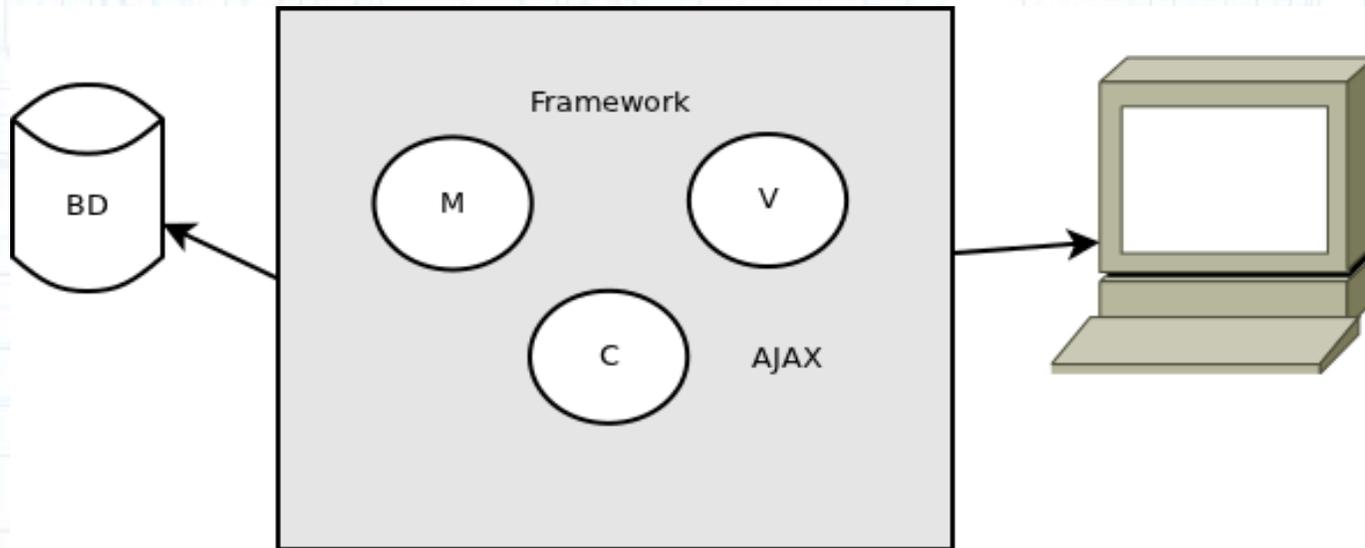
En los FWs: **Código JS en la vista**



Rompe la abstracción

Problema: Integración

Esta integración entre V y C es necesaria.
O lo hace el FW, o le queda al desarrollador.



**El FW debe proveer y
mantener las abstracciones**

Contenido

Introducción

Los problemas de las Aplicaciones Web

Estado del Arte

Ejemplo motivador

Diseño del Framework

Demostración

Implementación en PHP

Conclusiones

Trabajo Futuro

Diseño del framework



Un Framework distinto

Principio de diseño:

- Mantener las abstracciones
 - M,V,C
 - Manejar todas las interacciones
 - Restricciones al desarrollador

Además:

- Introducir nuevas abstracciones
 - Ejemplo: Transacciones largas, transacciones anidadas
- Facilitar el Reuso



Enfoque Metodológico

Composicionalidad

Ejemplos

- Composición de Funciones
- Garbage Collector

Beneficios

- Reuso
- Divide and Conquer

Programación Declarativa

Ejemplo

- Menú “Cortar” de editor de textos

Beneficios

- Hincapié en el qué y no en el cómo



Diseño del Framework – Modelo

Persistencia: Actualización de la estructura de la base de datos, Administración de Objetos, Persistencia por alcance y Garbage collection

Consultas: Consulta con OQL.

Transacciones: Integridad de los datos en Memoria, Observabilidad de los cambios

Concurrencia: Versionamiento



Diseño del Framework – Controlador

Reuso: Componentes

Navegación: Componentes en forma de Árbol.

Manejo de estado: Componentes Stateful

Transacciones: Transacciones Largas.



Diseño del Framework – Vista

Reuso: Templates por componentes, Mecanismo de asignación por tipo (herencia)

Integración de diseño y estilos: Extensión de XML

Forma de interacción: Templates declarativos, foma de interacción intercambiable entre Standard, Ajax, COMET.

Demostración

Demostración de una aplicación simple: Agenda de Contactos.

1 - Modelo	1:46
2 - Administrador	1:24
3 – Componentes y plantillas	1:34
4 – Componente de edición	1:37
5 – Transacciones anidadas	1:10
6 - Ajax	1:10
7 – Actualización de vista	1:08
8 – Transacciones largas	1:05

Contenido

Introducción

Los problemas de las Aplicaciones Web

Estado del Arte

Ejemplo motivador

Diseño del Framework

Demostración

Implementación en PHP

Conclusiones

Trabajo Futuro



Implementación en PHP

- Macros y Compilación
- Multiple dispatching
- DSL (OQL)
- Referencias Débiles
- Mixins
- COMET
- Variables Dinámicas



Conclusiones

- Desarrollamos un framework que resuelve los problemas de las aplicaciones Web de manera integrada.
- Este Framework es usable y performante.
 - Usado en múltiples proyectos de Eureka Consulting

Conclusión – Cont.

		 	 	 	 	Kusala
M	Persistencia (mapeo)	●	●	●	●	●
	Transacciones	●	●	●	●	●
	Consultas	●	●	●	●	●
	Concurrencia	●	●	●	●	●
C	Reuso	●	●	●	●	●
	Transacciones Largas	●	●	●	●	●
	Manejo de Estado	●	●	●	●	●
V	Integración de diseño	●	●	●	●	●
	Forma de interacción (Ajax)	●	●	●	●	●
	Reuso	●	●	●	●	●



Trabajo a Futuro

- Múltiples Bases de Datos
- Integración con Librerías
- Continuaciones
- Plantillas Avanzadas
- PHP 5.3