

MVC

(Model-View-Controller)

(...o la forma de traer orden al caos...)

(BORRADOR)

Universidad de los Andes

Demián Gutierrez

Diciembre 2010

¿MVC: Modelo – Vista – Controlador?

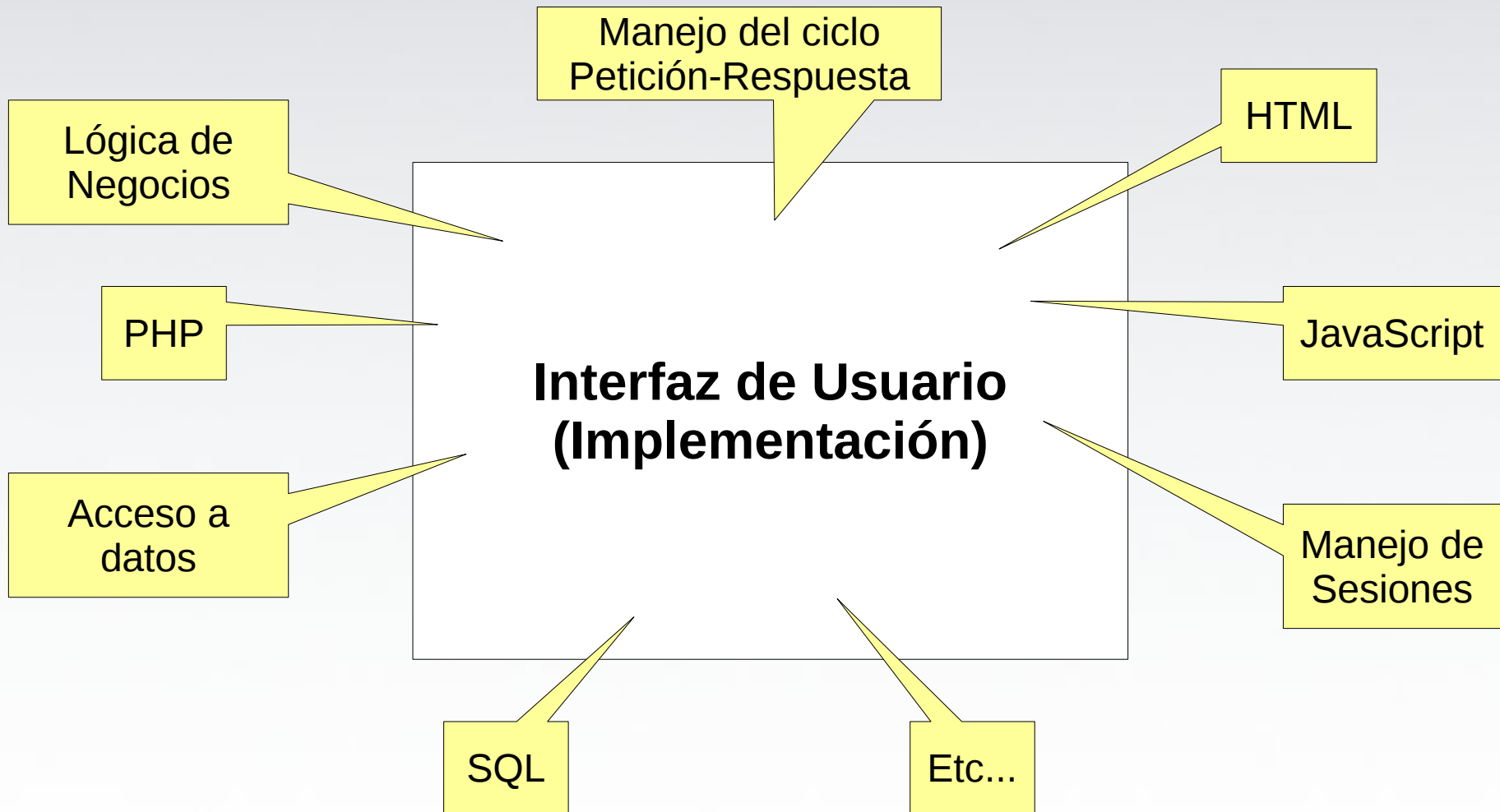
Desarrollado / Ideado originalmente en XEROX PARC (Palo Alto Research Center Incorporated) por Trygve Reenskaug entre 1978 y 1979

Ver: <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>

El objetivo del patrón MVC es desacoplar la presentación de la información (vista) de su representación (modelo), para así reducir la complejidad en el diseño arquitectónico (de IU) e incrementar la flexibilidad y mantenibilidad del código

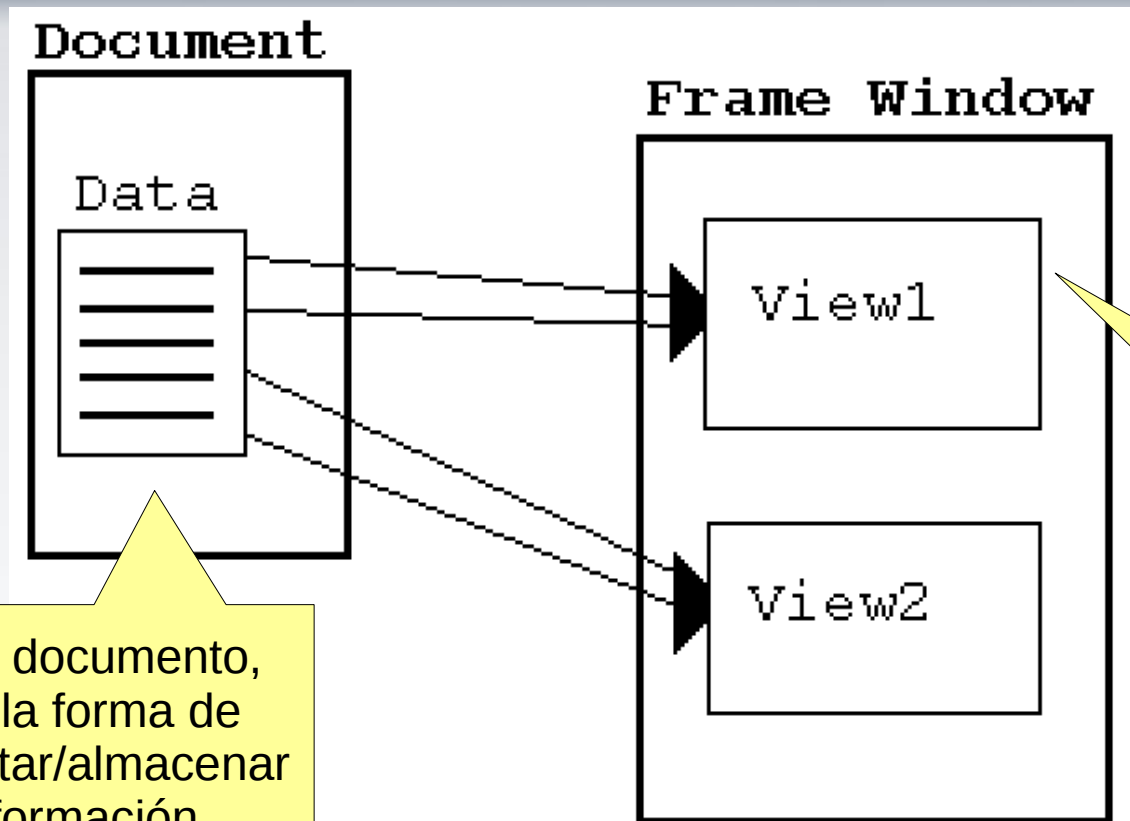
Tomado de: <http://en.wikipedia.org/wiki/Model-View-Controller>

Desde un punto de vista *muy personal* (DMI), el patrón MVC es lo que le trae orden al caos, estructura al espagueti, arquitectura a la anarquía en los aspectos relacionados a la implementación de interfaz de usuario



**Responsabilidades difusas,
todo incrustado en un mismo bloque**

¿Modelo Vista-Documento?



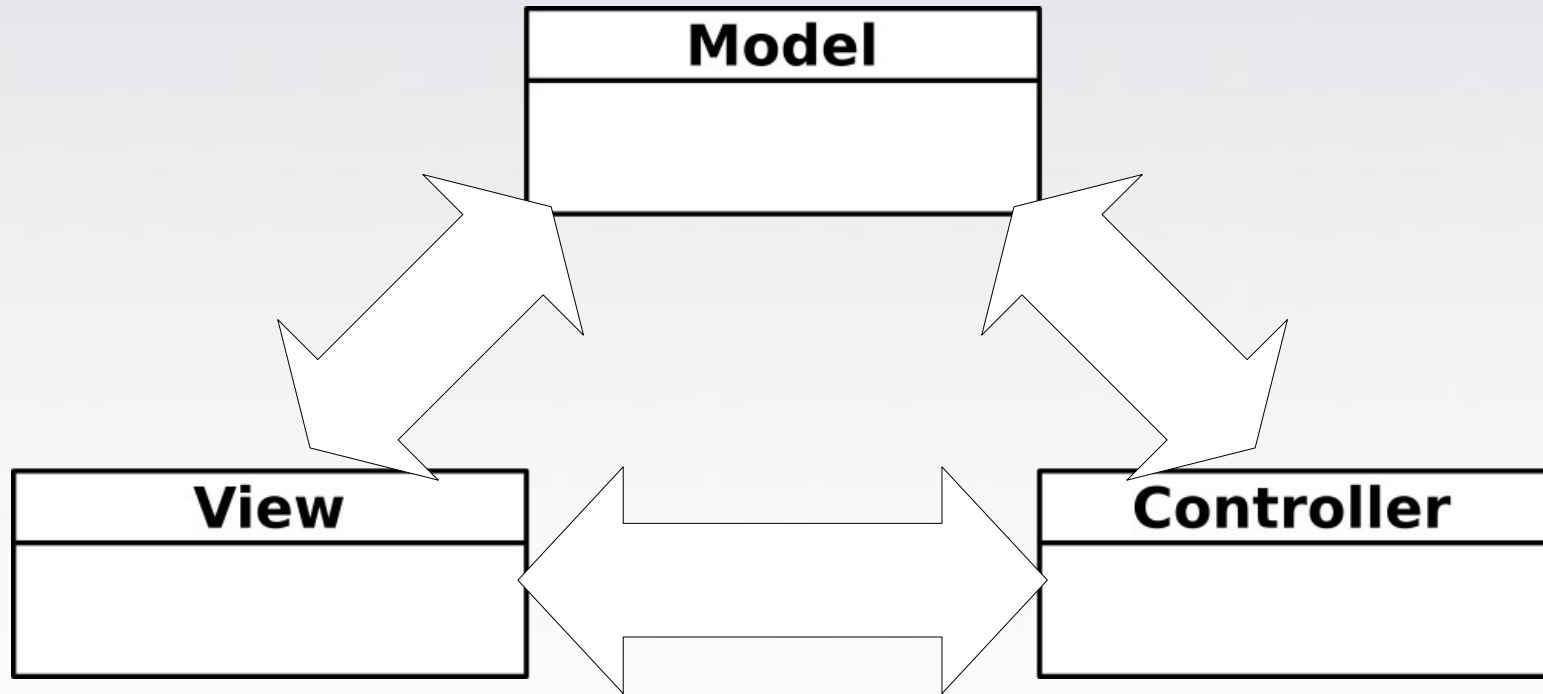
Un sólo documento,
una sola forma de
representar/almacenar
la información

Múltiples vistas,
múltiples formas de
mostrar la misma
información, por
ejemplo, un gráfico,
una hoja de cálculo,
un resumen
estadístico, etcétera

Documents, Views, & Frames

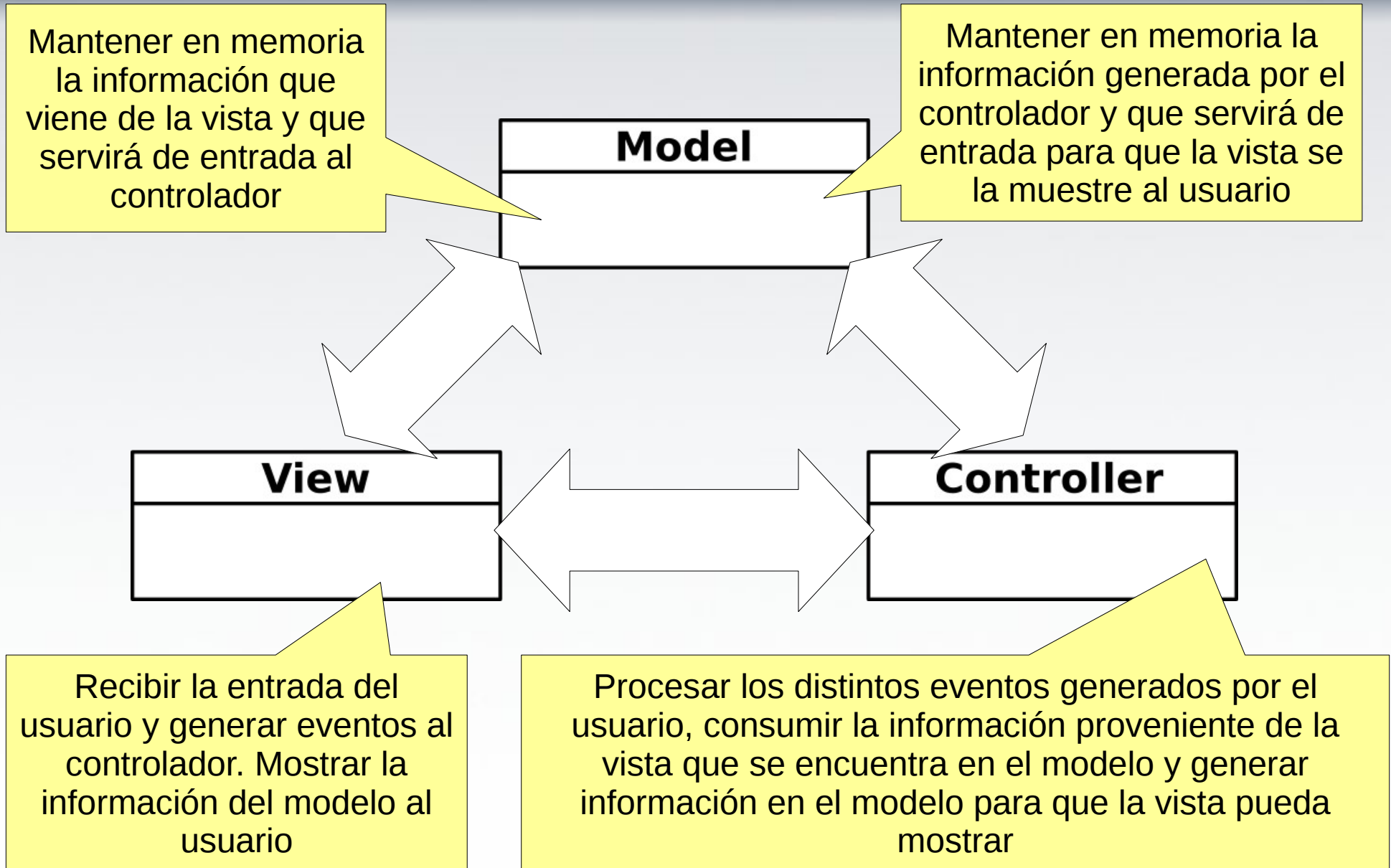
Si, yo se que es un plagio... (sue me if you can)
TODO: Hacer figura propia ;-)

¿MVC: Modelo – Vista – Controlador?

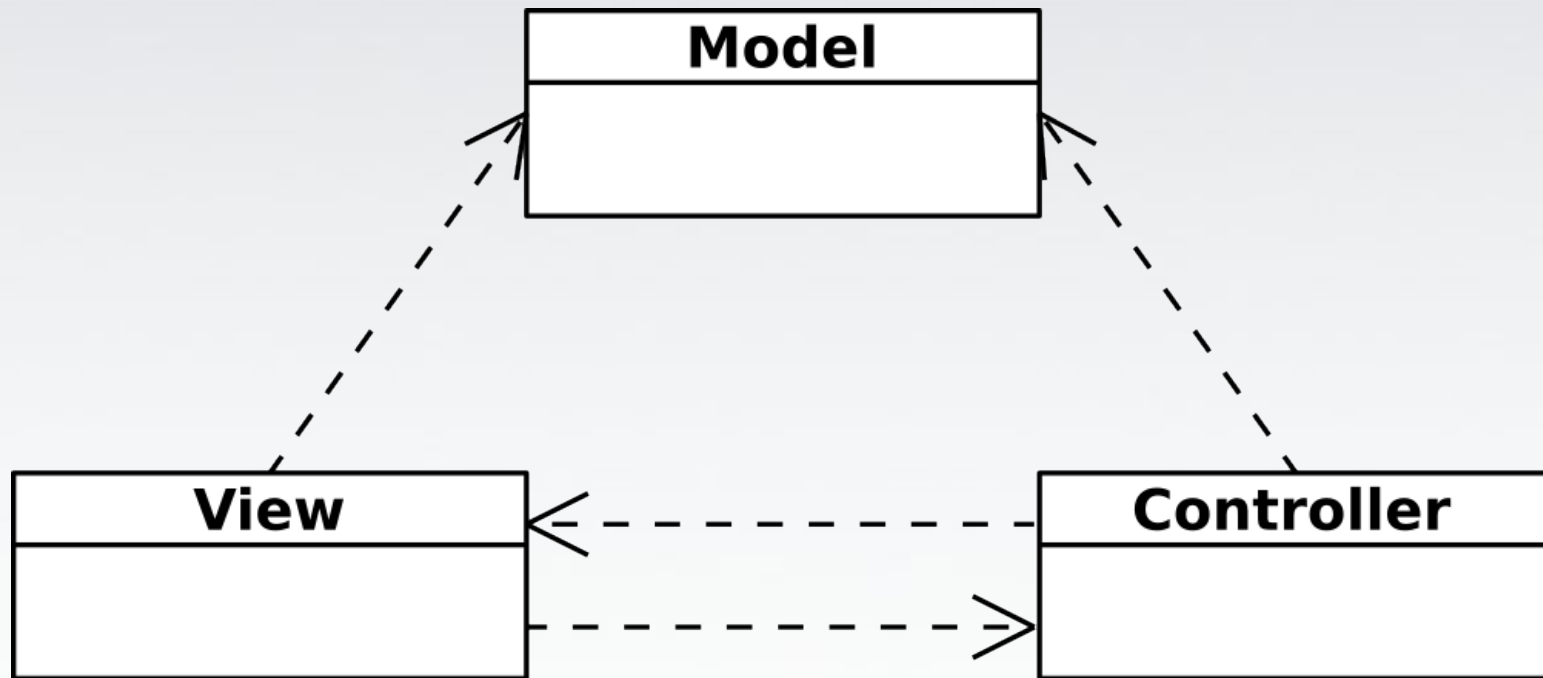


¿Responsabilidades?

¿MVC: Modelo – Vista – Controlador?



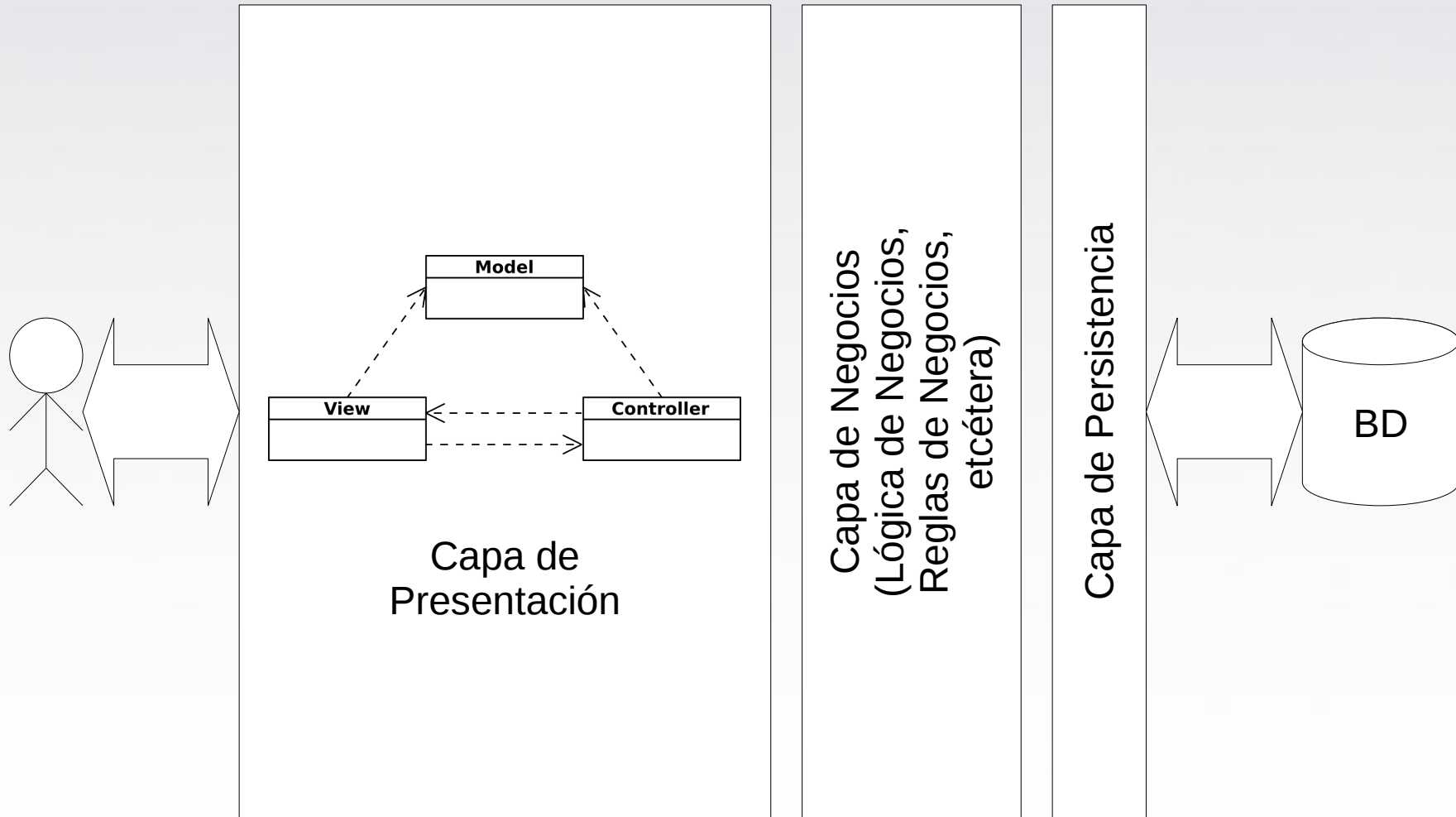
¿MVC: Modelo – Vista – Controlador?



En realidad es más de esta forma...

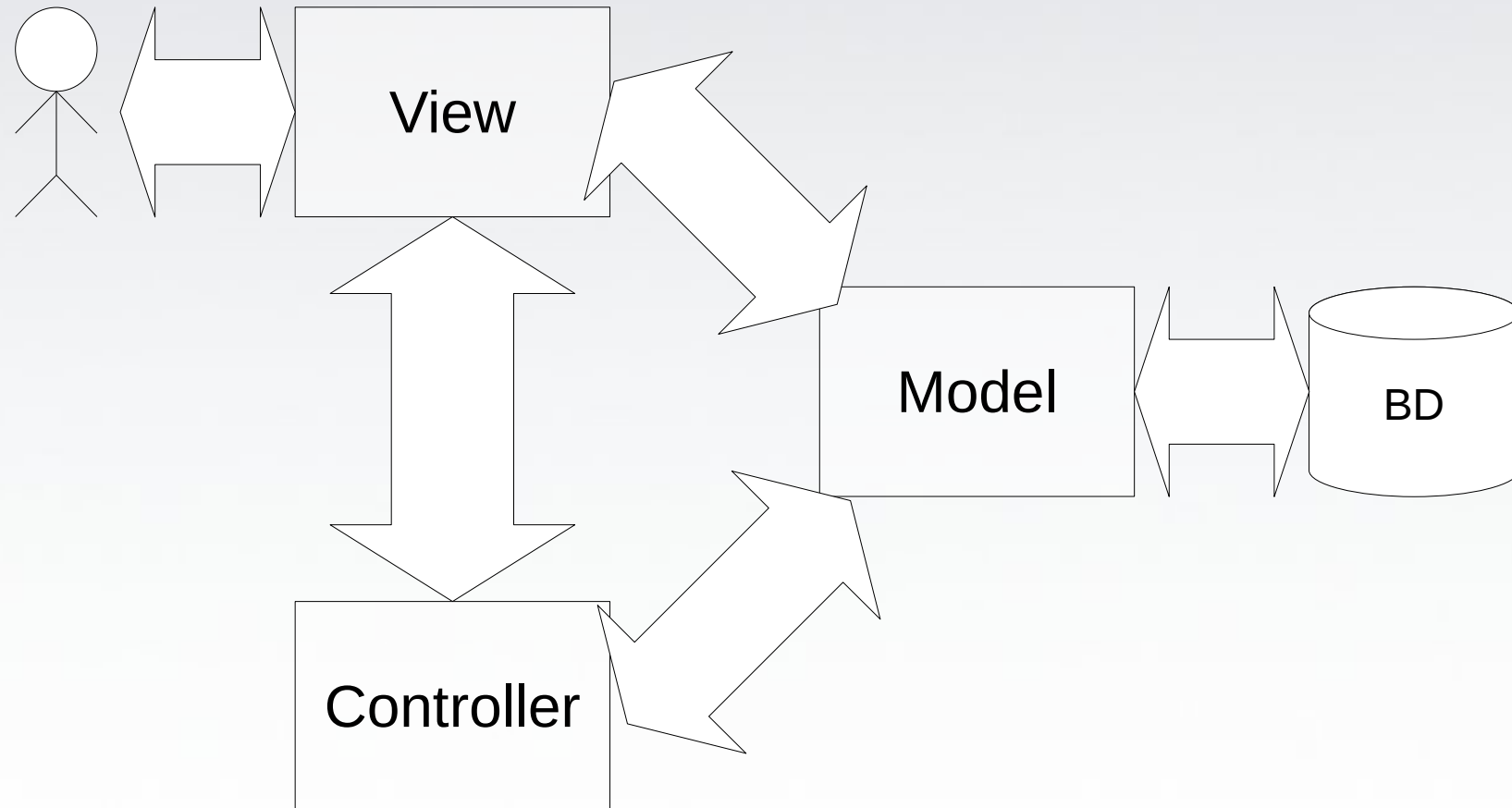
MVC y Arquitecturas / Aplicaciones por Capas

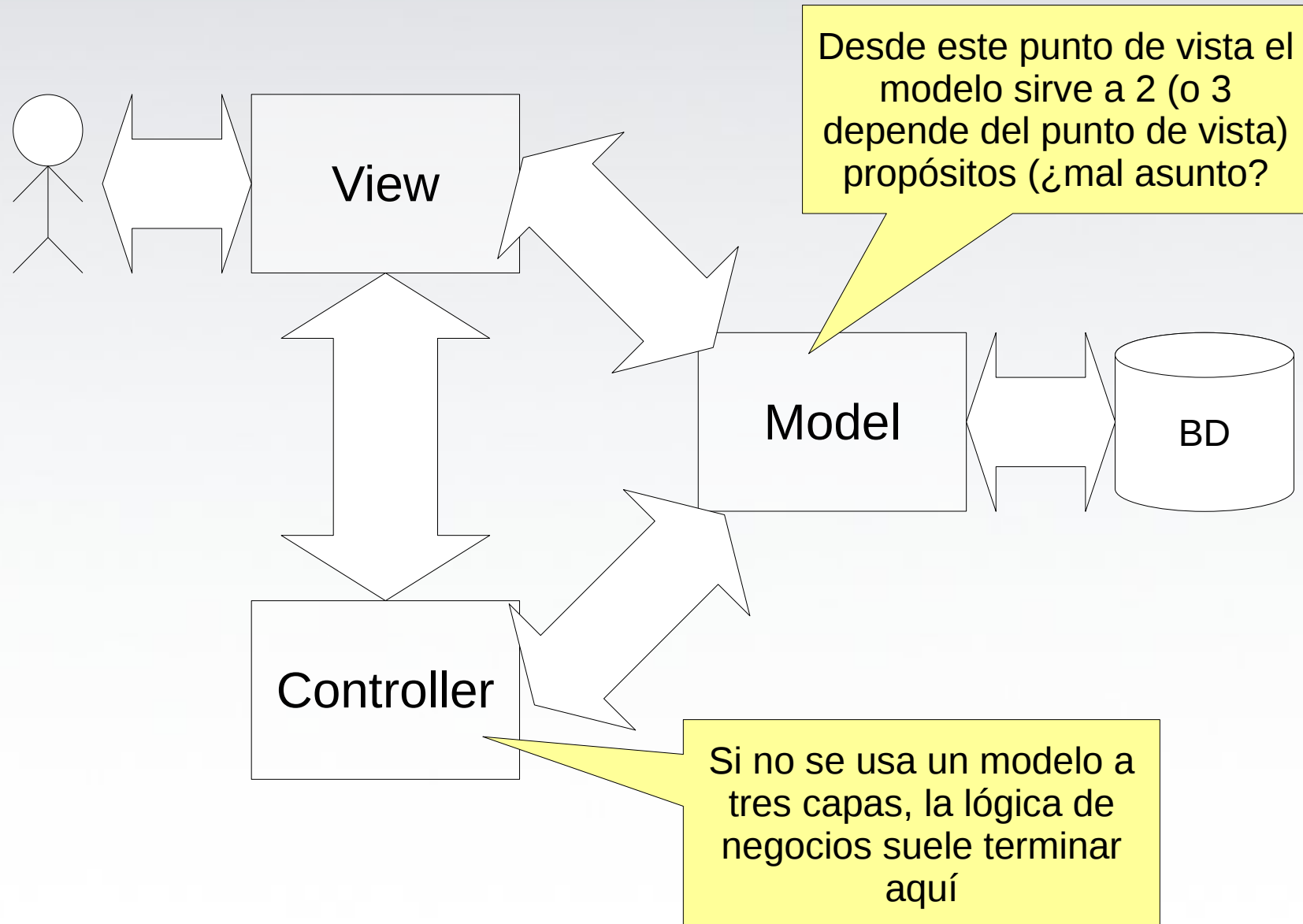
MVC != Arquitecturas a 3 Capas



MVC y Acceso a BD

(M) != Modelo de Datos (Aunque se puede)





The Model represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database

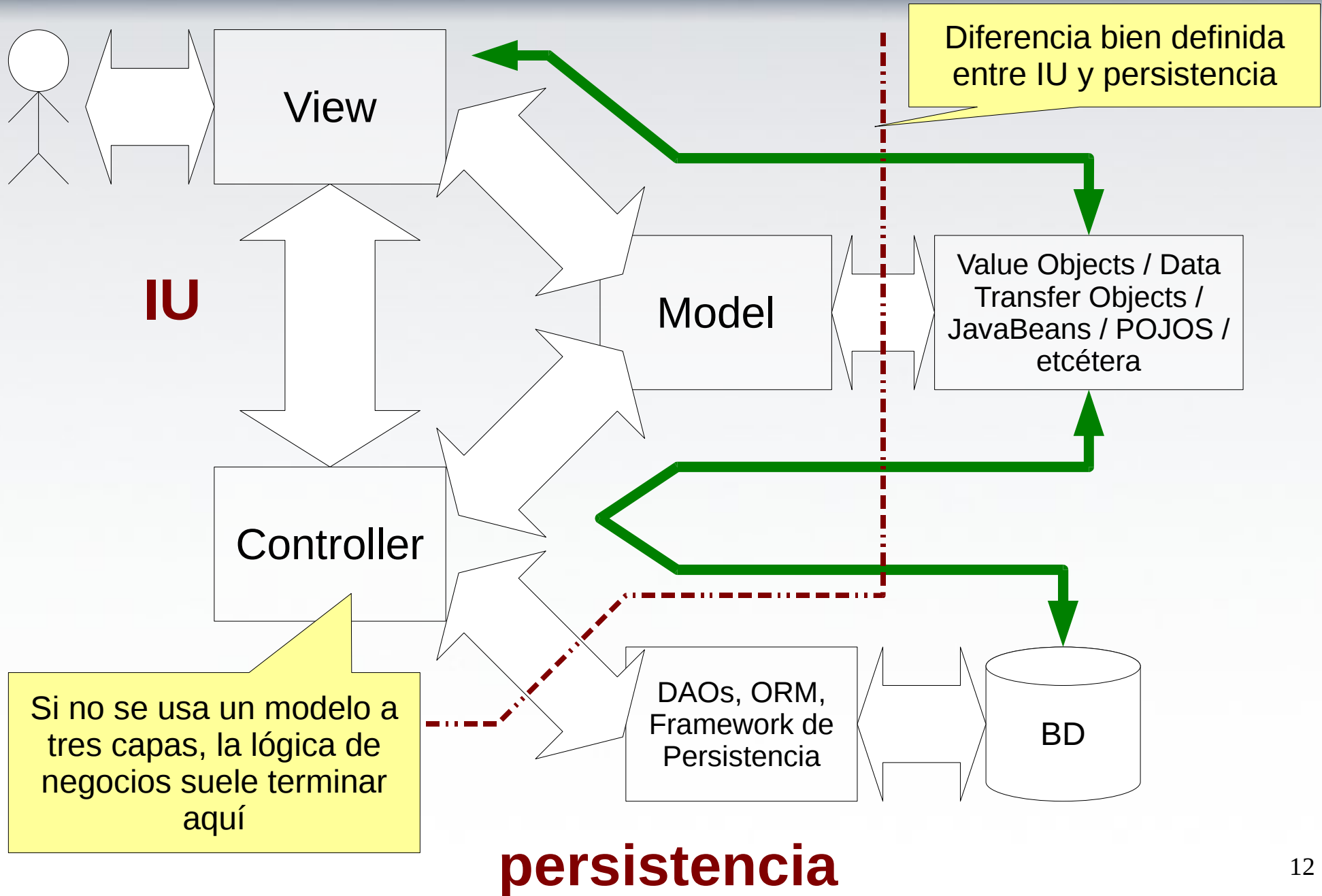
Tomado de: http://codeigniter.com/user_guide/overview/mvc.html

The model is not a database: the 'model' in MVC is both the data and the business/domain logic needed to manipulate the data in the application. Many applications use a persistent storage mechanism such as a database to store data. MVC does not specifically mention the data access layer because it is understood to be underneath or encapsulated by the model. Models are not data access objects; however, in very simple apps that have little domain logic there is no real distinction to be made. Active Record is an accepted design pattern which merges domain logic and data access code - a model which knows how to persist itself.

Tomado de: <http://en.wikipedia.org/wiki/Model-View-Controller>

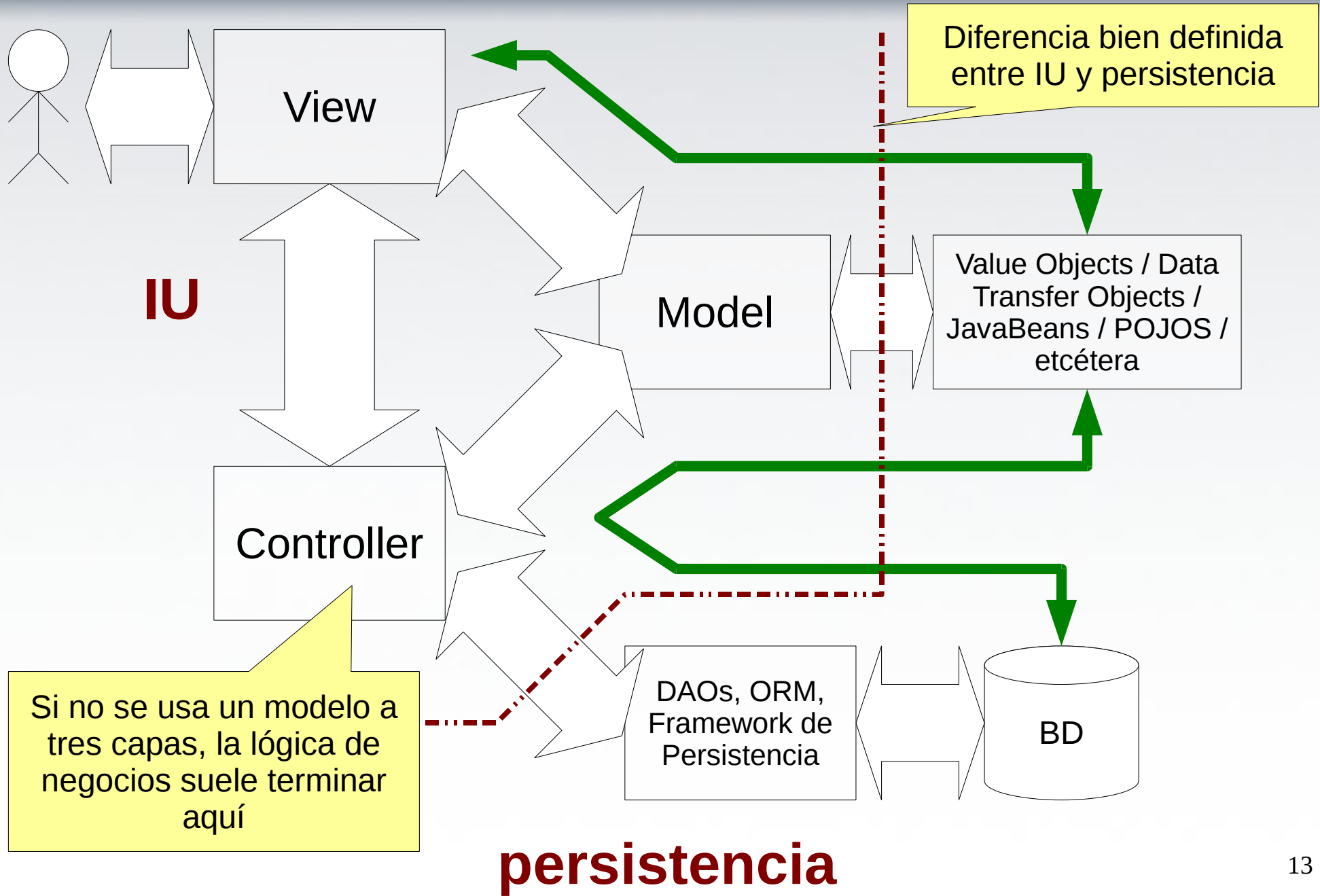
MVC y Acceso a BD

(M) != Modelo de Datos (Aunque se puede)

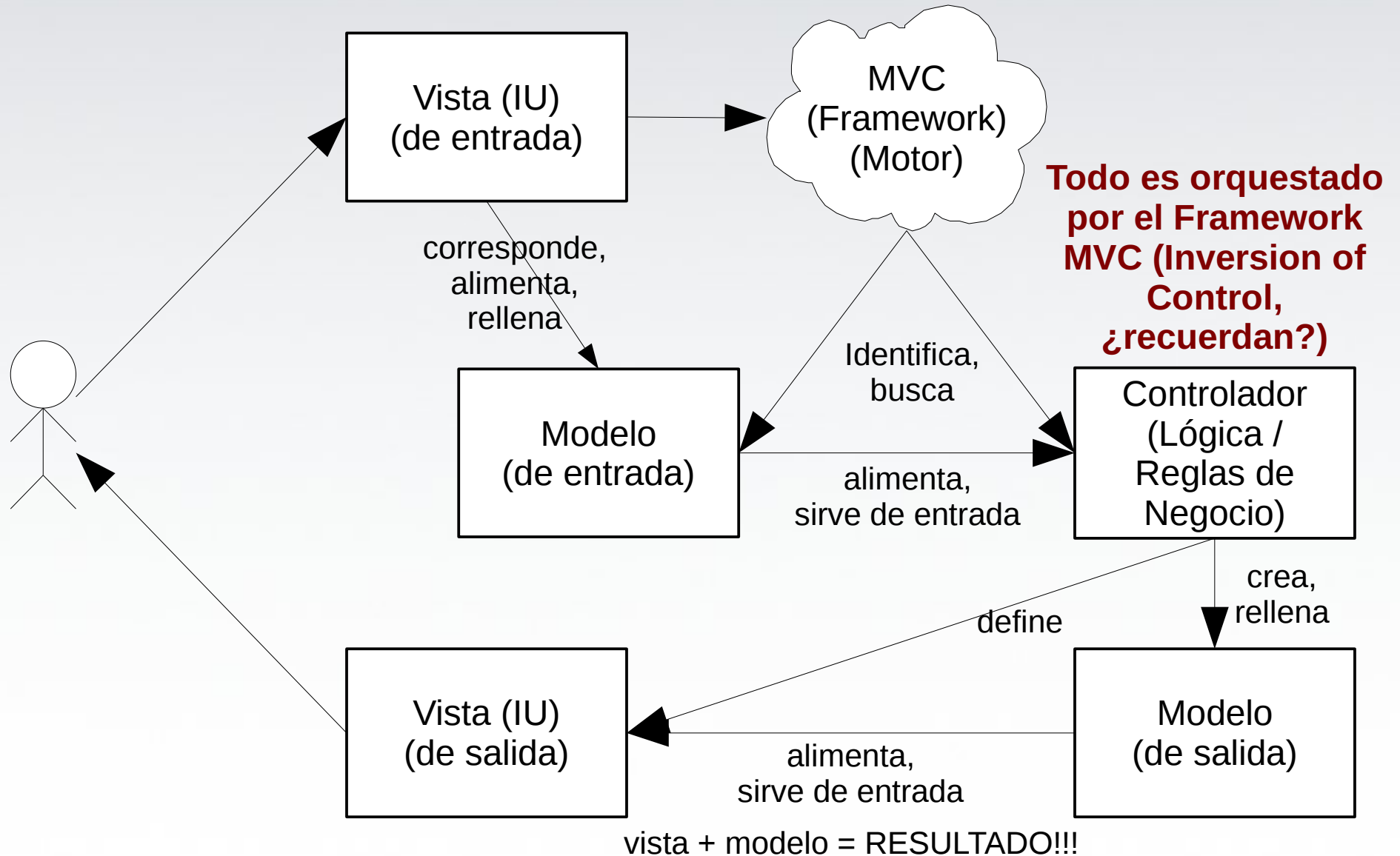


MVC y Acceso a BD

(M) != Modelo de Datos (Aunque se puede)

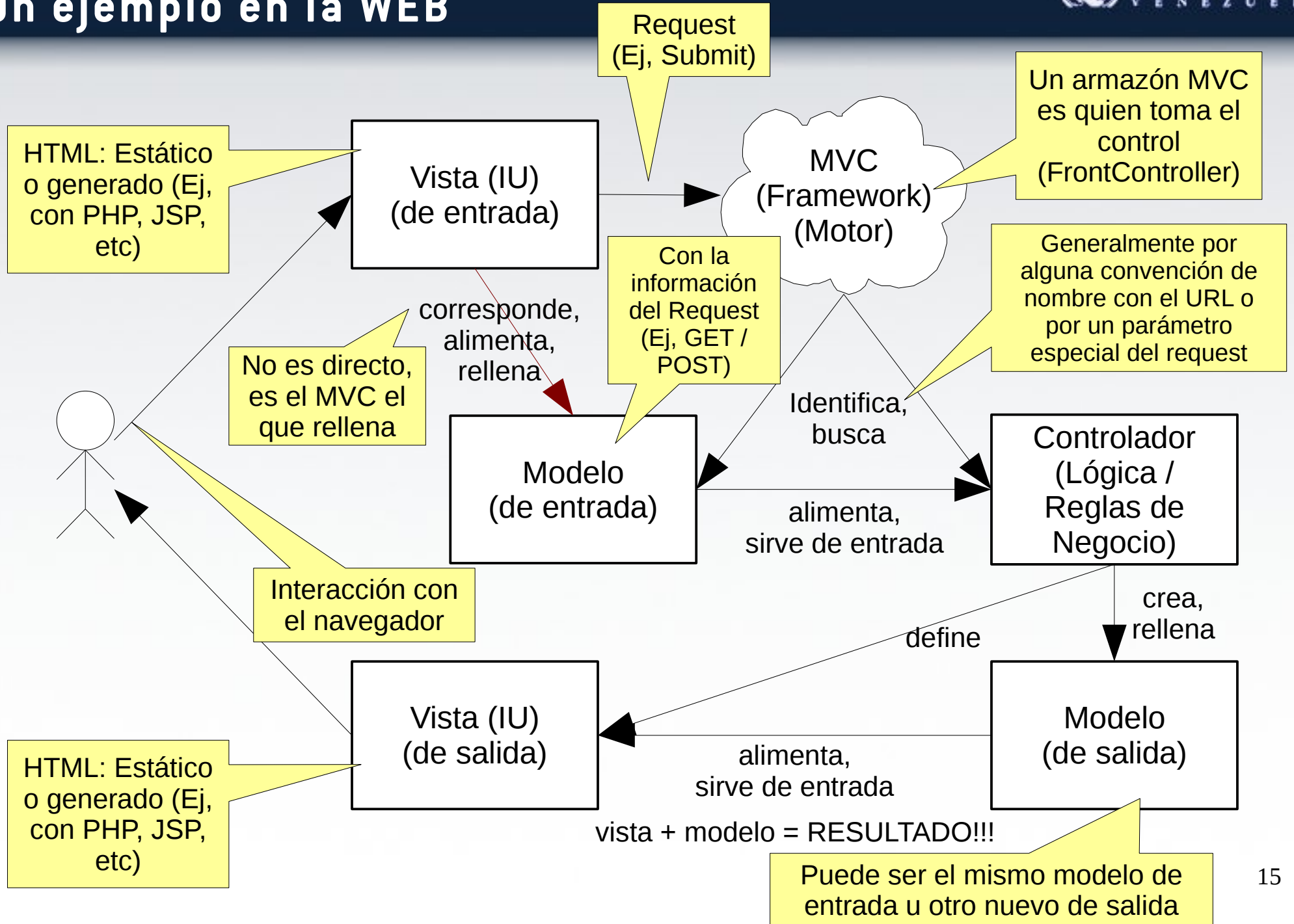


¿MVC: Modelo – Vista – Controlador? Un ejemplo en la WEB



¿MVC: Modelo – Vista – Controlador?

Un ejemplo en la WEB



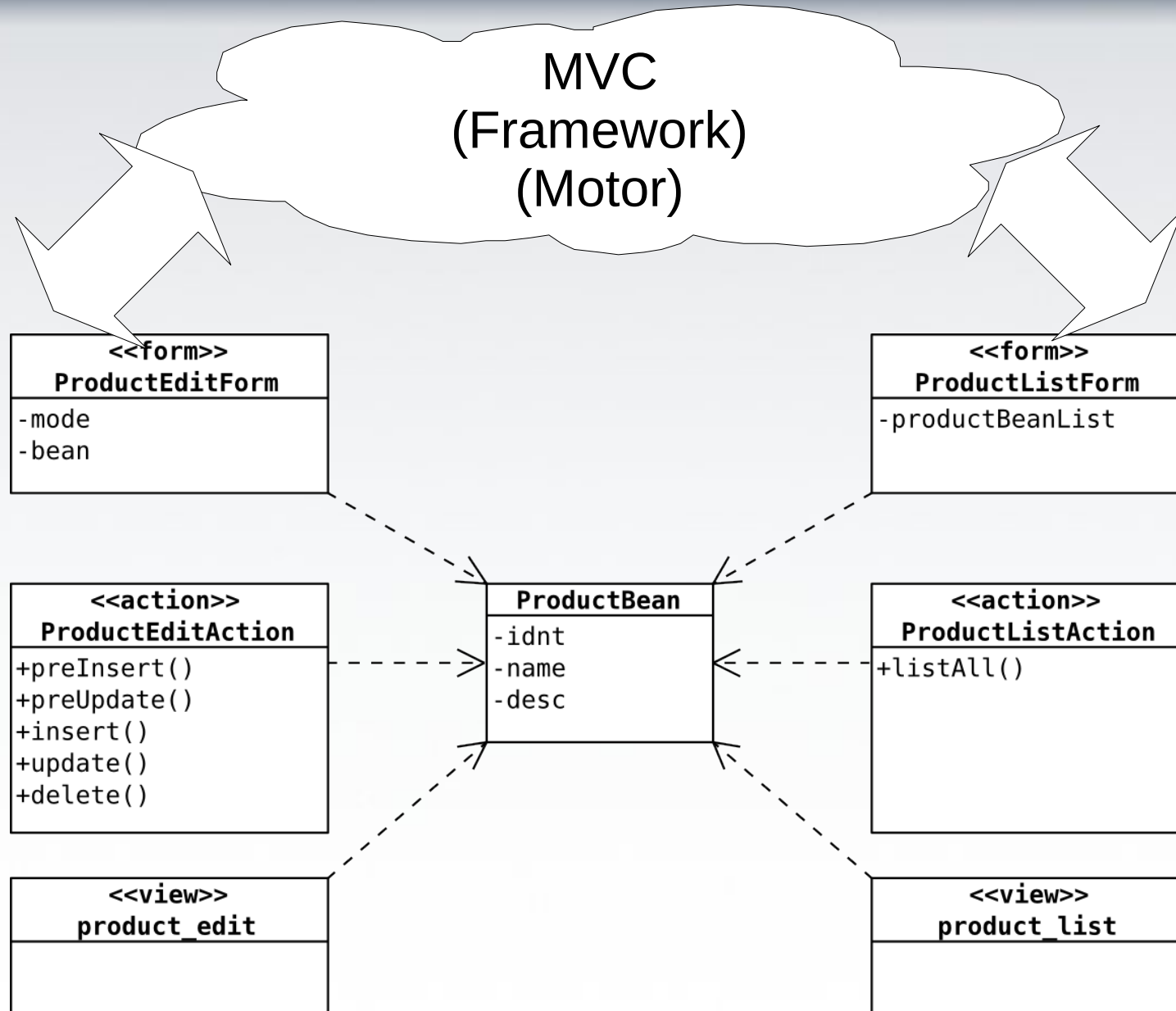
Ejemplo Struts1:

Uso de un MVC

Ejemplo PHP:

Uso e implementación (básica) de un MVC

MVC: Modelo – Vista – Controlador

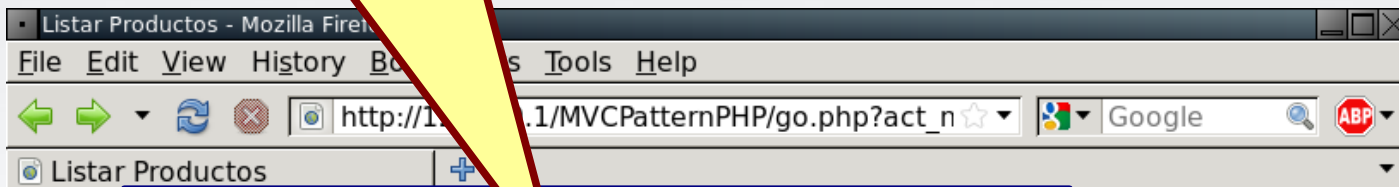









MVC: Modelo – Vista – Controlador

```
<<form>>  
ProductListForm  
-productBeanList
```

Contiene la información a mostrar por la vista (la lista de productos) así como la información ingresada por el usuario (si existe)

```
<<action>>  
ProductListAction  
+listAll()
```



Idnt	Name	Desc	Actions
6	Tuercas 1/2x5mm	Otra descripción de otra tuerca, etc	 
5	Tornillos 1/2x3	bla bla bla una descripción	 
	1/2x2	Una descripción sin sentido de un clavo	 
			

Maneja la acción de listar (Va a BD y genera la lista de productos)

```
<<view>>  
product_list
```

Genera este HTML en base al contenido del Form

http://127.0.0.1/MVCPatternPHP/go.php?act_name=product/ProductEditAction&method=prel...

MVC: Modelo – Vista – Controlador

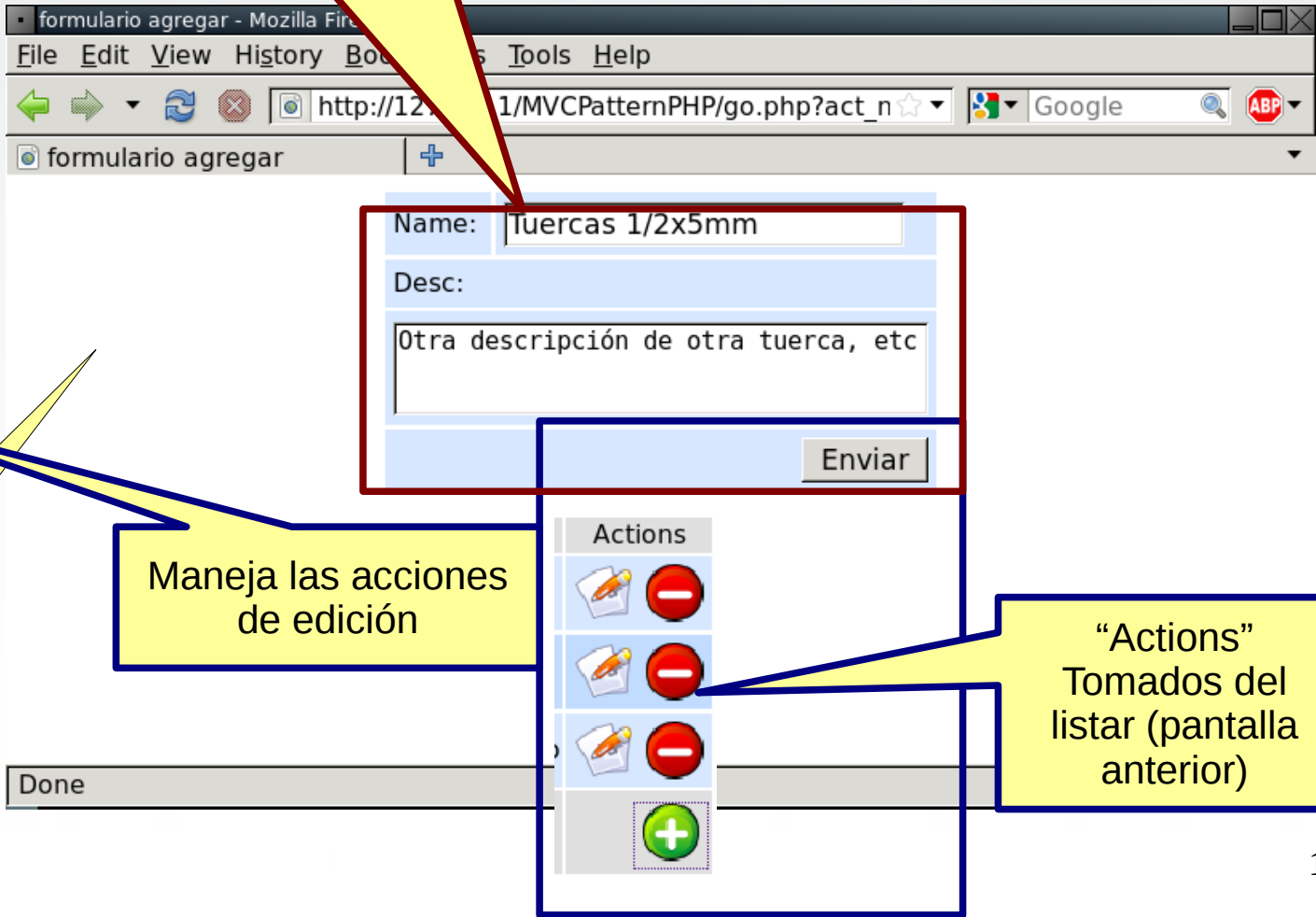
```
<<form>>  
ProductEditForm  
-mode  
-bean
```

Contiene la información a mostrar por la vista (el producto a editar) así como la información ingresada por el usuario (originada por los controles de la IU)

```
<<action>>  
ProductEditAction  
+preInsert()  
+preUpdate()  
+insert()  
+update()  
+delete()
```

```
<<view>>  
product_edit
```

Genera este HTML en base al contenido del Form



Maneja las acciones de edición

“Actions” Tomados del listar (pantalla anterior)

http://127.0.0.1/MVCPatternPHP/go.php?act_name=product/ProductListAction&method=listAll

Dirección
(¿Base?)

**http://127.0.0.1/MVCPatternPHP/
go.php**

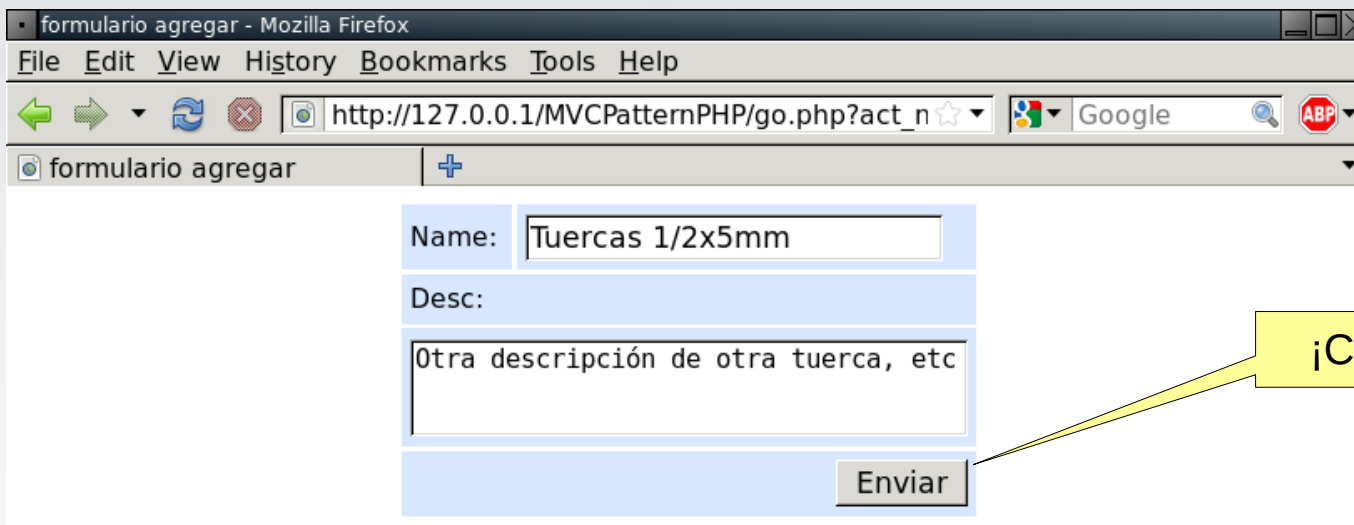
Script PHP
ejecutado
(go.php)

?act_name=product/ProductListAction&method=listAll

Parámetros de la petición
(los usa el MVC para
determinar el Form, el
Action y el método dentro
del Action a utilizar)

**De querer hacer
ingeniería inversa, y
comprender como
funciona el MVC ¿por
dónde comenzamos? Por ejemplo, que pasa si...**

... en este formulario hacemos click en enviar



formulario agregar - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://127.0.0.1/MVCPatternPHP/go.php?act_n

formulario agregar

Name: Tuercas 1/2x5mm

Desc:
Otra descripción de otra tuerca, etc

Enviar

¡Click!

```
<form method="GET" action="go.php">
```

```
<!-- CAMPOS DE CONTROL DEL MVC -->
```

```
<input type="hidden" name="act_name" value="product/ProductEditAction"/>
```

```
<input type="hidden" name="frm_name" value="product/ProductEditForm" />
```

```
<input type="hidden" name="php_self" value="product/product_edit.php" />
```

```
<input type="hidden" name="method" value="update" />
```

```
<!-- CAMPOS DE LA ACCIÓN PARTICULAR (Editar) -->
```

```
<input type="hidden" name="mode" value="update">
```

```
<input type="hidden" name="bean_idnt" value="6">
```

```
<!-- Los otros campos, el nombre, la descripción, etc -->
```

Diagrama de Secuencia

A desarrollar en clase...

Ver:

http://en.wikipedia.org/wiki/Model-View-Controller#Implementations_of_MVC_as_web-based_framework

Para una lista (probablemente incompleta) de frameworks que implementan el patrón MVC en distintos lenguajes de programación

Gracias

¡Gracias!

