

# Aplicaciones Internet Enriquecidas en el Contexto de una Intranet Universitaria.

---

Félix Oscar Fernández Peña.

CUJAE.





# Ecología del Aprendizaje en Evolución

*YouNiversity* (Jenkins, 2007)

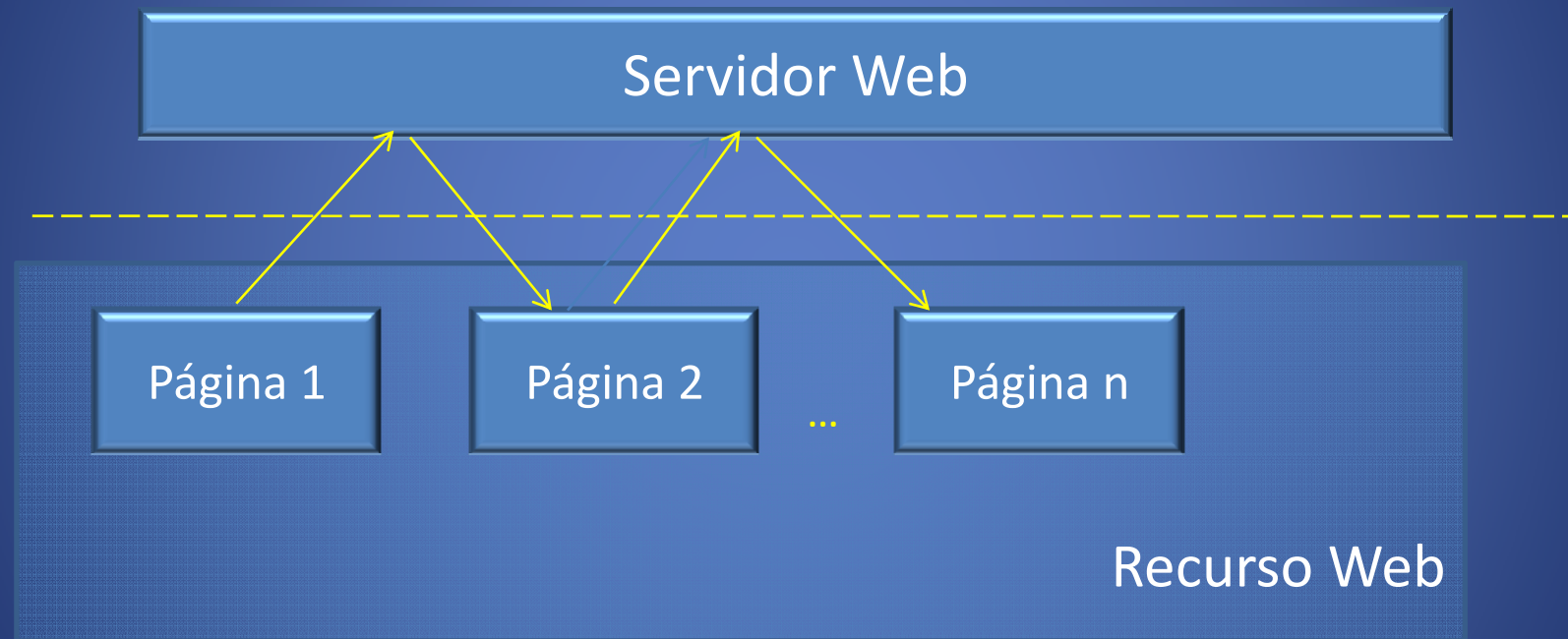
Red Intelectual donde los **estudiantes** interactúan con los **profesores**, la **industria** y la **comunidad**, provocando un **cambio** en la **ecología del aprendizaje**.



# Contenido

- Aplicaciones\_Web (Tradicionales\_Enriquecidas)
- Web (2.0, Intranet)
- Resultados ()
- Conclusiones ()

# Las Aplicaciones Web



Interacción Cliente-Servidor en una Aplicación Web “Tradicional”



# Descripción del Contexto

- No se necesita **instalación**.
- Las **actualizaciones** ocurren de manera **automática y transparente** para el **usuario**.
- Accesible desde cualquier terminal conectada a la red: **multiplataforma**.

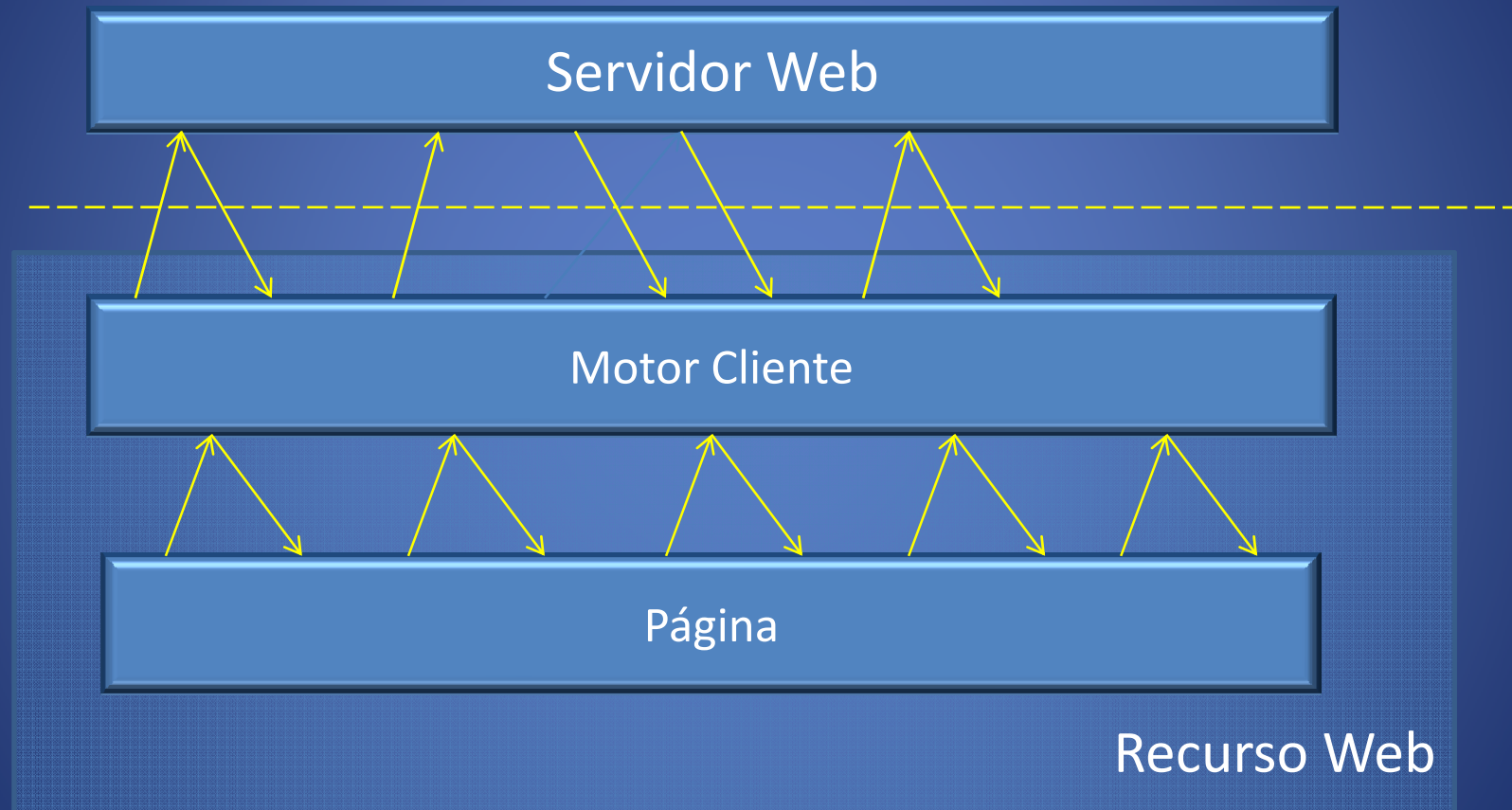


## Descripción del Contexto

- El **tiempo de espera** inherente a la ejecución de los procesos de cómputo .
- La **“ineficiencia”** del protocolo de comunicación.
- → Se afecta la **robustez** de la aplicación.

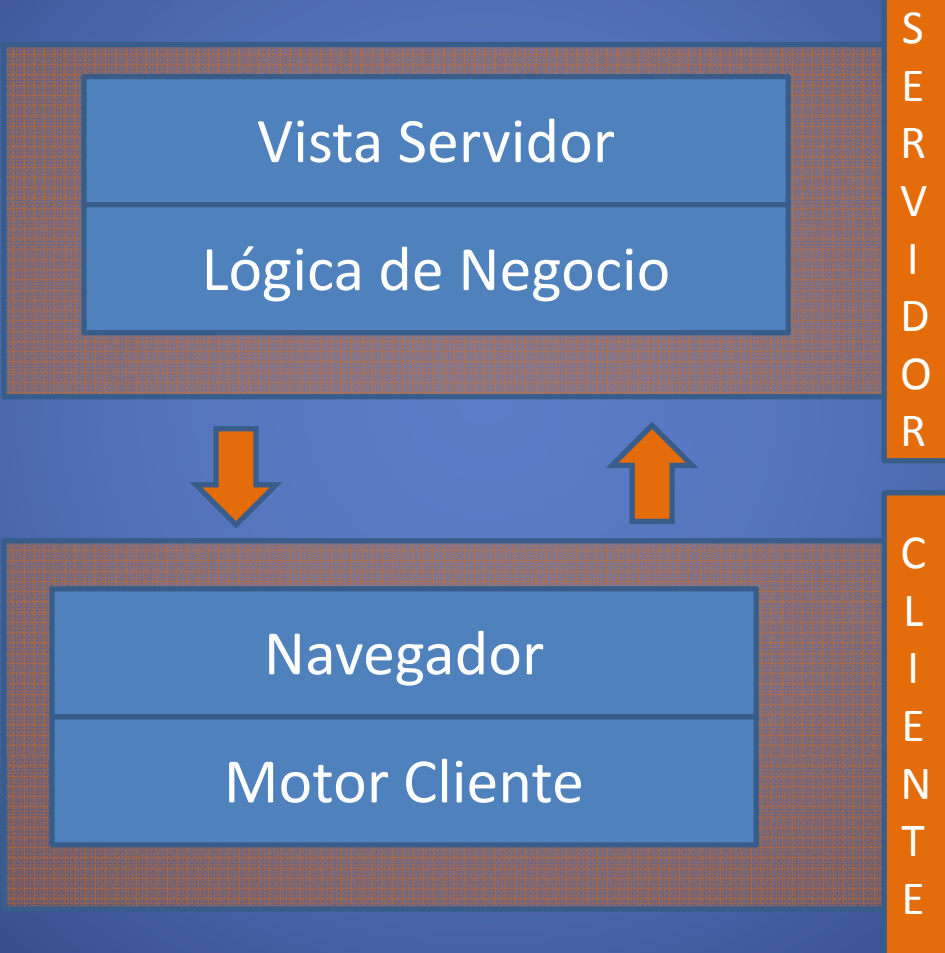


# Aplicaciones Internet Enriquecidas (RIAs)



Intenta combinar la **poderosa e intuitiva** experiencia de los usuarios de aplicaciones *desktop* con el **alcance** de Internet.

# Arquitectura de las RIAs





# Descripción del Contexto

- Reto: **explotar** la capacidad de cómputo del **lado del cliente**.
- Cambio de paradigma: protocolo de **comunicación asíncrona**.
- → Potencia el **cambio del rol** de los usuarios.



# Arquitectura de Software *Peer-Peer*

- Incremento de la **usabilidad** de los sistemas.
  - **Menor tiempo** de respuesta.
  - **Menor carga** de los procesos de cómputo para el proveedor del servicio.
- **Experiencia más rica** de los usuarios.
  - Nuevos servicios (blogs, wikis, repositorios).
  - Papel más activo. Se convierten en proveedores/consumidores de información.



# Aplicaciones RIAs

- Google Maps.
- Yahoo! mail.
- Gmail.

→ Se reduce el tráfico en la red.



# Tecnologías RIAs

- *Asynchronous JavaScript and XML (Ajax)*: JavaScript, XML como base del intercambio.
- *Adobe Flex*: Utiliza **MXML**, un lenguaje basado en XML para describir la interfaz gráfica de usuario y **Action Script** (Flash) para la interacción.
- *OpenLaszlo*: Compila programas escritos en el lenguaje LZX, una mezcla de etiquetas XML y JavaScript.
- *Curl*: **Lenguaje de programación de marcas**, orientado a objeto, diseñado para la creación de aplicaciones web interactivas.

# Creación y Uso de una Maquinaria JavaScript

*Asynchronous JavaScript and XML*

(Garett, 2005)





# JavaScript Orientado-a-Objetos

- Fácil manipulación de los datos y creación de componentes GUI reutilizables.
- Planeación, Ingeniería **vs.** Reinventar la rueda.
- Creación de Bibliotecas Personalizadas.



# Pseudo-Objetos

- El operador *new*.
- Notación Literal.
- Arreglos asociativos.
- Constructores.
- Prototipos.



## Operador *new*

```
var employee = new Object();  
employee.id = 001;  
employee.firstName = "Kris";  
employee.lastName = "Hadlock";
```

```
employee.getFullName = function(){  
    return this.firstName + " " + this.lastName;  
};
```

Sencillo pero

No es posible crear múltiples instancias del mismo objeto.



## Notación Literal. JavaScript 1.2

```
employee = {  
    id : 001;  
    firstName : "Kris";  
    lastName : "Hadlock";  
    getFullName : function() {  
        return this.firstName + " " +  
            this.lastName;  
    }  
}
```

Fácil construcción del objeto pero  
Sintaxis particular.



# Arreglos Asociativos

```
var employee = new Array();  
employees["firstName"] = "Kris";  
employees["lastName"] = "Hadlock";
```

```
var employee = new Array();  
employees[001]["firstName"] = "Kris";  
employees[001]["lastName"] = "Hadlock";  
employees[002]["firstName"] = "John";
```

Fácil acceso a elementos específicos pero  
El trabajo con muchos datos se hace engorroso.



# Constructores

```
function employee(_id, _firstName, _lastName, _jobTitle) {  
    this.id = _id;  
    this.fName = _firstName; this.lName = _lastName;  
    this.jobTitle = _jobTitle;  
    this.getFullName = function() {  
        return this.firstName + " " + this.lastName;  
    };  
    this.setJobTitle = function(_jobTitle){  
        this.jobTitle = _jobTitle;  
    };  
    this.getJobTitle = function() { return this.jobTitle; }  
}
```



## Prototipos. JavaScript 1.1

- Object.prototype / employee.prototype / kE

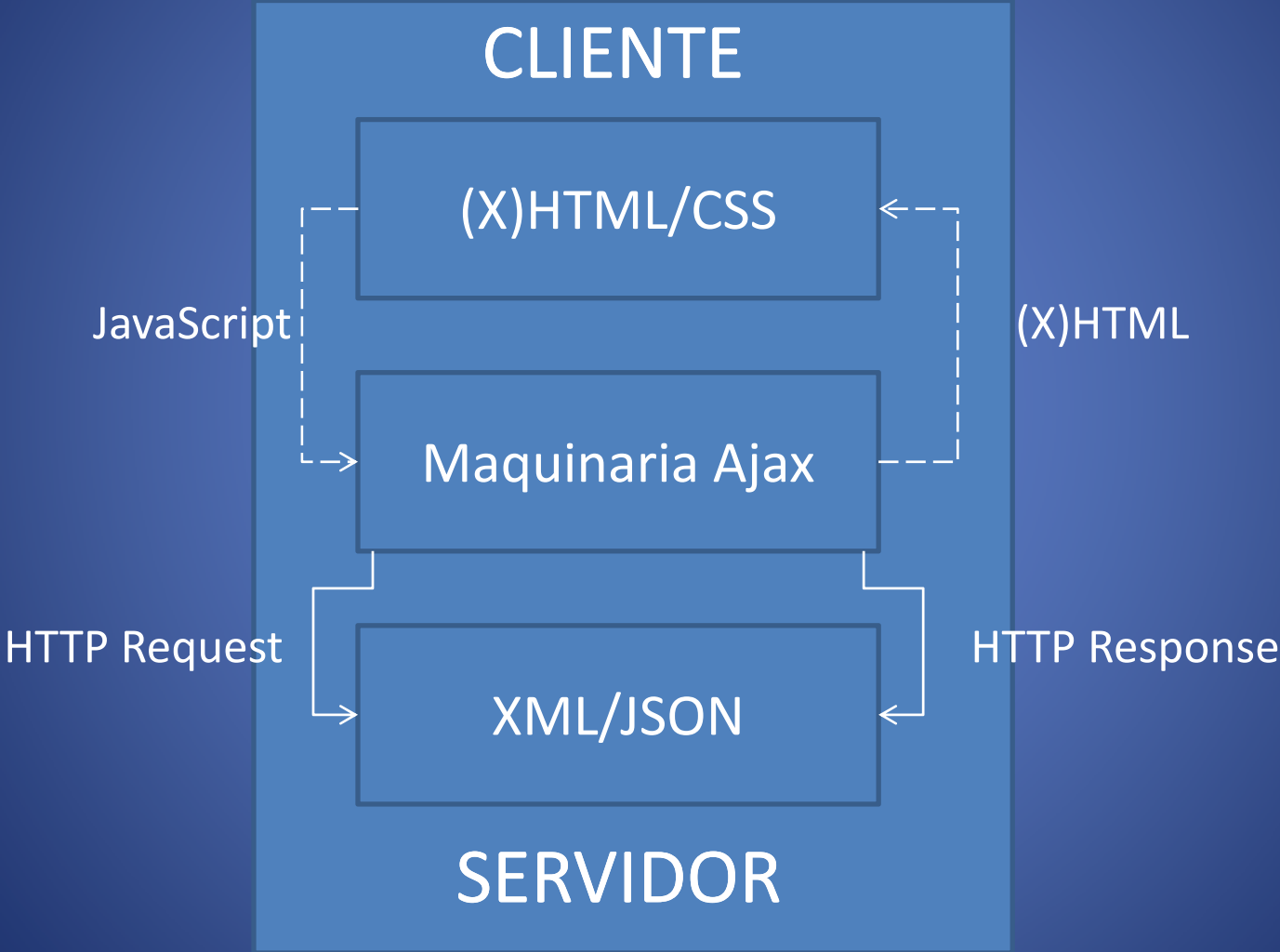
```
var kh = new employee(001, "Kris", "Hadlock");  
employee.prototype.emailAddress = "";  
kh.prototype.emailAddress = "my@emailAddress";
```



## Ajax en Uso

```
<script language="javascript"  
  type="text/javascript" src="actb.js"></script>  
  customarray = new Array('Guayaba', 'Piña',  
    'Mango', 'Papaya', 'Naranja');  
  actb(document.getElementById('tb'),  
    customarray);  
  
<input type='text' id='tb' value=''/>
```

# XMLHttpRequest (XHR)



→ Permite consultar el servidor sin refrescar la página en el cliente



# Desarrollo de Frameworks Ajax

Framework	% en uso
Prototype	68.4
Scriptaculous	58.7
jQuery	47.5
Yahoo UI	40.3
Dojo	38.3
Google Gears	22.0
Google Web Toolkit –GWT–	17.2
DWR	12.7
MooTools	11.3
Symfony	4.6
Qooxdoo	3.2
Tibco GI	2.9



# Prototype

- Framework que extiende DOM.  
Ejemplo: `$('#comments').addClassName('active').show()`
- Gran colección de módulos.
  - Ajax.
  - Array.
  - Event.
  - TimeObserver.



## IDE de Tibco GI

- Desarrollado enteramente en **JavaScript**.
- Vasto número de **componentes gráficos** para e desarrollo de una RIA.
- **WYSWYG** para la personalización de los componentes .



# Patrones de Diseño



## *Singleton Pattern*

- Resuelve el problema de tener que instanciar múltiples instancias de un objeto a través de una aplicación, dándole a la aplicación una referencia que nunca cambia.
- Singleton Object: No puede ser instanciado más de una vez.
- Ejemplo: `AjaxUpdater = {};` // instanciando un Singleton Object.



## Usando un *Singleton Object*

```
Ajax= {};
```

```
Ajax.makeReq = function(method, url, callback){  
  this.request = (window.XMLHttpRequest)?  
  new XMLHttpRequest():  
  new ActiveXObject ("MSXML2.XMLHTTP");  
  this.request.onreadystatechange = callback;  
  this.request.open(method, url, true);  
  this.request.send(url);  
}
```

```
Ajax.makeReq("POST", "./getElem.php", onResponse);
```

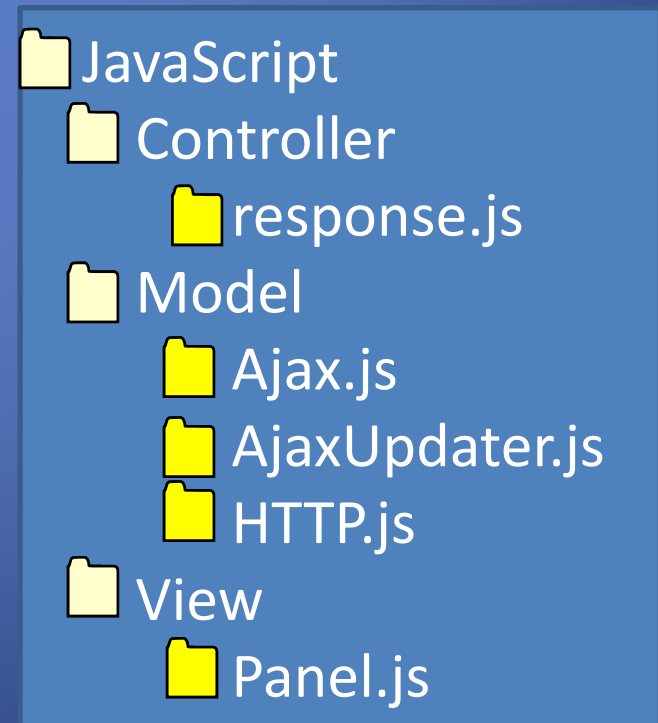
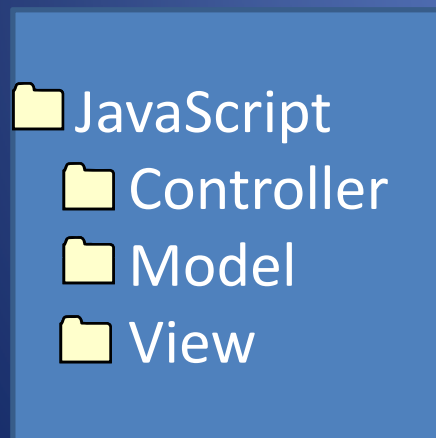


## *Model View Controller Pattern*

- Separación de la **GUI** con respecto a la **lógica central de la aplicación**.
- Objetivo: evitar tener que hacer **cambios** donde se programa la **capa de negocio** cuando en realidad solo se pretende un **cambio en la interfaz**.

# Implementación del *MVC*

- **Componente:** al menos un **controller** y un **view**.
- **Model:** objetos que **interactúan con el servidor**.



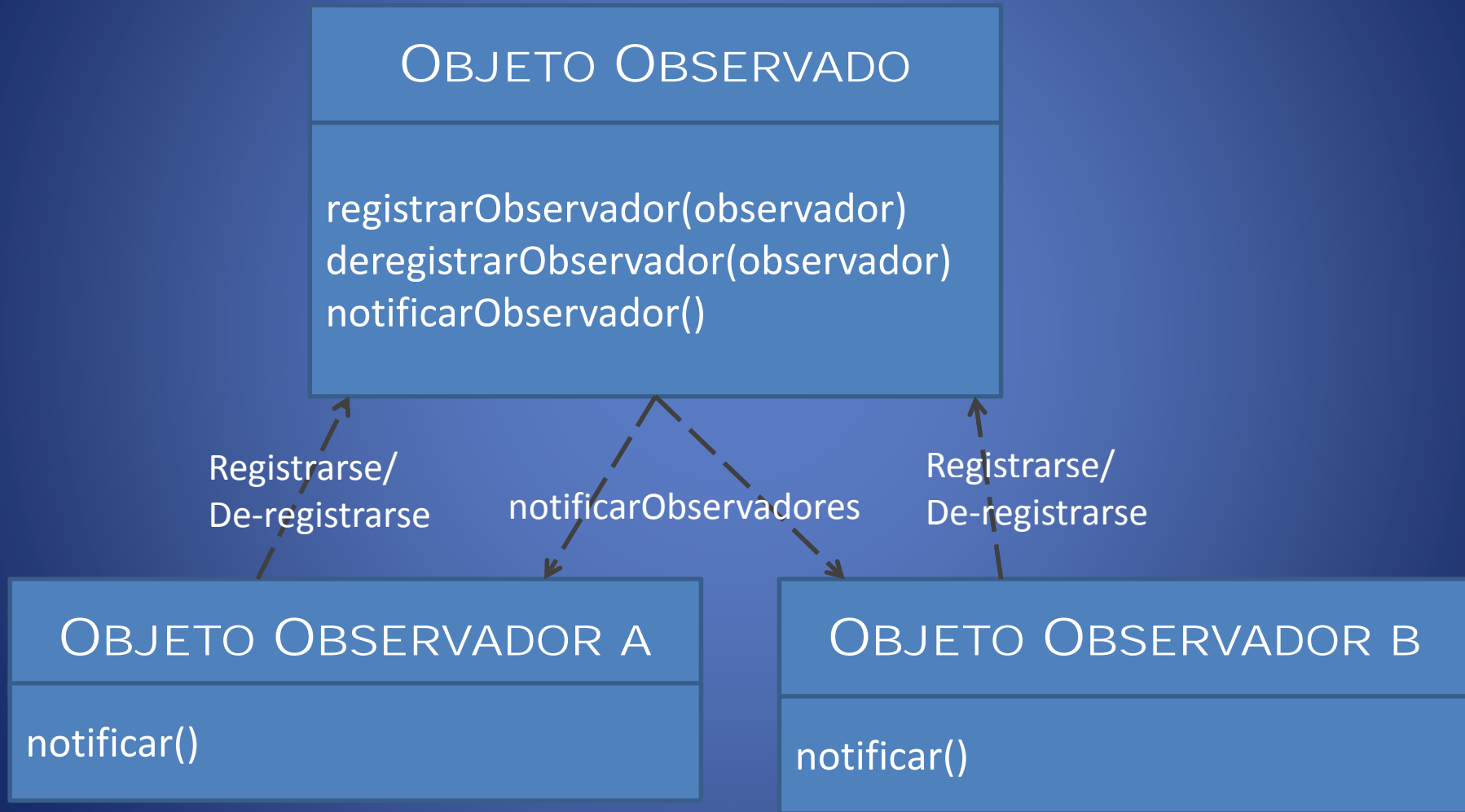
→ Reusabilidad de los objetos.



## *The Observer Pattern (OP)*

- Crea una forma para **manipular** un número **ilimitado de objetos** como una **colección con solo algunos métodos**.
- Utilizado para observar el **estado** de un **objeto** en una **aplicación**.

# Representación Gráfica del *OP*

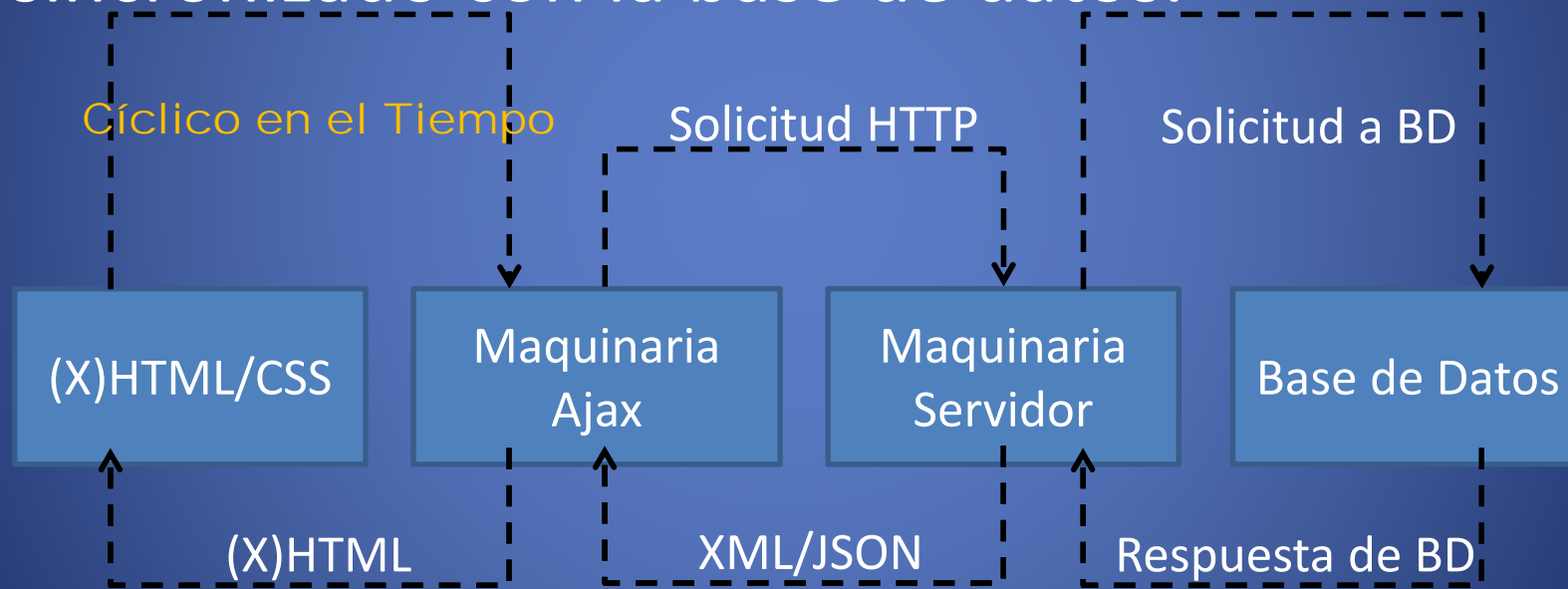


Ejemplo: Creación de un Objeto de Manipulación de Errores (ErrorManager)



# Data Reflection Pattern

- Mantiene el contenido de la aplicación sincronizado con la base de datos.





## *Multi-User Pattern*

```
Email.dataReflection =  
function(callbackMethod, delay) {  
  setTimeout(callbackMethod, delay); }  
}
```

```
Email.display = function(username) {  
  Email.dataReflection("Email.display(""+  
  username +""), 60000); }  
}
```



# Usability Pattern

**Please correct the following errors.**

- Enter a valid US city.
- ~~Select a state from the list.~~
- Enter a valid email address.

First Name:

Last Name:  Please enter a valid US city.

City:  State:

Country:

Email:

Web Site:



# Web 2.0

- **Web 2.0** (O'Reilly Media, 2003)  
Segunda generación de aplicaciones que permiten la **interacción entre comunidades Web**.
- Considerada como el punto de cambio de una **Web “de solo lectura”** en una **Web “de lectura-escritura”** (Gillmor, 2007).



# Web 2.0 en Intranet

- La red se concibe como plataforma de **aplicaciones accesibles por Web**.
- Los **usuarios** son **dueños de los contenidos** de los sitios.
- Los usuarios son **motivados a contribuir**.
- Una **interfaz rica, interactiva, y amistosa**.
- Funciones de **socialización en red**.



# Intranet. Valores Agregados

- Blogs, Forums.
- Wikis.
- Repositorio de Recursos.
- Gestión Semántica de Recursos de Información.



# Beneficios de los Blog

- Promueven un pensamiento analítico y crítico.
  - Pueden promover un pensamiento intuitivo asociativo y creativo.
- 
- Soporte al intercambio de comentarios sobre contenido, la lectura de bibliografía y otros comentarios.
  - Espacio colaborativo para que los estudiantes actúen como revisores del contenido impartido.
  - Registro de la actividad interactiva del curso.

P. Duffy. *“Engaging the YouTube Google-Eyed Generation: Strategies for Using Web 2.0 in Teaching and Learning”* Electronic Journal e-Learning Volume 6 Issue 2 2008 (119 - 130)



## Beneficios de las Wikis

- Espacio en línea para la **autoría y escritura colaborativa**.
  - **Control del historial** de determinado recurso de información.
- 
- Los estudiantes pueden crear sitios web sin conocimiento de HTML.
  - Propicia la colaboración y favorece una mejor gestión de la información.



## Beneficios de las Wikis

- Gestión de proyectos de investigación de los estudiantes.
- Creación de una bibliografía anotada colaborativa. [Course Advisor Wiki. Brown University, EUA.](#)
- Publicación de comentarios, noticias y su posterior edición y comentario colaborativo.
- Mapeo de conceptos. Constituye una red enlazada de recursos de información.

P. Duffy. *“Engaging the YouTube Google-Eyed Generation: Strategies for Using Web 2.0 in Teaching and Learning”* Electronic Journal e-Learning Volume 6 Issue 2 2008 (119 - 130)



# Tendencias. La Folcsonomía.

- Folcsonomía & Memética.
- Ontologías & Semántica.

## Folcsonomía.

Indización social; clasificación colaborativa por medio de etiquetas simples sin jerarquía.

<http://del.icio.us>

## Memética.

Acercamiento científico a los modelos evolutivos de transferencia de información basados en el concepto de "meme" (Dawkins, 76).



# Caso de Estudio

## Intranet Universitaria

# Gestión Semántica

Vistas

Descripción

REPOSITORIO DE RECURSOS

Recomendadas

- Google lanza edición de ...
- Nuevo backdoor para HP co...
- Apple se quejaba contra ...
- Nuevo estándar facilitará...
- CA se incorpora al grupo ...
- Hewlett Packard celebra H...
- Avaya ofrece a las empres...

Recomendadas


Noticias

The screenshot shows a web interface for semantic management. At the top, there is a search bar labeled 'Buscar Noticias' and a 'Buscar' button. Below this is a navigation bar with 'EMPRESA' and several icons. The main content area displays a list of news articles, including 'Symantec Cierra Año Fiscal 2006', 'Verbatim anuncia los USB HD Drives de 4GB y 8GB', and 'BenQ presenta el proyector digital inalámbrico más pequeño del mundo'. On the left side, there is a sidebar with a 'Vista Actual' section and a 'Menú Semántico' section containing various categories like 'Arquitectura', 'SQA', 'Empresas', 'Hardware', 'Internet', 'Multimedia', 'Periféricos', 'Procesadores', 'Seguridad', and 'Software'. A large orange diagonal banner with the text 'REPOSITORIO DE RECURSOS' is overlaid on the central part of the page. On the right side, there is a 'Recomendadas' section with a list of recommended items. The interface is designed with a light blue and white color scheme.



# Caso de Estudio

Bienvenido(a), Pepe Perez [Salir]



Facultad de Informática  
Cujae

Nuevo Tema Nueva Pregunta

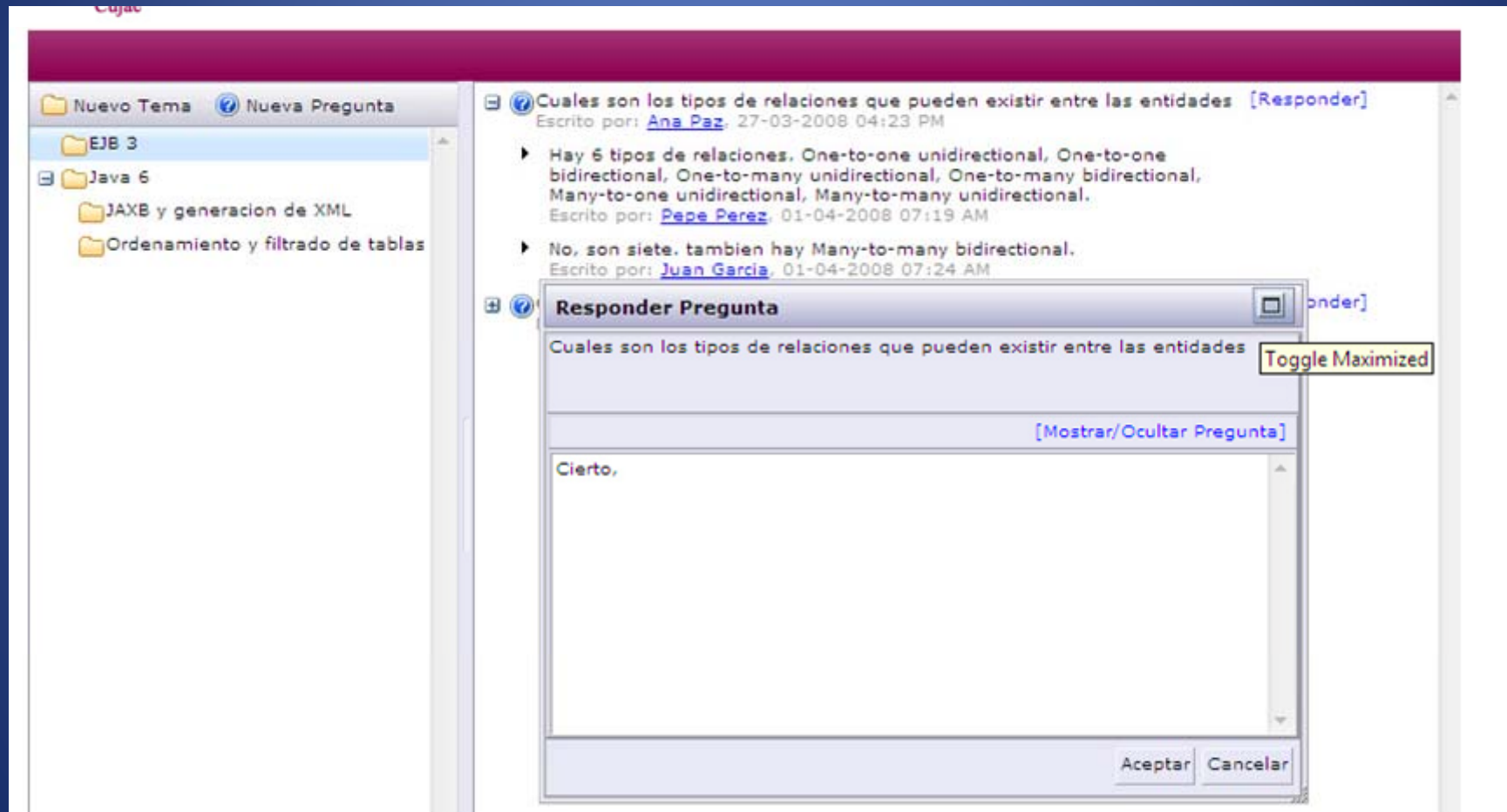
- EJB 3
- Java 6
  - JAXB y generacion de XML
  - Ordenamiento y filtrado de tablas

☑ Cuales son los tipos de relaciones que pueden existir entre las entidades [Responder]  
Escrito por: [Ana Paz](#), 27-03-2008 04:23 PM

- ▶ Hay 6 tipos de relaciones. One-to-one unidirectional, One-to-one bidirectional, One-to-many unidirectional, One-to-many bidirectional, Many-to-one unidirectional, Many-to-many unidirectional.  
Escrito por: [Pepe Perez](#), 01-04-2008 07:19 AM
- ▶ No, son siete. tambien hay Many-to-many bidirectional.  
Escrito por: [Juan Garcia](#), 01-04-2008 07:24 AM

☑ Cuales son las estrategias para mapear la herencia [Responder]  
Escrito por: [Ana Paz](#), 27-03-2008 05:16 PM

# Uso de Tibco GI y Prototype



The screenshot displays the Tibco GI (Global Interface) application. On the left, a navigation pane shows a tree structure with folders: 'Nuevo Tema', 'Nueva Pregunta', 'EJB 3', 'Java 6', 'JAXB y generacion de XML', and 'Ordenamiento y filtrado de tablas'. The main content area shows a list of questions. The selected question is: 'Cuales son los tipos de relaciones que pueden existir entre las entidades' with a '[Responder]' button. Below the question, two answers are listed:

- ▶ Hay 6 tipos de relaciones. One-to-one unidirectional, One-to-one bidirectional, One-to-many unidirectional, One-to-many bidirectional, Many-to-one unidirectional, Many-to-many unidirectional. Escrito por: [Pepe Perez](#), 01-04-2008 07:19 AM
- ▶ No, son siete. tambien hay Many-to-many bidirectional. Escrito por: [Juan Garcia](#), 01-04-2008 07:24 AM

A modal dialog box titled 'Responder Pregunta' is open, showing the question text: 'Cuales son los tipos de relaciones que pueden existir entre las entidades'. It includes a '[Mostrar/Ocultar Pregunta]' link and a text area containing the answer 'Cierto,'. The dialog has 'Aceptar' and 'Cancelar' buttons at the bottom. A 'Toggle Maximized' tooltip is visible over the dialog's title bar.



# YUI



## Ingeniería Informática Cujae

Reunión de colectivos de año: 23 de Septiembre de 2008...  
[Actividad en CasaBlanca...](#)  
[Inicio del curso escolar...](#)  
[Presentación de CD Playa Girón...](#)

Reunión de la FEU: 8 de septiembre de 2008...  
[El proyecto Sí a la Vida invita a concierto en la escalinata universitaria...](#)

[\[detalles\]](#)

[Autenticarse](#)

[Contraer menú](#)

### Mensajes

Reunión de la FEU: 8 de septiembre de 2008.  
Inicio del curso escolar

- ▶ FEU
- ▼ Proyecto Educativo
  - ▶ Deporte
  - ⊞ Cultura
  - Docencia

[webmaster@informatica.cujae.edu.cu](mailto:webmaster@informatica.cujae.edu.cu)



# YUI

### Herramientas de Edición de Página

**Tipo de letra y tamaño**  
Arial 13

**Estilo de letra**  
B I U A<sub>x</sub> A<sup>x</sup> [Color] [Background Color] [Link]

**Alineación**  
[Left] [Center] [Right] [Justify]

**Estilo de párrafo**  
Normal

**Identación y listas**  
[Increase Indent] [Decrease Indent] [Bulleted List] [Numbered List]

**Inserción...**  
[Link] [Image] [Table]

This is some more test text.  
This is some more test text.  
This is some more test text.  
This is some more test text.  
This is some more test text.  
This is some more test text.  
This is some more test text.

body



# Evaluación de Resultados

- Consultas asincrónicas.
  - Mayor **robustez** en la ejecución de la aplicación.
  - Incremento de la **accesibilidad** a la información.
  - **Gestión inteligente** de la información: el usuario recibe la información en el momento en que la necesita.
- Componentes gráficos.
  - **Ventanas** desplegadas.
  - **Estructura arbórea** para la clasificación de la información.



## RIA, la “cara” de SOA

- Reto: interacción de los servicios con los usuarios (**interactividad, intuición**).
- -> Los servicios pueden integrarse en una misma **página-interfaz** utilizando RIA.



# Perspectivas

- Computación ubicua.
- Servicios web: abstracción de la lógica de negocio.
- Interacción con el usuario de una aplicación en el contexto semántico que la describe.



# Limitaciones

- Javascript y las políticas de seguridad.
- Impacto negativo en la accesibilidad de usuarios discapacitados.
- Tiempo de descarga del motor cliente.
  - Compresión del código de las bibliotecas.
- Problemas de indexación en buscadores.



# Debilidades

- Establecer una comunicación asincrónica dificulta aislar los problemas de rendimiento.
- Paradójicamente, las acciones tomadas para incrementar la sensibilidad de la aplicación hacen más difícil entender, reportar y gestionar esta sensibilidad.
- Las diferencias entre las implementaciones del Modelo Objeto Documento de cada navegador.

→ Necesidad de un Estándar



# Conclusiones

- Convergencia Web –Aplicaciones *Desktop*.
- Solución a problemas comunes en el desarrollo de aplicaciones web.
  - Tiempo de espera de respuesta del servidor.
  - Granularidad en los procesos de actualización de la información en la capa cliente.
- Inconvenientes.
  - Incompatibilidades entre las implementaciones DOM de los navegadores.
  - La no existencia de estándares que normalicen el desarrollo de plataformas Ajax.



# Conclusiones

- Los estudiantes deben tomar la autoría de su propio aprendizaje y desarrollar habilidades en la gestión de medios e información.
- Común a Web 2.0: basado en lo social y comunitario.