

Desarrollo Web

Parte 2
Edición 2017

Taller de Programación
www.fing.edu.uy/inco/cursos/tprog
tprog@fing.edu.uy

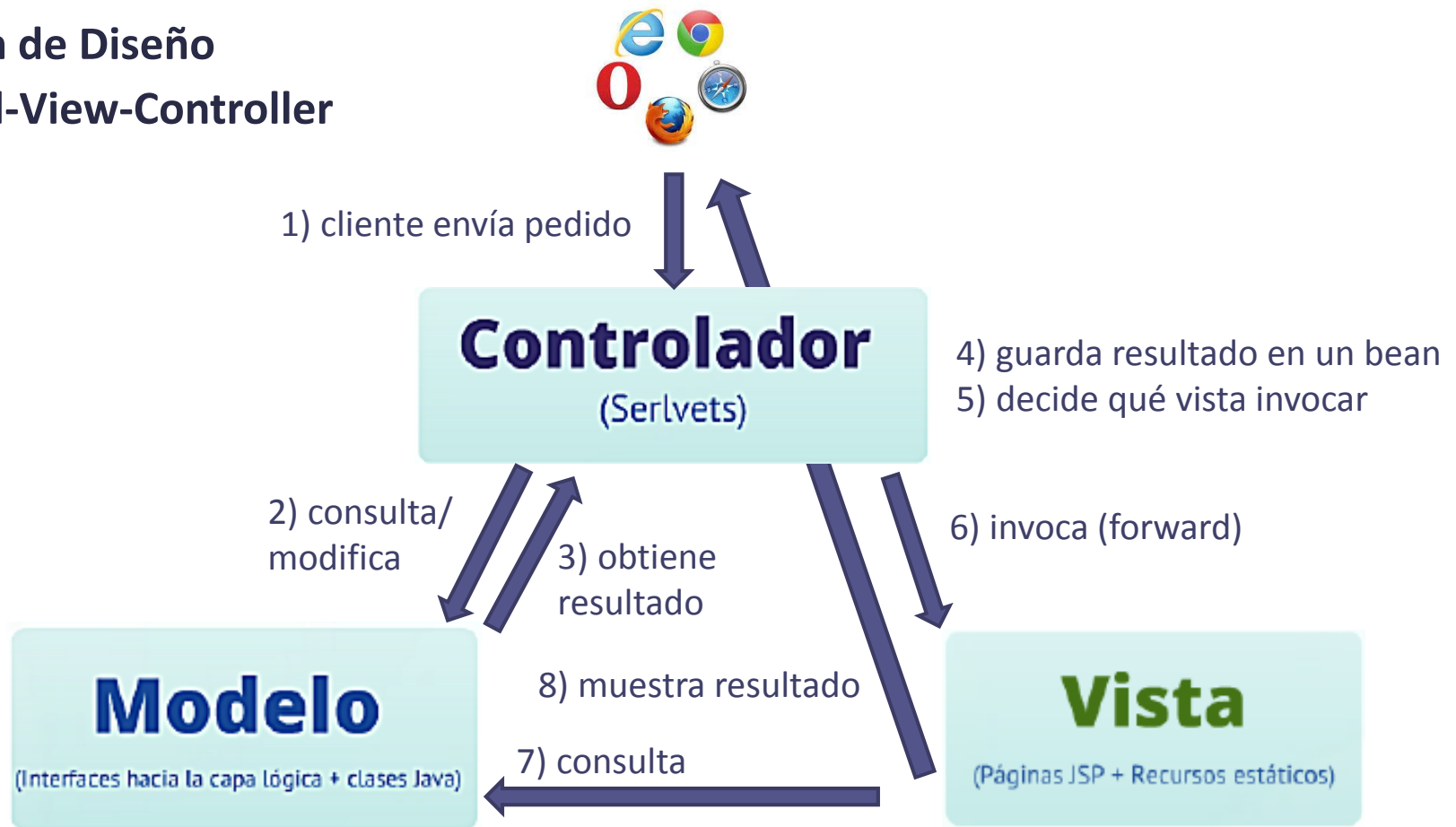
Temario

- MVC con JSP y Servlet
- CSS
- DOM
- JavaScript
- jQuery
- AJAX
- Bootstrap
- Demo

MVC con JSP y Servlet

Patrón de Diseño

Model-View-Controller



MVC con JSP y Servlet

1. El cliente completa un *form* y envía el pedido (GET/POST) al servlet indicado en el *action* del form (concide con el urlPatterns del @WebServlet).
2. El servlet (controlador) extrae los parámetros del HttpRequest e invoca la lógica de negocio (modelo) con esos parámetros.
3. La lógica de negocio devuelve resultado en un bean.
4. El servlet guara el bean utilizando *setAttribute* en *request*, *session* o *context*.
5. A continuación decide qué JSP se debe mostrar
6. Hace un *forward* a la JSP.
7. La JSP extrae el resultado del bean
8. La JSP muestra el resultado.

MVC con JSP y Servlet

- Alcance (**scope**):
 - **Request**: visible para el servlet y la pagina a la que hace forward
 - **Session**: visible para todos los servlets y páginas que participan en la misma sesión de usuario.
 - **Contexto** (aplicación): visible para todas los usuarios de la aplicación. Poco usado.
- Redirección (**forward**)

```
ServletContext context = getServletContext();  
RequestDispatcher dispatcher =  
context.getRequestDispatcher("resultados.jsp");  
dispatcher.forward(request, response);
```
- Ver también: **include**, **sendRedirect**.

CSS

- Cascading Style Sheets
- Permite definir la presentación de un documento HTML
- Independiza presentación de la información del documento
- Ej: colores, fuentes, distribución del texto, etc.

HTML



VS

HTML + CSS

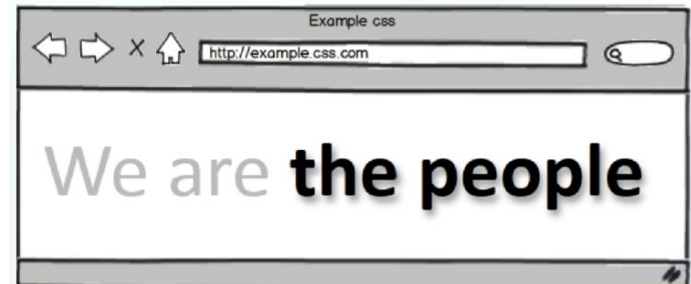


CSS - Ejemplo

```
<html>
  <style type="text/css">
    body { /* selector por etiqueta */
      color: #BBBBBB;
      font-family: calibri;
      font-size: 50pt;
    }

    #especial { /* selector por atributo id */
      color: #000000; /* regla sobrecargada */
      font-weight: bold;
      text-shadow: 3px 5px 5px #999999;
    }
  </style>

  <p>We are <span id="especial">the people</span></p>
</html>
```

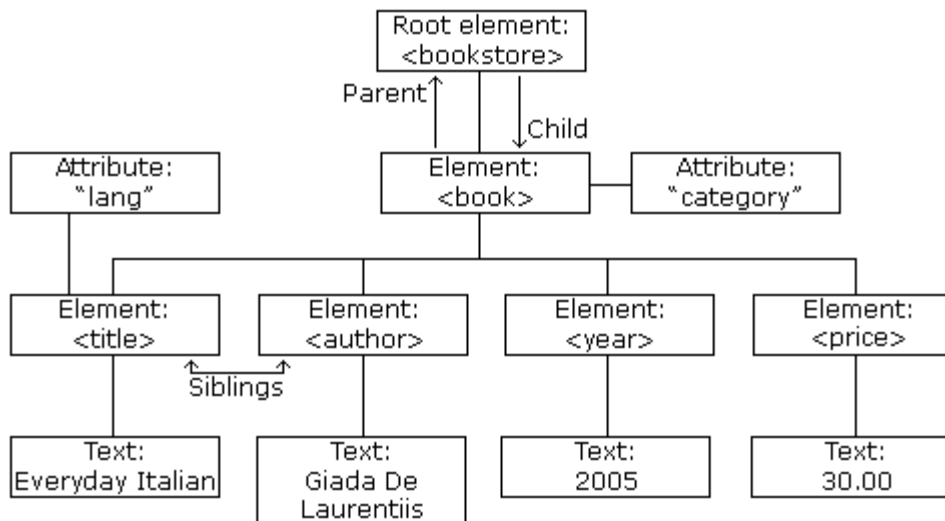


DOM

- Document Object Model
- Es un modelo que permite representar un documento en forma independiente del lenguaje.
- Define una interfaz estándar para manipular documentos con estructura arborescente
- En el caso de HTML, permite acceder a los elementos, atributos y texto (contenido) mediante objetos *Node*.
- Existe un nodo raíz denominado *document*.

DOM

- Se puede recorrer todo el documento mediante métodos de Node (*childNodes*, *firstChild*, *nextSibling*, etc).
- Se puede obtener un elemento con el método *getElementById*.
- Se pueden crear (*createElement*), agregar (*appendChild*) y borrar elementos (*removeChild*).



JavaScript

- Lenguaje de programación imperativo, interpretado, débilmente tipado y dinámico.
- Los navegadores ejecutan el código JavaScript que es referenciado desde una página Web.
- Se puede embeber en la página HTML mediante la etiqueta *script* en el head o en el body

```
<html>  
  <head>... <script>...</script> ... </head>  
  <body> .....<script>...</script></body>  
</html>
```

JavaScript

- Es una buena práctica incluir el código en archivos separados y referenciarlos, tanto si son locales a la página (camino relativo) o externos (URL)

```
<script src="scriptLocal1.js"></script>
```

```
<script src="js/scriptLocal2.js"></script>
```

```
<script src="https://www.example.com/js/scriptExterno.js"></script>
```

JavaScript

- Puede cambiar la estructura, contenido y estilo de la página Web manipulando el DOM.

```
var unSpan = document.getElementById("spanID");  
unSpan.innerHTML = "Hola JavaScript";  
unSpan.style.fontSize='35px';  
var unBoton = document.createElement('input');  
unBoton.setAttribute('type', 'button');  
document.getElementById("formID").appendChild(unBoton );
```

JavaScript

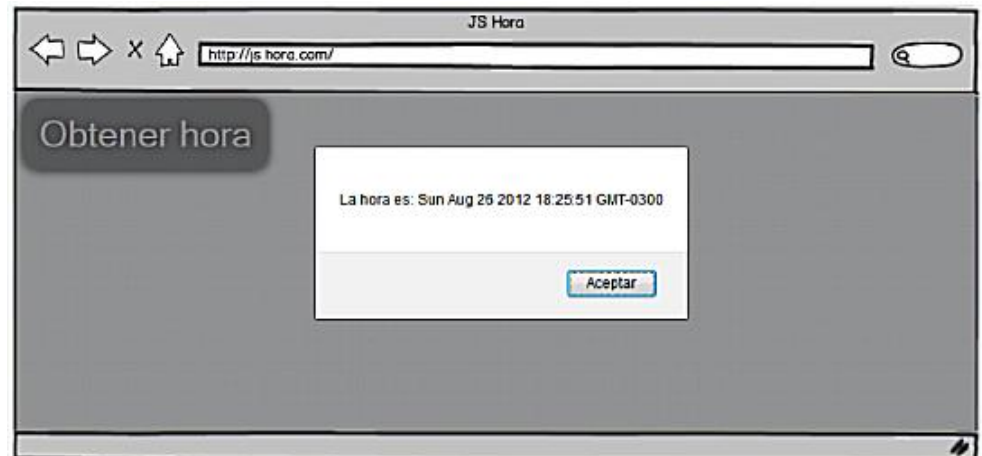
- El código JavaScript se utiliza también para manejar los eventos

```
<button id="botoncito">Fecha de hoy</button>
<script>
document.getElementById("botoncito").onclick = mostrarFecha;
function mostrarFecha() {
    document.getElementById("demo").innerHTML = Date();
}
</script>
```

- Ejemplos de eventos: *onclick*, *onload*, *onunload*, *onchange*, *onmouseover*, *onmouseout*, etc.

JavaScript - Ejemplo

```
<html>
<head>
  <script>
    function getHora() {
      hora = new Date();
      alert("La hora es: " + hora);
    }
  </script>
</head>
<body>
  <input type="submit" value="Obtener hora" onclick="getHora()" />
</body>
</html>
```



jQuery

- Es una librería de JavaScript.
- Permite simplificar la codificación de procedimientos habituales con una sintaxis más compacta.
- Es extensible mediante plugins.
- Brinda la siguientes características:
 - Manipulación de DOM
 - Manipulación de CSS
 - Manejo de eventos
 - Efectos y animaciones
 - AJAX

jQuery

- Utilizar los selectores CSS para realizar acciones sobre los elementos.
- Sintaxis:

`$(document)` – el documento

`$(this)` – el element actual

`$("tagName")` – todos los elementos `<tagName>`.

`$(".className")` – todos los elementos de `class="className"`.

`$("#id")` - el elemento con `id="id"`.

jQuery - Ejemplo

- Validación de datos de un formulario

```
<form id="submit_form"
action="submit_info" method="post"
onsubmit="return validate()">
```

Nombre:

Clave:

Debes ingresar la clave

☐ Recordarme

```
<script type="text/javascript">
function validate() {
    var correct = true;

    var name = $('#name').val();
    if (name == '') {
        $('#error_name').show();
        correct = false;
    } else
        $('#error_name').hide();

    var pass = $('#pass').val();
    if (pass == '') {
        $('#error_pass').show();
        correct = false;
    } else
        $('#error_pass').hide();

    return correct;
}
</script>
```

AJAX

- Asynchronous JavaScript And XML
- Permite que el browser haga pedidos en segundo plano, sin recargar la página ni cuando envía el pedido ni cuando recibe la respuesta.
- Una técnica que permiten el desarrollo de Single Page Application.
- Utiliza el objeto XMLHttpRequest que es proporcionado por los navegadores.
- Ej: Sugerencias de búsqueda (autocompletado)



taller de program	Buscar
taller de programacion fing	
taller de programacion	
taller de programacion 1	
taller de programacion web	
taller de programacion neurolinguistica	
taller de programacion y computo	
taller de programacion orientada a objetos	

AJAX - Ejemplo

- Crear objeto XMLHttpRequest

```
var xhttp = new XMLHttpRequest();
```

- Especificar el tipo de pedido (método HTTP, URL del script en el servidor, asíncrono o síncrono).

```
xhttp.open("GET", url, true);
```

- Enviar pedido

```
xhttp.send();
```

- Definir la función de callback (se ejecuta cuando llega la respuesta)

```
xhttp.onreadystatechange=procesarRespuesta;
```

AJAX - Ejemplo

- En la función de callback chequeamos que se haya recibido completamente la respuesta y que el código HTTP sea 200 (OK) antes de recuperar la respuesta del responseText.

```
function procesarRespuesta(){  
    if (xhttp.readyState == XMLHttpRequest.DONE  
    && xhttp.status == 200){  
        document.getElementById("spanID").innerHTML = xhttp.responseText;  
    }  
};
```

Bootstrap

- Conjunto de herramientas para facilitar el desarrollo de páginas Web.
- Incluye: CSS, fuentes, temas (templates) y extensiones JavaScript para realizar tareas comunes de desarrollo.
- Permite desarrollar páginas “responsivas” (Responsive Web Design)
- Mobile-first

Bootstrap – Algunos componentes

Botones



Navegación



Tablas

#	First Name	Last Name	Username
1	Walter	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@tbtst



Barras de progreso

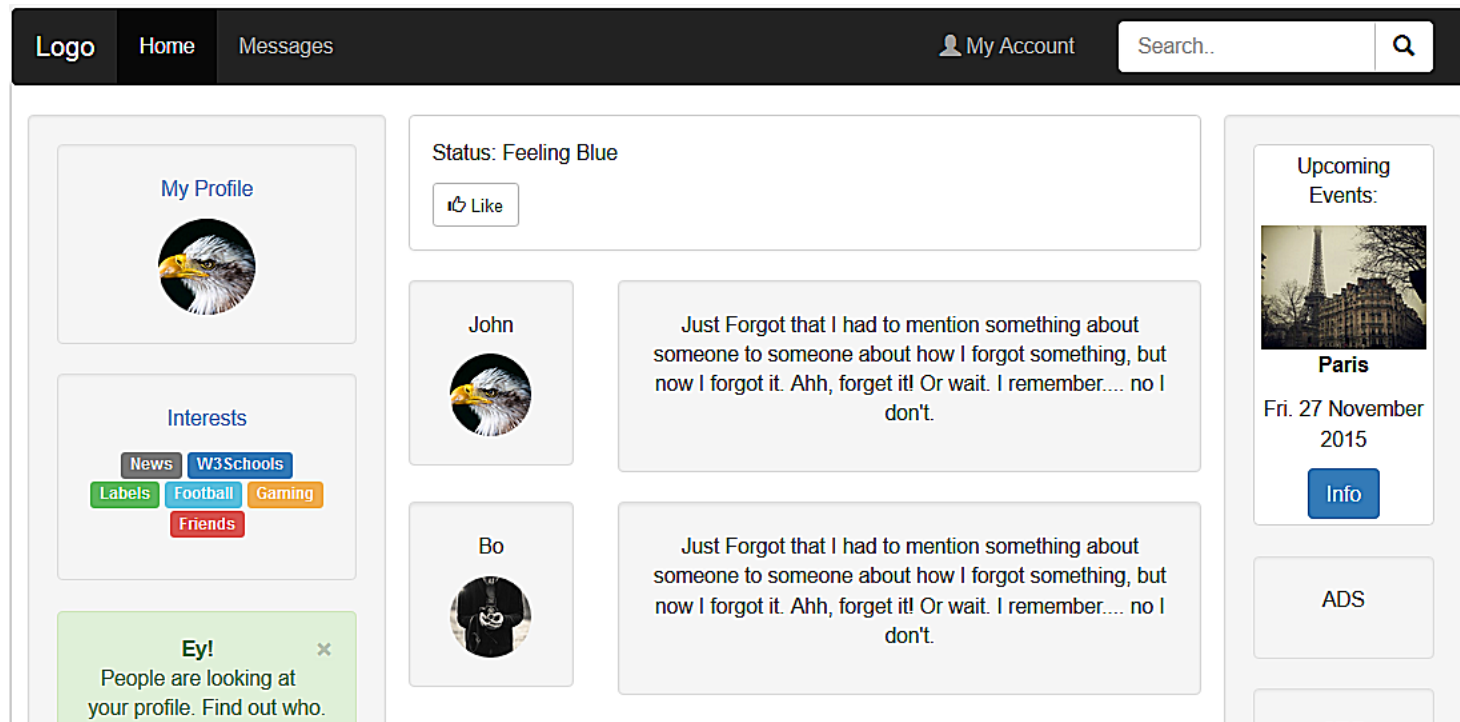


Menú desplegable



Bootstrap

- Templates



Demo

A series of horizontal lines in teal and light blue colors, some solid and some dashed, extending across the bottom of the slide.

Referencias (Parte 1 y 2)

- Redes de Computadores – Un enfoque descendente (J.F. Kurose & K. W. Ross)

<http://pearson.es/espa%C3%B1a/TiendaOnline/es-ebook-9788478291335->

- The Java EE 6 Tutorial (Oracle)

<http://docs.oracle.com/javaee/6/tutorial/doc/>

- Designing Enterprise Applications with the J2EE Platform (I. Singh et al.)

http://download.oracle.com/otn-pub/java/designing_enterprise_apps/2.0/designing_enterprise_apps-2_0-book.pdf?AuthParam=1505784380_6962734676a0d30fa873f2b2e708fc7aTM

- W3Schools Tutorials

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css>

https://www.w3schools.com/xml/dom_intro.asp

<https://www.w3schools.com/Js/>

<https://www.w3schools.com/jquery>

https://www.w3schools.com/js/js_ajax_intro.asp

<https://www.w3schools.com/bootstrap>