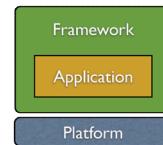


Temas de Hoy



Evoluciones

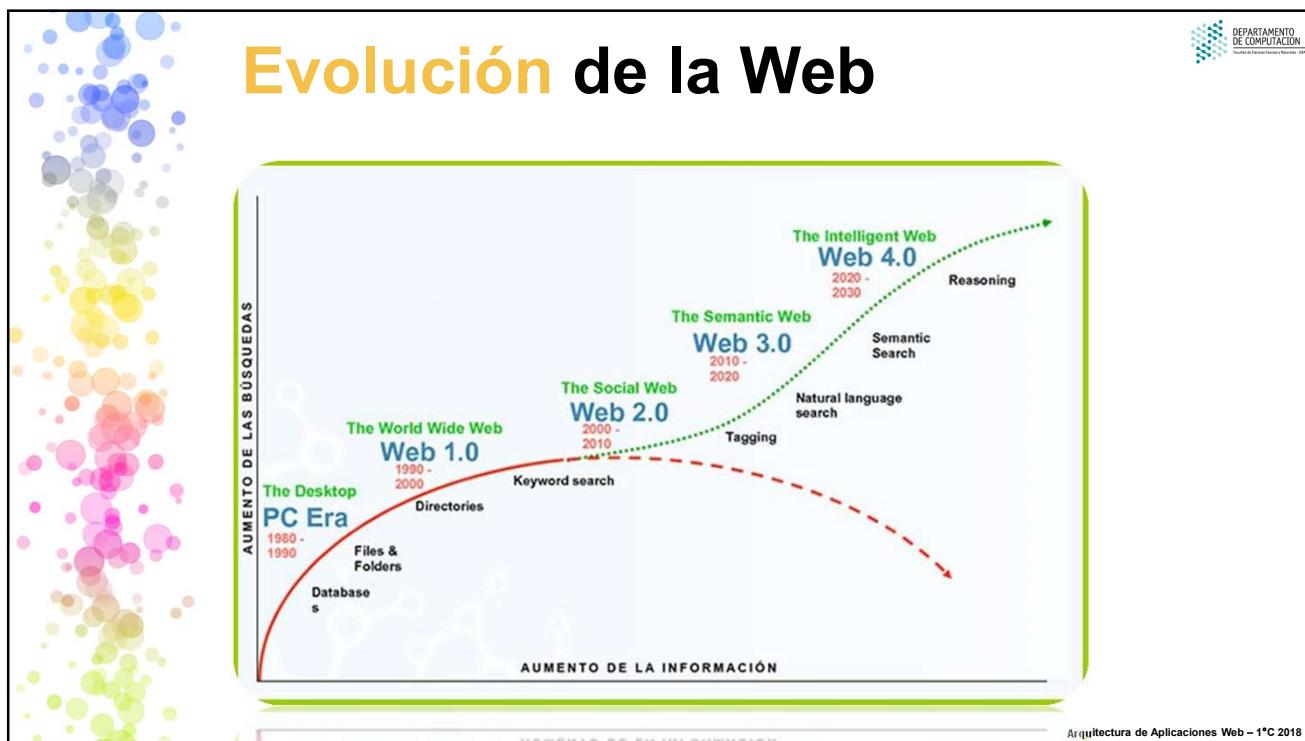
Frameworks

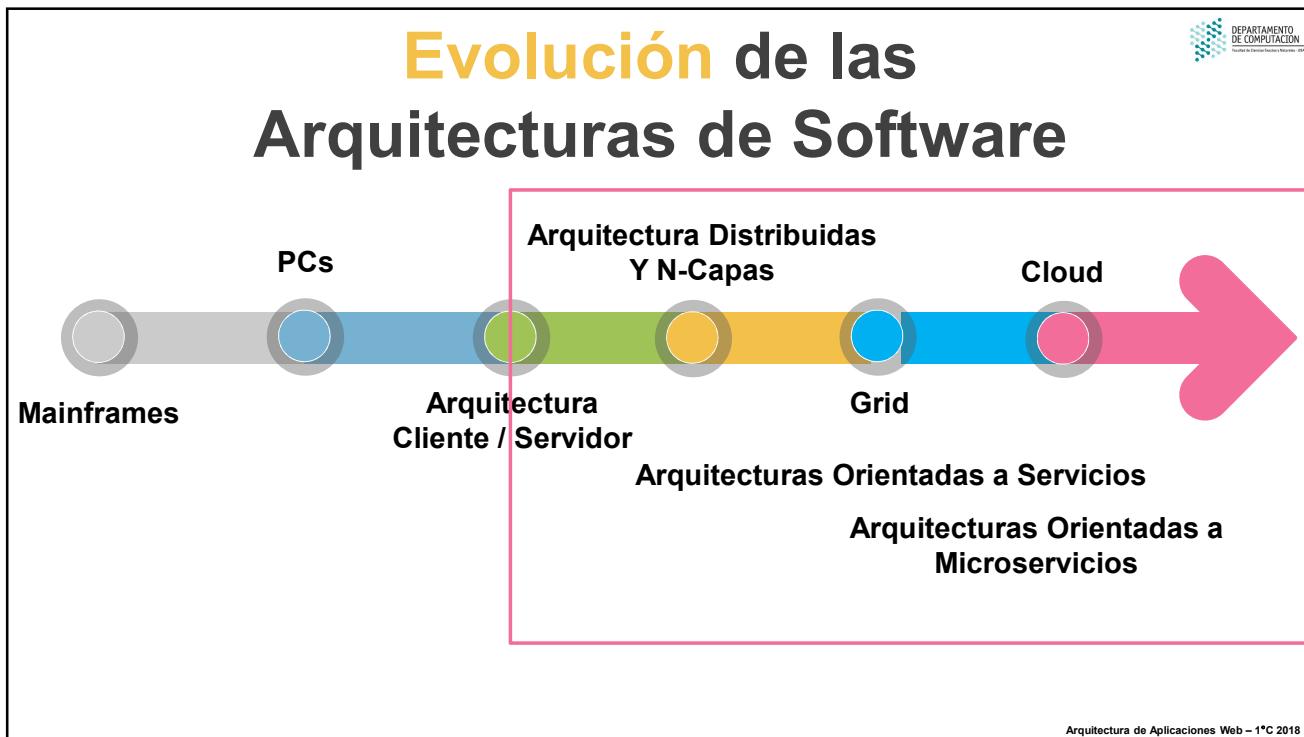
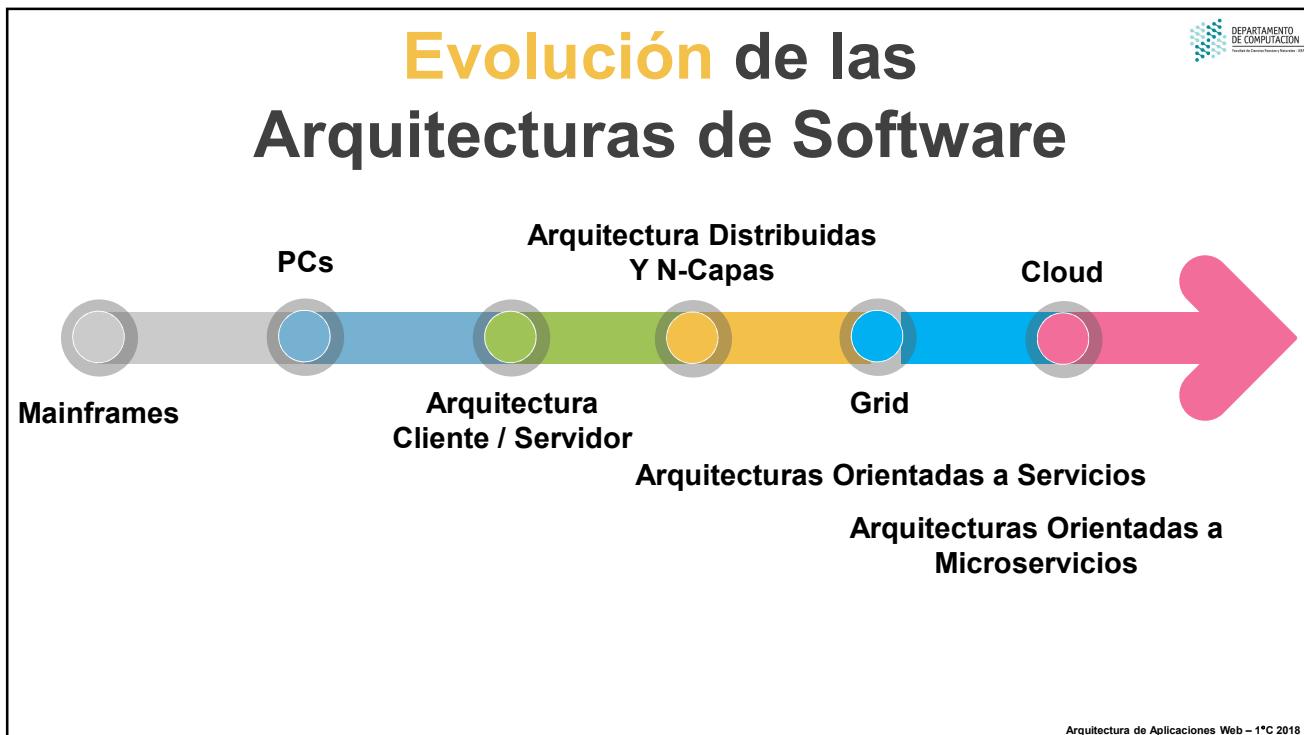


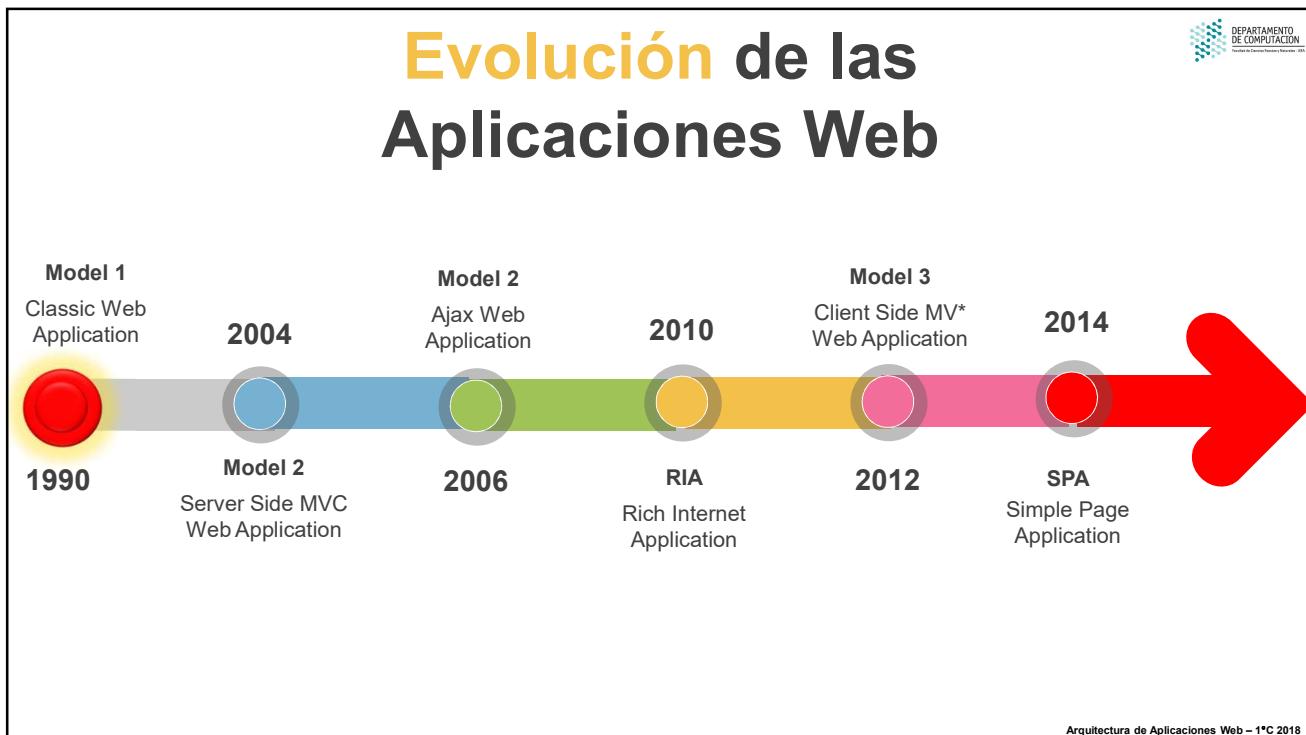
Arquitectura de Aplicaciones Web – 1*C 2018

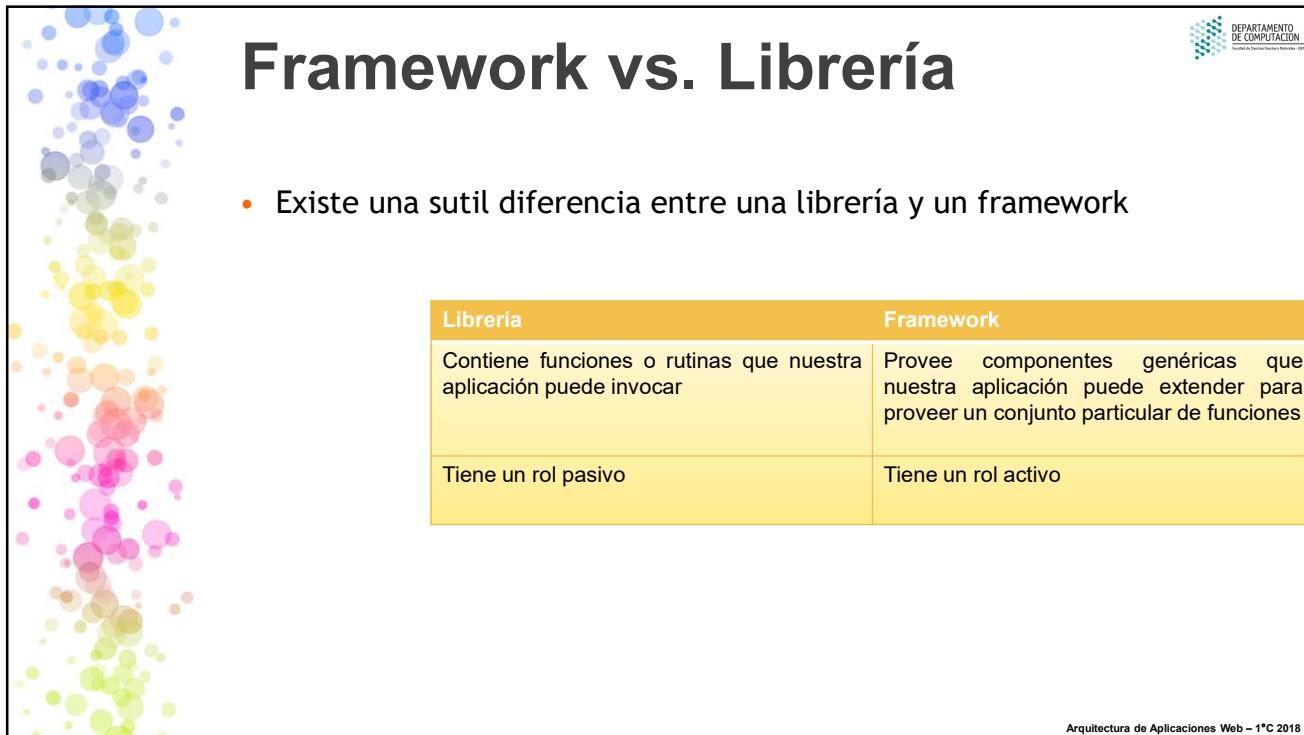
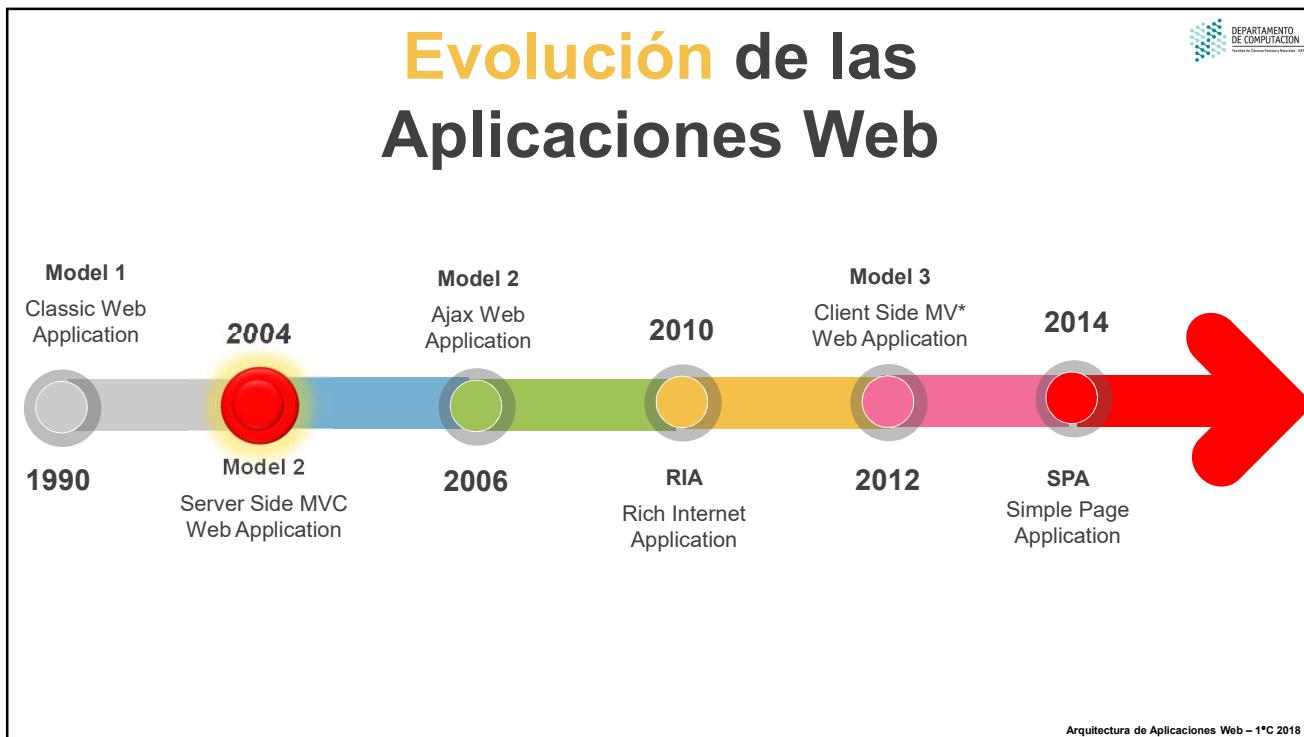
PREVIOUSLY...

Retomando...

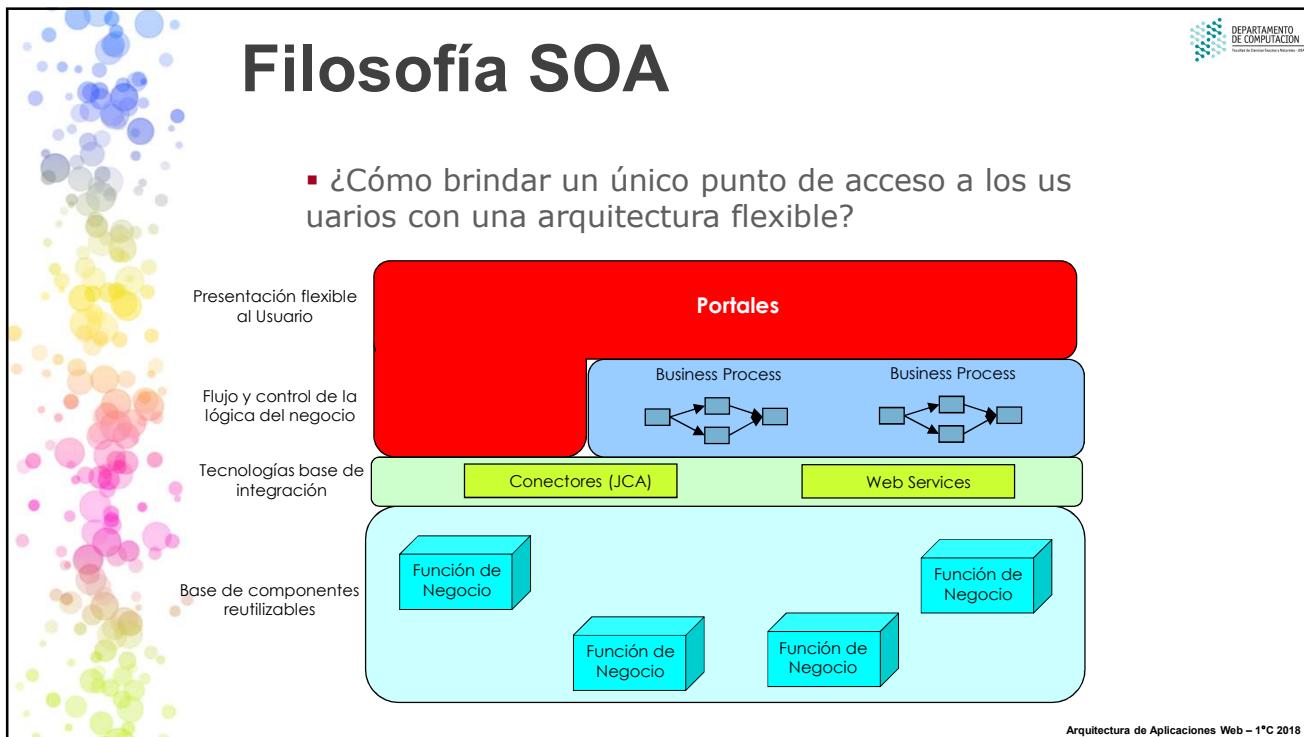


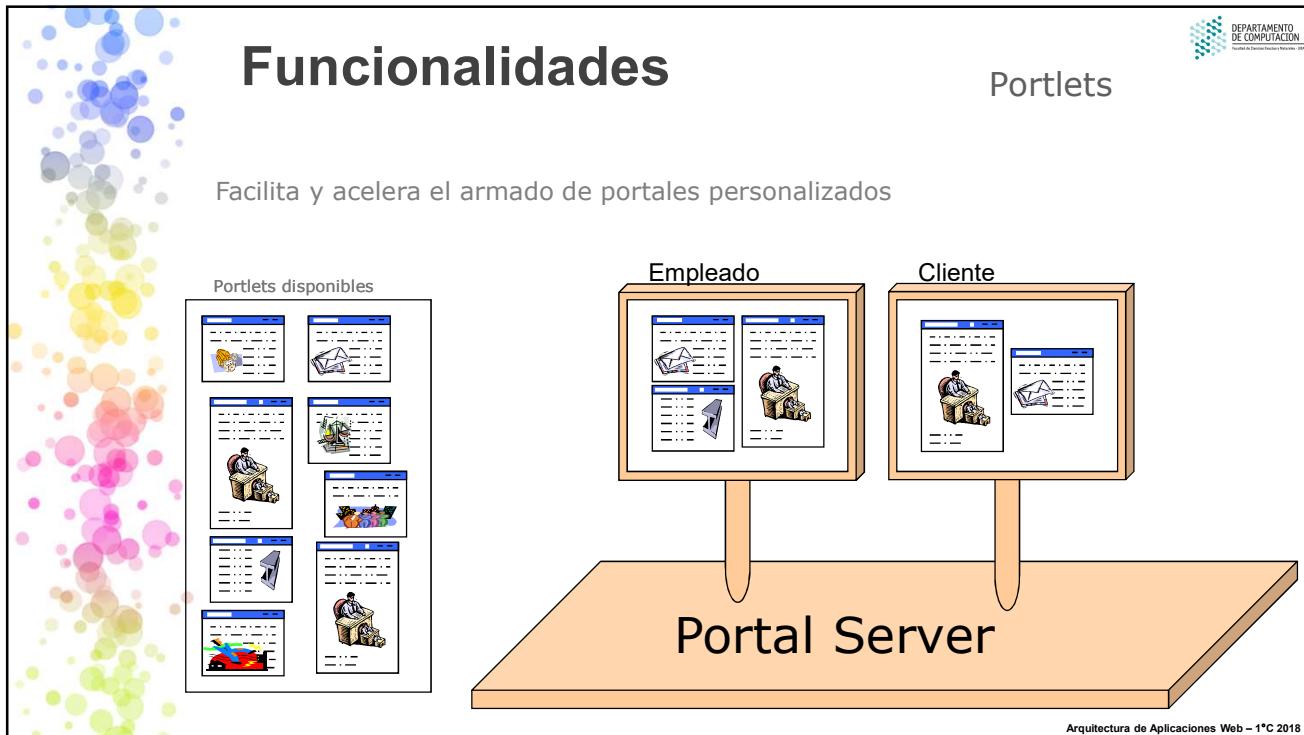
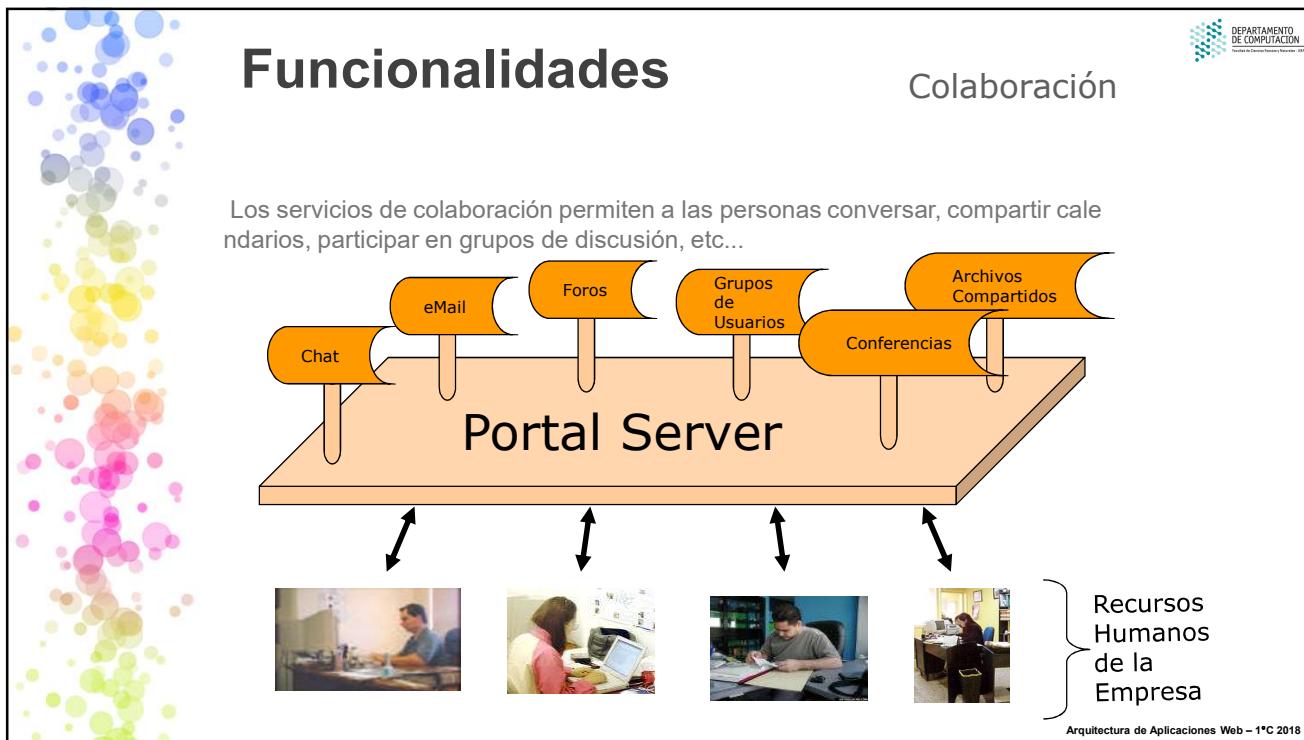


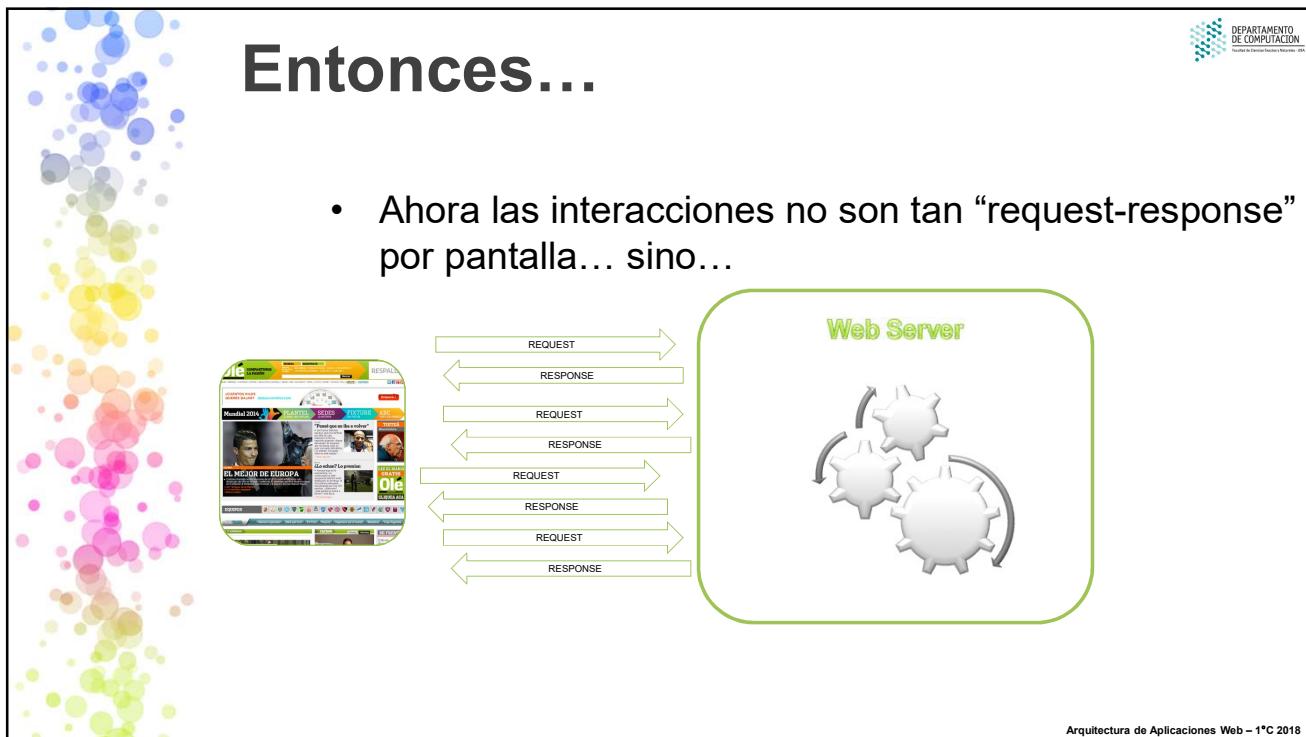
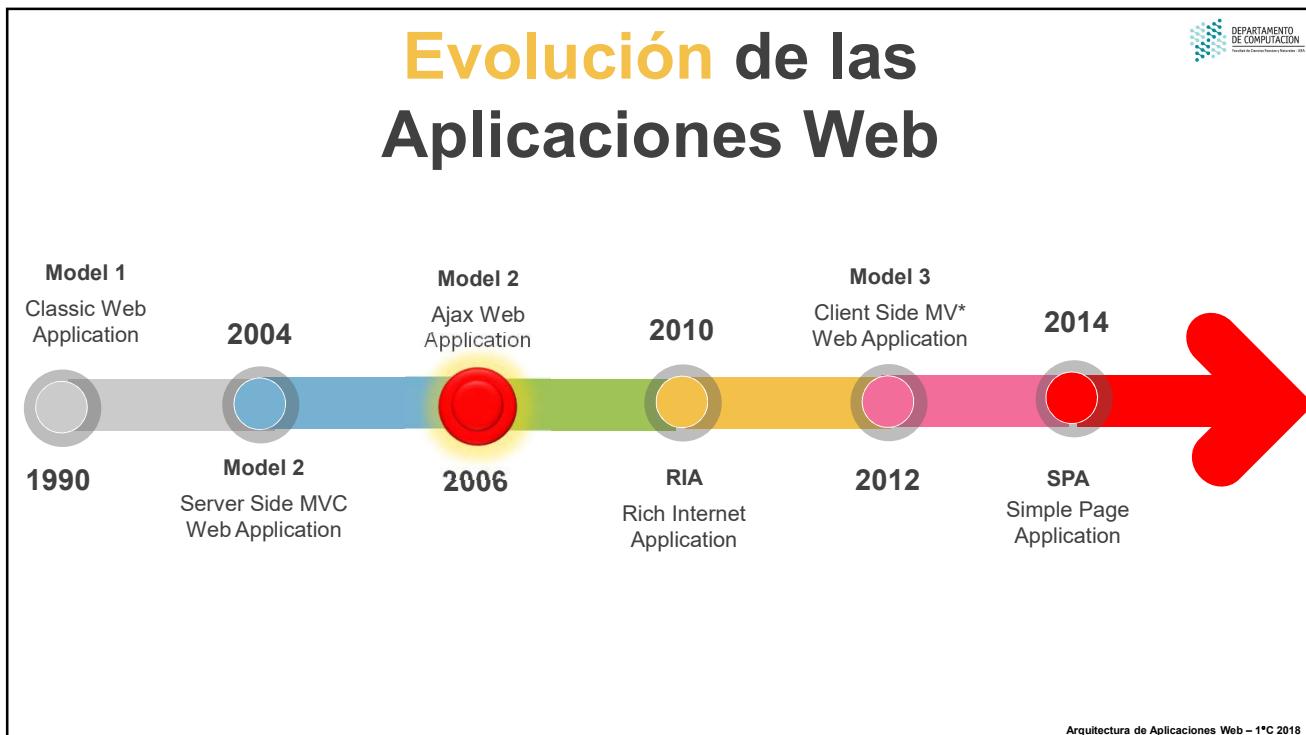












Otro detalle...

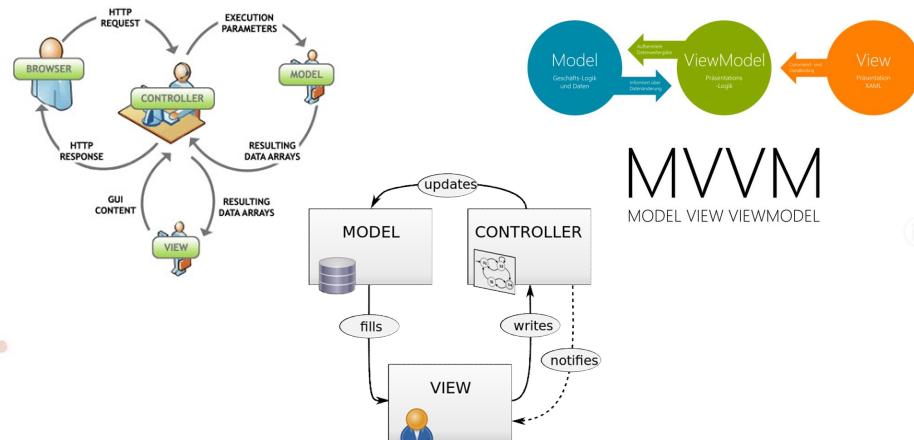
- Frameworks “server side” suman Ajax como elemento *First Class*



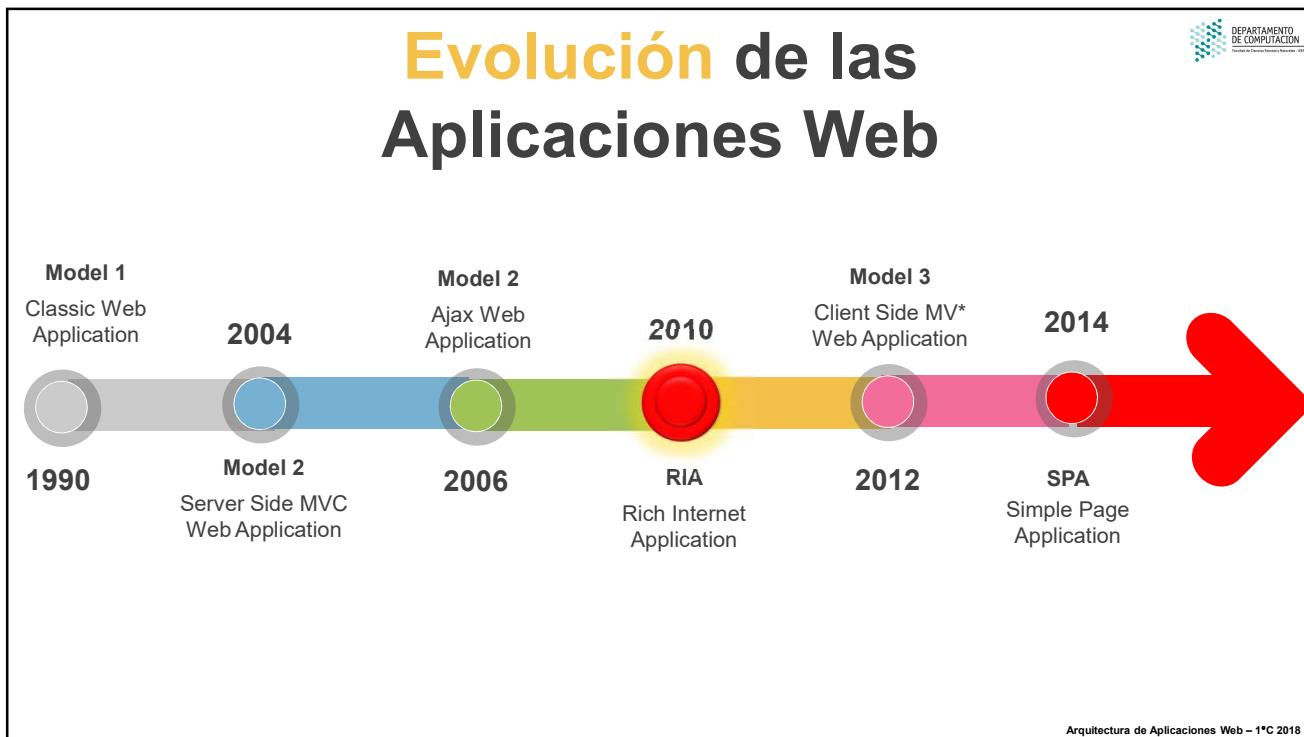
Arquitectura de Aplicaciones Web – 1*C 2018

Y más MV*...

- Model View Controller y derivados...



Arquitectura de Aplicaciones Web – 1*C 2018



Mientras tanto...

- Las Rich Internet Applications (RIA) son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales.
- Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales.

The slide features a decorative background on the left side consisting of a vertical column of colorful, semi-transparent circular bubbles in shades of blue, yellow, pink, and green. On the right side, there is a collection of logos for various Rich Internet Application frameworks and technologies, including Adobe AIR, Mozilla Prism, Fx (Firefox), Microsoft Silverlight, AJAX (Asynchronous JavaScript And XML), OpenLaszlo, and a coffee cup icon.

Arquitectura de Aplicaciones Web – 1*C 2018

Desktop vs Web



Arquitectura de Aplicaciones Web – 1*C 2018

¿Por que usar Flex?

- Multiplataforma
 - Las aplicaciones Flex corren sobre cualquier browser que soporte Flash Player
 - También corren en el escritorio utilizando Adobe AIR.
 - Adobe AIR puede acceder a datos locales y recursos del sistema en el escritorio.
- Interfaz interactiva para el usuario
 - Framework construido sobre Flash Player para brindar una experiencia única al usuario.

Arquitectura de Aplicaciones Web – 1*C 2018

Tecnología de Servidor

- ASP.NET
- ColdFusion
- J2EE
- PHP
- Ruby
- Perl

Arquitectura de Aplicaciones Web – 1ºC 2018

Componentes que integran Flex

- ActionScript 3.0
- Flash Player 9
- Flex Framework (SDK)
- Flex Builder 3
- MXML (Multimedia eXtended Markup Language)
- E4X (ECMAScript for XML)
 - ECMAScript 4 (Abandonado)
 - ECMAScript 6 → Typescript
 - ECMAScript 7 en progreso

Arquitectura de Aplicaciones Web – 1ºC 2018

¿Que es MXML?

- Adobe Flex 3
- ActionScript 3.0
- **MXML**
- HTML
- JavaScript
- HTML

Arquitectura de Aplicaciones Web – 1ºC 2018

ActionScript

- Similar a Java, C#.
- Orientado a objetos.
- Clases, interfaces, funciones anónimas.
- Eventos.

Arquitectura de Aplicaciones Web – 1ºC 2018

ActionScript: Eventos

- Flex (y por lo tanto ActionScript) es un lenguaje orientado a eventos.
- El cliente se subscribe al evento que quiere atender.
- Cuando el usuario desencadena una acción que tiene algún subscriptor, este es invocado para realizar alguna tarea deseada. (El subscriptor es mejor conocido como Event Handler)

Arquitectura de Aplicaciones Web – 1ºC 2018

Data providers

- Varios controles utilizan proveedores de datos para mostrar resultados.
- XML es un tipo de data provider. (También está ArrayCollection, Array).
- Controles que usan proveedores de datos:
 - DataGrid
 - List
 - Menu
 - Tree

Arquitectura de Aplicaciones Web – 1ºC 2018

Comunicación

- HTTPService es una de las formas de comunicarse con un servidor de datos.
- Se puede transmitir XML (de hecho cualquier stream de datos)
- Hay otras formas: WebServices, RemoteObjects.
- Acá vamos a ver HTTPService.

Arquitectura de Aplicaciones Web – 1*C 2018

HTTPService

- La clase HTTPService permite interactuar con un Web Server.
- Usualmente se utiliza con XML como objeto de transporte.
- El Web Server puede ser desarrollado con cualquier tecnología que permita crear XML. (Java, ASP.NET, PHP, Ruby, Python, Perl, etc).

Arquitectura de Aplicaciones Web – 1*C 2018

Backend

- XML via HTTP/HTTPS requests.
- Web Services.
- POJO
 - JavaBeans
 - DAO
 - Hibernate
 - EJBs
- ColdFusion Remote Objects
- Flex Data Services 2

Arquitectura de Aplicaciones Web – 1ºC 2018

Componentes

- Funcionalidad repetida en la interfaz de usuario sugiere Flex Components.
- Piezas de código creadas por el usuario para crear componentes que interactúan con el usuario más pequeños.
- Se puede crear a partir de cualquier componente Flex: Button, Canvas, TextInput, Panel, etc.

Arquitectura de Aplicaciones Web – 1ºC 2018



Componentes

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
<Application>
  <Script>
    function initStartPanel() {...}
    function initDetailPanel() {...}
  </Script>
  <HBox>
    <Panel id="startPanel" creationComplete="initStartPanel()">
      ...
    </Panel>
    <Panel id="detailPanel" creationComplete="initDetailPanel()">
      ...
    </Panel>
  </HBox>
</Application>
```

Arquitectura de Aplicaciones Web – 1ºC 2018



Componentes

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

MyApp.mxml

```
<Application>
  <HBox>
    <StartPanel id="startPanel"/>
    <DetailPanel id="detailPanel"/>
  </HBox>
</Application>
```

StartPanel.mxml

```
<Panel creationComplete="initStartPanel()">
  <Script>
    function initStartPanel() {...}
  </Script>
  ...
</Panel>
```

Arquitectura de Aplicaciones Web – 1ºC 2018

States

```
<Application initialize="doInit()">
  <states>
    <State name="small">
      < SetProperty target="{myPanel}" property="height" value="200"/>
      < SetProperty target="{myPanel}" property="width" value="300"/>
    </State>
  </states>
  <Style source="style.css"/>
  <Script source="Step5_code.as"/>
  <Panel id="myPanel" layout="absolute" width="80%" height="80%">
    <TextArea text="Hello, world!" top="10" bottom="71" left="10" right="30"/>
    <Button id="goodbyeButton" label="goodbye" right="30" bottom="41"/>
    <Button id="smallButton" label="small" left="10" bottom="41"/>
  </Panel>
</Application>
```

Arquitectura de Aplicaciones Web – 1*C 2018

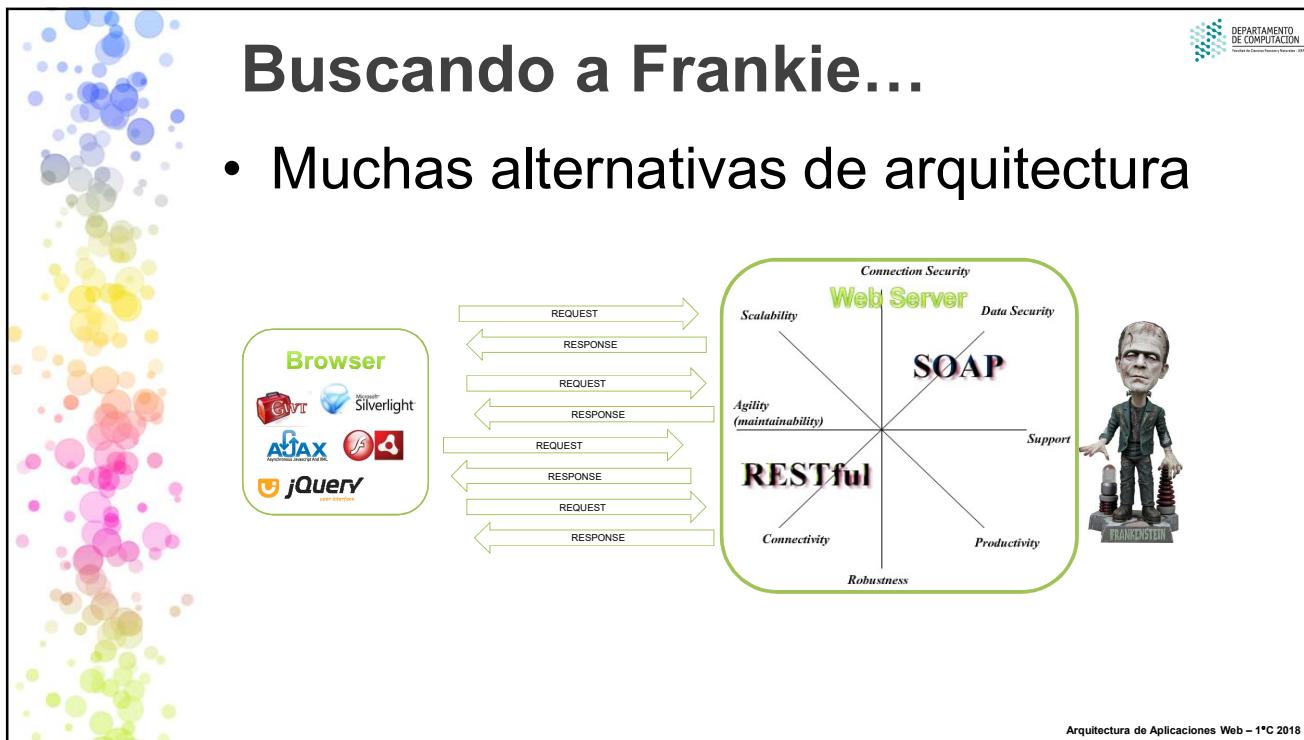
States

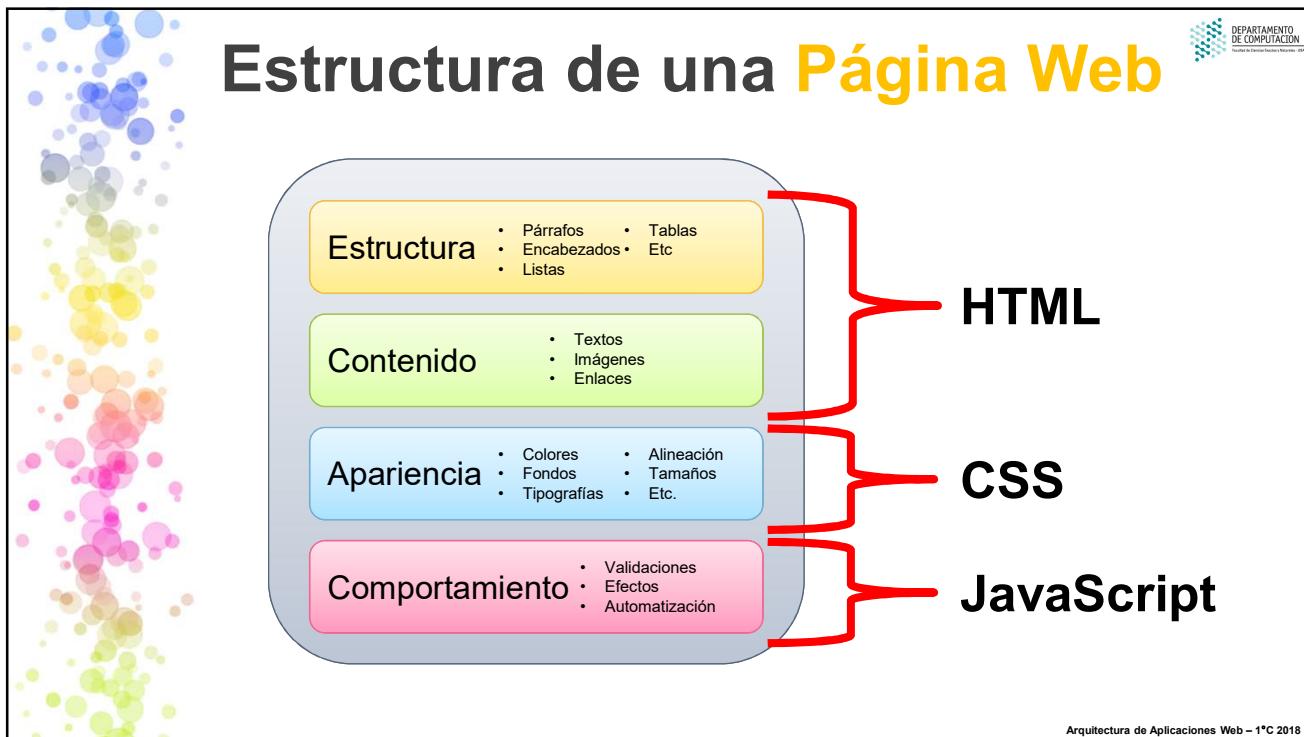
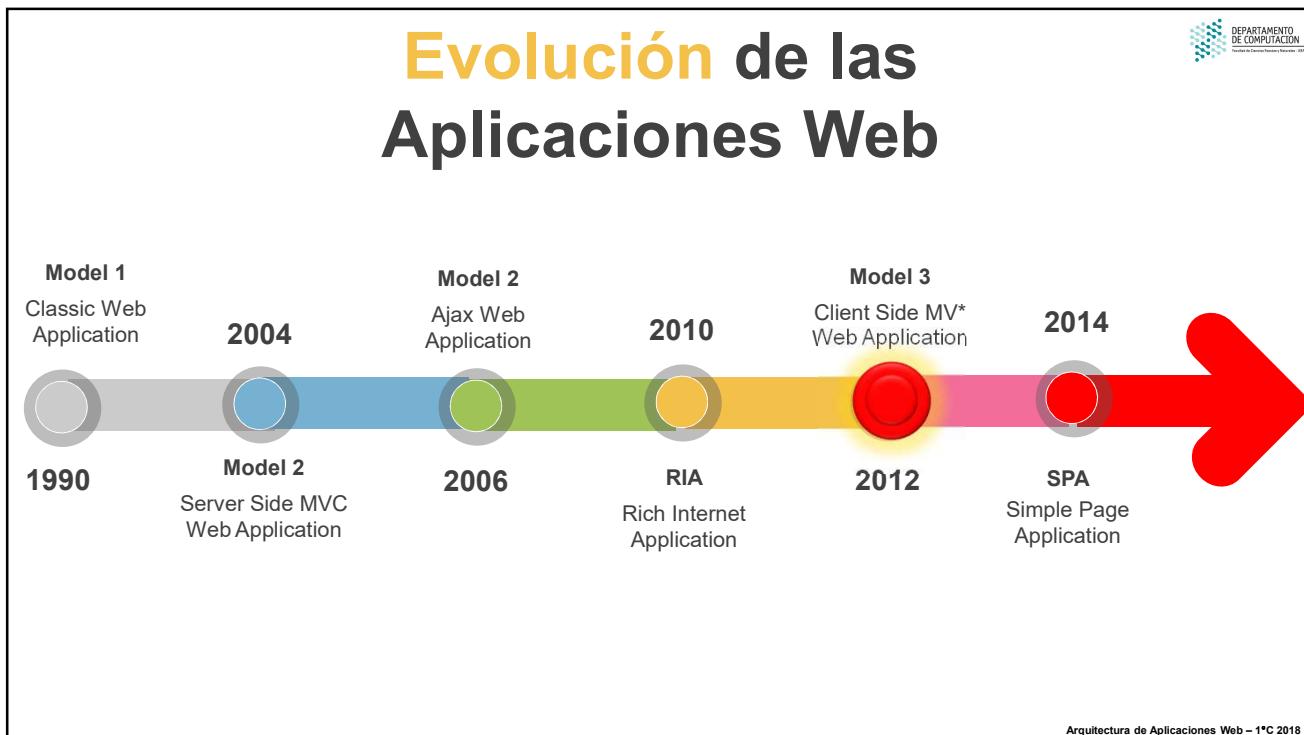
```
public function doInit() : void
{
  goodbyeButton.addEventListener("click", goodbye);
  smallButton.addEventListener("click", makeSmall);
}

public function goodbye(event: Event) : void
{
  myPanel.visible = false;
}

public function makeSmall(event: Event) : void
{
  currentState = "small";
}
```

Arquitectura de Aplicaciones Web – 1*C 2018

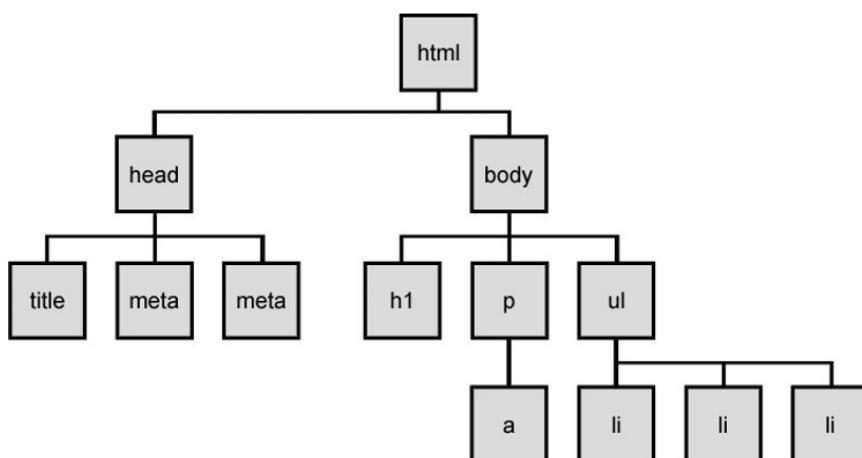




DOM

Cuando un navegador carga y despliega al usuario una página web cualquiera, lo que hace internamente es interpretar todo el HTML de la página web y traducir a una estructura de datos, donde el documento HTML queda representado en un estructura de tipo árbol, esto se denomina DOM (Document Object Model).

DOM



CSS



- Cómo se define un estilo en css:

Declaración

Selector { propiedad: valor; ...}

Ejemplo:

H1 {color: #CC9900; }

Arquitectura de Aplicaciones Web – 1*C 2018

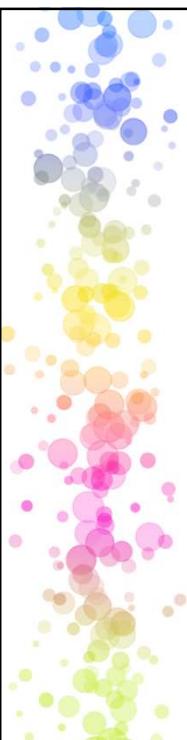
Selectores



- Tipos de selectores:
 - *De elemento HTML*
 - h1, table, div, span...
 - *De identificador*
 - Todos los elementos HTML cuya propiedad "id" tenga un determinado valor, tendrán ese estilo.
 - *De clase*
 - Todos los elementos HTML cuya propiedad "class" tenga un determinado valor tendrán ese estilo.

Arquitectura de Aplicaciones Web – 1*C 2018

Selectores





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Anatomía básica CSS

Ejemplo:0 Selector Declaración
`E {...}`

Ejemplo:1 Selector Propiedad1 Valor1 Propiedad2 Valor2
`p {color: red, background: #FF6600}`

Ejemplo:2 Camino + Elemento = Selector Declaración
`div p a {text-decoration: none}`

Ejemplo:3 almohadilla identación almohadilla de clase condicionante
`div#contenido h2.titulos:hover {...}`

Arquitectura de Aplicaciones Web – 1ºC 2018

JavaScript





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- ❑ Inicialmente se llamaba LiveScript.
- ❑ Con el respaldo de SUN en 1995 pasó a llamarse JavaScript.
- ❑ Es un lenguaje interpretado.
- ❑ Javascript se integra en HTML de dos maneras:
 - ❑ Como fichero *.js incluido en la misma sección que los CSS.
 - ❑ Como código directamente en el fichero HTML haciendo uso de la etiqueta <script>

Arquitectura de Aplicaciones Web – 1ºC 2018

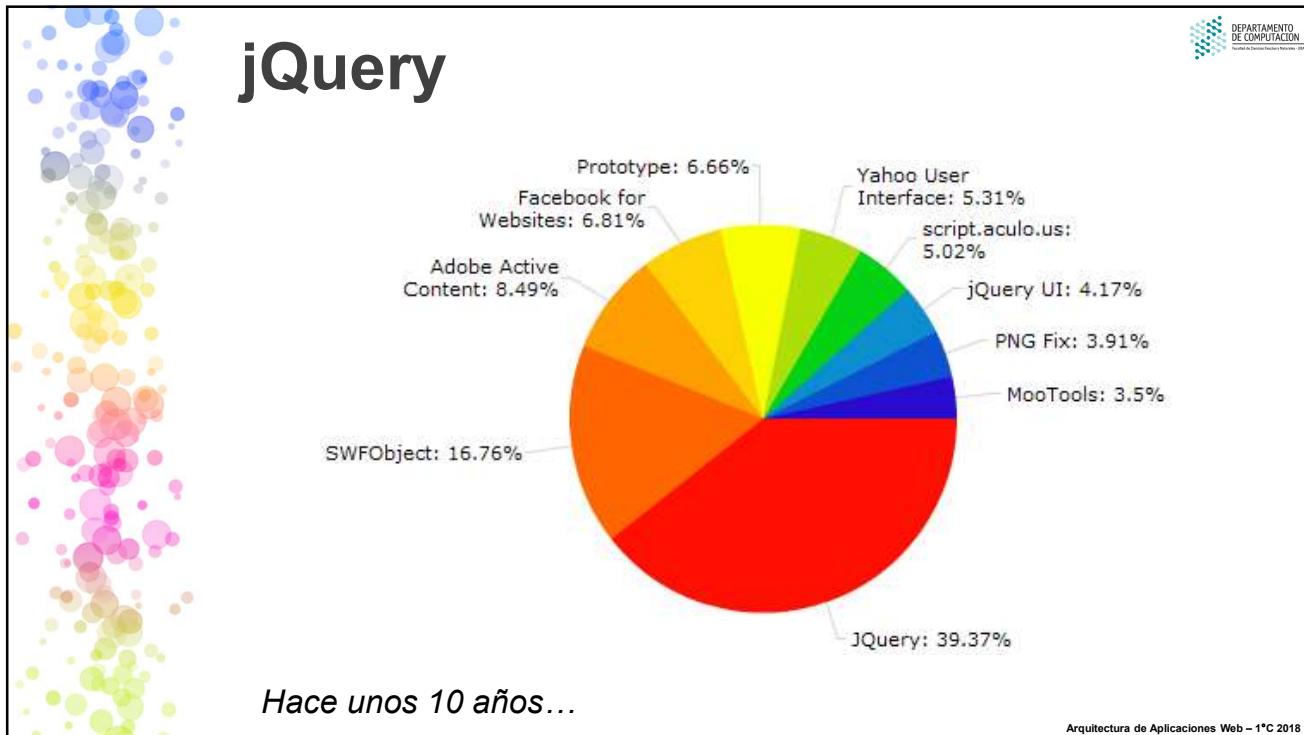
jQuery

Es una librería JavaScript Open-Source, que simplifica el desarrollo Client-Side, teniendo consideraciones cross-browser.

- Animations
- Manipular el DOM
- AJAX
- Extensible mediante plugins

 write less, do more.

Arquitectura de Aplicaciones Web – 1*C 2018



Volviendo al DOM...

- Resumiento... el DOM representa el documento desde el <!DOCTYPE> hasta el </html>
- Se *carga* de arriba hacia abajo.
- El DOM estará *ready* cuando todo el contenido de la página se haya cargado...
 - *Estilos*
 - *Javascript*
 - *Imagenes*

Arquitectura de Aplicaciones Web – 1ºC 2018

Entonces...

- Si con jQuery queremos manipular el DOM... tenemos que estar seguros que fue cargado:

```
$ (document) .ready(function () {  
    // inserte su código jQuery aqui...  
});
```

Arquitectura de Aplicaciones Web – 1ºC 2018

jQuery en acción

Html:

```
<p>Hola Mundo</p>
```

Script:

```
$(function() {  
    $("p").addClass("isCool");  
    ...  
});
```

Resultado:

```
<p class="isCool">Hola Mundo</p>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Selectores para tod@s

- Similar a CSS

```
$("p")
```

```
<p>
```

```
$("div")
```

```
<div>
```

```
$("#foo")
```

```
id="foo"
```

```
$(".foo")
```

```
class="foo"
```

Arquitectura de Aplicaciones Web – 1*C 2018

¿Qué podemos hacer?

- jQuery:

```
$ ("p") .addClass ("ejemplo");
```

- Antes:

```
<p>
```

- Despues:

```
<p class="ejemplo">
```

Arquitectura de Aplicaciones Web – 1*C 2018

¿Qué podemos hacer?

- jQuery:

```
$ ("p") .removeClass ("ejemplo");
```

- Antes:

```
<p class="ejemplo">
```

- Despues:

```
<p>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Ocultar / Mostrar

- **jQuery:**

```
$ ("div") .show () ;
```

- **Antes:**

```
<div style="display:none;">
```

- **Después:**

```
<div style="display:block;">
```

Arquitectura de Aplicaciones Web – 1*C 2018

Escribir

- **jQuery:**

```
$ ("#ingreso") .text ("Válido") ;
```

- **Antes:**

```
<p id="ingreso">Inválido</p>
```

- **Después:**

```
<p id="ingreso">Válido</p>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Encadenamiento...



```
$ ("p")
    .addClass("ejemplo")
    .text("Hola Mundo")
    .show();
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Eventos...



```
$("#clickeame").click(function() {
    $(this).text("Hola");
    alert("Gracias por hacer click!");
});
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Métodos básicos

.show ()	• Muestra un elemento oculto
.wrap ("<a>")	• Encierra un elemento con <a>
.parent ("p")	• Selecciona el parent <p>
.html ()	• Get/Set innerHTML
.val ()	• Get/Set Valor

Arquitectura de Aplicaciones Web – 1*C 2018

Podríamos seguir con jQuery varios días...

- jQuery Effect
- jQuery Attributes
- jQuery Plugins (UI Library)
- jQuery Selectors
- jQuery Traversing
- jQuery Manipulations
- jQuery CSS
- jQuery Effect
- jQuery AJAX

Arquitectura de Aplicaciones Web – 1*C 2018

Deja Vú



- Alcanza con Ajax?
- Alcanza con jQuery?
- Qué falta / qué tenemos?

Arquitectura de Aplicaciones Web – 1ºC 2018

Evolución



- La unión hace la fuerza...



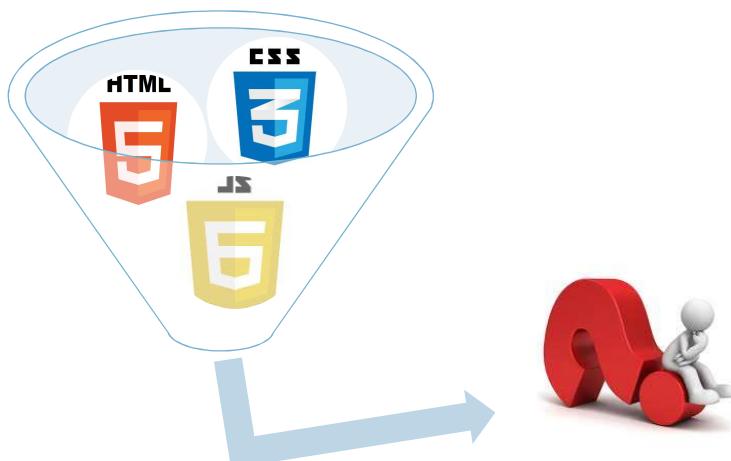
Arquitectura de Aplicaciones Web – 1ºC 2018

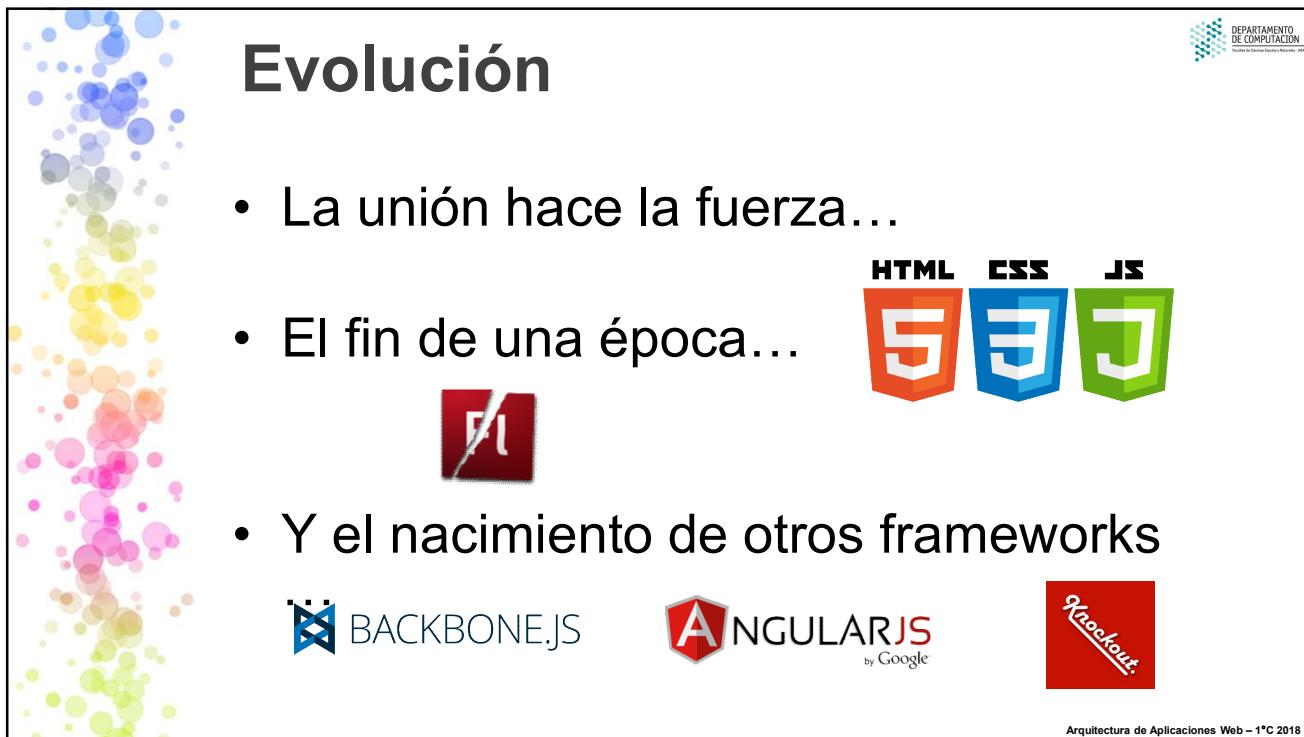
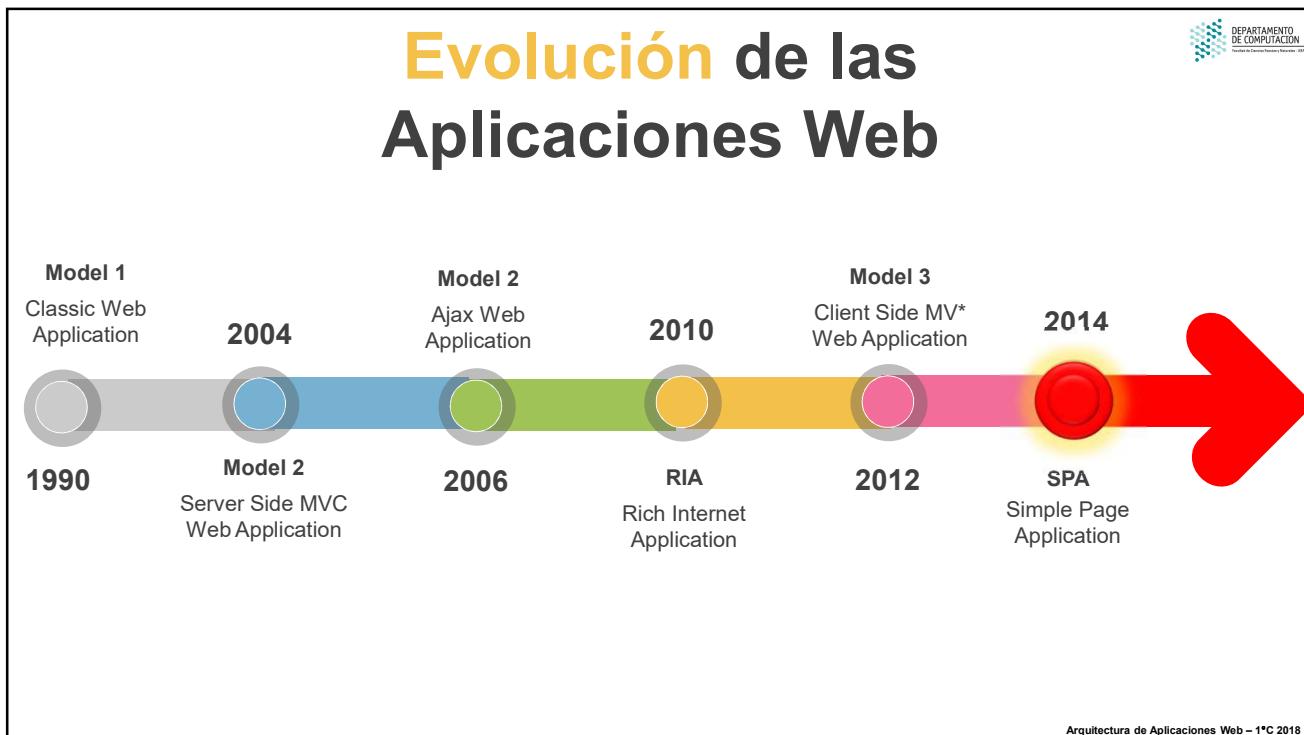
Evolución

- La unión hace la fuerza...
- El fin de una época...



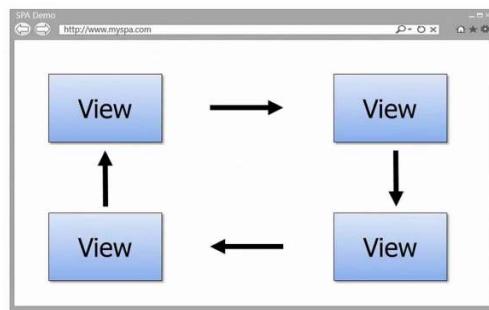
Y ahora...¿listo?





Single Page Applications

- En una SPA el usuario no navega por un sistema de enlaces tradicionales si no que en su lugar, mediante el uso de JavaScript, Ajax, HTML5 o una combinación de las anteriores, se actualiza lo que el usuario ve siempre desde la misma página (sin cambiar de URL ni refrescar el contenido entero).



Arquitectura de Aplicaciones Web – 1ºC 2018

Single Page Applications

- Aplicaciones web que se ejecutan en una única página, logrando así una experiencia de usuario más cercana a una aplicación de escritorio.

Arquitectura de Aplicaciones Web – 1ºC 2018

Próxima parada...



HTML5 – CSS3 - Javascript

Arquitectura de Aplicaciones Web – 1*C 2018

HTM.. Qué?



- **HTML:** siglas de **HyperText Markup Language** (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web



WIKIPEDIA
La enciclopedia libre

Aplicaciones Web – 1*C 2018



Paso a paso...



- 1991 HTML
- 1994 HTML 2
- 1996 CSS 1 + JavaScript
- 1997 HTML 4
- 1998 CSS 2
- 2000 XHTML 1
- 2002** Tableless Web Design
- 2005** AJAX
- 2009** HTML 5

Arquitectura de Aplicaciones Web – 1*C 2018



HTML5

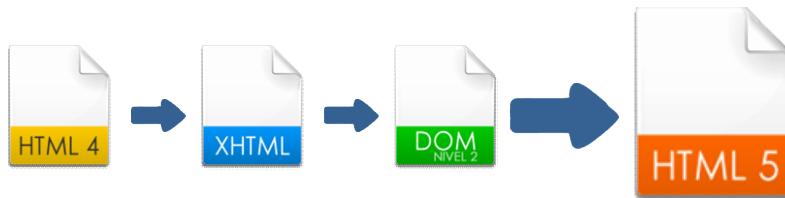


- 2004 Apple, Mozilla & Opera no consiguieron crear un estándar HTML4. En consecuencia forman la Web **WHATWG**
- 2005 Se publica el borrador **Web Applications 1.0**
- 2007 W3C “adopta” a WHATWG y **publica el borrador HTML5**
- 2009 **Finalización del borrador**
- Gracias a Google, Apple y los dispositivos móviles **HTML5 surge con fuerza**
- 2012 Primer Release Candidate de HTML5 [previsión W3C]
- 2020 Finalización de **test** [previsión W3C]
- 2022 Creación del **estándar HTML5** [previsión W3C]

Arquitectura de Aplicaciones Web – 1*C 2018

Qué es HTML 5?

- **HTML 5** es una nueva versión de HTML que proporciona mecanismos para simplificar el trabajo y facilitar la inclusión de elementos multimedia.



Arquitectura de Aplicaciones Web – 1ºC 2018

Nuevas etiquetas de presentación.

<div id="header">	<header>
<div id="menu">	<nav>
<div>	<article>
<div>	<section>
</div>	</article>
<div id="footer">	<aside>

Arquitectura de Aplicaciones Web – 1ºC 2018

Estandarización



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

video

geolocalización

drag & drop

audio

canvas

Arquitectura de Aplicaciones Web – 1ºC 2018

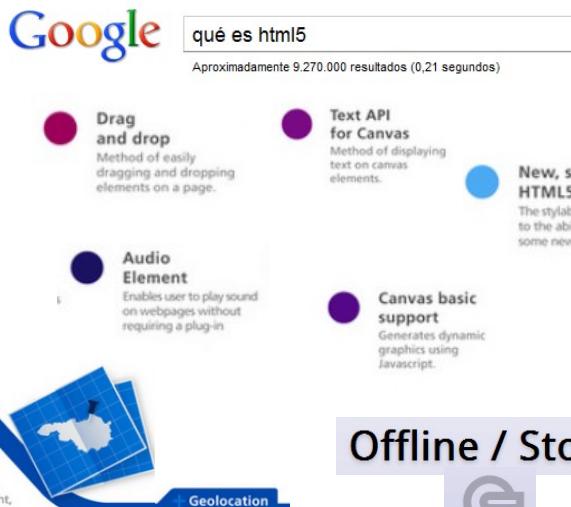
Estandarización



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Arquitectura de Aplicaciones Web – 1ºC 2018

No todo lo que brilla es HTML5...





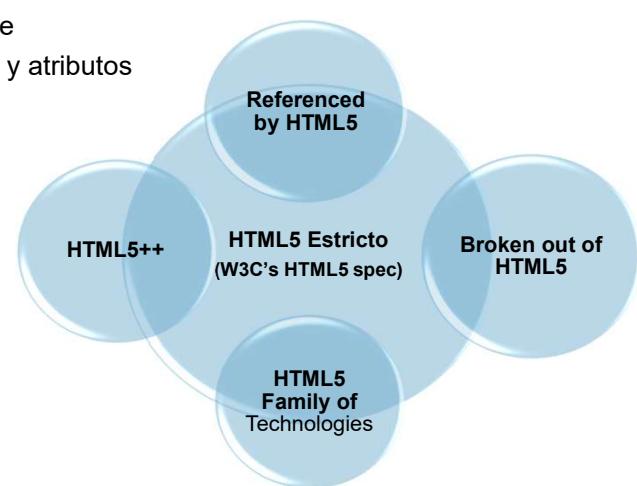




Arquitectura de Aplicaciones Web – 1ºC 2018

HTML5 entre nos...

- Sintaxis más simple
- Nuevos elementos y atributos
- Media embebida
- Canvas
- Storage offline
- Drag and Drop
- Geolocalización



Arquitectura de Aplicaciones Web – 1ºC 2018



HTML 5

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Referenced	Broken out	Family of Technologies	HTML5++
MathML	Canvas	CSS3	WebGL
SVG	WebSockets	Animations	O3D
	LocalStorage	Transforms	Firefox Audio APIs
	Device element	Transitions	XBL 2.0
	Ping Attribute	Gradients	
	Channel Messaging	W3C Geolocation	
	DataGrid	WebWorkers	
	SQL Storage	GPU acceleration of HTML	
	P2P connections	Canvas	
	Ajax Session History	SVG	
		CSS3	
		Faster Javascript	

Arquitectura de Aplicaciones Web – 1*C 2018



Un mundo ideal

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

SOPORTE HTML5 POR NAVEGADORES					
New Semantic tags	4.0*	3.6*	10.1*	4.0*	✗
Canvas	4.0*	3.5*	9*	3.0*	✗
Video element	4.0	3.0	10.5	3.2	✗
localStorage	4.0	3.0	10.5	4.0	8
Offline apps	4.0	3.6	✗	4.0	8*
HTML5 Forms	4.0*	✗	10.1*	5.0*	✗
Drag & drop	4.0	3.5	✗	4.0	7*

Arquitectura de Aplicaciones Web – 1*C 2018

Simplificación



DOCTYPE

XHTML 1.0	HTML5
-----------	-------

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

→

<!DOCTYPE html>



Arquitectura de Aplicaciones Web – 1ºC 2018



Simplificación



META

HTML 4.01	HTML5
-----------	-------

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

→

<meta charset="UTF-8">



Arquitectura de Aplicaciones Web – 1ºC 2018



Simplificación



SCRIPT

HTML 4.01	HTML5
<pre><script type="text/javascript" src="file.js"> </script></pre>	<pre><script src="file.js"></script></pre>
<pre><script type="text/javascript"> </script></pre>	<pre><script> </script></pre>



Arquitectura de Aplicaciones Web – 1ºC 2018

Simplificación



HOJAS DE ESTILO

HTML 4.01	HTML5
<pre><link rel="stylesheet" type="text/css" href="estilos.css"></pre>	<pre><link rel="stylesheet" href="estilos.css"></pre>



Arquitectura de Aplicaciones Web – 1ºC 2018

Simplificación

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

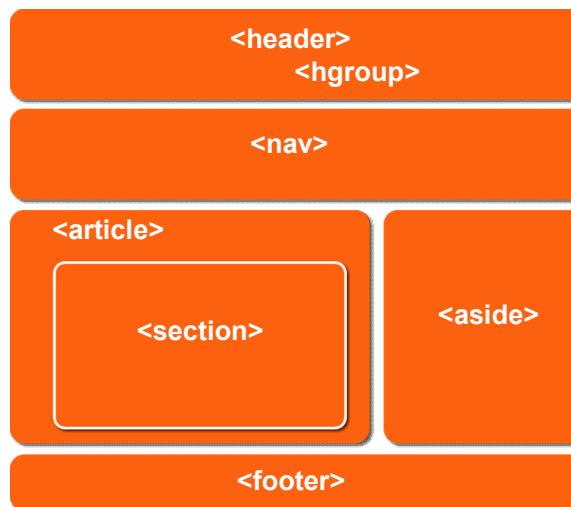
ETIQUETA “A”

HTML 4.01	HTML5
<pre><h2> Acerca de </h2> <p> Conoce quién es somos </p></pre>	<pre> <h2>Acerca de</h2> <p>Conoce quiénes somos</p> </pre> 

Arquitectura de Aplicaciones Web – 1ºC 2018

Elementos Estructurales

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA



<header>
<hgroup>

<nav>

<article>

<section>

<aside>

<footer>

Arquitectura de Aplicaciones Web – 1ºC 2018

Elementos Estructurales



```
<header> <hgroup>
          <nav>
<article>
    <section>
    <aside>
<footer>
```



<header>

representa la cabecera de un documento o sección

<hgroup>

representa el título de una sección. Se usa para agrupar conjuntos de elementos h1-h6 (títulos y subtítulos)

```
<header>
  <hgroup>
    <h1>Mi Blog</h1>
    <h2>Esforzándome para trabajar menos</h2>
  </hgroup>
</header>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales



```
<header> <hgroup>
          <nav>
<article>
    <section>
    <aside>
<footer>
```



<nav>

representa una sección del documento que contiene navegación

```
<nav>
  <ul>
    <li><a href="#">home</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">galería</a></li>
    <li><a href="#">contacto</a></li>
  </ul>
</nav>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales



```
<header>
  <hgroup>
```

```
<nav>
```

```
<article>
  <section>
    <img alt="Thumbnail image of a landscape" data-bbox="285 355 380 435"/>
  </section>
  <aside>
```

```
<footer>
```

<article>

representa una pieza de contenido independiente dentro de un documento

<section>

representa una sección del documento (un capítulo, un apartado, etc) que agrupa una serie de contenidos con una temática común

DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales



```
<header>
  <hgroup>
```

```
<nav>
```

```
<article>
  <section>
    <img alt="Thumbnail image of a landscape" data-bbox="285 765 380 845"/>
  </section>
  <aside>
```

```
<footer>
```

```
<article>
  <hgroup>
    <h1>Título del artículo</h1>
    <h2>Subtítulo del artículo</h2>
  </hgroup>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <section>
    <h1>Capítulo 1</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer bibendum scelerisque neque, ac facilisis neque</p>
  </section>
  <section>
    <h1>Capítulo 2</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer bibendum scelerisque neque, ac facilisis neque</p>
  </section>
</article>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales



`<header>` `<hgroup>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

???

!!!!

`<dialog>`

representa una transcripción de una conversación, o los subtítulos de algún elemento embebido (audio o video)

```
<dialog>
<dt>Pedro:</dt>
<dd>Toc, toc.</dd>
<dt>Jaime:</dt>
<dd>Sí?</dd>
<dt>Pedro:</dt>
<dd>¿Es aquí el Seminario de HTML5 y CSS3?</dd>
<dt>Jaime:</dt>
<dd>Pasa, pasa!</dd>
</dialog>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales



`<header>` `<hgroup>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

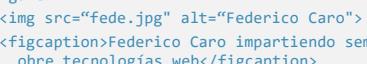


`<figure>`

representa un diagrama, una ilustración, una fotografía, etc

`<figcaption>`

representa la “nota al pie” del elemento incluido en `<figure>`

```
<figure>

<figcaption>Federico Caro impartiendo seminarios sobre tecnologías web</figcaption>
</figure>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales

The diagram illustrates the structure of a web page using semantic HTML elements. It shows a vertical stack of elements: <header>, <hgroup>, <nav>, <article> (containing <section>), <aside>, and <footer>. To the right, a callout box for <aside> defines it as content not directly related to the main page content or providing additional information. Below the diagram is a sticky note with placeholder text.

<aside>

representa contenidos que no están directamente relacionados con el resto de contenido de la página o que aporta información adicional

```
<article>
  <header><h1>Tecnologías web</h1></header>
  <p>bla bla bla</p>
  <aside>
    <ul>
      <li><a href="#">Links sobre HTML5</a></li>
      <li><a href="#">Links sobre CSS3</a></li>
    </ul>
  </aside>
</article>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos Estructurales

The diagram illustrates the structure of a web page using semantic HTML elements. It shows a vertical stack of elements: <header>, <hgroup>, <nav>, <article> (containing <section>), <aside>, and <footer>. To the right, a callout box for <footer> defines it as the page footer containing author and copyright information. Below the diagram is a sticky note with placeholder text.

<footer>

representa el pie de una sección o página. Suele contener información sobre el autor, copyright, etc

```
<footer>
  <p>© 2010 Bla bla bla bla</p>
</footer>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Formularios



DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

placeholder
`Hobbies: Indica tus hobbies`

autofocus

required

Debe especificar un valor

`<input>`

autocomplete { on off }

pattern [Expresiones regulares]

 <http://www.findmebyip.com/litmus/#html5-forms-inputs>

Arquitectura de Aplicaciones Web – 1*C 2018

Formularios



DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

`<input>`

type {

- search
- email
- url
- tel
- range (*)
- number (*)
- date
- datetime
- datetime-local
- month
- color

min (*) max (*)

 <http://www.findmebyip.com/litmus/#html5-forms-inputs>

Arquitectura de Aplicaciones Web – 1*C 2018

Input types e interfaces

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Type	Purpose
<code>tel</code>	Telephone number
<code>email</code>	Email address text field
<code>url</code>	Web location URL
<code>search</code>	Term to supply to a search engine. For example, the search bar atop a browser.
<code>range</code>	Numeric selector within a range of values, typically visualized as a slider

```

<input type="text" />
<input type="email" />
<input type="url" />
    
```

Arquitectura de Aplicaciones Web – 1*C 2018

Nuevos atributos y funciones

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

- **datalist y list**

```

<input type="email" id="contacts" list="pedidos">

<datalist id="pedidos">
  <option value="recubote@gmail.com" label="Bote">
  <option value="recubalsa@gmail.com" label="Balsa">
  <option value="recupile@gmail.com" label="Pileta">
</datalist>
    
```

Sueldo:

```

Pedidos
recubote@gmail.com Bote
recubalsa@gmail.com Balsa
recupile@gmail.com Pileta
    
```

```

<script>
  function showValue(newVal) {
    document.getElementById("sueldoDisplay").innerHTML = newVal;
  }
</script>

<label for="sueldo">Sueldo:</label>
<input id="sueldo" name="level" type="range"
  min="0" max="10000" step="1000" value="3000"
  onchange="showValue(this.value)">
<span id="sueldoDisplay">3000</span>
    
```

- **min, max, step (range input type)**

Arquitectura de Aplicaciones Web – 1*C 2018

Nuevos atributos y funciones

- Spellcheck

```
<p contenteditable="true" spellcheck="true">
This is a praggagraph. It is editable.
Try to change this text.
Hola eu no hablo spanish very bien
</p>
```

This is a praggagraph. It is editable. Try to change this text. Hola eu no hablo spanish very bien



```
<div id="columns" >
  <div class="column" draggable="true">
    <header>A</header></div>
    <div class="column" draggable="true">
      <header>B</header></div>
  </div>
```

```
var cols = document.querySelectorAll('#columns .column');
[].forEach.call(cols, function(col) {
  col.addEventListener('dragstart', handleDragStart, false);
  col.addEventListener('dragenter', handleDragEnter, false);
  col.addEventListener('dragover', handleDragOver, false);
  col.addEventListener('dragleave', handleDragLeave, false);
  col.addEventListener('drop', handleDrop, false);
  col.addEventListener('dragend', handleDragEnd, false);
});
```

- draggable / dropzone

Validaciones

- ValidityState

```
var valCheck = document.myForm.myInput.validity;
valCheck.valid
```

- valueMissing Constraint

```
<input type="text" name="myText" required>
```

- typeMismatch

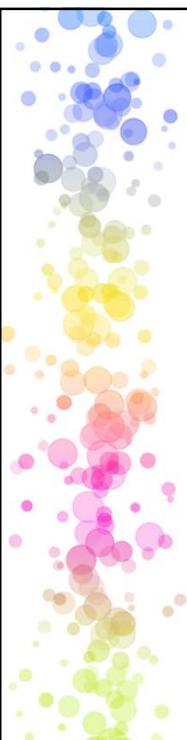
```
<input type="email" name="myEmail">
```

- patternMismatch

```
<input type="text" name="creditcardnumber" pattern="[0-9]{16}" title="A credit card number is 16 digits with no spaces or dashes">
```

- tooLong

```
<input type="text" name="limitedText" maxLength="140">
```



Validaciones

- rangeUnderflow

```
<input type="range" name="ageCheck" min="18">
```

- rangeOverFlow

```
<input type="range" name="kidAgeCheck" max="12">
```

- stepMismatch

```
<input type="range" name="confidenceLevel" min="0" max="100" step="5">
```

- customError

```
passwordConfirmationField.setCustomValidity("Password values do not match.");
```

Arquitectura de Aplicaciones Web – 1*C 2018



Javascript

NUEVOS MÉTODOS

- document.getElementsByClassName: acceso a todos los elementos del DOM que compartan la clase especificada
- document.querySelectorAll(selector): acceso a todos los elementos del DOM con el selector especificado

```
<script>
    elementos = document.querySelectorAll("section div.wrapper");
    elementos = document.querySelectorAll("div.content, div.wrapper");
    inputsText = document.querySelectorAll('[type=text]');
</script>
```

- document.querySelector(selector): acceso al primer elemento del DOM con el selector especificado

```
<script>
    var elemento = document.querySelector("section div.wrapper");
</script>
```



Arquitectura de Aplicaciones Web – 1*C 2018



Video





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
<video controls width="360" height="240">
  <source src="movie.mp4">
  <source src="movie.ogv" type="video/ogg codecs='theora, vorbis'">
  <source src="movie.webm" type="video/webm codecs='vp8, vorbis'">
  <object type="application/x-shockwave-flash"
    width="360" height="240" data="player.swf?file=movie.mp4">
    <param name="movie" value="player.swf?file=movie.mp4">
    <a href="movie.mp4">Descargar video</a>
  </object>
</video>
```



SOPORTE VIDEO POR NAVEGADORES

	6	3.6	10.6	5	9
mp4	✓	✗	✗	✓	✓
ogg	✓	✓	✓	✗	✗
webM	✓	✓	✓	✗	✓

Arquitectura de Aplicaciones Web – 1*C 2018

Audio





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
<audio controls>
  <source src="song.ogg" type="audio/ogg" />
  <source src="song.mp3" type="audio/mpeg" />
  <object type="application/x-shockwave-flash" data="player.swf?soundFile=song.mp3">
    <param name="movie" value="player.swf?soundFile=song.mp3">
    <a href="song.mp3">Descargar canción</a>
  </object>
</audio>
```



SOPORTE AUDIO POR NAVEGADORES

	6	3.6	10.6	5	9
mp3	✓	✗	✗	✓	✗
ogg	✗	✓	✓	✗	✗

Arquitectura de Aplicaciones Web – 1*C 2018



Canvas

- *HTML 5 defines the <canvas> element as “a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly.”*
- Un canvas es un rectángulo en la página en donde se puede usar Javascript para dibujar lo que quieras.

```
<canvas height="200" width="200"></canvas>
```

Arquitectura de Aplicaciones Web – 1*C 2018



Canvas

- No requiere plugins, ni codecs
- Mapa de bits
- El contenido no se añade al DOM
- Puede ser exportado

```
<canvas id="miLienzo" width="360" height="240">
<p>Tu navegador no soporta canvas</p>
</canvas>
<script>
var lienzo = document.getElementById('miLienzo')
var contexto = lienzo.getContext('2d');
</script>
```

 <http://code.google.com/p/explorercanvas/>

 <http://billmill.org/static/canvastutorial/index.html>



Arquitectura de Aplicaciones Web – 1*C 2018



Soporte para Html 5 Canvas



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Browser	Details
Chrome	Supported in version 1.0 and greater
Firefox	Supported in version 1.5 and greater
Internet Explorer	Not supported
Opera	Supported in version 9.0 and greater
Safari	Supported in version 1.3 and greater

Se puede usar canvas en Internet Explorer a través del proyecto open-source [explorercanvas](http://code.google.com/p/explorercanvas), (<http://code.google.com/p/explorercanvas>). Para usarlo solo hay que chequear que internet explorer se a nuestro navegador, e incluir un script en la página para iniciararlo:

```
<head>
<!--[if IE]><script src="excanvas.js"></script><![endif]-->
</head>
```

Arquitectura de Aplicaciones Web – 1ºC 2018



Canvas API



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- void rotate(in float angle);
- void scale(in float x, in float y);
- void setTransform(in float m11, in float m12, in float m21, in float m22, in float dx, in float dy);
- void transform(in float m11, in float m12, in float m21, in float m22, in float dx, in float dy);
- void translate(in float x, in float y);
- void clearRect(in float x, in float y, in float w, in float h);
- void fillRect(in float x, in float y, in float w, in float h);
- void strokeRect(in float x, in float y, in float w, in float h);
- void arc(in float x, in float y, in float radius, in float startAngle, in float endAngle, in boolean anticlockwise);
- void arcTo(in float x1, in float y1, in float x2, in float y2, in float radius);
- Etc...

Arquitectura de Aplicaciones Web – 1ºC 2018

No va más...

ETIQUETAS

<acronym>	<frameset>
<applet>	<isindex>
<basefont>	<noframes>
<big>	<s>
<center>	<strike>
<dir>	<tt>
	<u>
<frame>	

REDEFINID AS

<small>	<cite>
	<i>
<a>	

ATRIBUTOS

abbr	codetype	scheme
align	compact	scope
alink	compact	shape
archive	declare	size
axis	hspace	standby
background	link	target
bgcolor	longdesc	text
border	marginheight	type
cellpadding	marginwidth	type
cellspacing	name	valign
char	nohref	valuetype
charoff	noshade	version
charset	nowrap	vlink
classid	profile	vspace
clear	rev	width
codebase	rules	



Arquitectura de Aplicaciones Web – 1ºC 2018

Almacenamiento local

- Web <> Online?
- Cache <> Storage?



localStorage

Arquitectura de Aplicaciones Web – 1ºC 2018

Almacenamiento local

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Técnicas de Almacenamiento Offline Previas

The diagram illustrates three techniques for offline storage:

- Cookies:** Represented by a cookie icon. It stores "Poca información (4 Kb)" and "Reduce velocidad". A note indicates it requires a plugin and has issues with firewalls.
- Gears:** Represented by a gear icon. It is described as "Basado en Plugins". It shows an upward arrow between the two columns.
- Navegador:** Represented by icons of various web browsers. It forces users to use a specific browser and is shown with a downward arrow between the two columns.

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Técnicas de Almacenamiento con **HTML5**

Diferentes APIs:

- Web Storage (Local Storage or DOM Storage)
- Web SQL Database
- IndexedDB
- File Storage

Principios:

- Normas estándar para “todos” los navegadores.
- Información solo accesible desde el propio navegador.
- Interacción de la API y la Base de Datos es asíncrona

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Web Storage



- ➔ El más compatible.
- ➔ Estructura clave-valor
- ➔ Integridad de los datos.
- ➔ Race conditions

TIPOS DE DATOS:

- localStorage ➔ Permanecen hasta que el usuario los borra
- sessionStorage ➔ Desaparecen al cerrarse el navegador



<http://html5demos.com/storage>
<http://playground.html5rocks.com/#localStorage>
<http://playground.html5rocks.com/#sessionStorage>

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local





DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Web SQL Database



- ➔ Como las DDBB tradicionales.
- ➔ Estructura relacionada (joins)
- ➔ Más complejas.
- ➔ IE y Firefox no son compatibles



http://playground.html5rocks.com/#async_transactions

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local



IndexedDB

- Mezcla entre Web Storage y Web SQL Database
- Estructura relacionada (joins)
- Transacciones de información.
- Ningún navegador lo soporta, pero está previsto ser soportado por todos

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local



• FileStorage

- Puedes guardar información binaria (como texto), y **grandes cantidades de información**.
- **File Reader** soportado solo por Chrome
- **FileWriter** todavía no soportado por nadie
- Cuando sea soportado será bueno para almacenar grandes cantidades de información

Arquitectura de Aplicaciones Web – 1*C 2018

Almacenamiento local

- Por defecto usar WEB STORAGE (+ simple & + compatible)
- Ayudarse de librerías como persis.js ó MODERNIZR
- Protegerse contra la perdida de datos
 - No guardar información privada
 - Los usuarios pueden borrar los datos fácilmente
 - Sincronizarse con el servidor frecuentemente
- Securizar los datos
 - Alto riesgo en navegadores compartidos
 - SessionStorage mejor que LocalStorage
 - Encriptar información
 - Evitar guardar cierto tipo de datos

Arquitectura de Aplicaciones Web – 1*C 2018

Yo... me lo banco?



```
<script src="js/modernizr-1.6.min.js"></script>
```



True (sopotada)

False (no soportada)

```
if (Modernizr.draganddrop) {
  // Browser supports HTML5 DnD.
} else {
  // Fallback to a library solution.
}
```

Modernizr detects support for:

@font-face ✓	Geolocation API ✓
Canvas ✓	localStorage ✓
Canvas Text ✓	sessionStorage ✓
HTML5 Audio ✓	SVG ✓
HTML5 Video ✓	SMIL ✓
rgba() ✓	SVG Clipping ✓
hsla() ✓	Inline SVG ✓
border-image: ✓	Drag and Drop ✓
border-radius: ✓	hashchange ✓
box-shadow: ✓	X-Window Messaging ✓
text-shadow: ✓	History Management ✓
opacity: ✓	applicationCache ✓
Multiple backgrounds ✓	Touch events
Flexible Box Model ✓	Web Sockets ✓
CSS Animations ✓	Web Workers ✓
CSS Columns ✓	Web SQL Database ✓
CSS Gradients ✓	WebGL
CSS Reflections ✓	IndexedDB
CSS 2D Transforms ✓	Input Types†
CSS 3D Transforms	Input Attributes‡
CSS Transitions ✓	

Arquitectura de Aplicaciones Web – 1*C 2018

CSS



- **Hoja de estilo en cascada o CSS** (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML² (y por extensión en XHTML).
- El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.
- La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Arquitectura de Aplicaciones Web – 1*C 2018

Historia CSS 3

1996	CSS 1: permite dar estilos independientemente del navegador y del HTML
1998	CSS2: nuevas funcionalidades, pero implementación lenta
	Semilla del CSS3.
	Se plantea una lista de mejoras de CSS2
2000	Borrador de CSS3
2002	CSS2.1: Crea lo que ahora consideramos el estándar
2005	Empieza el desarrollo de CSS3
2009	Implementación en algunos navegadores de algunas partes de CSS3

1991	HTML
1994	HTML 2
1996	CSS 1 + JavaScript
1997	HTML 4
1998	CSS 2
2000	XHTML 1
2002	Tableless Web Design
2005	AJAX
2009	HTML 5

Arquitectura de Aplicaciones Web – 1*C 2018



css 3 vs. Css 2

- ▲ Mejora en los selectores
- ▲ Nuevos estilos
 - Sombra
 - Opacidad
 - esquinas redondeadas
 - ...
- ▲ Mejora en tipografías
- ▲ Transformaciones
- ▲ Reduce la cantidad de HTML (divitis)
- ▲ Reduce las peticiones de imágenes
- ▲ ¡Nuestra pagina va a ser más rápida!

Arquitectura de Aplicaciones Web – 1*C 2018



Bordes

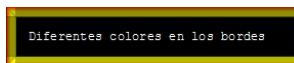
border-radius

border-radius: 15px;



border-color

border: 5px solid #000;
border-color:#e00 #c30 #c50 #c60 #c70



Diferentes colores en los bordes

border-image

border-image: url(border.png)
27 27 27 27 round round;



Lorem ipsum dolor sit amet.

box-shadow / text-shadow

box-shadow: 10px 10px 5px #888;





HTWK Leyendecker 01

Multiple Backgrounds



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
background:
url(..../topImage.jpg) top left      no-repeat,
url(..../centerImage.jpg) top right   repeat-y,
url(..../bottomImage.jpg) bottom center no-repeat;
```



[!\[\]\(f59842166a12d90a8a0108ac365de408_img.jpg\) http://www.css3.info/wp-content/uploads/2007/09/multiple-backgrounds-example.html](http://www.css3.info/wp-content/uploads/2007/09/multiple-backgrounds-example.html)



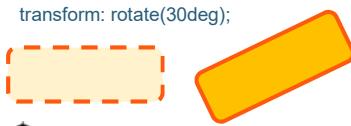
Arquitectura de Aplicaciones Web – 1ºC 2018

Transform



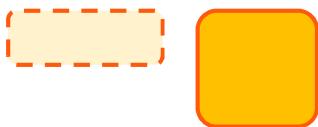
DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- **rotate :**
`transform: rotate(30deg);`



[!\[\]\(20b91340b38c2118e833e5fbfc26d7aa_img.jpg\) http://www.ejhansel.com/transform/](http://www.ejhansel.com/transform/)

- **scale :**
`transform: scale(0.5,2.0);`



[!\[\]\(58b4dfce06ed9513161599cad2d4f2bf_img.jpg\) http://lab.simurai.com/css/tilt-shift/](http://lab.simurai.com/css/tilt-shift/)

- **skew :**
`transform: skew(-30deg);`



- **translate**
`transform: translate(30px,10px);`



Arquitectura de Aplicaciones Web – 1ºC 2018



Color



DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- Opacity

Opacity: 1.0
Opacity: 0.5

- RGBA:
(Red, Green, Blue, Alpha)

rgb(255,192,0,1);
rgb(255,192,0,0.5);

- HSL:
(Hue, Saturation, Lightness)

hsl(21,97%,52%)
hsl(21,97%,52%)

- HSLA:
(Hue, Saturation, Lightness, Alpha)

hsla(21,97%,52%,1);
hsla(21,97%,52%,0.5);

Arquitectura de Aplicaciones Web – 1*C 2018



Multi Column Layout



DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- column-count
- column-width
- column-gap
- column-rule

column-width

column-width: 300px;

column-width: 3em;

column-width: 1fr;

column-rule

column-rule: 1px solid black;

column-rule: 1px solid red;

column-gap

column-gap: 10px;

column-gap: 1em;

column-gap: 1fr;

column-count: 3;

column-count: 3;

column-width: 300px;

column-width: 3em;

column-width: 1fr;

column-gap: 10px;

column-gap: 1em;

column-gap: 1fr;

column-rule: 1px solid black;

column-rule: 1px solid red;

column-gap: 10px;

column-gap: 1em;

column-gap: 1fr;

Arquitectura de Aplicaciones Web – 1*C 2018

Web Fonts



HTML 5 FONTS
HTML 5 FONTS
HTML 5
HTML 5 FONTS
HTML 5 FONTS
HTML 5 FONTS
HTML 5 FONTS
HTML 5 FONTS
HTML 5 FONTS
HTML 5 FONTS

```
@font-face {
    font-family: 'FontName';
    src: url('Gondola_SD-webfont.eot');
    src: local('☺'), url('FontName.woff')
        format('woff'), url('FontName.ttf')
        format('truetype'),
    url('FontName.svg#webfontgM4b18D') format(
    'svg');
    font-weight: normal;
    font-style: normal;
}

div {
    font-family: FontName;
}
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Textos



Text-shadow

Text-shadow: Xpos Ypos Blur Color;

Text-overflow

Text-overflow: ellipsis-word;

Lorem ipsum dolor sit...

Word-wrap

word-wrap: break-word;

This paragraph has long words thisisaveryverylongwordthatistn'treally one word and again a longwordwithno spacesinit

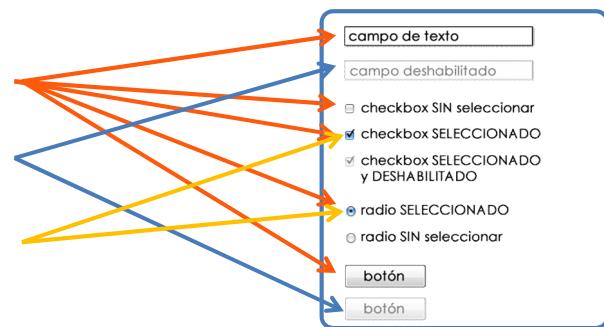
Arquitectura de Aplicaciones Web – 1°C 2018

Pseudo Clases



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

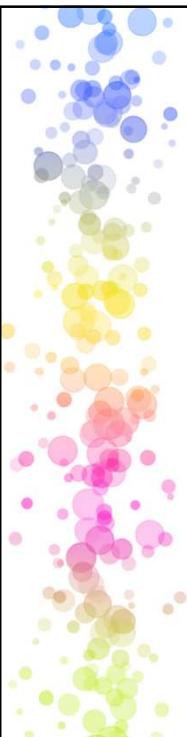
- :enabled
- :disabled
- :checked



campo de texto
campo deshabilitado
checkbox SIN seleccionar
checkbox SELECCIONADO
checkbox SELECCIONADO y DESHABILITADO
radio SELECCIONADO
radio SIN seleccionar
botón
botón

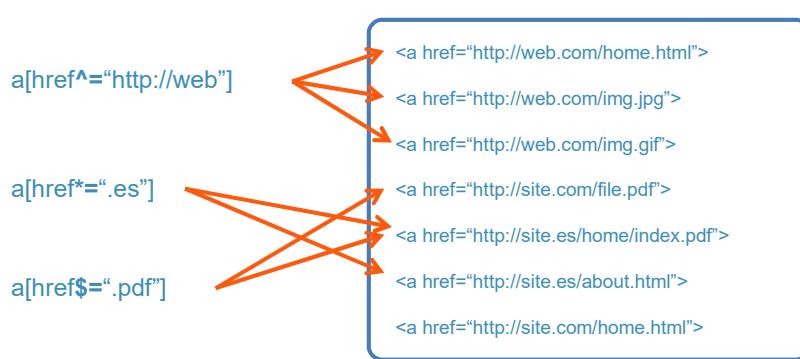
Arquitectura de Aplicaciones Web – 1*C 2018

Selectores de atributos



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

[att*=val] contiene val	[att^=val] empieza por val	[att\$=val] termina por val
-----------------------------------	--------------------------------------	---------------------------------------

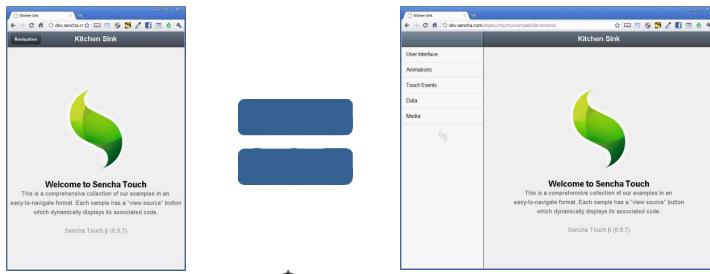


a[href^="http://web"]
a[href*=".es"]
a[href\$=".pdf"]

Arquitectura de Aplicaciones Web – 1*C 2018

Media Queries

min-width & max-width
diferentes estilos según el tamaño de la pantalla



<http://dev.sencha.com/deploy/touch/examples/kitchensink/>

Arquitectura de Aplicaciones Web – 1*C 2018

Estado actual

- Desarrollo dividido en módulos
 - Selectores, fuentes, colores, ...

CSS3 Properties Support by Browsers					
	5, 6	3.6 - 4.02	10.6	5	6-8
CSS Animations	✓	✗ ✗	✗	✓	✗
CSS Columns	✓	✓ ✓	✗	✓	✗
CSS Gradients	✓	✓ ✓	✗	✓	✗
CSS Reflections	✓	✗ ✗	✗	✓	✗
CSS Transforms	✓	✓ ✓	✓	✓	✗
CSS Transforms 3D	✗	✗ ✗	✗	✓	✗
CSS Transitions	✓	✗ ✓	✓	✓	✗
CSS FontFace	✓	✓ ✓	✓	✓	✓

<http://www.w3.org/Style/CSS/current-work#table>
<http://www.findmebyip.com/litmus/#target-selector>

HTTP/2.5, 02/08/18
 Arquitecturas de Aplicaciones Web – 1*C 2018

Propiedades de Navegador



 ➔ `-o-border-radius: 10px;` ➔ `-webkit-border-radius: 10px;`

 ➔ `-ms-border-radius: 10px;` ➔ `-moz-border-radius: 10px;`

-  ¡podemos usar CSS3 desde hoy!
-  Aumentamos y ensuciamos el código
-  No todos los navegadores lo soportan
-  Posibles soluciones
 - css3generator.com
 - [less.js](http://lesscss.org)
 - Ficheros .css para cada navegador
 - css3tender.org

Arquitectura de Aplicaciones Web – 1ºC 2018



Modelo de Eventos de JavaScript



- Los eventos suceden a tres niveles:
 - A nivel del documento HTML
 - A nivel de un formulario individual
 - A nivel de un elemento de un formulario
- El evento es gestionado por una sección de código en *JavaScript* (Gestor de Eventos)
- Declaración de Gestores de Eventos: similar a los atributos en HTML

Arquitectura de Aplicaciones Web – 1ºC 2018



Evento - Acción



```
<BODY onLoad="cargarfuncion()"  
      onUnload="descargarfuncion()">  
  
<FORM name="nombre_del_formulario" ...  
      onSubmit="función_o_sentencia">  
  
<INPUT type="button" name="mycheck" value="HA!"  
      onClick="alert('Gracias por apretar')">
```

Arquitectura de Aplicaciones Web – 1*C 2018

Event Handlers



Evento	Ocurre Cuando	Gestor
blur	El usuario quita el cursor de un elemento de formulario	onBlur
click	El usuario clica un link o un elemento de formulario	onClick
change	El usuario cambia el valor de un texto, un área de texto o selecciona un elemento.	onChange
focus	El usuario coloca el cursor en un elemento de formulario.	onFocus
load	El usuario carga una página en el Navegador	onLoad
Mouseover	El usuario mueve el ratón sobre un link	onMouseOver
Select	El usuario selecciona un campo del elemento de un formulario	onSelect
Submit	Se envía un formulario	onSubmit
Unload	Se descarga la página	onUnload

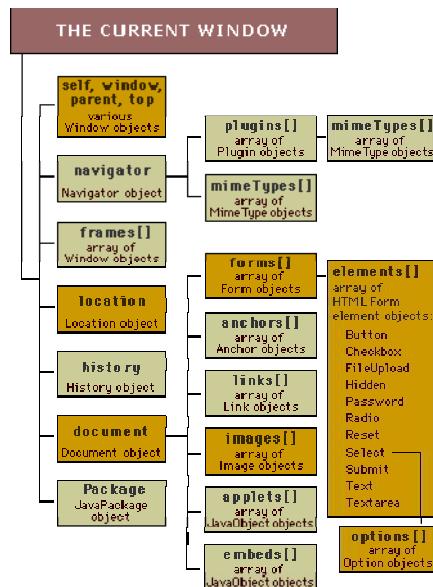
Arquitectura de Aplicaciones Web – 1*C 2018

Clases en Javascript

- ❑ Clases Predefinidas
 - ❑ Clase **String**: Cada vez que se asigna una cadena de caracteres a una variable, se crea un objeto de la clase String
 - ❑ Clase **Math**: Se usa para efectuar cálculos matemáticos
 - ❑ Clase **Date**: Para el manejo de fechas y horas
- ❑ Clases del Browser o Navegador
 - ❑ Tienen que ver con la navegación
- ❑ Clases del Documento HTML
 - ❑ Están asociadas con cualquier elemento de una página Web (link, ancla, formulario, etc)
- ❑ Clases definidas por el usuario

Arquitectura de Aplicaciones Web – 1*C 2018

Jerarquía de Clases



Arquitectura de Aplicaciones Web – 1*C 2018



HTML5 y Javascript



DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

NUEVOS MÉTODOS

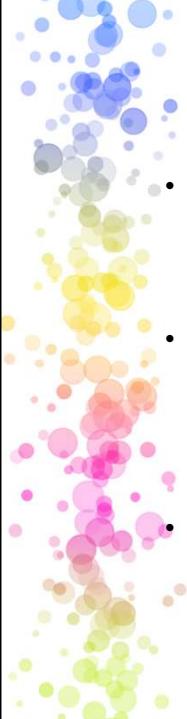
- `document.getElementsByClassName`: acceso a todos los elementos del DOM que comparten la clase especificada
- `document.querySelectorAll(selector)`: acceso a todos los elementos del DOM con el selector especificado

```
<script>
    elementos = document.querySelectorAll("section div.wrapper");
    elementos = document.querySelectorAll("div.content, div.wrapper");
    inputsText = document.querySelectorAll('[type=text]');
</script>
```

- `document.querySelector(selector)`: acceso al primer elemento del DOM con el selector especificado

```
<script>
    var elemento = document.querySelector("section div.wrapper");
</script>
```

Arquitectura de Aplicaciones Web – 1*C 2018



JSON



DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- **JSON**, acrónimo de "JavaScrip Object Notation", es un formato ligero para el intercambio de datos. **JSON** es un subconjunto de la notación literal de objetos de Javascript pero no requiere el uso de XML.
- La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de **JSON** sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador semántico de JSON.
- En Javascript, JSON puede ser analizado trivialmente usando el procedimiento eval(), lo cual ha sido fundamental para la aceptación de JSON por parte de la comunidad de desarrolladores Ajax, debido a la ubicuidad de Javascript en casi cualquier navegador Web.

Arquitectura de Aplicaciones Web – 1*C 2018

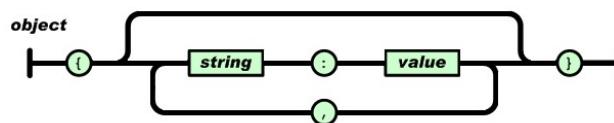
JSON

- Está constituido por dos estructuras
 - Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un *objeto*, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
 - Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Arquitectura de Aplicaciones Web – 1ºC 2018

Objetos en JSON

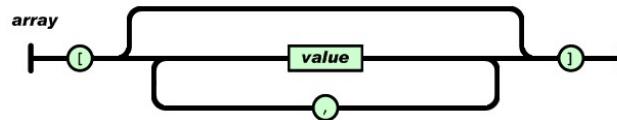
- Un objeto es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termina con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).



Arquitectura de Aplicaciones Web – 1ºC 2018

Arrays en JSON

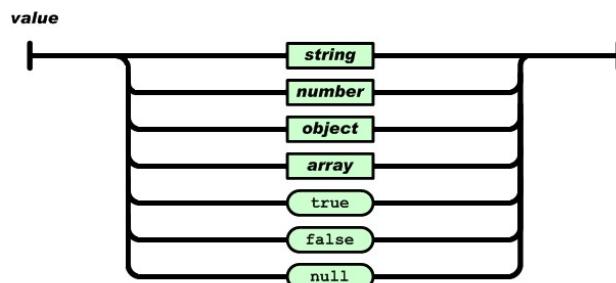
- Un array es una colección de valores. Un array comienza con [(corchete izquierdo) y termina con] (corchete derecho). Los valores se separan por , (coma).



Arquitectura de Aplicaciones Web – 1*C 2018

Valores en JSON

- Un *valor* puede ser una *cadena de caracteres* con comillas dobles, o un *número*, o true o false o null, o un *objeto* o un *array*. Estas estructuras pueden anidarse



Arquitectura de Aplicaciones Web – 1*C 2018

JSON – Equivalencia con XML

```
{"menu": {  
    "id": "file",  
    "value": "File",  
    "popup": {  
        "menuitem": [  
            {"value": "New", "onclick": "CreateNewDoc()"},  
            {"value": "Open", "onclick": "OpenDoc()"},  
            {"value": "Close", "onclick": "CloseDoc()"}  
        ]  
    }  
}}
```

```
<menu id="file" value="File">  
  <popup>  
    <menuitem value="New" onclick="CreateNewDoc()" />  
    <menuitem value="Open" onclick="OpenDoc()" />  
    <menuitem value="Close" onclick="CloseDoc()" />  
  </popup>  
</menu>
```

Arquitectura de Aplicaciones Web – 1*C 2018

JSON – Solicitud AJAX

- **Formato**

```
var jSonRequest = new Json.Remote("script que hace el servicio",  
  {onComplete: function(objeto json){  
  }}).send({variables a enviar});
```

- **Ejemplo**

```
var jSonRequest = new Json.Remote("servicios/personas_texto.asp",  
  {onComplete: function(jsonpersonas){  
    acabaAccion('JSON', jsonpersonas.personas);  
  }}).send({'filtro': ''});
```

Arquitectura de Aplicaciones Web – 1*C 2018

Y ahora...¿listo?

The diagram illustrates the process of web development. On the left, there is a vertical column of colorful bubbles (blue, yellow, pink, green) that funnel down into a blue funnel shape. Inside the funnel are three icons: HTML5, CSS3, and JS. A large blue arrow points from the funnel to a red question mark. A small white 3D person is sitting on top of the question mark.

Arquitectura de Aplicaciones Web – 1ºC 2018

¿Cómo describir estas aplicaciones?

The diagram illustrates the Model-View-ViewModel (MVVM) architectural pattern. It shows the following components and their interactions:

- DOM**: Represented by a rectangle at the top.
- View**: Represented by a rectangle below DOM.
- Model**: Represented by a rectangle at the bottom.
- Template**: Represented by a rectangle to the right of the View.
- Storage**: Represented by a rectangle to the right of the Model.

Interactions are labeled as follows:

- A curved arrow points from a box labeled **???** to the View, labeled "emits events".
- An arrow points from the View to the DOM, labeled "changes".
- An arrow points from the View to the Template, labeled "renders".
- An arrow points from the View to the Model, labeled "observes".
- An arrow points from the Model to the Storage, labeled "queries and writes to".

- Y todo esto sin contar las llamadas al servidor...

Arquitectura de Aplicaciones Web – 1ºC 2018

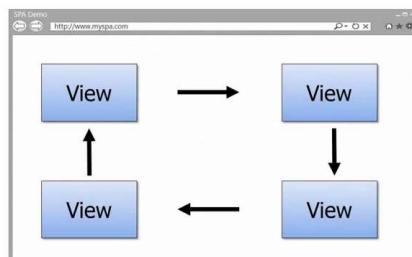
Single Page Applications

- Aplicaciones web que se ejecutan en una única página, logrando así una experiencia de usuario más cercana a una aplicación de escritorio.

Arquitectura de Aplicaciones Web – 1ºC 2018

Single Page Applications

- En una SPA el usuario no navega por un sistema de enlaces tradicionales si no que en su lugar, mediante el uso de JavaScript, Ajax, HTML5 o una combinación de las anteriores, se actualiza lo que el usuario ve siempre desde la misma página (sin cambiar de URL ni refrescar el contenido entero).



Arquitectura de Aplicaciones Web – 1ºC 2018



Desafíos SPA

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Problemas a resolver

- Manipulación DOM
- Historia
- Module loading
- Routing
- Caching
- Diseño Orientado a Objetos
- Data Binding
- Ajax
- View loading

Arquitectura de Aplicaciones Web – 1*C 2018



Javascript++

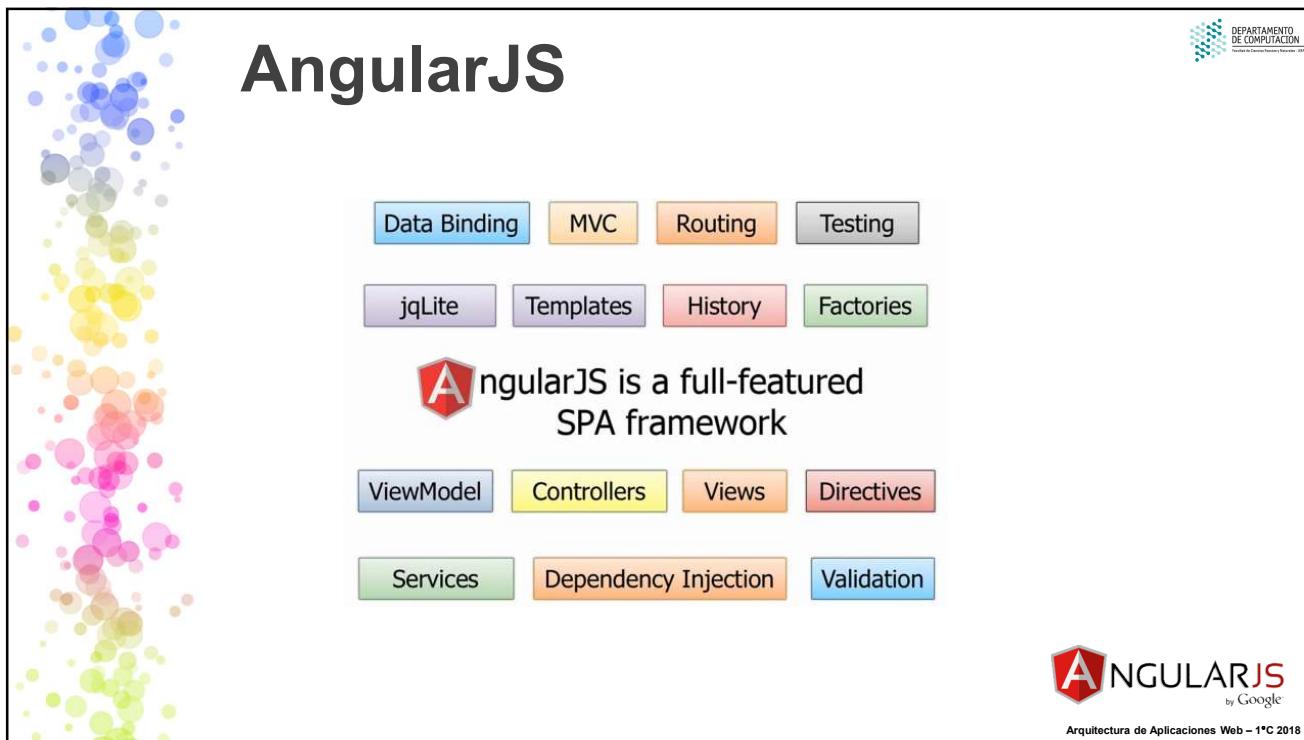
DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

- A medida que javascript obtuvo mayor popularidad y aceptación, surgen de él distintas librerías y frameworks.

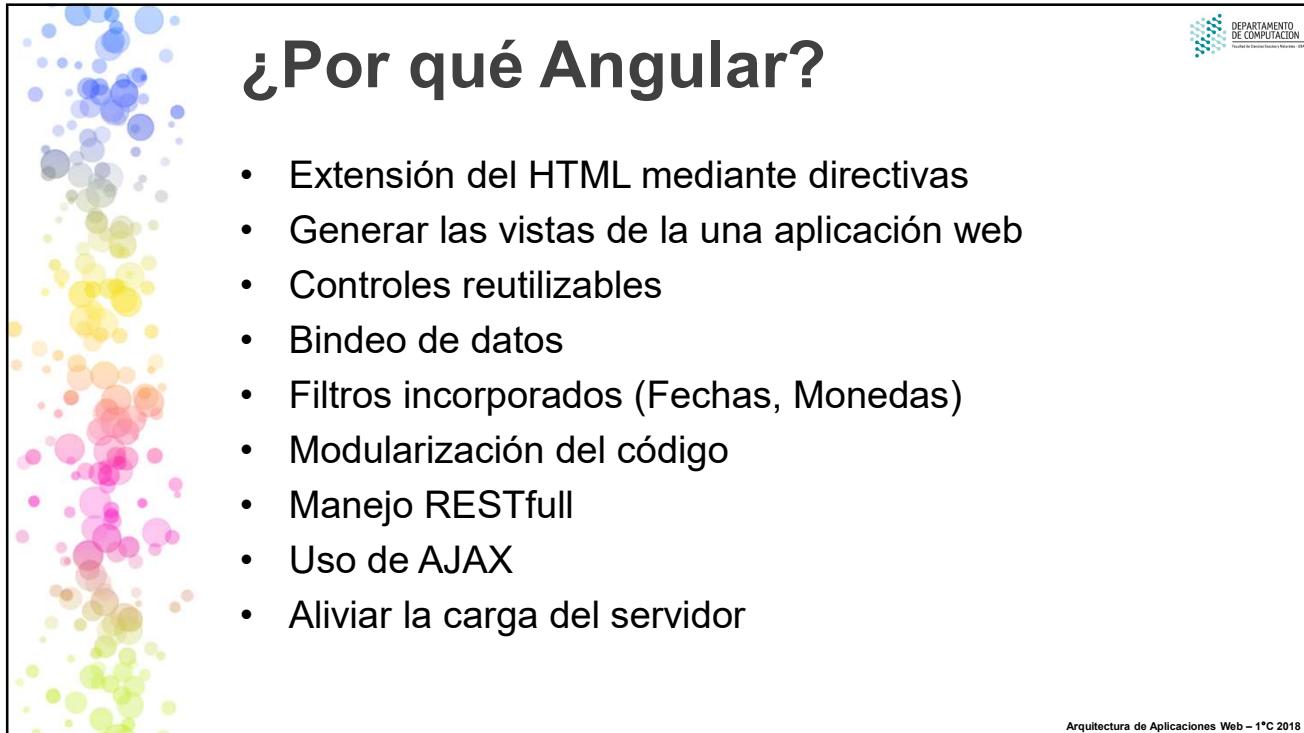
BACKBONE.JS

ANGULARJS
by Google

Arquitectura de Aplicaciones Web – 1*C 2018



The image shows the AngularJS landing page. On the left, there is a vertical decorative column of colored circles (blue, yellow, red, green) of varying sizes. The main title "AngularJS" is in large, bold, dark gray font at the top center. Below it is a subtitle "AngularJS is a full-featured SPA framework" with a red "A" icon. A grid of colored boxes lists features: Data Binding (blue), MVC (yellow), Routing (orange), Testing (gray), jqLite (purple), Templates (light purple), History (pink), Factories (green), ViewModel (blue), Controllers (yellow), Views (orange), Directives (red), Services (green), Dependency Injection (orange), and Validation (blue). At the bottom right is the AngularJS logo and the text "Arquitectura de Aplicaciones Web – 1ºC 2018". The top right corner features the "DEPARTAMENTO DE COMPUTACION" logo.



The image shows a slide titled "¿Por qué Angular?" (Why Angular?). It features a vertical decorative column of colored circles on the left. The title is in large, bold, dark gray font. Below the title is a bulleted list of reasons:

- Extensión del HTML mediante directivas
- Generar las vistas de la una aplicación web
- Controles reutilizables
- Bindeo de datos
- Filtros incorporados (Fechas, Monedas)
- Modularización del código
- Manejo RESTfull
- Uso de AJAX
- Aliviar la carga del servidor

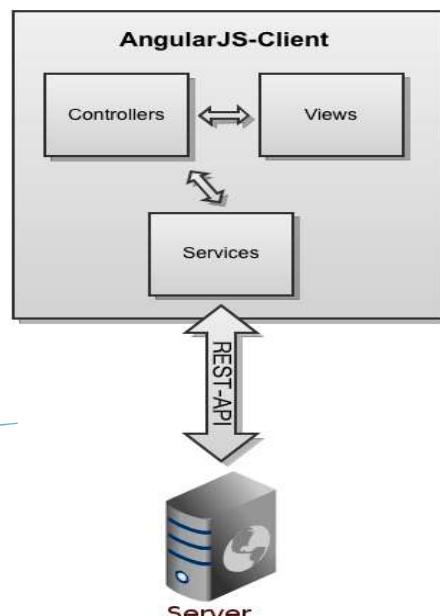
At the bottom right is the text "Arquitectura de Aplicaciones Web – 1ºC 2018". The top right corner features the "DEPARTAMENTO DE COMPUTACION" logo.

Y el servidor?

- Angular está hecho para que se ejecute exclusivamente en el cliente, por lo que, es totalmente independiente del código del lado del backend. El servidor puede estar hecho con cualquier otra tecnología, ya sea Java, .Net, Rails, etc...
 - En particular, Angular utiliza objetos JSON para la obtención de datos a través de un servicio.
 - Idealmente el servidor deberá definir todos los servicios Rest que la aplicación Angular vaya a necesitar

Arquitectura de Aplicaciones Web – 1ºC 2018

Estructura



Arquitectura de Aplicaciones Web – 1ºC 2018

¿Cómo empezamos?

- Para utilizar Angular, debemos cargarlo en el archivo HTML que representará la vista de nuestra Web.

- Puede cargarse desde un directorio del servidor....

```
<head>
  <script src="files/angular.js"></script>
</head>
```

- O bien directamente desde un enlace externo, Angular pesa alrededor de 100kb en su versión minificada.

```
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.18/angular.min.js"
  "></script>
</head>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Hello World

```
<!DOCTYPE html >
<html>
  <head>
    <script type="text/javascript" src="js/angular.js"></script>
    <script type="text/javascript"
      src="js/helloangular.js"></script>
    <title>Hello world example</title>
  </head>
  <body ng-app="helloangular">
    <helloworld></helloworld>
  </body>
</html>
```



```
Código Javascript de "helloangular.js"
(function() {
var app = angular.module("helloangular", []);

app.directive("helloworld", function() {
  return { restrict: 'E', template: "<h1>
Hello world!</h1>" };
});
}());
```

Aplicaciones Web – 1*C 2018

Hello World



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="js/angular.js"></script>
    <script type="text/javascript"
           src="js/helloangular.js"></script>
    <title>Hello world example</title>
  </head>
  <body ng-app="helloangular">
    <helloworld></helloworld>
  </body>
</html>
```

Código Javascript de "helloangular.js"

```
(function() {
  var app = angular.module("helloangular", []);

  app.directive("helloworld", function() {
    return { restrict: 'E', template: "<h1> Hello world!</h1>" };
  });
})();
```

Aplicaciones Web – 1*C 2018

Hello World



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

```
(function() {
  var app = angular.module("helloangular", []);

  app.directive("helloworld", function() {
    return { restrict: 'E', template: "<h1> Hello world!</h1>" };
  });
})();
```

- Se define el nombre de la aplicación, asignándola a una variable, el corchete indica las dependencias a **inyectarse**, en este caso ninguna.
- Se define una directiva personalizada de la aplicación: **helloworld**.
 - Para definir que la directiva será un nuevo elemento HTML se la define como restrict: 'E', análogamente, un atributo, se lo define como restrict:'A'
 - Se describe el código HTML al que corresponderá el uso de la directiva (también se podría estar en un archivo aparte, pudiendo referenciarlo de la forma: templateUrl: "nombre")

Arquitectura de Aplicaciones Web – 1*C 2018

Elementos de AngularJS

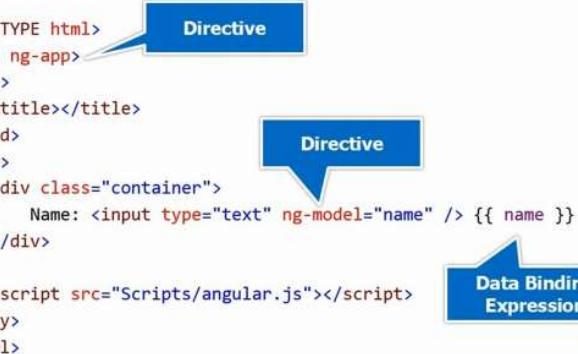
- Directivas
- Expresiones
- Controllers
- Scope
- Filtros
- Servicios

Arquitectura de Aplicaciones Web – 1ºC 2018

Directivas

- Las directivas permiten darle nuevos comportamientos al HTML

```
<!DOCTYPE html>
<html ng-app>
<head>
    <title></title>
</head>
<body>
    <div class="container">
        Name: <input type="text" ng-model="name" /> {{ name }}
    </div>
    <script src="Scripts/angular.js"></script>
</body>
</html>
```



Arquitectura de Aplicaciones Web – 1ºC 2018

Directivas

- Son el corazón del framework y definen gran parte de la funcionalidad de Angular.
- Las directivas propias se emplean como atributos de elementos html, de la siguiente forma

```
<tag ng-directiva="expresión"></tag>
```

- Principales directivas:
 - Ng-app: Define el nombre de la aplicación
 - Ng-repeat: Permite repetir una porción de código HTML
 - Ng-controller: Define un controlador
 - Ng-show: Muestra el elemento HTML en función de un booleano
 - Ng-src: Reemplaza al src estandar
 - Ng-route: Funciona de nexo entre directivas y servicios
 - Ng-view: Actualiza los html dados por ng route
 - Ng-model: Bindea los campos de un input a un controlador
 - Ng-bind: Bindea manualmente un elemento a una propiedad

Views, Controllers & Scopes



- \$scope es la forma de comunicar/unir una view con su controllers.
- Similar al concepto de ViewModel

Expresiones

- Angular permite la inserción de código Angular directamente en el HTML, utilizando `{{código}}` que luego es renderizado por el navegador.

```
<p>{{ contador }}</p>
```

Imprimirá el valor de la variable `contador` del controller asociado a la porción de HTML.

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-model

- Relaciona un elemento input,select, textarea (or custom form control) con una propiedad del scope asociado.

```
<script>
    angular.module('inputExample', [])
        .controller('ExampleController', ['$scope', function($scope) {
            $scope.val = '1';
        }]);
</script>
Update input to see transitions when valid/invalid. Integer is a valid value.
<form name="testForm" ng-controller="ExampleController">
    <input ng-model="val" ng-pattern="/^\d+$/" name="anim"/>
</form>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-bind

- El atributo ngBind reemplazará el contenido de texto del elemento HTML especificado con el valor de una expresión dada.
- También actualizará el contenido cuando el valor de la expresión cambie.

```
<script>
    angular.module('bindExample', [])
        .controller('ExampleController', ['$scope'])
        $scope.name = 'Whirled';
    </script>
    <div ng-controller="ExampleController"> Enter name:
        <input type="text" ng-model="name"><br> Hello
        <span ng-bind="name"></span>!
    </div>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-model vs Ng-bind

- **ng-bind** es unidireccional:
 - \$scope --> view
 - Tiene un shortcut: {{ val }} inserta el valor de \$scope.val en el html.
- **ng-model** se utiliza dentro de formularios y en su caso es bi-direccional:
 - \$scope --> view & view --> \$scope

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-repeat

- Instancia una plantilla una vez por cada elemento de una colección.
- Cada instancia de plantilla posee su propio scope.

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-repeat

```
<div
    ng-init="friends = [ {name:'John', age:25, gender:'boy'}, {name:'Jessie', age:30, gender:'girl'}, {name:'Jo
    hanna', age:28, gender:'girl'}, {name:'Joy', age:15, gender:'girl'}, {name:'Mary', age:28, gender:'girl'}, {name:'Pet
    er', age:95, gender:'boy'}, {name:'Sebastian', age:50, gender:'boy'}, {name:'Erika', age:27, gender:'girl'}, {name:'P
    atrick', age:40, gender:'boy'}, {name:'Samantha', age:60, gender:'girl'} ]">
I have {{friends.length}} friends. They are:
<input type="search" ng-model="q" placeholder="filter friends..." />
    <ul class="example-animate-container">
        <li class="animate-repeat"
            ng-repeat="friend in friends | filter:q as results"> [{{$index + 1}}]
            {{friend.name}} who is {{friend.age}} years old. </li>
            <li class="animate-repeat" ng-if="results.length == 0">
                <strong>No results found...</strong>
            </li>
        </ul>
    </div>
```

Arquitectura de Aplicaciones Web – 1*C 2018



Ng-repeat

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales UBA

Variable	Type	Details
\$index	number	iterator offset of the repeated element (0..length-1)
\$first	boolean	true if the repeated element is first in the iterator.
\$middle	boolean	true if the repeated element is between the first and last in the iterator.
\$last	boolean	true if the repeated element is last in the iterator.
\$even	boolean	true if the iterator position \$index is even (otherwise false).
\$odd	boolean	true if the iterator position \$index is odd (otherwise false).

Arquitectura de Aplicaciones Web – 1*C 2018



Además

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales UBA

```

<div
    ng-init="friends = [ {name:'John', age:25, gender:'boy'}, {name:'Jessie', age:30, gender:'girl'}, {name:
    'Johanna', age:28, gender:'girl'}, {name:'Joy', age:15, gender:'girl'}, {name:'Mary', age:28, gender:'girl'}, {nam
    e:'Peter', age:95, gender:'boy'}, {name:'Sebastian', age:50, gender:'boy'}, {name:'Erika', age:27, gender:'girl'},
    {name:'Patrick', age:40, gender:'boy'}, {name:'Samantha', age:60, gender:'girl'} ]">
I have {{friends.length}} friends. They are:
<input type="search" ng-model="q" placeholder="filter friends..." />
<ul class="example-animate-container">
    <li class="animate-repeat"
        ng-repeat="friend in friends | filter:q as results"> [{{$index + 1}}]
        {{friend.name}} who is {{friend.age}} years old. </li>
        <li class="animate-repeat" ng-if="results.length == 0">
            <strong>No results found...</strong>
        </li>
    </ul>
</div>

```

Arquitectura de Aplicaciones Web – 1*C 2018

Ng-if

- Elimina o inserta una porción del árbol DOM en función de la evaluación de una expresión.

Click me:

```
<input type="checkbox" ng-model="checked" ng-init="checked=true" />
<br/> Show when checked:
<span ng-if="checked" class="animate-if">
    I'm removed when the checkbox is unchecked.
</span>
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Ng-show

- Muestra o esconde una porción del árbol DOM en función de la evaluación de una expresión.

```
<!--when $scope.myValue is truthy (element is visible)-->
<div ng-show="myValue"></div>
```

```
<!--when $scope.myValue is falsy (element is hidden)-->
<div ng-show="myValue" class="ng-hide"></div>
```

Arquitectura de Aplicaciones Web – 1ºC 2018

@Override

- AngularJS modifica el comportamiento de muchos tags estandares:

a	input[radio]
form	input[text]
input	input[time]
input[checkbox]	input[url]
input[dateTimeLocal]	input[week]
input[date]	script
input[email]	select
input[number]	textarea

Arquitectura de Aplicaciones Web – 1ºC 2018

Controladores

- Los controladores, son los módulos de una aplicación que definen un comportamiento específico para una porción del HTML.
- Todos los controladores se definen mediante la directiva `ng-controller="nombre"`. El archivo.js en el que esté definido puede o bien ser el mismo en el que se define la aplicación o bien ser un módulo distinto.

Arquitectura de Aplicaciones Web – 1ºC 2018

Controladores



```
<script>
    function SimpleController($scope) {

        $scope.customers = [
            { name: 'Dave Jones', city: 'Phoenix' },
            { name: 'Jamie Riley', city: 'Atlanta' },
            { name: 'Heedy Wahlin', city: 'Chandler' },
            { name: 'Thomas Winter', city: 'Seattle' }
        ];

    }
</script>
```

Basic controller

Arquitectura de Aplicaciones Web – 1*C 2018

Controladores



```
<div class="container" data-ng-controller="SimpleController">
    <h3>Adding a Simple Controller</h3>
    <ul>
        <li data-ng-repeat="cust in customers">
            {{ cust.name }} - {{ cust.city }}
        </li>
    </ul>
</div>
```

\$scope injected dynamically

```
<script>
    function SimpleController($scope) {

        $scope.customers = [
            { name: 'Dave Jones', city: 'Phoenix' },
            { name: 'Jamie Riley', city: 'Atlanta' },
            { name: 'Heedy Wahlin', city: 'Chandler' },
            { name: 'Thomas Winter', city: 'Seattle' }
        ];

    }
</script>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Controladores

- El \$scope esta siempre disponible dentro del fragmento donde esta definido (en este caso: desde el inicio del DIV hasta su cierre).

```
<div class="container" data-ng-controller="SimpleController">
  <h3>Adding a Simple Controller</h3>
  <ul>
    <li data-ng-repeat="cust in customers">
      {{ cust.name }} - {{ cust.city }}
    </li>
  </ul>
</div>                                $scope injected
                                         dynamically
<script>
  function SimpleController($scope) {
    $scope.customers = [
      { name: 'Dave Jones', city: 'Phoenix' },
      { name: 'Jamie Riley', city: 'Atlanta' },
      { name: 'Heedy Wahlin', city: 'Chandler' },
      { name: 'Thomas Winter', city: 'Seattle' }
    ];
  }
</script>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Controladores

- El \$scope esta siempre disponible dentro del fragmento donde esta definido (en este caso: desde el inicio del DIV hasta su cierre).

```
<ul>
  <li data-ng-repeat="cust in customers">
    {{ cust.name }} - {{ cust.city }}
  </li>
</ul>
...                                Access $scope
```

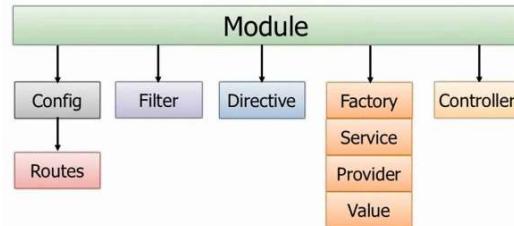
```
<div class="container" data-ng-controller="SimpleController">
  <h3>Adding a Simple Controller</h3>
  <ul>
    <li data-ng-repeat="cust in customers">
      {{ cust.name }} - {{ cust.city }}
    </li>
  </ul>
</div>                                $scope injected
                                         dynamically
<script>
  function SimpleController($scope) {
    $scope.customers = [
      { name: 'Dave Jones', city: 'Phoenix' },
      { name: 'Jamie Riley', city: 'Atlanta' },
      { name: 'Heedy Wahlin', city: 'Chandler' },
      { name: 'Thomas Winter', city: 'Seattle' }
    ];
  }
</script>
```

Arquitectura de Aplicaciones Web – 1*C 2018

Módulos

- Un módulo pueden pensarse como un contenedor para las diferentes partes de su aplicación - controladores, servicios, filtros, directivas, etc

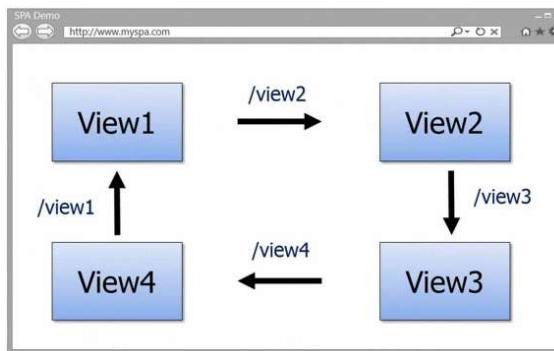
```
<html ng-app="moduleName">
```



Arquitectura de Aplicaciones Web – 1ºC 2018

Routes

- Dado un modulo y un controller, necesitaremos alguna manera de diferenciar diferentes vistas de una misma pagina.



Arquitectura de Aplicaciones Web – 1ºC 2018

Routes

- Qué hacer cuando el path es “/”
- Qué hacer cuando el path es “/partial2”

```

var demoApp = angular.module('demoApp', []);
demoApp.config(function ($routeProvider) {
    $routeProvider
        .when('/', {
            controller: 'SimpleController',
            templateUrl:'View1.html'
        })
        .when('/partial2',
        {
            controller: 'SimpleController',
            templateUrl:'View2.html'
        })
        .otherwise({ redirectTo: '/' });
});

```

Define Module Routes

Arquitectura de Aplicaciones Web – 1*C 2018

Routes & ng-view

- En el modulo demoApp
- Definimos un placeholder para los fragmentos HTML

```

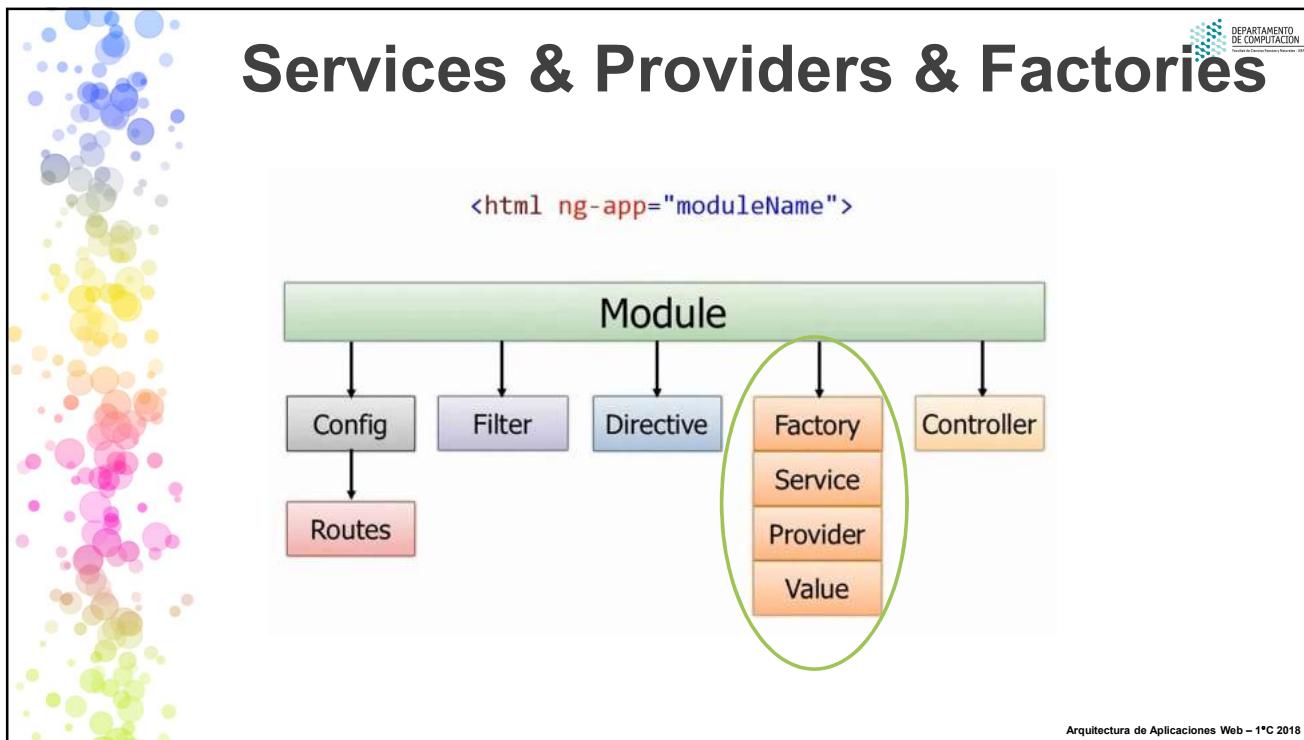
<div>
    <!-- Placeholder for views -->
    <div data-ng-view=""></div>
</div>
<script src="Scripts/angular.min.js"></script>

<script>
    var demoApp = angular.module('demoApp', []);

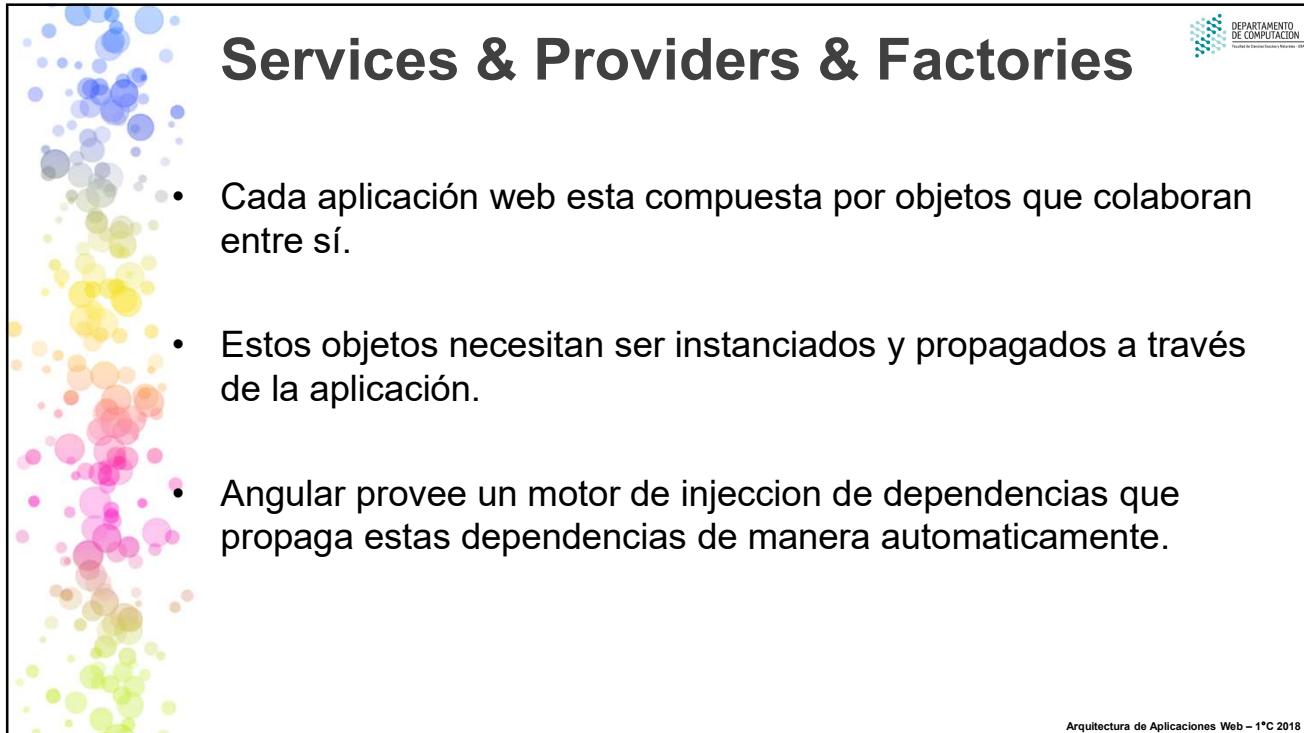
    demoApp.config(function ($routeProvider) {
        $routeProvider
            .when('/', {
                controller: 'SimpleController',
                templateUrl: 'Partials/View1.html'
            })
    });

```

Arquitectura de Aplicaciones Web – 1*C 2018



Arquitectura de Aplicaciones Web – 1*C 2018



Arquitectura de Aplicaciones Web – 1*C 2018

Services & Providers & Factories

- En resumen
 - Sirven para encapsular y reutilizar código
 - Se injecan en los controladores
 - Son funcionalidades de la aplicación no relacionadas con el interfaz gráfico.
 - Nunca acceden a la página (DOM)
 - Son singleton

\$http

- El servicio de \$http es un servicio que facilita la comunicación con los servidores HTTP remotos a través objeto XMLHttpRequest del navegador o mediante JSONP.

```
$http({method: 'GET', url: '/someUrl'}). success(function(data, status, headers, config) {
    // this callback will be called asynchronously
    // when the response is available
}).
error(function(data, status, headers, config) {
    // called asynchronously if an error occurs
    // or server returns response with an error status.
});
```

\$location

- El servicio \$location analiza la URL en la barra de direcciones del navegador (basado en la window.location) y deja la URL a disposición de su aplicación.
- Los cambios en la dirección URL en la barra de direcciones se reflejan en el servicio de localización y viceversa.

```
$location.path(expresion)
```

Arquitectura de Aplicaciones Web – 1ºC 2018

\$exceptionHandler

- Cualquier excepción no detectada en las expresiones angulares se delega a este servicio.
- La implementación por defecto simplemente delega en el \$ log.error que registra en la consola del navegador.

Arquitectura de Aplicaciones Web – 1ºC 2018

\$exceptionHandler

```
var myApp = angular.module('myApp', ['ng']).provider({

  $exceptionHandler: function() {
    var handler = function(exception, cause) {
      alert(exception);
      //I need rootScope here
    };

    this.$get = function() {
      return handler;
    };
  }
);

myApp.controller('MyCtrl', function($scope, $exceptionHandler) {
  console.log($exceptionHandler);
  throw "Fatal error";
});
```

Arquitectura de Aplicaciones Web – 1*C 2018

Services Core

\$anchorScroll	\$location
\$animate	\$log
\$cacheFactory	\$parse
\$compile	\$q
\$controller	\$rootElement
\$document	\$rootScope
\$exceptionHandler	\$sce
\$filter	\$sceDelegate
\$http	\$templateCache
\$httpBackend	templateRequest
\$interpolate	\$t
\$interval	\$timeout
\$locale	\$window

Arquitectura de Aplicaciones Web – 1*C 2018

Services & Providers & Factories

- **Services**
 - Sintaxis : module.service('serviceName', function);

Devuelve una instancia de la función(como alto orden)
- **Factories**
 - Sintaxis : module.factory('factoryName', function);

Devuelve el resultado (valor) despues de invocar la función.
- **Providers**
 - Sintaxis: module.provider('providerName', function);

Devuelve la implementación de la función \$get

Arquitectura de Aplicaciones Web – 1*C 2018

Provider

Provider Function

```
angular.module('myApp', [])
  .provider('myPro', function() {
    console.log('myApp => Create provider => return object with $get');
    return {
      isTrue: false,
      $get: function b($http) {
        var self = this;
        console.log('myApp => my Provider $get => return func');
        return function c(msg) {
          console.log(msg + " isTrue: " + self.isTrue);
        };
      }
    };
  });
app.config(function (myProProvider) {
  myProProvider.isTrue = true;
  console.log('myApp --> config');
});
```

Step 1: Invoke the function before the config stage. No args.

Step 2: This object will be available in config stage as injectable service. The name is "myProProvider".

Step 3: \$get func is a factory func for the service, invoke only if needed and only once. Available after the config stage.

Step 4: The injectable service.

ANGULARJS

Arquitectura de Aplicaciones Web – 1*C 2018

Validaciones



- Angular trae por defecto varias validaciones que son útiles a la hora de implementar formularios.
- Como contrapartida de esto, es fuertemente recomendado desactivar (`<form novalidate/>`) la validación por defecto de HTML5 para que Angular pueda trabajar sin problemas.

Arquitectura de Aplicaciones Web – 1ºC 2018

Validaciones



- Validan el contenido de los `<input>`, no del modelo en el `$scope`
- Se compone de directivas
- Clases CSS
- JavaScript Formulario
- JavaScript `<input>`

Arquitectura de Aplicaciones Web – 1ºC 2018

Directivas en <input>

- ❑ **min:** Valor mínimo de un número
- ❑ **max:** Valor máximo de un número
- ❑ **ngRequired:** Si es requerido
- ❑ **ngMinlength:** Tamaño mínimo de un String
- ❑ **ngMaxlength:** Tamaño máximo de un String
- ❑ **ngPattern:** Expresión regular que debe cumplir

Arquitectura de Aplicaciones Web – 1*C 2018

Clases CSS

- AngularJS establece estas clases en el tag <input>
 - **ng-valid:** Cuando es válido
 - **ng-invalid:** Cuando es inválido
 - **ng-pristine:** Cuando NO se ha modificado su valor
 - **ng-dirty:** Cuando ha sido modificado su valor

Arquitectura de Aplicaciones Web – 1*C 2018

Javascript Formulario

- **nombreFormulario.\$pristine**: Vale 'true' si no se ha modificado ningún valor.
- **nombreFormulario.\$dirty**: Valor 'true' si se ha modificado algún valor.
- **nombreFormulario.\$valid**: Vale 'true' si todos los campos del formulario son válidos.
- **nombreFormulario.\$invalid**: Vale 'true' si algún campo del formulario es inválido.

Arquitectura de Aplicaciones Web – 1ºC 2018

Javascript Formulario

- `nombreFormulario.nombreInput.$error.nombreError` : Vale 'true' si ese error ha fallado.

Siendo nombreError:

- min
- max
- required
- minlength
- Etc.

Arquitectura de Aplicaciones Web – 1ºC 2018

Filtros

- Angular provee un set de filtros útiles para presentar información de la forma que queramos.
- Estos filtros se utilizan de la forma:
`{ expresion | filtro }`
- Angular también permite la definición de filtros customizados.

Arquitectura de Aplicaciones Web – 1*C 2018

Extensiones de Angular

- Existe una gran comunidad que apoya y trata de extender Angular para hacerlo aún más versátil.
- <http://ngmodules.org/>

The screenshot shows the homepage of ANGULAR MODULES, which is a repository for finding reusable AngularJS tools and directives. The page features a search bar at the top right, a 'Sign in with GitHub' button, and navigation links for 'Modules', 'Submit a Module', and 'Tags'. Below the header, there's a section for 'Popular' modules, currently sorted by newest. Three modules are listed:

- abourget-angular** — abourget: A reusable AngularJS tools and directives, mostly Swatch fighters for now. It has 1 person using it. Tags include: directive, service, bootstrap, angular, filter, mobile, AngularJS, backbone, REST, model, activerecord, backbone.model, placeholder.animation, nested.suggestions, input.directive.
- ac-fancy-input** — alexcppns: Enhanced search input with 'placeholder animation' and categorized suggestions for AngularJS. There is no jQuery dependency, just pure angularJS. It has 1 person using it. Tags include: placeholder, input, component, fancyinput, animated.input, placeholder.animation, nested.suggestions, input.directive.
- activerecord** — bfanger: A Backbone Model inspired modellayer for AngularJS. It has 2 people using it. Tags include: backbone, REST, model, activerecord, backbone.model.

To the right of the module list is a 'Popular Tags' sidebar, which lists various Angular-related terms with their counts:

- directive (361)
- service (78)
- bootstrap (43)
- angular (41)
- ui (33)
- filter (27)
- input (24)
- mobile (22)
- AngularJS (21)
- HTML (21)
- form (21)
- upload (19)
- table (18)
- jQuery (16)
- tree (16)
- validation (16)
- forms (16)
- selected (16)
- http (16)
- model (16)
- model (15)
- scroll (15)
- directives (14)
- grid (14)

Extensiones

- ¿Qué significa incorporar una extensión?

```

<script src="vendor/angular-growl/angular-growl.js"></script>
    * Main module of the application.
    */
(function() {
    var app = angular.module('intelliTrackApp', [ 'n
        'ngResource', 'ngRoute', 'ngSanitize', 'l...
        'angularFileUpload', 'ngTable', 'http-auth-interceptor',
        'loginModule', 'contentModule',
        'localStorageModule', 'ngResource
        ('angular-growl', 'ui.bootstrap', 'pascalprecht.translate' ))
            .module('intelliTrackApp')
            .controller(
                'SectorCtrl',
                function($scope, $location, $http, ngTableParams, $rootScope,
                    growl, dialogs) {
                    $scope.paginaActual = 1;
                    $scope.registrosTotales = 0;

                    if (!codigoOk(response)) {
                        var descripcion = response['desc'];
                        growl.error(descripcion);
                    }
                }
            )
        app
        .controller(
    
```

Arquitectura de Aplicaciones Web – 1*C 2018

ngTable

The screenshot shows the official ngTable documentation page. At the top, there's a navigation bar with links for 'Getting Started', 'Examples', 'Need Help?', 'Who is using ngTable?', and social sharing buttons for Facebook, Google+, Twitter, GitHub, and Forks.

The main content area has a sidebar on the left listing various examples: #1: Simple table with pagination, #2: Pagination template, #3: Table with sorting, #4: Table with filters, #5: Saving params in url, #6: **AJAX Data Loading** (which is currently selected), #7: Table with hidden pagination, #8: Cell Template, #9: Row Template, #10: Table with checkboxes, #11: Custom filters, #12: Table with grouping, #13: Table with grouping with callback, #14: Table with external control of data, #15: Export table data to csv, and #16: Editable demo.

The main content area features a section titled 'Example #6: AJAX Data Loading' which displays a table with data from an API. The table has columns for 'Name' and 'Age'. The data includes rows for Enos (false, 34), Jacob (false, 27), Moroni (false, 50), and Nephi (false, 29). Below the table is a pagination control with pages 1, 2, and 3, and a total of 100 pages.

Arquitectura de Aplicaciones Web – 1*C 2018



angularFileUpload

Angular file upload Demo

Visit [angular-file-upload](#) on github

http method: post put

How to upload:

- Multipart/form-data upload using \$upload.upload() service cross browser
- File binary content with the file type as Content-Type header using \$upload.http() service

The second option could be used to upload files to [CouchDB](#), [imgur](#), etc... for HTML5 FileReader browsers.

myModel: model object to be sent with the file.

choose a single file: Seleccionar archivo Ningún archivo seleccionado

or multiple files: Elegir archivos Ningún archivo seleccionado

or only images: Elegir archivos Ningún archivo seleccionado

[Click here to select](#)

You can have any element as an upload button: file

or drop files here

Upload right away

Progress:

Arquitectura de Aplicaciones Web – 1*C 2018



checklist-model

Checklist-model

[AngularJS directive for list of checkboxes](#)

Why this is needed?

In Angular one checkbox `<input type="checkbox" ng-model="...">` is linked with one model. But in practice we usually want one model to store array of checked values from several checkboxes. **Checklist-model** solves that task without additional code in controller. You should play with attributes of `<input type="checkbox">` tag:

1. set `checklist-model` instead of `ng-model`
2. set `checklist-value` - what should be picked as array item

Please, try out demos below:

Array of primitives

demo	user.roles
<input checked="" type="checkbox"/> guest <input checked="" type="checkbox"/> user <input checked="" type="checkbox"/> customer <input type="checkbox"/> admin	<pre>["user", "guest", "customer"]</pre>

Arquitectura de Aplicaciones Web – 1*C 2018

DEPARTAMENTO DE COMPUTACION

Angular Local Storage

Give it a try

guardame

guardame
— Local storage value

Key	Value
demoPrefix.localStorageDemo	guardame

guardame

DEPARTAMENTO DE COMPUTACION

dialogs.main

Editor Log In

HTML

```
<html ng-app="modalTest">
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.1.5/angular.min.js"></script>
    <script src="http://angular-ui.github.io/bootstrap/ui-bootstrap-tpls-0.6.0.js" type="text/javascript"></script>
    <script src="http://m-e-conroy.github.io/angular-dialog-service/javascripts/dialogs.min.js" type="text/javascript"></script>
  </head>
  <body ng-controller="dialogServiceTest" class="pad">
```

CSS

```
/* Fix for Bootstrap 3 with Angular UI Bootstrap */
.modal {
  display: block;
}
.dialog-header-error { background-color: #d2322d; }
.dialog-header-wait { background-color: #428bc4; }
.dialog-header-notify { background-color: #eeeeee; }
```

JS

```
angular.module('modalTest', ['ui.bootstrap','dialogs'])
.controller('dialogServiceTest',function($rootScope,$timeout,$dialogs){
  $scope.confirmed = 'You have yet to be confirmed!';
  $scope.name = '"Your name here."';
  $scope.launch = function(which){
    var dlg = null;
    switch(which){
      case 'error':
        dlg = $dialogs.error('This is m');
      break;
      case 'wait':
        dlg = $dialogs.wait('This is a wait dialog');
      break;
      case 'notify':
        dlg = $dialogs.notify('This is a notify dialog');
      break;
      case 'confirm':
        dlg = $dialogs.confirm('This is a confirmation dialog');
      break;
    }
    if(dlg) dlg.open();
  }
});
```

Bootstrap 3 & AngularJS Dialog/Modals

Error Dialog Wait Dialog Notify Dialog Confirm Dialog Custom Dialog

From Confirm Dialog: Shame on you for not thinking this is awesome!

The screenshot shows a CodePen interface with the following details:

- Header:** The title "dialogs.main" is displayed prominently.
- Code Editor:** The code is divided into three sections: HTML, CSS, and JS.
 - HTML:** Contains the AngularJS template for a modal test, including imports for Bootstrap and Angular UI Bootstrap, and a controller named "dialogServiceTest".
 - CSS:** Provides custom styles for modal components like ".modal" and ".dialog-header-error".
 - JS:** Contains the AngularJS controller logic for handling modal interactions.
- Preview Area:** A modal dialog titled "Please Confirm" is displayed, asking "Is this awesome or what?". It includes buttons for "Yes" and "No".
- Footer:** A note at the bottom of the preview area says "From Confirm Dialog: Shame on you for not thinking this".

The screenshot shows a web browser window with a decorative background of colored circles. The address bar shows 'localhost:8080/verific...'. The main content is a login form for 'Intellipos VERIFICADORES'. The form has two input fields, one containing 'noexiste' and another containing '...', and a blue 'Ingresar' button. Above the form is a red alert box with a shield icon and the text 'El usuario y/o contraseña son inválidos'. To the right of the browser window, a portion of the underlying HTML and JavaScript code is visible. A green arrow points from the error message in the browser to the 'response' variable in the code.

```
        response) {  
  
    ];  
    varPermisos');  
    s']);  
  
    ged', true);  
    , token);  
    ario', usuario);  
    , nivelPermisos);  
    , acciones);  
  
    ...;  
  
    $location.path('/sincronizacion');  
    swindow.location.reload();  
} else {  
    growl.error(response['desc']);  
}  
}
```

The screenshot shows the Angular Modules website interface. On the left, there's a decorative sidebar with a vertical column of colored circles (blue, green, yellow, red) of varying sizes. The main content area has two sections:

- ngFacebook**: A treeview menu under the heading "Angular facebook service". It includes a "Homepage:" link to [GoDiscord](#), an "Author:" link to [AlmogBaku](#), and a "View the Project on GitHub" button.
- ng-nvd3**: A section for the "ng-nvd3" module. It features a pie chart with three segments: 47% (blue), 40% (orange), and 13% (light blue). The legend indicates: ● Worse (light blue), ● Same (white), ● Better (orange). Below the chart is a "How to use?" section with the following code snippet:

```
Angular directive in html:  
<nvd3-pie-chart ...></nvd3-pie-chart>
```

At the bottom right of the main content area, it says "Arquitectura de Aplicaciones Web – 1*C 2018". In the top right corner of the slide, there's a logo for "DEPARTAMENTO DE COMPUTACION Facultad de Ciencias Exactas y Naturales UBA".

The screenshot shows the Angular Modules website interface. On the left, there's a decorative sidebar with a vertical column of colored circles (blue, green, yellow, red) of varying sizes. The main content area has one section:

Un mundo feliz?

- Es muy fácil, a medida que se complejiza la vista, terminar con un código spaghetti, mezclándose javascripts, directivas, expresiones, html y demás yerbas, terminando por hacer más compleja la aplicación de lo que debería ser. Angular permite y promociona una modularización intensa, la cual debería ser aplicada.

At the bottom right of the main content area, it says "Arquitectura de Aplicaciones Web – 1*C 2018". In the top right corner of the slide, there's a logo for "DEPARTAMENTO DE COMPUTACION Facultad de Ciencias Exactas y Naturales UBA".

Un mundo feliz?

- Al igual que cualquier herramienta, a mayor poder y flexibilidad, mayor complejidad. Angular no es la excepción y su uso comprende una curva de aprendizaje superior a la mayoría de los restantes frameworks javascripts.

Arquitectura de Aplicaciones Web – 1ºC 2018

Seguridad

- En este punto el problema es la naturaleza misma del lenguaje sobre el que está hecho Angular: Javascript.
- Es muy importante tener en cuenta que cualquier característica de seguridad en la aplicación debe ser repetida en el servidor.
- El código javascript que se ejecuta en el cliente puede ser perfectamente modificado por el cliente, por lo que se debe pensar en términos de seguridad e n profundidad y siempre verificar permisos y cualquier aspecto de seguridad en el servidor.

Arquitectura de Aplicaciones Web – 1ºC 2018

Seguridad: hay herramientas

- Extensión que intercepta todos los requerimientos HTTP para validar su origen, permisos y autenticación.

The screenshot shows the Angular Modules website with a search bar and navigation links for 'Modules', 'Submit a Module', and 'Tags'. A specific module, 'http-auth-interceptor', is highlighted. The module's page includes its GitHub link, download button, and popularity metrics (69 people use it). Below the header, there are tabs for 'README' and 'Discussion'. The main content area is titled 'HTTP Auth Interceptor Module for AngularJS' and describes it as an implementation of authentication management for AngularJS. It also provides a demo link and a note about switching to the gh-pages branch for source code.

Arquitectura de Aplicaciones Web – 1*C 2018

AngularJS Batarang

The screenshot shows the Chrome Web Store page for 'AngularJS Batarang'. The page has a rating of 4.5 stars from 300 reviews and is categorized under 'Herramientas para desarrolladores'. It is described as a tool that extends the Developer Tools for debugging and profiling AngularJS applications. A video thumbnail in the center of the page shows a developer using the extension in the browser's developer tools interface.

Arquitectura de Aplicaciones Web – 1*C 2018

The screenshot shows a web browser window with the URL localhost:8080/verificadores-web/#/nuevaEmpresa. The page title is "IntelliJ Verificadores". The main content is a form titled "Nueva empresa" with fields for "Código" (with placeholder "Ingrese el código"), "Razón Social" (placeholder "Ingrese la razón social"), and "Dirección" (placeholder "Ingrese la dirección"). To the left is a sidebar with links: Empresas, Sectores, Sucursales, Terminales, and Usuarios. Below the sidebar is a "Scopes" section showing a tree view of scopes from 002 to 008. To the right is a "Models for (007)" section showing a JSON-like code block:

```
{  
  messages: []  
  inlineMessage: false  
  deletedMessage: null  
  alertClasses: null  
  wrapperClasses: null  
  computeTitle: null}
```

At the bottom of the browser window, there are tabs for Models, Performance, Dependencies, Options, Help, and a checked checkbox for Enable. The browser's address bar shows multiple tabs related to the application.

Arquitectura de Aplicaciones Web – 1ºC 2018



AngularJS Batarang

The screenshot shows the Bootstrap logo, which consists of a large purple square containing a white letter 'B', followed by the word "Bootstrap" in a bold, purple, sans-serif font.

Arquitectura de Aplicaciones Web – 1ºC 2018



¿Qué es Bootstrap?



- **Twitter Bootstrap** es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.
- En agosto del 2011, Twitter liberó a Bootstrap como código abierto

Arquitectura de Aplicaciones Web – 1*C 2018

Responsive Web Design



- El objetivo es brindar una experiencia visual óptima para la diversa variedad de dispositivos actualmente existente.



Arquitectura de Aplicaciones Web – 1*C 2018

Elementos del RWD



- Fluid Grid
- Resizable Images
- Media Queries

Arquitectura de Aplicaciones Web – 1ºC 2018

960 Grid System



- El sistema de cuadrícula 960 es simplemente una manera de diseñar sitios web usando una rejilla de 960 píxeles de ancho. La razón es 960 píxeles de ancho se debe a que el número 960 simplifica las divisiones en columnas y los márgenes de la plantilla de diseño.
- El 960 GS viene en dos variantes principales: una cuadrícula de 12 columnas y una cuadrícula de 16 columnas. También incluye una versión de 24 columnas para diseños web en los que necesitemos incluir una mayor densidad de contenidos.

Arquitectura de Aplicaciones Web – 1ºC 2018

960 Grid System

- En la versión de 12 columnas, la columna más estrecha es de 60 píxeles de ancho, incrementándose después en 80 píxeles. De esta forma, los anchos de las columnas disponibles son: 60, 140, 220, 300, 380, 460, 540, 620, 700, 780, 860 y 940.

The diagram illustrates the 12 Column Grid System. It consists of 12 horizontal pink bars representing columns. The total width is 940 pixels. The widths of the columns are: 60, 140, 220, 300, 380, 460, 540, 620, 700, 780, 860, and 940. The first column is 60 pixels wide, followed by 140, 220, 300, 380, 460, 540, 620, 700, 780, 860, and the last one is 940 pixels wide. The grid is labeled "12 Column Grid" at the top.

960 Grid System: disposición en 12 columnas

960 Grid System

- En la versión en 16 columnas, la columna más estrecha es de 40 píxeles de ancho y después se incrementa en 60 píxeles. Así que el ancho de las columnas disponibles son: 40, 100, 160, 220, 280, 340, 400, 460, 520, 580, 640, 700, 760, 820, 880 y 940.

The diagram illustrates the 16 Column Grid System. It consists of 16 horizontal pink bars representing columns. The total width is 940 pixels. The widths of the columns are: 40, 100, 160, 220, 280, 340, 400, 460, 520, 580, 640, 700, 760, 820, 880, and 940. The first column is 40 pixels wide, followed by 100, 160, 220, 280, 340, 400, 460, 520, 580, 640, 700, 760, 820, 880, and the last one is 940 pixels wide. The grid is labeled "16 Column Grid" at the top.

960 Grid System

- Cada una de las barras horizontales podemos relacionarla con una clase CSS en el sistema 960 Grid, de forma que le asignaremos una capa (div) con el ancho adecuado.
- La nomenclatura de cada uno de los elementos del CSS será acorde con el ancho del elemento de la rejilla, de forma que la clase CSS grid_1 representará a la columna más estrecha y la clase grid_12 será la que aplicaremos a la columna más ancha para un sistema de 12 columnas.



Aplicaciones Web – 1ºC 2018

960 Grid System

- Por ejemplo, si utilizamos el sistema de 12 columnas, simplemente tendremos que utilizar la clase grid_4 para incluir uniformemente tres capas de texto que ocupen el diseño de lado a lado:



Arquitectura de Aplicaciones Web – 1ºC 2018

Resizable Images

- Con CSS3 es posible redimensionar las imágenes dinámicamente.

```
img {  
    max-width: 100%;  
    height: auto;  
}  
• Esto se complementa con:  
    – Relative Font Size  
    – Relative Margin  
    – Etc...
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Media Queries

- Es un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo como la resolución de pantalla (por ejemplo, un smartphone frente a pantallas de alta definición) o la presencia de características de accesibilidad como el braille.

```
<!-- CSS media query on a link element -->  
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />  
  
<!-- CSS media query within a style sheet -->  
<style> @media (max-width: 600px)  
{  
    .facet_sidebar { display: none; }  
}  
</style>
```

Arquitectura de Aplicaciones Web – 1ºC 2018



Media Queries



DEPARTAMENTO
DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales

- Media Types
 - braille
 - embossed
 - handheld
 - print
 - projection
 - screen
 - speech
 - tty
 - tv

Atributos Media

La siguiente tabla contiene los atributos media recibidos de la última recomendación de la W3C para media queries

Atributo	Valor	Min/Max
color	integer	si
color-index	integer	si
device-aspect-ratio	integer/integer	si
device-height	length	si
device-width	length	si
grid	integer	no
height	length	si
monochrome	integer	si
resolution	resolution ("dpi" or "dpcm")	si
scan	"progressive" or "interlaced"	no
width	length	si

Arquitectura de Aplicaciones Web – 1*C 2018



Ejemplo



Granada, España

Donec condimentum, id velit dignissim duis id congue sapien leo tincidunt est. Mauris consectetur tempus lorem id aliquet. Proin eu fructibus massa. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In magna ligula, ornare sed posuere faucibus, consequat ac lacus. Fusce sodales fermentum nibh, a imperdiet nisi blhendum eget. Donec gravida iaculis sapien eu consectetur. Curabitur id augue augue. Nam mauris urna, suscipit eget faucibus sit amet, mollis vitae felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris ipsum lectus, imperdiet id aliquam nec, vulputate vitae mauris. Integer grava, neque eu placerat egestas, urna mutus blhendum nisi, quis congue felis velit id diam. Quisque nisl titor at turpis facilisis placerat quis sed felis. Ut eget ipsum dolor, id lacinia leo. Vivamus vitae blandit libero. Integer ultrices grava leo quis lobortis. Morbi ultrices risus vulputate magna dignissim sed ultricies arcu tristique. Sed non facilisis sapien.



```

body{
  background:#f5f5f5;
  font-family: Arial, sans-serif;
  font-size:13px;
  line-height:1.6em;
  color:#444;
}

p{
  margin:15px 0;
}

h2{
  margin-top:20px;
}

#container{
  background:#fff;
  border-left:1px #ddd solid;
  border-right:1px #ddd solid;
  border-bottom:1px #ddd solid;
  width:600px;
  margin:0 auto;
}

header h1 a{
  text-indent:-9999px;
  display:block;
  width:600px;
  height:98px;
  background:url(image-med.jpg) no-repeat 50% 0;
}

#content{
  padding:0 15px;
}

```

Arquitectura de Aplicaciones Web – 1*C 2018

Ejemplo

Granada, España

Constituyentes:

```

body{
  background:#f5f5f5;
  font-family: Arial, sans-serif;
  font-size:13px;
  line-height:1.6em;
  color:#444;
}

p{
  margin:15px 0;
}

h2{
  margin-top:20px;
}

#container{
  width:1100px;
  margin:0 auto;
  padding:0 15px;
}

@media screen and (min-width:1200px) {
  img {
    max-width:100px;
  }

  #container{
    width:1100px;
  }

  header h1 a{
    width:1100px;
    height:180px;
    background:url(image-small.jpg) no-repeat 0 0;
    position: absolute;
    left: 50% 0;
    top: 50% -50px;
  }
}

```

Arquitectura de Aplicaciones Web – 1*C 2018

Editores Online

- <http://www.bootply.com/>

Bootply Run Save Visuals

Settings Bootstrap 3.1.1 jQuery 2.0.2 More includes Mermaid

Name, describe tag:

Free web app performance monitoring and alert when you try New Redis, Redis via Carbon

Find premium templates & themes

Community

Templates

Brand Navbar Item 1 Navbar Item 2 Navbar Item 3

Sidebar Item 1 Admin

Sidebar Item 2

1 body {
2 padding-top: 50px;
3 overflow: hidden;
4 }
5 #expper {
6 height: 100%;
7 width: 100%;
8 position: absolute;
9 }

Arquitectura de Aplicaciones Web – 1*C 2018

Editores Online

- <http://www.bootply.com/>

The screenshot shows the Bootply online editor interface. On the left, there's a sidebar with navigation options like 'Themes', 'Communities', 'Templates', 'Brand', 'Sidebar 1', and 'Sidebar 2'. The main area has a preview window showing a mobile device icon and some icons for download and cloud storage. Below the preview is a code editor with CSS and JavaScript code. The code editor contains snippets such as 'Brand' (with items 1, 2, and 3), 'Admin' (with a placeholder for a logo), and a dropdown menu section.

Editores Online

- <http://www.layoutit.com/build?r=85243880>

The screenshot shows the LayoutIt! online editor interface. On the left, there's a sidebar with 'GRID SYSTEM', 'RAISE CSS', 'COMPONENTS', and 'JAVASCRIPT'. The main area has a preview window showing a 'Hello, world!' message and a 'Hero' component. Below the preview is a 'Row' component with two 'Column' sections, each containing a 'Heading' and another 'Column' section. The code editor on the right shows the HTML and CSS for these components.

CSS con Superpoderes

The logo for Sass, featuring the word "Sass" in a stylized, handwritten font. The letters are primarily pink with some purple and white highlights.

Arquitectura de Aplicaciones Web – 1*C 2018

Syntactically Awesome Stylesheets



- ¿Que pasaría si pudiéramos usar variables en las hojas de estilo?
- ¿Nunca han sentido que en lugar de estar repitiendo decenas de líneas sería mejor anidarlas?

Arquitectura de Aplicaciones Web – 1*C 2018

Sass

- Sass es un pre-procesador CSS.
- La idea es “compilar” los archivos Sass para generar archivos CSS.

Arquitectura de Aplicaciones Web – 1ºC 2018

Características de Sass

- **Variables**

- Las variables nos permiten guardar datos para ser reutilizados

```
$color:red;  
$size:10px;  
  
p{  
    color:$red;  
    font-size:$size + 6;  
}
```

Arquitectura de Aplicaciones Web – 1ºC 2018

Características de Sass

- **Mixins**

- Los mixins nos permiten guardar un conglomerado de datos para ser reutilizados

```
@mixin caja{  
  display:inline-block;  
  margin:10px;  
  padding:10px;  
  vertical-align:top;  
}  
#cajaUno{  
  width:50%;  
  @include caja;  
}  
#cajaDos{  
  width:35%;  
  @include caja;  
}
```

Arquitectura de Aplicaciones Web – 1*C 2018

Características de Sass

- **Anidar elementos**

- Los mixins nos permiten guardar un conglomerado de datos para ser reutilizados

```
ul.menu{  
  background:rgba(51,51,51,0.9);  
  list-style:none;  
  
  li{  
    display:inline-block;  
  
    a{  
      color:#ffff;  
      text-decoration:none;  
      padding:10px 20px;  
  
      &:hover{  
        color:white;  
      }  
    }  
  }  
}
```

Arquitectura de Aplicaciones Web – 1*C 2018

Compass

- Compass nos facilita un montón de mixins que se le agregan a Sass, como border-radius, box-shadow y gradientes



Arquitectura de Aplicaciones Web – 1ºC 2018

Mixins en Compass

- Compass son un montón de mixins para reutilizar

Código

```
box{  
  border:1px solid black;  
  height:100px;  
  width:100px;  
  @include border-radius(5px);  
}
```

CSS

```
box{  
  border:1px solid black;  
  height:100px;  
  width:100px;  
  border-radius:5px;  
  -webkit-border-radius:5px;  
  -moz-border-radius:5px;  
}
```



Arquitectura de Aplicaciones Web – 1ºC 2018

Mixins en Compass

- Compass son un montón de mixins para reutilizar

Código

```
box{  
width:100px;  
height:100px;  
@include border-radius(5px);  
@include box-shadow($shadow-1);  
}
```

CSS

```
box{  
border:1px solid black;  
height:100px;  
width:100px;  
border-radius:5px;  
-webkit-border-radius:5px;  
-moz-border-radius:5px;  
box-shadow:0px 1px 0px 1px black;  
-webkit-box-shadow:0px 1px 0px 1px black;  
-moz-box-shadow:0px 1px 0px 1px black;  
}
```



Arquitectura de Aplicaciones Web – 1ºC 2018

Desarrollando Aplicaciones

Server Side – Client Side
Frontend – Backend – Fullstack
Consideraciones....

Arquitectura de Aplicaciones Web – 1ºC 2018

