

Curso de Programación para Internet con Java EE (Parte IV - Aplicaciones Web)

Antonio Santos Ramos

antoniosantosramos@gmail.com

Curso de Programación para Internet con Java EE

(29 Octubre 2018 – 14 Diciembre 2018)_{v14}

Java (Lucatc) (8h-17h) Puente 2/11, 9/11 y 07/12

© Antonio Santos Ramos 2018

1

Curso de Programación para Internet con Java EE (Parte IV - Aplicaciones Web)

T0 - Índice

© Antonio Santos Ramos 2018

2

Contenido

Fundamentos de Programación en Java (Parte IV - Aplicaciones Web)

Temario

- T00 – Web-based Programming
- T01 – HTML
- T02 – Composición de páginas
- T03 – CSS. Hojas de estilo
- T04 – Javascript
- T05 – AngularJS



© Antonio Santos Ramos 2018

3

Curso de Programación para Internet con Java EE (Parte IV - Aplicaciones Web)

T00 – Web Based Programming

Principales lenguajes de edición

Lenguajes de lado cliente (páginas estáticas)

- HTML
- XHTML
- JavaScript
- VBScript
- CSS

DHTML (Dynamic Hypertext Mark-Up Language)
Es una combinación de HTML, CSS, JS y DOM

Lenguajes de lado servidor (páginas dinámicas)

- ASP
- JSP
- PHP
- ASP.NET

Editores Web (I/IV)

■ Editores Web

- Herramientas para gestionar páginas HTML (y sitios)
- Básicamente es un editor de texto “con esteroides”

■ Ejemplos

- **Adobe DW CC:** <http://www.adobe.com/es/products/dreamweaver.html>
- **Brackets:** <http://brackets.io/>
- **Visual Studio Code:** <https://code.visualstudio.com/>
- **Atom:** <https://atom.io/>
- **Sublime Text:** <https://www.sublimetext.com/>
- **Blue Griffon:** <http://bluegriffon.org/>
- **Aptana Studio, BBEdit, Kompozer, Coda (Mac), WebStorm, Rapidweaver (Mac), ...**

<http://www.emezeta.com/articulos/los-mejores-editores-de-texto-para-programar>

<http://www.campusmvp.es/recursos/post/Los-10-mejores-editores-gratuitos-de-HTML-CSS-y-JavaScript.aspx>

<http://www.genbeta.com/web/los-7-mejores-editores-de-codigo-online-para-programar>

Editores Web (II/IV)

- Brackets (Nov 2014)
 - Escrito en HTML, CSS y Js por Adobe. Evolución de Edge Code
 - Incluye Live Preview y uso de extensiones
- Uso
 - Código + TAB: expandir
 - CTRL + E: edición CSS
 - CTRL + K: Información adicional
- Extensiones
 - <http://www.johnpapa.net/my-recommended-brackets-extensions/>
 - <https://brackets-registry.aboutweb.com/>
 - <https://github.com/adobe/brackets/wiki/Brackets-Extensions>

© Antonio Santos Ramos 2018

7

Vídeos: <https://www.youtube.com/user/CodeBrackets>

Editores Web (III/IV)

- Emmet. Abreviaturas HTML <http://docs.emmet.io/cheat-sheet/>
 - Elementos: p, div.resumen, dir#header
 - Elementos Hijo: div.header > div.main
 - Elementos hermanos: .outer>.inner>h1+p
 - Varios elementos: .wrap>ul.list>.sites*4
 - Varios elementos con nombre: ul>li.item\$*3
 - Añadiendo texto: ul>li.item\${item \$\$}*3
 - Mejoras en etiquetas: a:link.acciones{Haz clic \$}
 - Agrupaciones: (.one>h1)+(.two>h1)
 - Climb Up: header>#main^footer



Emmet.io
emmet.io

<http://www.cheat-sheets.org/saved-copy/ZenCodingCheatSheet.pdf>

Editores Web (IV/IV)



■ Extension Emmet. Elementos rápidos para estructura HTML

■ html:5 meta:vp link:css link:favicon

■ Ejemplo 01

Home
About
Portfolio
Contact

Seleccionar
+
CTRL+MAY+A

```
nav>ul>
li[title=$#]*>
a[href=$#.html]{$#}>
img[src=$#.png alt="$#"].nav-bg
```

■ Ejemplo 02

```
html:5>
(header>h1+.titular+nav>ul>
    (li>a[href=#]{Enlace $})*4)+ 
(main>(article>(header>h2{Titular $})+
    (p>lorem25)+footer>span>(a[href=#].acciones{Haz clic $})*2)*4)+ 
(aside>img[src="images/im$.jpg"]*4)+ 
(footer>p>lorem5)
```

© Antonio Santos Ramos 2018

<https://scotch.io/tutorials/write-html-crazy-fast-with-emmet-an-interactive-guide>

9

<http://www.adictosaltrabajo.com/tutoriales/emett/>

Code Playgrounds



<http://www.editoresdecodigo.com/2014/06/5-mejores-editores-codigo-online.html>

<http://alternativeto.net/software/jsfiddle/>

<http://www.genbeta.com/web/los-7-mejores-editores-de-codigo-online-para-programar>

<https://codegeekz.com/best-code-playgrounds-for-developers/>

<http://techslides.com/30-code-playgrounds-and-sharing-tools>

<http://mediaunmasked.com/reviews/software/codepen-jsfiddle-cssdeck-liveweave-jsbin-dabblet/>

<http://www.sitepoint.com/7-code-playgrounds/>

https://en.wikipedia.org/wiki/Comparison_of_online_source_code_playgrounds

© Antonio Santos Ramos 2018

10

Curso de Programación para Internet con Java EE

(Parte IV - Aplicaciones Web)

T01 – HTML

© Antonio Santos Ramos 2018

11

El lenguaje HTML-XHTML

HTML 4.01 (1999)
XHTML 1.1 (2010)
HTML 5 (28/10/2014)

- Es un lenguaje de marcas
 - <p> ... </p>
 -
 -
- El lenguaje se forma con tres tipos de elementos
 - **Etiquetas** o tags
 - Ej.- <p> Esto es un párrafo</p>
 - **Atributos**: Son modificadores o argumentos y dan forma a las etiquetas. **Siempre** se colocan en las etiquetas de apertura
 - Ej.- Web
 - **Valores**: propiedades con un valor concreto
 - Ej.- en el caso anterior sería el valor "http://www..."

```
<etiqueta attr1="val1" attr2="val2" ... > ... </etiqueta>
```

Estructura de un documento HTML (I/VI)

- Los documentos HTML sólo tienen una única raíz
 - Viene indicada por la etiqueta `<html> </html>`
- El elemento raíz puede contener dos elementos:
 - Cabecera:
 - Indicada por la etiqueta `<head></head>`
 - Contiene el título de la página, tipo de documento, meta información y las funciones de script.
 - Cuerpo:
 - Indicado por la etiqueta `<body></body>`
 - Contiene el cuerpo del documento, es decir, lo que se mostrará en la pantalla del navegador.

Estructura de un documento HTML (II/VI)

Ejemplo

```
<html>
  <head><title>Pagina HTML</title></head>
  <body>
    <h1>Cabecera del documento</h1>
    <p>Primera linea del documento</p>
    <p>Segunda linea del documento<br/>
      <a href="http://www.increibleperociento.es">Web</a></p>
    </body>
</html>
```



Estructura de un documento HTML (III/VI)

Cabecera

- Describe parámetros y características generales.
- En la cabecera de un documento HTML puede aparecer:
 - **El título del documento.** Etiqueta: **<title>**
 - Aparecerá en el título de la ventana del navegador.
 - **Código script** a ejecutar en el navegador. Etiqueta: **<script>**
 - JavaScript es el lenguaje de script más difundido. Cualquier navegador puede ejecutarlo
 - **Meta-information** sobre el documento. Etiqueta: **<meta>**
 - Estas etiquetas debe interpretarlas el navegador.
 - Informan sobre aspectos auxiliares: descripción del contenido, comportamiento de bots, mapas de caracteres, etc.

Estructura de un documento HTML (IV/VI)

Cabecera (Ejemplo)



```
<!doctype html> 5
<html>
<head>
    <title>Mi Primera pagina</title>
    <meta charset="utf-8" />
    <meta name="Autor" content="Antonio Santos" />
    <meta name="keywords" content="HTML, curso, web, internet" />

    <script language="JavaScript" type="text/javascript">
        src="prueba.js"</script>
</head>
<body>
    <h1><strong>Hola clase</strong></h1>
    <p><a href="http://www.increibleperociertito.es">Web</a></p>
</body>
</html>
```



Estructura de un documento HTML (V/VI)

DOCTYPE



HTML 4.01 Transicional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
      "http://www.w3.org/TR/html4/loose.dtd">
```

■ XHTML 1.0 Transitional

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

■ XHTML 1.0 Strict

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

■ XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
      "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

✓ HTML 5

```
<!DOCTYPE html>
```

Estructura de un documento HTML (VI/VI)

Cuerpo

- El cuerpo de un documento HTML es la parte que se mostrará en el navegador.
- Viene delimitado por la etiqueta **<body>**
- Los elementos principales incluidos en el cuerpo son:
 - Elementos de texto
 - Párrafos
 - Encabezados
 - Listas
 - Enlaces
 - Imágenes

HTML. Validaciones

■ Direcciones

- <https://validator.w3.org/>
- <https://html5.validator.nu/>

The screenshot shows a web-based HTML validator. At the top, there are three tabs: 'Validate by URI' (which is selected), 'Validate by File Upload', and 'Validate by Direct Input'. Below the tabs, there's a section titled 'Validate by URI' with the sub-instruction 'Validate a document online:'. A text input field labeled 'Address:' contains a URL. To the right of the input field is a 'Check' button. Below the input field, there's a link 'More Options'.

■ HTML 5 permite...

-
 ... <p> ... <p> ... </P>
- Void Elements: br, img, input, hr, link, meta, ...
 - XHTML: <foo /> significa <foo> </foo>
 - HTML 5: <foo /> significa <foo>. Soporta ambos

<http://www.impressivewebs.com/understanding-html5-validation/>

<http://stackoverflow.com/questions/3558119/are-self-closing-tags-valid-in-html5>

Consejos (subjetivos) extraídos de normas XHTML

- Etiquetas y atributos en **minúsculas** (XML es sensible)
- **Elementos se cierran** (
<p>... no necesitan en HTML5)


```
<p>Esto es un párrafo</p>
<br /> (lleva un espacio)

```
- Valores de atributos **entrecomillados**

```
<table width="100%">
```
- Elementos **bien emparejados**

```
<div><p>Este texto es bold e italic

```
- Se puede **minimizar atributos** (en algunos casos)


```
<input checked>
```

HTML.

Elementos de texto

■ Etiquetas para textos:

- Encabezados: <h1> ... <h6>
- Párrafos: <p> ... </p>
- Listas (lo veremos más adelante):
- Enlaces: enlace

■ Para enfatizar: http://www.w3schools.com/tags/tag_phrase_elements.asp

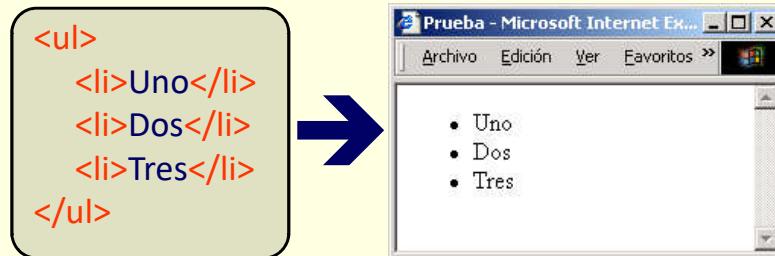
```
<strong> <address>
<cite>
<dfn> <code>
<samp> <var> <q>
<kbd> <abbr>
<blockquote> <em>
```

HTML. Listas

■ 1º) Definir el tipo de lista que se va a utilizar

- Listas numeradas, etiqueta
- Listas sin numerar, etiqueta

■ 2º) Marcar cada elemento de la lista mediante la etiqueta



- Las listas pueden tener sublistas
- Para definiciones usar <dl><dt><dd>

HTML.

Imágenes

■ Elementos clásicos

<http://www.w3.org/wiki/HTML/Elements/img>

- Etiquetas: **img**
- Atributos básicos: **width**, **height**, **src**
- Descripciones se usan **alt** (si no existe), **title** (tooltip)

```
<p><p>
```

■ Elementos nuevos en HTML 5: **figure** y **figcaption**



```
<figure>
  
  <figcaption>Fig1-Smiley face</figcaption>
</figure>
```

<http://html5doctor.com/the-figure-figcaption-elements/>

<https://longhandpixels.net/blog/2014/02/complete-guide-picture-element>

© Antonio Santos Ramos 2018

23

HTML

<div>

- La etiqueta **<div>** define una división en la página
- Es un elemento de bloque y permite agrupar dentro cualquier elemento

```
<html>
<body>

  <p>Texto antes de la caja</p>

  <div id="noticia" style="background-color:#dddddd">
    <h3>Este es el encabezado</h3>
    <p>Este es el párrafo</p>
  </div>

  <p>Texto fuera de la caja</p>

</body>
</html>
```

Texto antes de la caja

Este es el encabezado

Este es el párrafo

Texto fuera de la caja

© Antonio Santos Ramos 2018

24

HTML.

Tablas

■ Elementos principales de HTML:

- Elemento raíz: **<table>**
- Inicio de fila: **<tr>**
- Columna: **<td>**
- Cabeceras: **<th>** (en vez de **<td>**)

Columna 1	Columna 2
Fila 1 Col 1	Fila 1 Col 2
Fila 2 Col 1	Fila 2 Col 2
Fila 3 Col 1	Fila 3 Col 2

```

<table border ="1">
  <tr>
    <th>Columna 1</th>
    <th>Columna 2</th>
  </tr>
  <tr>
    <td>Fila 1 Col 1</td>
    <td>Fila 1 Col 2</td>
  </tr>
  <tr>
    <td>Fila 2 Col 1</td>
    <td>Fila 2 Col 2</td>
  </tr>
  <tr>
    <td>Fila 3 Col 1</td>
    <td>Fila 3 Col 2</td>
  </tr>
</table>

```

© Antonio Santos Ramos 2018

25

HTML. Formularios (I/IV)

- **Formulario:** conjunto de elementos que permite al usuario enviar información al servidor para su procesamiento.
- HTML utiliza la etiqueta **<form>** con los siguientes atributos:
 - **action:** indica la URL a la que se enviará el formulario.
 - Ej.- `action="http://www.respuesta.com/validacion.php"`
 - **method:** indica el método HTTP con el que se enviará
 - Su valor debe ser GET o POST.
 - Siempre debe ser POST. GET sólo para pruebas

```

<form action="http://www.respuesta.com/v.php" method="POST">
  http://www.10sharpdesign.com/accessibility/forms/index.html

```

HTML.

Formularios (II/IV)

■ Piezas

- Cuadro de texto
- Área de texto
- Lista de selección
- Casillas de verificación
- Radio-button
- Botones
 - Botón de envío
 - Botón de borrado
 - Botón de función

■ TODO lleva un **id** y **name**

SUSCRIPCIÓN A REVISTA

Cliente: Asantos

Contraseña: *****

Indique la publicación

Revista 1
Revista 2

Forma de pago VISA/MasterCard Contra reembolso

Formato

Recibirla en PDF
 Recibirla impresa

Observaciones

Creo que no disponen de mi dirección de correo electrónico

Enviar Restablecer

© Antonio Santos Ramos 2018

27

HTML. Formularios (III/IV)

[Validaciones HTML 5]

Input Type	Description	HTML	HTML Markup
date	A control for entering the date.		<input type="date">
datetime	Date and time using UTC date and time format		<input type="datetime">
datetime-local	Date and time according to your local time		<input type="datetime-local">
month	Month and year		<input type="month">
time	The time of day		<input type="time">
week	Allows you to pick the week and year.		<input type="week">
color	Allows you to enter a simple color value (which is in hexadecimal color notation)		<input type="color">
email	Validates the input using the standard email format		<input type="email">
tel	Gives you the ability to validate telephone numbers format against a pattern		<input type="tel">
search	Searches a data set (like a <datalist> HTML element)		<input type="search">
range	A slider control for picking a number in between two numbers		<input type="range">
number	Accepts numbers only		<input type="number">
url	Accepts URLs only		<input type="url">

<http://sixrevisions.com/html5/new-html5-form-input-types/><http://www.coreservlets.com/html5-tutorial/input-types.html><http://diveintohtml5.info/forms.html><http://html5doctor.com/html5-forms-input-types/>

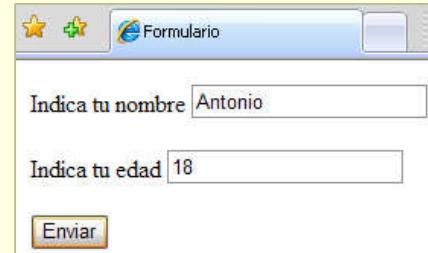
© Antonio Santos Ramos 2018

28

HTML. Formularios (IV/IV)

[Funcionamiento]

```
<form id="form1" method="get" action="respuesta.htm">
  <div>
    <label for="nombre">Indica tu nombre</label>
    <input type="text" name="nombre" id="nombre" />
  </div>
  <div>
    <label for="edad">Indica tu edad</label>
    <input type="text" name="edad" id="edad" />
  </div>
  <div>
    <input type="submit" name="enviar"
           id="enviar" value="Enviar" />
  </div>
</form>
```



<http://.../respuesta.htm?nombre=Antonio&edad=18&enviar=Enviar>

© Antonio Santos Ramos 2018

29

HTML

Audio y Video



■ Elementos multimedia

- Incluye etiquetas para trabajo multimedia (y así no emplear Flash)
- Atributos (video): autoplay, autobuffer, controls, height, width, loop
- Atributos (audio): autoplay, autobuffer, controls, loop

```
<video src="mivideo.mp4"></video>
<audio src="audio.mp3"></audio>
```

■ Elementos para dibujar

- Incluyen la etiqueta **<canvas>**

<http://www.chromeexperiments.com>
<http://www.storiesinflight.com/html5/>

© Antonio Santos Ramos 2018

30

Curso de Programación para Internet con Java EE

(Parte IV - Aplicaciones Web)

T02 – Composición de páginas

© Antonio Santos Ramos 2018

31

Composición de páginas Capas

- Características de las capas (DIV)
 - La posiciones pueden ser absolutas, relativa o flotantes.
 - Las capas pueden ocultarse, ser transparentes, anidarse, etc.
 - Los contenidos pueden solaparse (gracias a **z-index**).
 - Las capas con mayor Z aparecen delante.
- Se emplea CSS para trabajar con capas y...
 - Definir el aspecto del contenido
 - Definir posición
 - Definir tamaño.
 - Definir la colocación (flotante, absoluta, etc.)

http://www.w3schools.com/css/css_positioning.asp

Composición de páginas con HTML 5 (I/II)



- **section**: sección “general” dentro de un documento.
- **article**: un contenido independiente en un documento.
- **aside**: un contenido que está muy poco relacionado con el resto de la página
- **header**: cabecera de una sección
- **footer**: pie de una sección
- **nav**: sección dedicada a la navegación
- **main**: contenido principal. Es único
- **figure/figcaption**: elementos media
- **details/summary**: elementos desplegables

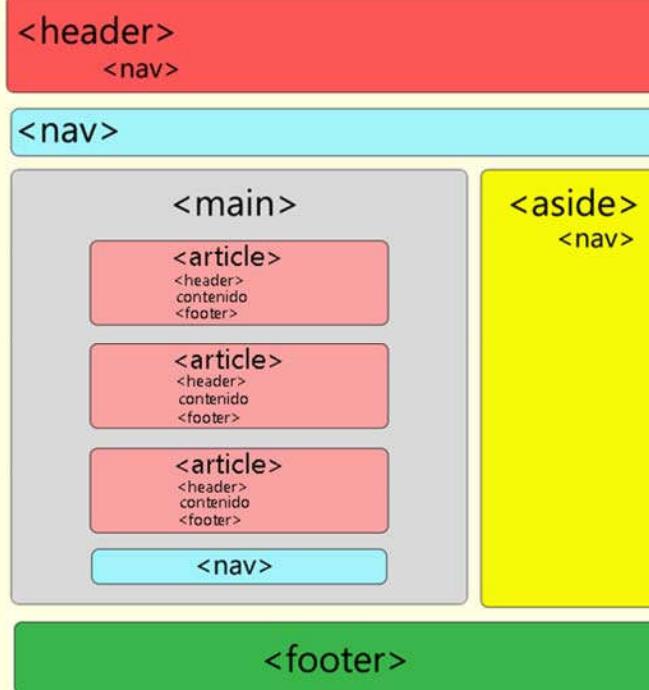


© Antonio Santos Ramos 2018

33

<http://html5doctor.com/avoiding-common-html5-mistakes/>

Composición de páginas con HTML 5 (II/II)



© Antonio Santos Ramos 2018

34

Diseño adaptado (I/II)



Ethan Marcotte

- **Responsive Design (RWD):** Técnica que emplea CSS para ajustar el diseño de una página al tamaño de pantalla.
- Es la suma de 3 ideas
 - Diseño fluido: float + %
 - Imágenes flexibles: imágenes escalables y porcentuales


```
img {max-width: 100%;}
```
 - **CSS media queries:** método para ejecutar distintos códigos dependiendo del tipo de dispositivo

```
@media screen and (min-width:480px) and (orientation:landscape) {
    //los nuevos estilos para ese media query
}
```

<http://quirktools.com/screenfly/>

http://www.page2images.com/mobile_phone_emulator

<http://designshack.net/articles/css/which-is-right-for-me-22-responsive-css-frameworks-and-boilerplates-explained/>

<http://www.flashuser.net/responsive-web-design-showcase-of-30-sites>

© Antonio Santos Ramos 2018

35

Diseño adaptado (II/II)



```
@media screen and (max-width:520px) {
    //los nuevos estilos para Iphone
}

@media screen and (max-width:768px) {
    //los nuevos estilos para Tablet
}
```

© Antonio Santos Ramos 2018

Cambios

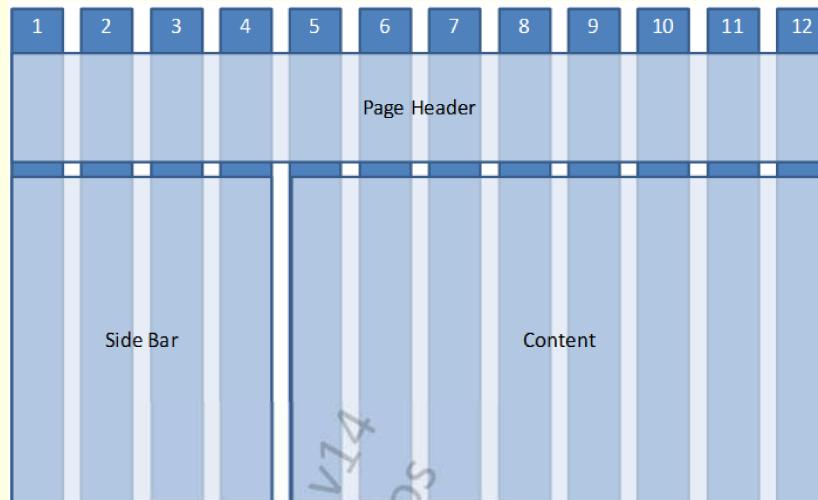
- colocación (float),
- fuentes (↑ al ser más pequeño)
- padding, margin
- width

36

<http://www.ifuckinglovecoding.com/es/fragmentosdecodigo/media-queries-cheat-sheet>

Grid Layout

- Plantilla que define una serie de estilos que ayudan a organizar el contenido de una página web.
 - Similar a una hoja cuadriculada.



© Antonio Santos Ramos 2018

37

Herramientas Front End Twitter Bootstrap



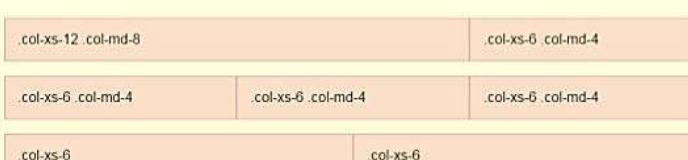
■ Twitter Bootstrap

- Es un framework con elementos CSS/JS de Twitter para el desarrollo de interfaces de usuario.
- Incluye numerosos elementos gráficos
- Dispone de grid adaptable a distintas resoluciones (RWD)

- xs: 0 – 767px
- sm: 768 – 991px
- md: 992 – 1199px
- lg: 1200+ pixels

Bootstrap 3

<http://getbootstrap.com/>
<https://hackerthemes.com/bootstrap-cheatsheet/>



Device	Ems	Pixels	Class
Extra Small	Less than 34em	Less than 544px	col-xs
Small	34em and up	544px and up	col-sm
Medium	48em and up	768px and up	col-md
Large	62em and up	992px and up	col-lg
Extra-Large	75em and up	1200px and up	col-xl

Bootstrap 4

© Antonio Santos Ramos 2018
<http://tutsme-webdesign.info/best-bootstrap-visual-editor/>
<https://www.agriya.com/blog/2016/07/28/10-most-popular-bootstrap-editors-tools/>

38

Curso de Programación para Internet con Java EE

(Parte IV - Aplicaciones Web)

T03 – CSS. Hojas de estilo

© Antonio Santos Ramos 2018

39

CSS. Características (I/III)

- CSS: Hojas de estilo en cascada (**Cascading Style Sheets**)
 - Define la presentación de un documento escrito en HTML o XHTML
 - Separa la estructura de un documento de su presentación
- Versiones:
 - CSS Nivel 1: recom. diciembre 1996 (rev. Abril 2008)
 - CSS Nivel 2: recom. mayo 1998. Actualmente v2.1 (rev. Junio 2011)
 - CSS Nivel 3: desde 2005
- Cada navegador emplea un motor de diseño

Firefox	Gecko
Chrome (antes WebKit) Opera (antes WebKit y Presto)	Blink
Explorer y Windows Phone (Edge tiene un motor mejorado a partir de Trident)	Trident
Safari, DW, Android, Symbian, AIR, Maxthon, iOS	WebKit

[http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(CSS\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(CSS))

http://en.wikipedia.org/wiki/List_of_web_browsers

<http://www.masadelante.com/faqs/listado-buscadores>

CSS. Características (II/III)

```
selector {  
    propiedad_1:valor_1;  
    propiedad_2:valor_2;  
    ...  
}
```

- **Selector** → a que elemento le afecta el estilo
- **Propiedad** → atributo que quieras cambiar
- **Valor** → “naturaleza” que acepta esa propiedad
- Ej:

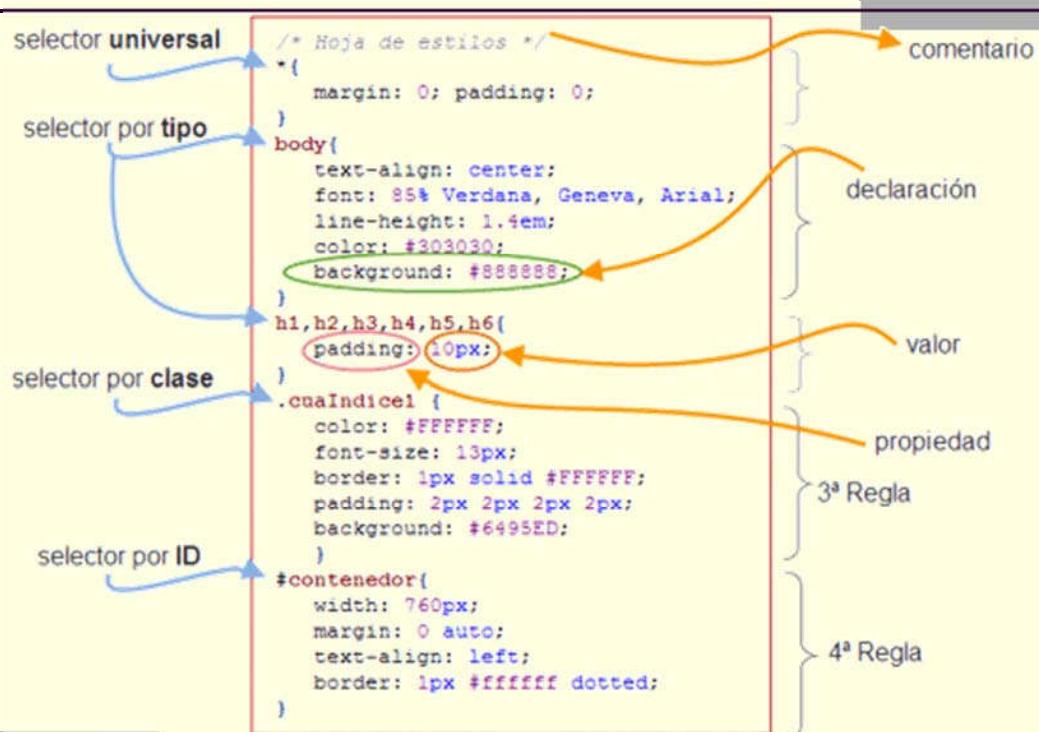
```
p {  
    text-align: center;  
    color: #000000;  
    font-family: arial;  
}
```

<http://es.html.net/tutorials/css/>

© Antonio Santos Ramos 2018

41

CSS. Características (II/III)



© Antonio Santos Ramos 2018

http://www.w3schools.com/cssref/css_selectors.asp

42

<http://code.tutsplus.com/es/tutorials/the-30-css-selectors-you-must-memorize--net-16048>

CSS. Características (III/III)

- CSS integrado en elemento html (estilo en línea). NO se usa.

```
<p style="color:#ff6699;">Hola mundo</p>
```

- CSS integrado en documento html (interno o embebido)

```
<head>
  <style type="text/css">
    h1{color:#ff6699;}
  </style>
</head>
```



- CSS independiente del documento html

```
<head>
  <link rel="stylesheet" type="text/css"
        href="hoja.css">
</head>
```

Documentos HTML que enlazan con la misma hoja de estilo

(externo)

CSS. Sintaxis (I/II)

- Puedo declarar más de un selector a la vez. Van separados por comas.
- Comentarios: /*un comentario */
- No distingue mayúsculas o minúsculas, pero se utilizan minúsculas (xhtml)
- Formatos para representar colores
 - #ffcc00
 - #fc0
 - rgb(255,204,0)
 - hsl(48,100%,50%)
 - rgb(100%,80%,0%)
 - Nombres en inglés

→ rgba(255,204,0, 1) → hsla(48,100%,50%,1)

CSS. Sintaxis (II/II)

■ Unidades de medida

UNIDAD	DESCRIPCIÓN	VALOR	EJEMPLO
in	Pulgada que equivale a 2,54 cm.	real	0.1in
cm	Centímetro	entero	6cm
mm	Milímetro	entero	60mm
pt	Punto equivale a 1/72 pulgada	entero	48pt
pc	Pica equivale a 12 puntos (1/6 pulgada)	real	4.5pc
px	Píxel	entero	800px
%	Porcentaje	entero	30%
em	Tamaño de fuente asignada	ex em Tamaño de la fuente	1.2em
ex	Altura de la “x”		0.8ex

Aprox. $12\text{pt} = 1/6\text{in} = 1\text{pc} = 16\text{px} = 100\% = 1\text{em} = 1.5\text{ ex}$

© Antonio Santos Ramos 2018

45

CSS. Formato Selectores (html)

```
ElementoHtml{
    propiedad1:valor1;
    propiedad2:valor2;
    ...
}
```

HTML

```
<p> Párrafo azul</p>
<h1>Encabezado verde </h1>
```

CSS

```
p {color: blue;}
h1 {color: green;}
```

CSS. Formato Selectores (class)

```
.NombreClase{  
    propiedad1:valor1;  
    propiedad2:valor2;  
    ...  
}
```

- Se asigna con el atributo “class”.
- “class” NUNCA puede ir suelto. Es un atributo, no una etiqueta

HTML	CSS
<p class="noticia"> Párrafo azul</p> <p class="titular">Título verde </p>	.noticia {color: blue;} .titular {color: green;}

- Puede llevar más de una clase <etq class="c1 c2 c3">...</etq>

CSS. Formato Selectores (id)

```
#NombreIdentificador{  
    propiedad1:valor1;  
    propiedad2:valor2;  
    ...  
}
```

- Se asigna con el atributo “id”.
- “id” NUNCA puede ir suelto. Es un atributo, no una etiqueta

HTML	CSS
<p id="noticia"> Párrafo azul</p> <h1 id="titular">Título verde </h1>	#noticia {color: blue;} #titular {color: green;}

CSS. Selectores (elementos que cumplen a la vez dos o más requisitos)

- Formato del selector → AB (rara vez es ABC)
 - Cualquiera de los elementos es una de las piezas comentadas anteriormente ('A' suele ser una etiqueta html)
- El estilo debe indicarse en CSS escribiendo los elementos seguidos, **sin espacios**.
- En HTML deben emplearse las etiquetas HTML y los atributos "id" y/o "class" para que tenga sentido.

HTML

```
<p class="texto">Párrafo azul</p>
<h2 id="titulo">Párrafo verde </p>
```

CSS

```
p.texto {color: blue;}
h2#titulo {color: green;}
```

CSS. Selectores (Elementos contenidos unos dentro de otros)

- Formato del selector → A B C D
 - Cualquiera de los elementos es una de las piezas comentadas anteriormente (o una combinación AB)
- El estilo debe indicarse en CSS escribiendo los elementos seguidos, **con espacios** entre elementos
- En HTML deben emplearse las etiquetas HTML y los atributos "id" y "class" para que tenga sentido

HTML

```
<h1>Un título con <a href="#">enlace</a></h1>
<div><p class="texto"> Párrafo azul</p></div>
```

CSS

```
h1 a {color: green;}
div p.texto {color: blue;}
```

CSS. Propiedades Fuentes y párrafos

■ Propiedades para párrafos

- line-height
- text-decoration
- text-align
- text-indent
- text-transform
- word-spacing
- letter-spacing
- vertical-align

■ Propiedades para fuentes

- color
- font-size
- font-family
- font-weight
- font-style
- font-variant
- @font-face

<https://escss.blogspot.com/2012/05/font-face-y-sus-problemas-guia-de-uso-y.html>

<https://escss.blogspot.com/2016/03/modern-system-fonts.html>

CSS. Propiedades Otros elementos de texto

■ Formatos para fuentes

- eot, woff, ttf, otf, svg

<http://www.w3.org/TR/css3-fonts/>

■ Propiedades CSS3



- Sombras en textos: text-shadow

■ Columnas

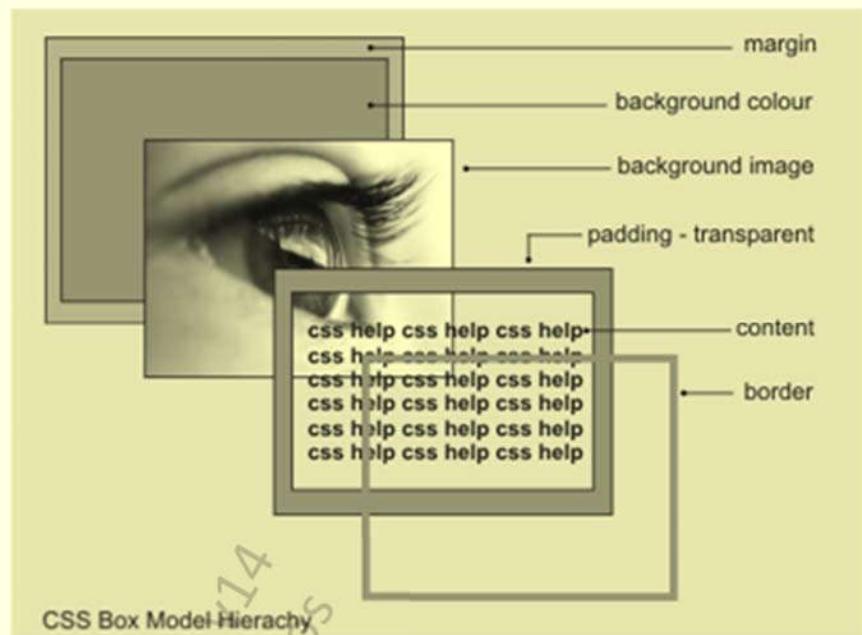
- column-count: nº columnas
- column-width
- column-gap: espacio entre columnas
- column-rule

- word-wrap:break-word → para romper palabras muy largas

CSS. Propiedades Box model (I/II)

■ Formatos para elementos de bloque

- width
- height
- margin
- padding
- border-color
- border-style
- border-width
- border
- min-width
- max-width



© Antonio Santos Ramos 2018

53

CSS. Propiedades Box model (II/II)

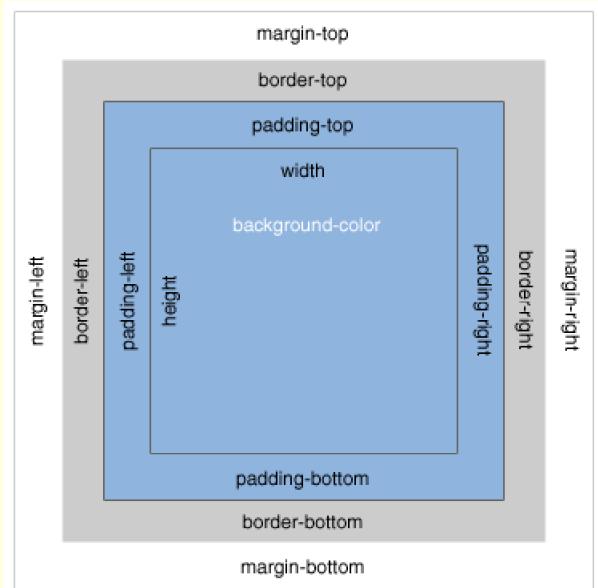
■ ¿Cómo modificar la presentación?

- overflow
- display
- visibility

■ Novedades CSS3



- display:flex
- border-radius
- box-shadow
- border-image
- resize
- outline
- box-sizing: border-box|content-box



© Antonio Santos Ramos 2018

54

CSS. Propiedades Listas y tablas

■ Fondos

- background-color
- background-image
- background-position
- background-repeat
- background-attachment

■ Fondos

- background-origin
- background-clip
- background-size (contain,cover)
- Uso de
 - Gradienes: linear-gradient
 - Varias imágenes de fondo



■ Listas

- list-style-type
- list-style-position
- list-style-image

■ Tablas

- empty-cells
- border-spacing
- border-collapse

CSS. Propiedades Transformaciones



■ Transformaciones 2D: translate(), rotate(), scale(), skew(), matrix() http://www.w3schools.com/css/css3_2dtransforms.asp

■ Transformaciones 3D: rotateX(), rotateY() http://www.w3schools.com/css/css3_3dtransforms.asp

■ Transiciones: transition http://www.w3schools.com/css/css3_transitions.asp

■ Animaciones: @keyframes, animation http://www.w3schools.com/css/css3_animations.asp <http://www.creativebloq.com/css3/animation-with-css3-712437> <http://css3 bradshawenterprises.com/transitions/> <http://learn.shayhowe.com/advanced-html-css/transitions-animations> <http://www.1stwebdesigner.com/css/50-awesome-css3-animations/>

CSS. Formato Pseudoclases

```
selector:pseudoclas {
    propiedad: valor
}
```

- Clasifican elementos según el estado del mismo
- Ejemplos:

■ Enlace no visitado:	a:link {propiedad: valor}
■ Enlace visitado:	a:visited {propiedad: valor}
■ Foco sobre enlace:	a:hover {propiedad: valor}
■ Enlace activado:	a:active {propiedad: valor}
■ Enlace seleccionado:	a:focus {propiedad: valor}

CSS. Formato Pseudo-elementos

```
selector:pseudoelemento {
    propiedad: valor
}
```



En CSS3
se emplean ::

- Seleccionan parte de un elemento
 - Ej.- p:first-letter {color:#ff0000; }
- Ejemplos:
 - Primer carácter: :first-letter {propiedad: valor}
 - Primera línea: :first-line {propiedad: valor}
 - El elemento anterior: :before {propiedad: valor}
 - El elemento posterior: :after {propiedad: valor}

CSS para imprimir

- Se puede especificar una hoja de estilo diferente para la impresión.
- Se emplea (al enlazar la hoja css) el atributo media para el diseño en pantalla (`media="screen"`) , diseño para móviles (`media="handheld"`), imprimir (`media="print"`), etc.
 - `<link rel="stylesheet" href="style.css" type="text/css" media="screen" />`
 - `<link rel="stylesheet" href="print.css" type="text/css" media="print" />`
 - `<link rel="stylesheet" href="movil.css" type="text/css" media="handheld" />`

© Antonio Santos Ramos 2018

59

CSS3 (I/II)

- Nuevos selectores
 - <http://bit.ly/cKO24D>
- Pseudo-elementos
 - Cambia sintaxis a ::
 - Ej.- `p::first-letter`
 - `p::selection`
- Pseudo-clases
 - Varias nuevas
- Textos y fuentes
 - `@font-face`
 - Estructuras con columnas
 - `Word-wrap`
- Color
 - Uso de transparencias RGBA
 - Soporte HSL
- Bordes
 - `border-radius`
 - `border-image`
- Fondos
 - `background-origin`
 - `background-clip`
 - Se pueden incluir gradientes con `background-image`
- Sombras y efectos
 - `box-shadow`
 - `text-shadow`
- Transformaciones (rotaciones, inclinaciones, escalas)
 - `transform`
- Animaciones
 - `@keyframes`

© Antonio Santos Ramos 2018

60

CSS3 (II/II)



■ Reglas propietarias (vendor prefixes)

```
.box_gradient {
    background-color: #444444;
    background-image: -webkit-linear-gradient(top, #444444, #999999); /* Chrome 10+, Saf5.1+, iOS 5+ */
    background-image: -moz-linear-gradient(top, #444444, #999999); /* FF3.6+ */
    background-image: -ms-linear-gradient(top, #444444, #999999); /* IE10 */
    background-image: -o-linear-gradient(top, #444444, #999999); /* Opera 11.10+ */
    background-image: linear-gradient(to bottom, #444444, #999999);
}
```

<http://leaverou.github.com/prefixfree/>

- <http://www.cristalab.com/blog/la-nueva-guerra-de-navegadores-del-2012-y-los-prefijos-de-css3-c1053591/>
- <http://coding.smashingmagazine.com/2010/06/17/start-using-css3-today-techniques-and-tutorials/>
- <http://www.w3schools.com/css3/default.asp>
- <http://www.css3.com/>
- <http://www.css3.me/>
- <http://css3generator.com/>
- <http://css3clickchart.com>
- <http://www.findmebyip.com/litmus/>
- <http://caniuse.com/>

© Antonio Santos Ramos 2018

CSS3 Properties Support by Browsers					
CSS Animations	✓	✗	✗	✗	✓
CSS Columns	✓	✓	✓	✗	✓
CSS Gradients	✓	✓	✓	✗	✓
CSS Reflections	✓	✗	✗	✗	✗
CSS Transforms	✓	✓	✓	✓	✓
CSS Transforms 3D	✗	✗	✗	✗	✗
CSS Transitions	✓	✗	✓	✓	✓
CSS FontFace	✓	✓	✓	✓	✓

Curso de Programación para Internet con Java EE (Parte IV - Aplicaciones Web)

T04 – JavaScript

© Antonio Santos Ramos 2018

Versiones de JS

ES5 (ECMAScript 5)

ES6 (2015)

ES2016

ES2017

JavaScript

■ JavaScript (1996)

- Lenguaje de tipo script
 - Se interpreta en cliente
- ## ■ Funcionamiento
- El código JavaScript se integra en HTML o se anexa en un fichero Js
 - Trabaja a través de funciones y eventos que desencadenan esas funciones
 - Puede trabajar con los elementos del HTML por medio del DOM

■ Características

- Se valida al ejecutar
- Permite POO
- No se declaran los tipos

■ Comandos de JavaScript:

- Variables
- Expresiones
- Estructuras de control
- Funciones
- Clases, objetos y arrays (agrupaciones de datos)

© Antonio Santos Ramos 2018

63

La TAG <script>... </script>

■ Sintaxis archivo externo

```
<script type="text/javascript" src="fuente.js"></script>
```

■ Sintaxis archivo interno

```
<head>
...
<script type="text/javascript">
    function valor_abs(form) {
        var num
        num = eval(form.expr.value)
        if (num >= 0){ form.result.value = num}
        else {form.result.value = num }
    }
</script>
...
</head>
```

■ Puede declararse en otras partes del HTML

© Antonio Santos Ramos 2018

64

<https://rolandocaldas.com/html5/como-incluir-javascript-en-html5>

Modelo de Eventos de JavaScript

- Los eventos son capturados por los gestores de eventos (tienen un **on** delante)
- Pueden existir tres tipos de eventos:
 - A nivel del documento HTML
 - A nivel de un formulario individual
 - A nivel de un elemento de un formulario

```
<body onLoad="cargarfuncion()" onUnload="descargarfuncion()>

<input type="button" name= "test" value= "Pasa de mi"
       onClick= "alert('Que no me aprietas, melon')">
```

Gestores de Eventos (Event Handlers)

Evento	Ocurre cuando	Gestor
blur	El usuario quita el cursor de un elemento de formulario	onBlur
click	El usuario hace clic en 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

Ejemplo

Mensaje emergente (I)

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hola Mundo v1</title>
</head>
<body onLoad="javascript:alert('Hola Mundo v1');">
</body>
</html>
```

Ejemplo

Mensaje emergente (II)

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hola Mundo v2</title>
</head>
<body>
    <script type="text/javascript">
        document.write("<h1>Bienvenido a mi página v2</h1>");
        alert("Hola mundo v2")
    </script>
</body>
</html>
```

Ejemplo

Mensaje emergente (III)

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hola Mundo v3</title>
    <script type="text/javascript">
        document.write("<h1>Bienvenido a mi página v3</h1>");
        function msgHolaMundo() {
            alert("Hola mundo v3");
        }
    </script>
</head>
<body onLoad="msgHolaMundo()">
</body>
</html>
```

© Antonio Santos Ramos 2018

69

Ejemplo

Mensaje emergente (IV)

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hola Mundo v4</title>
    <script type="text/javascript" src="mensajes04.js">
</head>
<body onLoad="msgHolaMundov4()">
</body>
</html>
```

```
function msgHolaMundov4() {
    document.write("<h1>Bienvenido a mi página v4</h1>");
    alert("Hola mundo v4.");
}
```

mensajes04.js

© Antonio Santos Ramos 2018

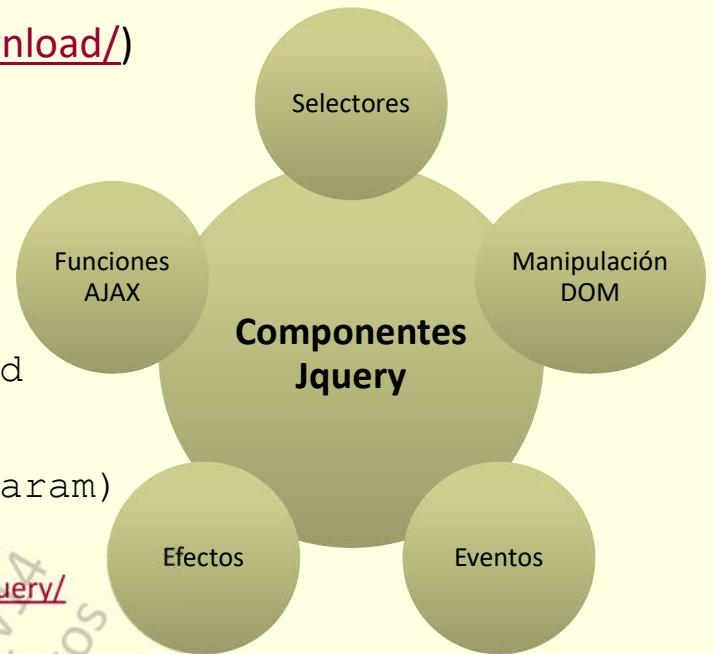
70

Componentes Jquery



- ¿Necesito librerías externas?
 - Sí (<http://jquery.com/download/>)
- ¿Dónde se incluye?
 - En el head
 - En el body
- Sintaxis básica
 - \$(selector).propiedad
 - \$(selector).accion()
 - \$(selector).accion(param)

<http://oscarotero.com/jquery/>
http://librosweb.es/libro/fundamentos_jquery/
<http://librojquery.com/>



© Antonio Santos Ramos 2018

71

Componentes Jquery Selectores

- \$("*")
- \$("<etiqueta_html>")
- \$("#id_elemento")
- \$(".clase")
- \$(selector1,selector2,...)
- \$(this)

```

alert($(".p").length);
$('.bloque1').addClass('fondo');
$("#datos").html("<strong>nuevo</strong>");
$("h1", #contenido)    <- Necesita algo detrás
  
```

© Antonio Santos Ramos 2018

72

Componentes Jquery

Manipulación DOM

■ .css("propiedad") .css("propiedad",valor)

```
$(".texto").css("background-color");
$(".texto").css("background-color","red");
```

■ .text() .html()

```
alert($("#capa").text());
```

■ .append(contenido)

```
$(".noticia").append("<b>Autor:antonio</b>");
```

■ .remove()

```
$("#lateral").remove();
```

■ .val() .val(valor)

```
$("#apellidos").val("Indica apellidos");
```

© Antonio Santos Ramos 2018

73

Componentes Jquery

Eventos

■ .on

```
$("p").click(function() {
    // action goes here!!
});
```

```
function f1(){ ... }
```

■ .hover(f1,f2)

```
$(.tareas 'li').click(f1);
```

■ .click()

```
$('.tareas li').on('click', f1);
```

■ .change()

```
$(document).ready(function(){
    // jQuery methods go here...
});
```

■ .bind()

```
$(function() { ... });
```

```
$("#menu").hover(
    function() {
        $("#submenu").show();
    },
    function() {
        $("#submenu").hide();
    }
);
```

■ .unbind()

■ ...

■ .each(f_anonima)

© Antonio Santos Ramos 2018

74

Componentes Jquery

Efectos

- hide()
 - show()
 - show(duracion)
 - toggle()
 - fadeOut()
 - slide()
 - animate()
- ```
$("p").toggle();
$("p").hide(1000);
$("#c1").slideDown('slow');
$("#c1").fadeIn('slow');
$("#c1").fadeTo('slow', 0.25);
$('#picTwo').delay(5000).fadeIn(1500);
$(".modal").delay(100).fadeTo(200, 1);
$("div").animate({left:'250px'});
$('div').animate({fontSize: 11, opacity: 0});
$("#info").animate({marginTop:"-120px"}, 400);
```

# Componentes Jquery

## Funciones Ajax

- Permite funciones ajax trabajando con \$.ajax, \$.get, \$.post, \$.getJSON, ...

<http://api.jquery.com/jQuery.ajax/>  
<http://api.jquery.com/jQuery.get/>  
<http://api.jquery.com/jQuery.post/>

```
$(function() {
 $("#button").click(function() {
 $.ajax({
 type: "POST",
 url: "InsertMessage",
 data: dataString,
 success: function(data) {
 $("ol#update").prepend(data);
 $("ol#update li:first").slideDown("slow");
 document.getElementById('cont').value = '';
 $("#content").focus();
 $("#flash").hide();
 }
 })
 })
})
```

# Curso de Programación para Internet con Java EE

## (Parte IV - Aplicaciones Web)



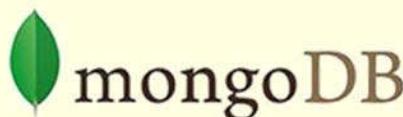
T05 – Introducción a Angular y AngularJS

© Antonio Santos Ramos 2018

77

## Aplicaciones MEAN (Foto Básica)

BASE DE DATOS



SERVIDOR con  
una API REST

3. Respuesta de la  
Base de Datos



FRONTEND

4. Devuelve un objeto  
en JSON

1. Llamadas AJAX al  
API REST

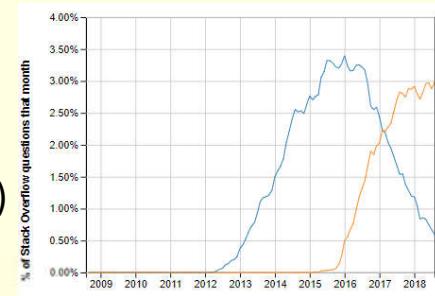


ANGULARJS

# ¿Qué es AngularJS?



- Framework JavaScript open-source
  - Creado por Google (2009) por Misko Hevery para desarrollar Aplicaciones Web para entornos móviles y de escritorio.
  - Solo se requieren HTML, CSS y JavaScript en el cliente.
  - Todo lo ejecuta del lado del browser
  - Intenta definir el comportamiento de la aplicación mediante nuevas “construcciones” dentro del HTML
- Nomenclatura
  - AngularJS (1.x): Octubre 10
  - Angular 2: Septiembre 16
  - Angular 4: Marzo 17 (+ Sep 18)
  - Angular 5: Noviembre 17 (+ May 19)
  - Angular 6: Mayo 18
  - Angular 7: Octubre 18
  - Angular 8: Marzo 19



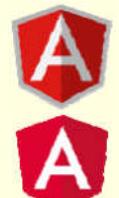
<https://angular.io/guide/styleguide>

© Antonio Santos Ramos 2018

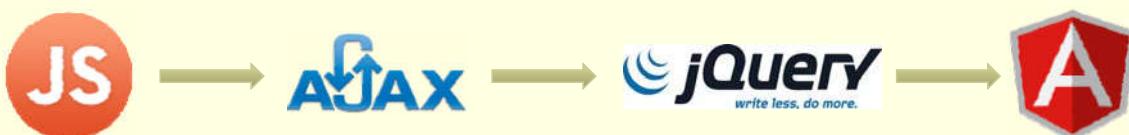
79

<https://auth0.com/blog/migrating-an-angular-1-app-to-angular-2-part-1/>

## AngularJS Single-page Apps (SPAs)

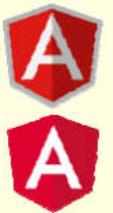


- Una aplicación web que está conformada por una sola página HTML
  - La aplicación se carga en una sola llamada al server
  - No es necesario refrescar la página para ir utilizando la aplicación
  - No se transfiere el control a ninguna otra página
- Provee una experiencia similar a las aplicaciones desktop-only



© Antonio Santos Ramos 2018

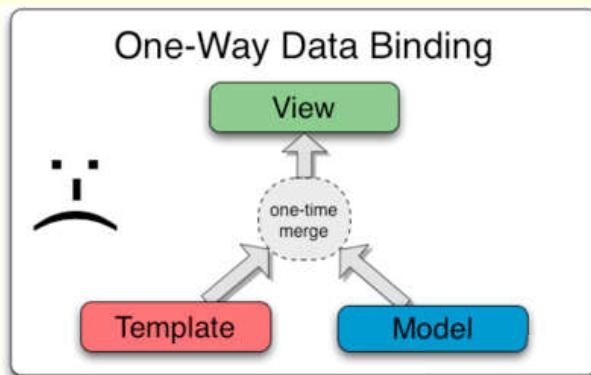
80



# Data Bindings

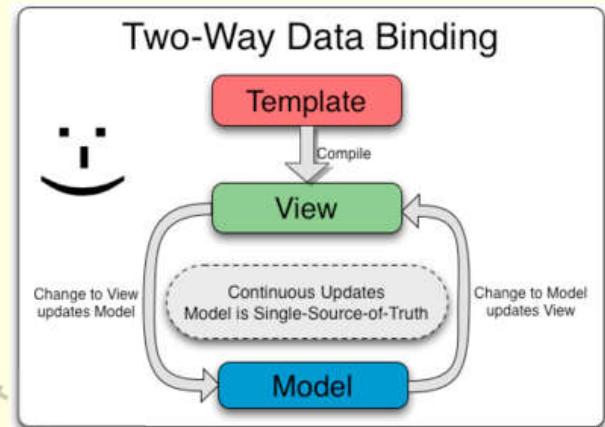
## ■ ANTES

- Líneas de código extras para reflejar los cambios en un lado u otro



## ■ AHORA

- Un cambio en el modelo se refleja en la vista y viceversa mediante {{ }}



<http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-ii-binding-directivas.html#data-binding>

© Antonio Santos Ramos 2018

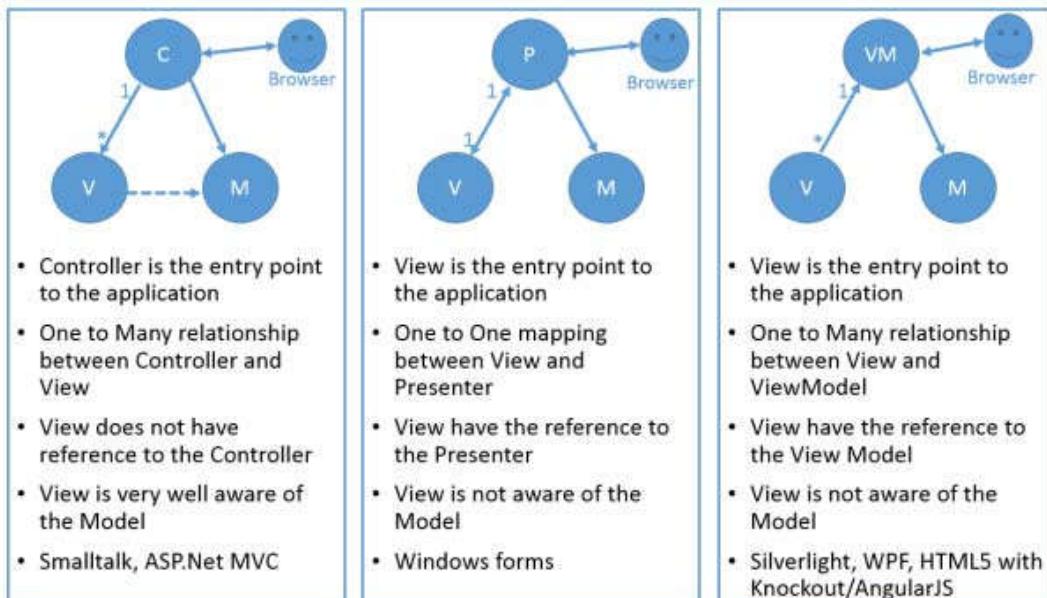
81

<https://dev.to/odigietony/the-very-simplified-concept-of-binding-in-programming-languages-op>



# Patrones MV\*

## MVC – MVP – MVVM



© Antonio Santos Ramos 2018

82

<https://stackoverflow.com/questions/36950582/angular2-mvc-mvvm-or-mv>



# Elementos y conceptos principales en AngularJS

|              |                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Data-Binding | • Sincronización automática de datos entre los componentes de la vista y el modelo.                                                  |
| Scope        | • Los objetos que se refieren al modelo. Actúan como enlace entre el controlador y la vista.                                         |
| Controller   | • Son funciones de Javascript que están atadas a un scope en particular.                                                             |
| Services     | • AngularJS tiene por defecto varios servicios ya construidos. Son objetos que permiten obtener información. Siempre son Singleton.  |
| Filters      | • Los filtros se agregan a las expresiones y a las directivas para transformar datos.                                                |
| Directives   | • Son marcas sobre los elementos del DOM, que pueden ser utilizadas para crear etiquetas HTML y/o widgets personalizados.            |
| Templates    | • Se utilizan para la generación de vistas dinámicas a partir de información manipulada por el controlador y el modelo.              |
| Routing      | • Es la forma de cambiar de vistas.                                                                                                  |
| MVV          | • AngularJS no implementa el MVC tradicional y se acerca más la MVVM (Model-View-ViewModel).                                         |
| Deep Linking | • Esta propiedad permite codificar el estado de una aplicación en una URL y puede ser restaurada desde la URL y con el mismo estado. |

© Antonio Santos Ramos 2018

83

<https://docs.angularjs.org/guide/concepts>



## AngularJS Directivas (I/II)

```
<!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>
```

Directive

Directive

Data Binding  
Expression

En Angular hay directivas que ya no son necesarias

© Antonio Santos Ramos 2018

84



# AngularJS

## Directivas (II/II)

ng-app	Autoarranca la aplicación Angular. Suele colocarse en <html>
ng-controller	Indica a la vista dónde trabaja el controller
ng-model	Define las variables del modelo a usar
ng-click	Asociada al evento clic (sobre un elemento DOM)
ng-init	Define los datos iniciales de la aplicación
ng-repeat	Permite iterar sobre una colección de datos
ng-change	Detecta cambios en la entrada
ng-show	Muestra una parte de la vista (un elemento DOM)
ng-hide	Oculta una parte de la vista (un elemento DOM)
ng-bind	La misma funcionalidad que {{ }}
ng-disabled	Deshabilita un elemento DOM



# AngularJS

## Implantación de MV\*

```
<html lang="en" data-ng-app="myApp">
<div class="container" data-ng-controller="SimpleController">
 var myApp = angular.module('myApp', []);
 function SimpleController($scope) {
 $scope.personas = [
 {name:'Antonio',city:'Madrid'},
 {name:'Luis',city:'Barcelona'},
 {name:'Virginia',city:'Valencia'},
 {name:'Maria',city:'Murcia'}
];
 }
 myApp.controller('SimpleController', SimpleController);

```

Nombramos la App en la Vista

Se declara el controlador

Se define el controlador

Se embebe el modelo

Se crea la App y se asocia el controlador



# AngularJS

## Módulos

- Un módulo agrupa diferentes partes de la aplicación: controladores, servicios, directivas, filtros, etc.
  - El código se puede empaquetar como código reusable.
  - Se cargan en cualquier orden.
  - Las pruebas de unidad solo cargan los módulos requeridos.
- Permiten desacoplar el código
- Se define de esta forma

```
angular.module('Nombre_del_modulo', []);
```
- Las dependencias que necesite se inyectan entre los [ ]

```
angular.module('Nombre_del_modulo', ['dep1','dep2']);
```

  - Se inyectan otros módulos de los que dependa el nuestro



# AngularJS

## \$scope

- Definido literalmente indica ámbito o alcance
  - Dentro de este alcance se tiene acceso a las variables y los métodos que enlazan el controlador con la vista.
  - Por lo tanto, todo controlador tiene asociado un \$scope que le permite interactuar con la vista
- Para monitorizarla se emplea \$watch
- La tendencia actual es a la minimización de su uso
  - Angular 2 prescinde del \$scope
  - La idea es emplear la técnica del controller as combinado con this

<https://toddmotto.com/digging-into-angulars-controller-as-syntax/>



# AngularJS Controladores

- Son objetos que permiten desarrollar la lógica de la aplicación, enlaza el ámbito, \$scope, con la vista y permite tener un control total de los datos.

- En pocas palabras: encargado de gestionar los eventos.

- Se enlazan con la vista mediante la directiva ng-controller

```
var app = angular.module('MyApp', []);
app.controller('mainController', function($scope) {
 //contenido
});
```

- Colocaremos los Controllers dentro de MODULE

- En Angular un componente sería algo parecido a un controlador AngularJS ligado a una vista



# AngularJS Servicios

- Los servicios son objetos singleton, injectables por Dependency Injection, donde definimos la lógica de negocio de la aplicación, con el objetivo de que sea reutilizable e independiente de las vistas.
- Los servicios incluyen métodos para mantener los datos en el ciclo de vida y se comunican a través de los controladores.
- Hay 5 formas diferentes de crear servicios en AngularJS
  - constant, provider, service, value, factory
  - Su uso depende del momento del ciclo de vida y de lo que se quiera realizar

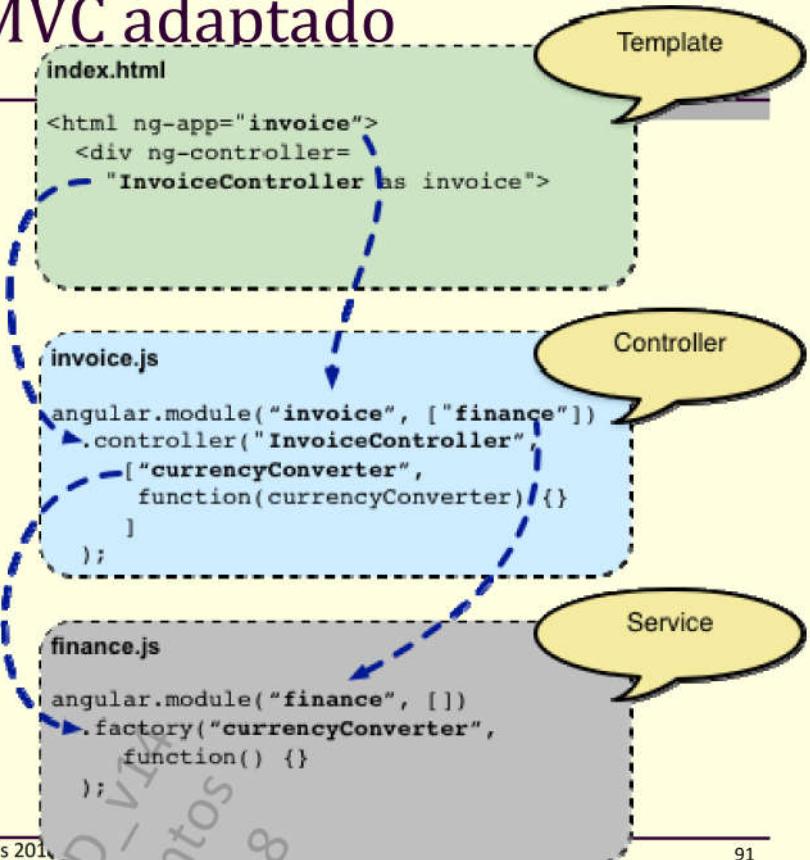


# AngularJS

## Mi MVC adaptado

- En Angular se emplea un modelo más modular con respecto a AngularJS.
- En Angular los elementos principales serán los componentes y los servicios asociados

© Antonio Santos Ramos 2018



91

## ¿Qué es Angular?



- Framework JavaScript open-source
  - Liberado a partir de Septiembre 2016
  - Está construido desde cero. Es más rápido, tiene soporte de estándares, soporte de módulos ES6 (con webpack y SystemJS), componentes como pieza principal, etc.
- Nomenclatura
  - AngularJS (1.x): Octubre 10
  - Angular 2: Septiembre 16
  - Angular 4: Marzo 17
  - Angular 5: Noviembre 17
  - Angular 6: Mayo 18



<https://angular.io/guide/styleguide>

© Antonio Santos Ramos 2018

92



# Instalación Angular (I/II)

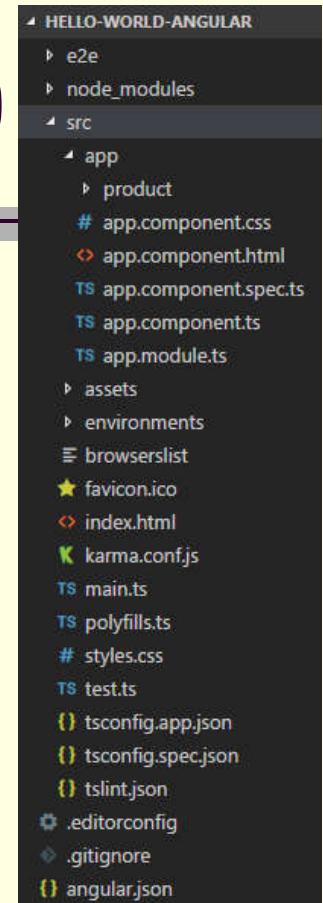
- Node (nodejs.org)    node -v
- Node Package Manager (NPM)
  - Angular CLI (Global)
    - npm install -g @angular/cli
    - Acceso ahora con ng

*AngularJS es “sólo” una librería, con lo que el entorno no es tan predefinido*

## ■ Para instalar Angular

- ng new hola-mundo              Crear proyecto
- cd hola-mundo
- npm install                      instalar dependencias
- ng serve (start)                http://localhost:4200

Para instalar Angular Material    <https://www.toptal.com/angular/angular-5-tutorial>  
 npm install --save @angular/material @angular/cdk @angular/animations



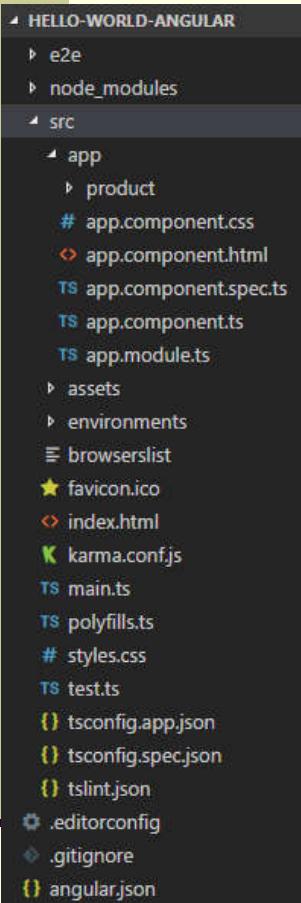
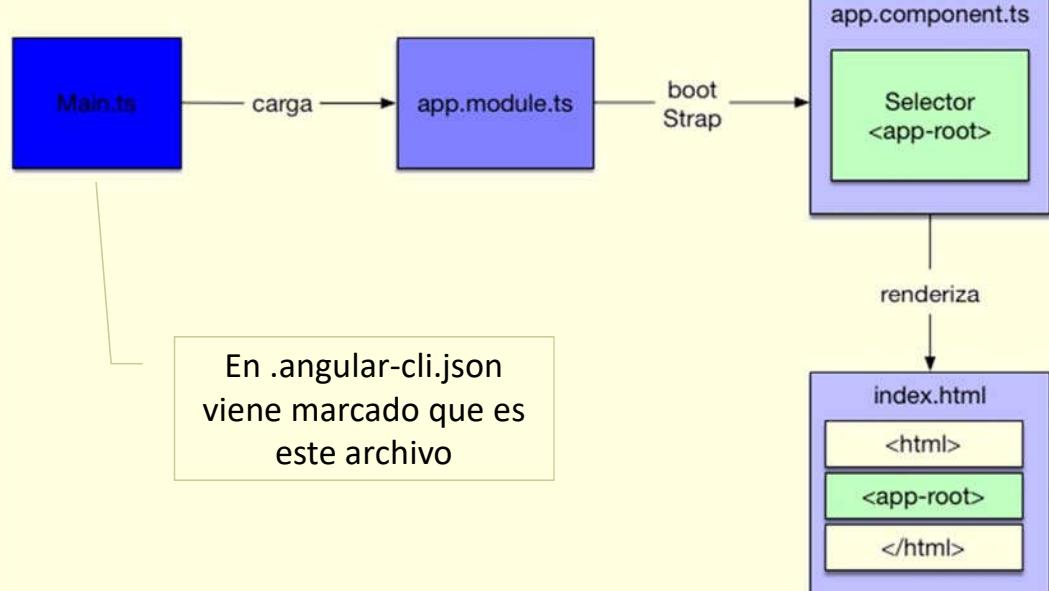
<https://programmingwithmosh.com/angular/angular-4-tutorial/>

© Antonio Santos Ramos 2018

93

<https://coursetro.com/courses/19/Learn-Angular-5-from-Scratch---Angular-5-Tutorial>

# Instalación Angular (II/II)



<https://www.arquitecturajava.com/angular-5-hello-world-y-su-funcionamiento/>

© Antonio Santos Ramos 2018

94

<https://jaxenter.com/angular-2-intellij-netbeans-eclipse-128461.html>

Versiones de JS

ES5 (ECMAScript 5)

ES6 (2015)

ES2016

ES2017

# TypeScript

## (Para Angular)



- Es un superset de JavaScript (<https://www.typescriptlang.org/>)
  - Strong typing, POO, errores compile-time, etc.
  - Incluye constructores, clases, interfaces, enum, módulos, etc.
  - Semejante a C#
- Se compila (transpilador) el TypeScript a JavaScript para que lo entienda el navegador
- Para instalar TypeScript
  - `npm install -g typescript` (para ver la versión `tsc -v`)
- Para trabajar
  - `code hola.ts` crear fichero
  - `tsc hola.ts` transpilar

© Antonio Santos Ramos 2018

95

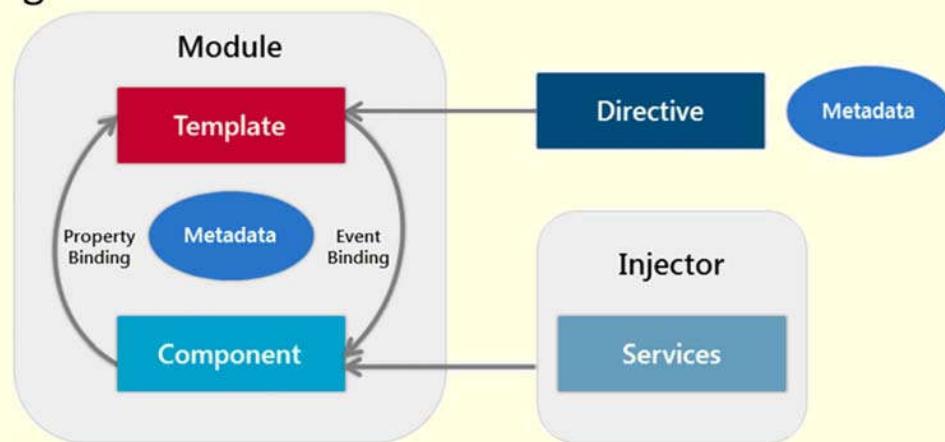
<http://blog.enriqueoriol.com/2016/06/angular2-aprendo-es6-o-typescript.html>

# Angular. Arquitectura



## ■ Elementos de Angular

- Modules
- Components
- Templates
- Metadata
- Data binding
- Directives
- Services
- Dependency injection
- Decorator ≈ Anotaciones Java



<https://angular.io/guide/architecture>

© Antonio Santos Ramos 2018

96

<http://blog.enriqueoriol.com/2017/03/introduccion-angular-modulo-y-componente.html>

<http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-ii-binding-directivas.html>

# Angular.

## Funcionamiento básico

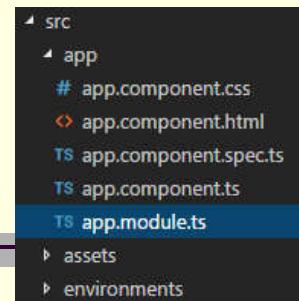


- Para construir App
  - Se crean plantillas HTML (templates) con marcas Angular
  - Se escriben componentes para gestionar esas plantillas y Directivas que afectan al comportamiento de los componentes (programan la lógica).
  - Se encapsula la lógica de la aplicación en Servicios
    - Los servicios se inyectan como dependencias
  - Se define un módulo principal que indica a Angular qué es lo que incluye tu app (otros módulos), y cómo compilarlo y lanzarlo (NgModule).
- Un componente define varias vistas y el servicio Router define la navegación entre ellas

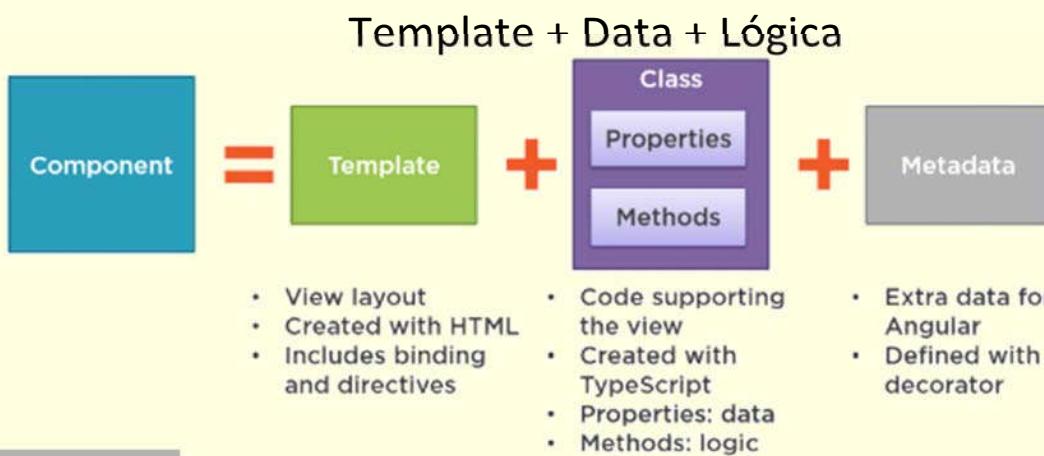
© Antonio Santos Ramos 2018

97

## Componentes en Angular (I/II)



- Son las piezas principales de Angular: muestran datos en pantalla, escuchan iteraciones del usuario y toman acciones basadas en esa entrada.
- Un componente = HTML + Data + Behavior (Method)



© Antonio Santos Ramos 2018

98

# Componentes en Angular (II/II)



### ng g c Nombre-Componente

Crea la carpeta incluida y los 4 ficheros

## Componente principal

## Módulo de la aplic.

```
ts app/components.ts
 1 import { Component } from '@angular/core';
 2
 3 @Component({
 4 selector: 'app-root',
 5 templateUrl: './app.component.html',
 6 styleUrls: ['./app.component.css']
 7 })
 8 export class AppComponent {
 9 title = 'hello-wold-angular';
 10 }
```

Define el estado (datos a mostrar) y el comportamiento (lógica del componente)

```
1 <div style="text-align:center">
2 <h1>
3 Welcome to {{ title }}!
4 </h1>
5
 {{todo.subject}}
</div>
<todo-detail *ngIf="selectedTodo" [todo]="selectedTodo"></todo-detail>
```

# Angular Domain Models



- Debes aislar la estructura a usar del componente
- Pasos para trabajar
  - Exportar en fichero: ng g class Modelo (crea sólo modelo.ts)
  - Importar a componente
  - Inicializarlo en el componente
  - Usarlo desde HTML

```
export class Model {
 param1: string;
}
```

Puede tener constructores

```
Import { Model } from './model';
```

```
export class testWidget {
 public model: Model; Atributo
 constructor(){
 this.model = new Model();
 this.model.param1 = "your value";
 }
}
```

Para el ejemplo

```
@Component({
 selector: "testWidget",
 template: "<div>The {{model.param1}} is my param.</div>"
})
```

# Angular Directives



- Es un concepto de AngularJS pero modificado en Angular
  - Basicamente son clases con el decorador @Directive
  - Realmente un componente es un caso de directiva que va asociado a un template. @Component es un decorador @Directive con características de los templates
- Directivas estructurales:
  - comienzan con \* y alteran el DOM
  - Ej: \*ngIf, \*ngFor (ng-for de AngularJS)
- Directivas atributo
  - Altera la apariencia o comportamiento de un elemento del DOM
  - Ej: ngModel, ngClass, ngStyle...

# Angular Services (I/II)



- Se crean clases servicios para los datos y la lógica no asociada a una vista específica (es como asociar las capas de lógica y datos en un único sitio)
- A diferencia de AngularJS, en Angular los servicios no tienen una sintaxis específica, son simples Clases
- Un servicio viene precedido por @Injectable() si usa otros servicios
  - Se recomienda incluirlo aunque un servicio no esté formado por otros y no haya DI
- Se crean
  - Primero creas la carpeta y dentro de la carpeta escribe
  - ng g s nombre-servicio

```
//app/todos/shared/todo.service.ts
import { Injectable } from '@angular/core';
import { Todo } from './todo.model';
import { Logger } from '../../shared/logger.service';

@Injectable()
export class TodoService{

 todos:Todo[] = [];

 constructor(public logger: Logger){}

 addTodo(todo:Todo){
 this.todos = [...this.todos, todo];
 this.logger.log(this.todos);
 }

 getTodos(){
 this.logger.log(this.todos);
 return this.todos;
 }
}
```

## Angular Services (II/II)

```
//app/shared/logger.service.ts
export class Logger {
 log(msg: any) { console.log(msg); }
 error(msg: any) { console.error(msg); }
 warn(msg: any) { console.warn(msg); }
}
```

- `@Injectable` indica que por DI se le inyecta otro servicio (en este caso es desde el constructor)

## Angular Routing

- El NgModule “Router” proporciona un servicio que refina la ruta de navegación entre los estados de la aplicación y las vistas de la app.
- Angular dispone de un potente sistema de routing para facilitar toda la operativa de las single page applications. Está compuesto por varios actores:
  - El módulo del sistema de rutas: `RouterModule`.
  - Rutas de la aplicación: un array con un listado de rutas que nuestra aplicación soportará.
  - Enlaces de navegación: enlaces HTML con una directiva para indicar que deben funcionar usando el sistema de routing.
  - Contenedor: donde colocar las pantallas de cada ruta.

# Anexo (I/VIII)

## Node.js



- Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.
- Creado en 2009 por Ryan Dahl y podría asemejarse a Java EE

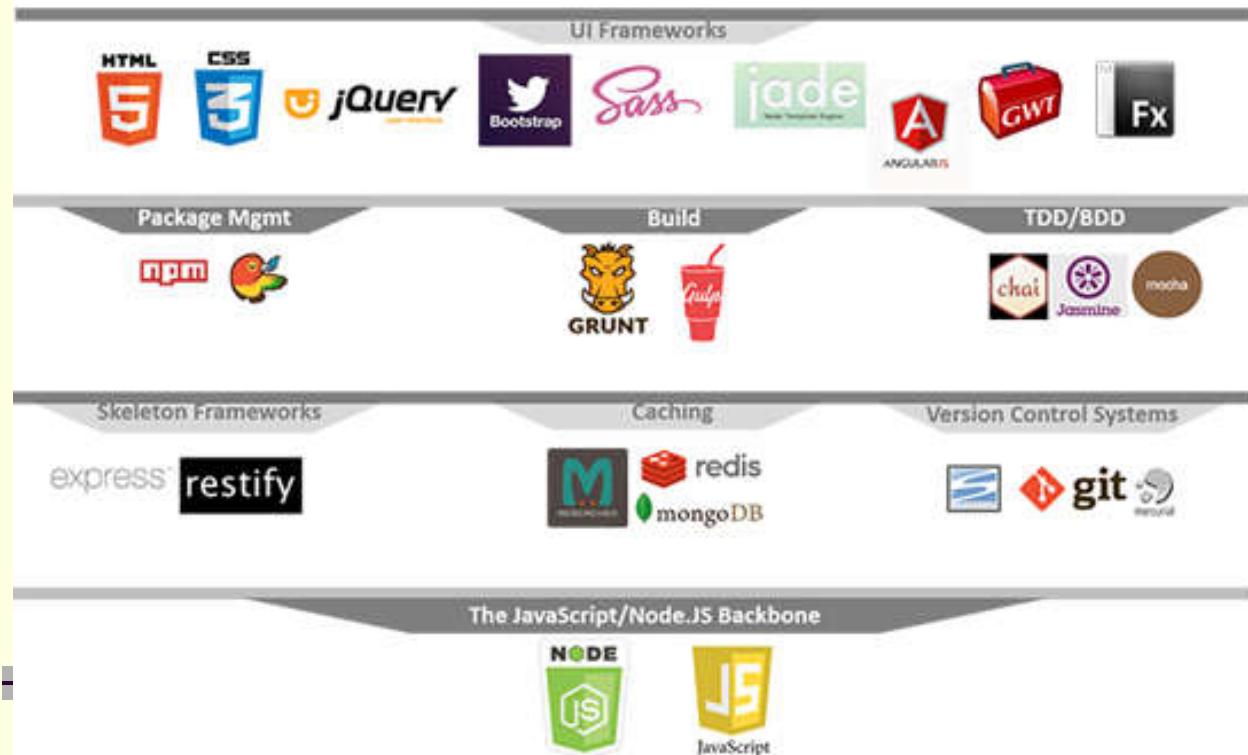
```
var http = require('http');
var server = http.createServer();
function control(petic, resp) {
 resp.writeHead(200, {'content-type': 'text/plain'});
 resp.write('Hola, Mundo!');
 resp.end();
}
server.on('request', control);
server.listen(8080);
```

<https://nodejs.org/es/>

# Anexo (II/VIII)

## Aplicaciones MEAN (Foto Completa)

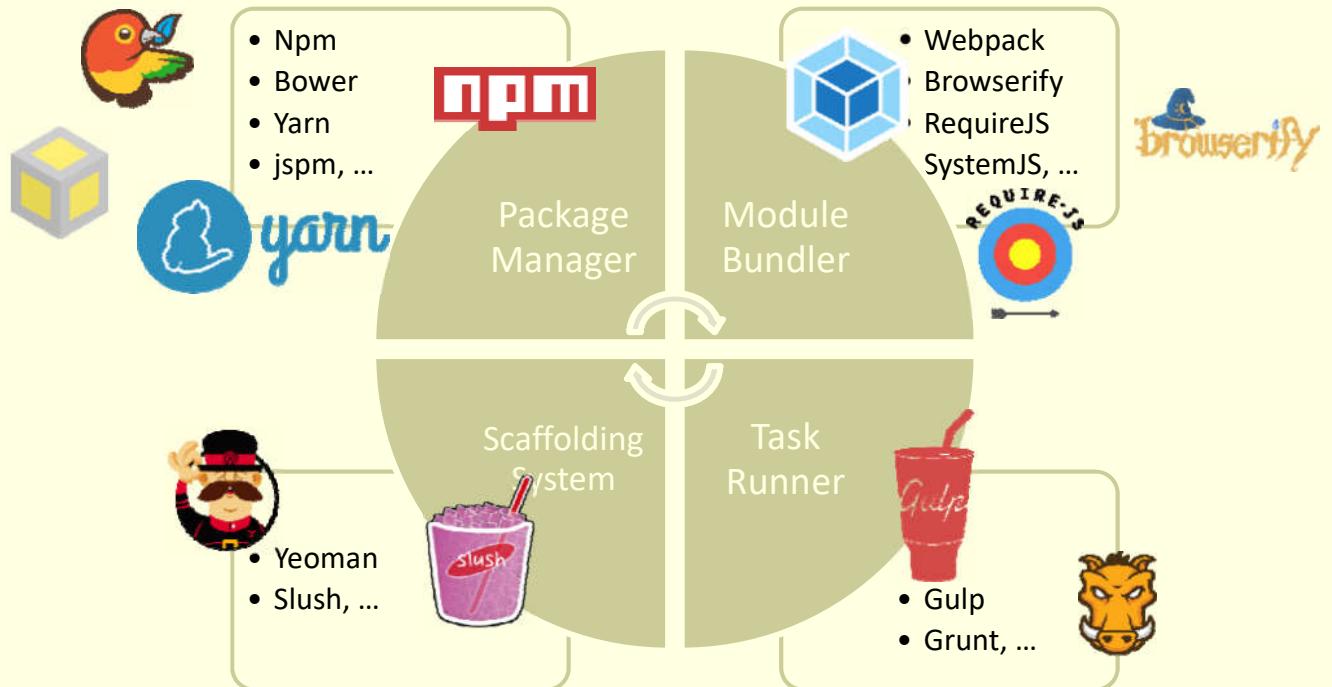
### MEAN Stack Expertise



# Anexo (III/VIII) Full Stack Developer



# Anexo (IV/VIII) Tools para nuevas web



# Anexo (V/VIII)

## Administradores de paquetes (package manager)

- Mantienen bases de datos de paquetes y dependencias que pueden ser utilizados en nuestros proyectos web.
- **Npm:** <https://www.npmjs.com/>
  - Significa Node JS Package Manager
  - El administrador de paquetes de JS más importante.
  - Es el gestor de paquetes de Node ( $\approx$  Composer PHP)
  - Utilizamos npm para instalar, compartir y distribuir código. Existen casi medio millón de paquetes
    - Se ejecuta con `npm init` que crea un archivo `package.json` con todas las librerías del proyecto
    - Para descargar librerías: `npm install`



# Anexo (VI/VIII)

## Administradores de paquetes (package manager)

- Mantienen bases de datos de paquetes y dependencias que pueden ser utilizados en nuestros proyectos web.
- **Bower:** <https://bower.io/>
  - Mantiene un registro actualizado de los paquetes (librerías, frameworks, components) del sitio web. Usado con Font-Awesome, Bootstrap y jQuery.
  - Utiliza un archivo `bower.json`, pero está en desuso este tool
  - Instalar: `npm install -g bower`
- **Yarn:** <https://yarnpkg.com/lang/en/>
  - Es un administrador de paquetes (a veces llamados módulos) rápido y seguro
  - ~~Paraleliza operaciones para ser más rápido.~~



# Anexo (VII/VIII)

## Empaquetadores de módulos (Module bundler)

- Empaquetan todos los módulos usados en simples archivos.
- **Webpack:** <https://webpack.js.org/>
  - Condensador de módulos para aplicaciones y sitios web
  - Usado internamente por Angular CLI
  - Básicamente convierte módulos, imágenes y archivos de estilo usados en simples archivos JavaScript o CSS ligeros.
  - Instalar: `npm install -g webpack`
- **Browserify:** <http://browserify.org/>
  - Permite escribir módulos con la misma sintaxis que Node.
  - Permite utilizar, importar y exportar módulos y sus dependencias al igual que en Node pero en el navegador.



© Antonio Santos Ramos 2018

115

# Anexo (VIII/VIII)

## Ejecutadores de tareas (Task runner)

- Permiten ejecutar tareas en tus proyectos; simples o combinadas para procesos más complejos. Ej.- "minificar CSS", "optimizar imágenes", "realizar tests unitarios".
- **Gulp:** <http://gulpjs.com/>
  - Herramienta creada en Node para automatizar tareas mediante un sistema de pipes (como Unix)
  - Instalar: `npm install --global gulp-cli`
- **Grunt:** <https://gruntjs.com/>
  - Herramienta para crear y gestionar tareas, similar a Gulp y desarrollada en Node. Usa un fichero "Gruntfile.js"
  - Instalar: `npm install -g grunt-cli`



© Antonio Santos Ramos 2018

<https://da-14.com/blog/gulp-vs-grunt-vs-webpack-comparison-build-tools-task-runners>

<https://www.keycdn.com/blog/gulp-vs-grunt>

116

# Material Design

- Angular Material es un “UI component framework” que implementa las especificaciones “Material Design” de Google

<https://material.angular.io/>

<https://material.angular.io/components/categories>

- Para instalar

- `npm install --save @angular/material @angular/cdk @angular/animations`

- Permite crear páginas que cumplen especificaciones PWA

<https://auth0.com/blog/creating-beautiful-apps-with-angular-material/>

- Ideas

- <https://angularexpo.com/>
  - <https://www.madewithangular.com/>