

Web UI

Introducción y conceptos básicos

HTML

- Es un lenguaje formado por **etiquetas** (tags) que sirve para **describir** documentos.
- **H**yper-**T**ext **M**arkup **L**anguage
- Con HTML se describen aspectos estáticos de un documento web
- HTML utiliza tags para delimitar estructuras
- Su sintaxis deriva de XML

Algunos tipos de tags:

- Separadores de párrafos / Saltos de líneas
- Encabezados
- Tablas / Columnas / Celdas
- Listas
- Bloques
- y más...



HTML

ESTRUCTURA

```
1 <html>
2   <head>
3     <title>DDS - UTN</title>
4     <!-- Acá se pueden agregar tags de control -->
5   </head>
6   <body>
7     <!-- Acá va el contenido que se muestra en pantalla -->
8   </body>
9 </html>
```

HTML

Ejemplo: BODY

```
1 <h1>DDS - Intro WebUI</h1>
2 <p>Esta presentación es una breve introducción a WebUI</p>
3 <div>
4   Algunos temas son:
5   <ul>
6     <li>HTML</li>
7     <li>CSS</li>
8     <li> ... </li>
9   </ul>
10 </div>
```

← Código HTML

DDS - Intro WebUI

Esta presentación es una breve introducción a WebUI

Algunos temas son:

- HTML
- CSS
- ...

Output →

HTML

ALGUNAS CARACTERÍSTICAS DE LOS TAGS

- Los tags tienen atributos asociados:
 - Atributos comunes a todos los tags
 - id: el valor debe ser único dentro de un documento
 - name: puede repetirse dentro del documento
 - class: sirven para agrupar elementos. Se usan para aplicarle estilos (visual) a los bloques. Pueden repetirse dentro del documento.
 - etc.
 - Atributos propios de cada tag

CSS

- Es un lenguaje que permite definir **reglas** de estilo que se aplican a ciertos elementos del documento HTML.
- Cada **regla** es un conjunto de propiedades
- Mediante las propiedades se pueden configurar, por ejemplo:
 - Tipo, color, y tamaño de tipografía
 - Bordes, márgenes y paddings de los bloques
 - Colores o imágenes de fondo
 - Alineación
 - Posicionamiento
 - y muchas cosas más.
- En las versiones más modernas se pueden definir transiciones, transformaciones y efectos.

CSS

REGLAS CSS

- Las reglas constan de:
 - **Selector**: sirve para especificar a qué se aplicará la regla
 - **Propiedades**: indica las propiedades que se aplicarán
- Algunos selectores
 - **#<id>** : selecciona el elemento usando el atributo “id”
 - **.<class>** : selecciona los elementos que contengan la clase indicada
 - **<tag>**: selecciona los elementos cuyo tag sea el indicado.

CSS

Ejemplo: HTML + CSS

```
1 <div class="encabezado">
2   <h1>DDS - Intro WebUI</h1>
3   <p>Esta presentación es una breve introducción a WebUI</p>
4   
5 </div>
6
7 <div class="listado">
8   Algunos temas son:
9   <ul>
10    <li>HTML</li>
11    <li>CSS</li>
12    <li> ... </li>
13  </ul>
14 </div>
```



```
1 .encabezado {
2   background: #ffffff;
3   padding: 10px;
4 }
5 .encabezado p {
6   color: #bb0000;
7 }
8 .encabezado img {
9   position: absolute;
10  right: 8px;
11  top: 20px;
12  width: 50px;
13  height: 50px;
14 }
15
16 .listado {
17  border: solid 1px #0000aa;
18  border-radius: 5px;
19  padding: 5px;
20 }
```


CSS

Ejemplo: HTML + CSS

DDS - Intro WebUI



Esta presentación es una breve introducción a WebUI

Algunos temas son:

- HTML
- CSS
- ...

Ejemplo disponible en <https://jsfiddle.net/v0gnvtx0/3/>

JavaScript

- Es un lenguaje de programación interpretado, creado en 1995 por Netscape.
- No tiene **nada** que ver con Java! (sólo parte de la sintaxis)
- La motivación inicial de este lenguaje fue la de poder aplicarle cierta **dinamismo** a los documentos HTML.
- Todos los navegadores modernos tienen la capacidad de interpretar JavaScript.
- **No** se limita sólo al uso del lado del cliente.
- Entre otras cosas, con javascript se puede:
 - Acceder a elementos del documento HTML y modificar sus atributos.
 - Realizar requests a un servidor

HTML 5 como concepto

Comúnmente se denomina HTML5 a la combinación de:

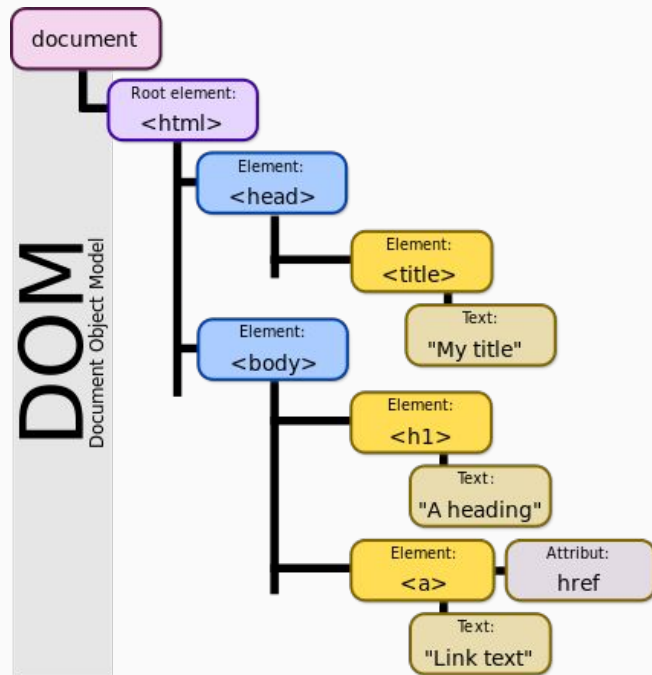
- HTML en su versión 5, que incluye nuevos tags para el manejo de contenido multimedia y web semántica.
- CSS en su versión 3
- JavaScript

Algunas características adicionales agregadas en HTML5 son el soporte de web-workers y storage local



DOM Tree

- Domain-Object-Model
- Es un modelo que representa en forma lógica la estructura de un documento HTML como un árbol.
- Mediante este modelo se puede “navegar” a través de la estructura del documento para poder manipular los nodos.



DOM Tree

MANIPULACIÓN

Mediante la manipulación del DOM se pueden crear o eliminar nodos, iterarlos, seleccionar nodos (usando los selectores), modificar atributos, etc.

Algunas formas de manipular el DOM Tree:

- JavaScript nativo: utilizando las funciones de selección propias del lenguaje.
- jQuery: es una librería en JavaScript que simplifica la manipulación del DOM, además de agregar otras funcionalidades.



Maquetado como herramienta de prototipado

- La maquetación es el proceso a través del cual se realizan prototipos de la aplicación (o documento) a desarrollar.
- Estos prototipos buscan acercarse al resultado final desde un enfoque visual, relegando la funcionalidad o contenido real.
- Utilizando HTML + CSS se pueden realizar prototipos con un esfuerzo relativamente bajo. Además, las reglas CSS generadas podrían ser reutilizadas.

Maquetado como herramienta de prototipado



Ejemplo

Maquetado de una aplicación a modo de demostración, utilizando CSS y HTML.

SPA - Single Page Application

Se conoce como **SPA** a las aplicaciones web cuyo contexto de ejecución corresponde con un único documento HTML.

Características

- Proveen una experiencia de usuario más fluida, similar a una aplicación desktop.
- Las acciones transcurren dentro del mismo contexto.
- El contenido dinámico se carga en background. Se evita el efecto visual de la recarga de la página
- Suelen implementar arquitecturas del tipo MVC (Model-View-Controller)

Ventaja principal

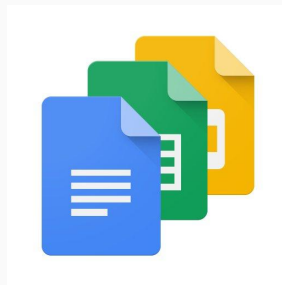
La experiencia de usuario es más fluida, similar a una aplicación desktop

SPA - Single Page Application

Ejemplos

SPA

- Google docs
- Gmail
- Twitter



No SPA

- Campus Virtual
- Diario Ole



Diseño web para dispositivos móviles

Concepto “MOBILE FIRST”

Problema

Existen muchos tipos de dispositivos y muchas resoluciones de pantalla diferentes. Cómo adaptamos nuestra aplicación para que se visualice correctamente en todos ellos?

- El enfoque **Mobile First** propone que se aborde la aplicación orientándola desde un principio a los dispositivos móviles.
- Utilizando este enfoque, se reducen los tiempos y costos relacionados con la adaptación de una aplicación a las diferentes resoluciones de los dispositivos existentes en la actualidad

Diseño web para dispositivos móviles

RESPONSIVE DESIGN

Responsive design surge como solución al problema de adaptación de la aplicación a diferentes resoluciones.

1. Propone dividir el área visible de la pantalla (viewport) en una grilla de 12 columnas
2. Se determinan rangos de ancho del tamaño del viewport: XS, S, M, L
3. En cada elemento de la UI se indica cuántas columnas ocupará en cada uno de los tamaños del viewport

De esta manera, se pueden adaptar los bloques a la resolución del dispositivo

Diseño web para dispositivos móviles

RESPONSIVE DESIGN

Photovoltaic PV Kit

Size: 6

£299.99

2

£599.98

X

Solar Heater Panel

Size: 8"

£100.00

1

£100.00

X

Inverter Pack

Size: Medium

£40.00

1

£40.00

X

Subtotal (inc. VAT)

£739.98


Discount Code

ADD

Discount Value 10%


- £73.99

Continue Shopping




Need help? Call on:

0844 544 9553



Secure Shopping Guarantee

We guarantee that every transaction you make with us will be safe.



Privacy Guarantee

We will never give, lease or sell your personal information.

FREE Delivery*

*Delivery for

Diseño web para dispositivos móviles

RESPONSIVE DESIGN



Diseño web para dispositivos móviles

Concepto “OFFLINE FIRST”

Alineado con el enfoque Mobile First, el concepto **Offline First** busca maximizar la experiencia del usuario, permitiendo que la aplicación mantenga su funcionalidad a pesar de que el dispositivo pierda conectividad.

Algunas técnicas utilizadas para cumplir con este enfoque están asociadas con el uso de local storage para almacenar datos y luego sincronizar cuando el dispositivo esté conectado



Diseño web para dispositivos móviles

Web Apps vs Native Apps

Aplicación nativa

- Optimizada para hacer uso eficiente de los recursos del dispositivo
- Interfaz gráfica asociada al *theme* del dispositivo
- Disponibles desde AppStores
- Sólo sirve para una familia de sistema operativo (iOS, Android, etc)

Aplicación web

- Se pueden obtener muy buenos resultados en términos de UX
- Usa HTML5 *standard*
- Se puede hacer un único desarrollo para cualquier tipo de dispositivo
- Disponible a través de su URL



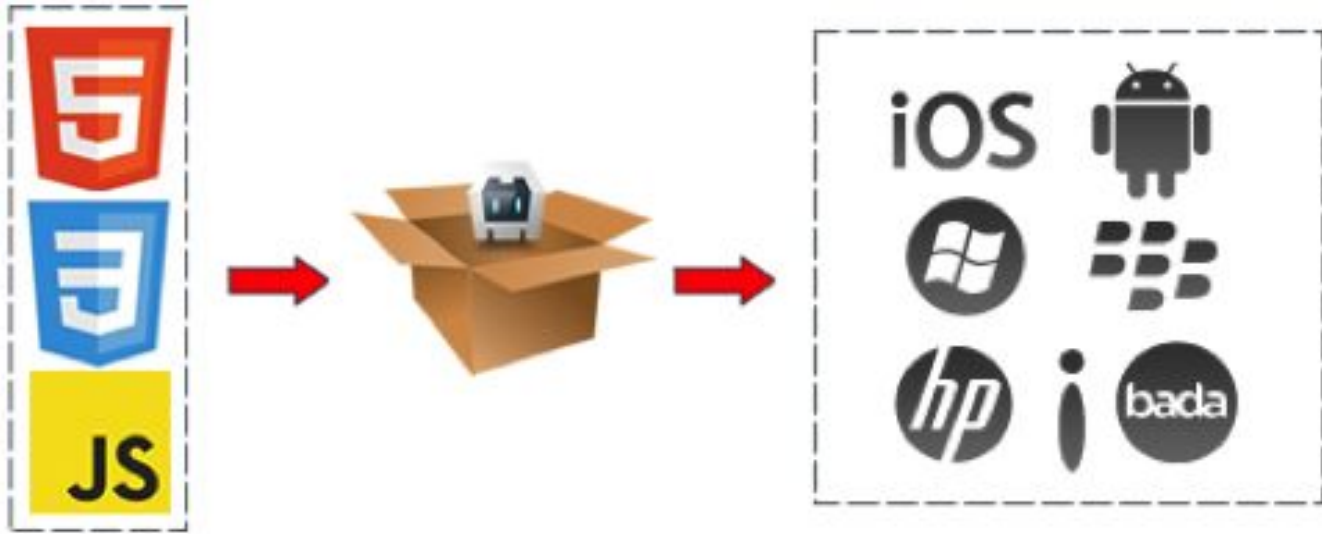
Diseño web para dispositivos móviles

Web containers

- Existen web containers que permiten *empaquetar* una aplicación web, generando aplicaciones nativas.
- De esta manera se fusionan las ventajas del mundo **nativo** y **web**.
- Los web containers además poseen plugins para acceder a funcionalidades propias del dispositivo móvil (cámara, teléfono, geolocalización, etc).
- Por ejemplo: PhoneGap, Apache Cordova.

Diseño web para dispositivos móviles

Web containers

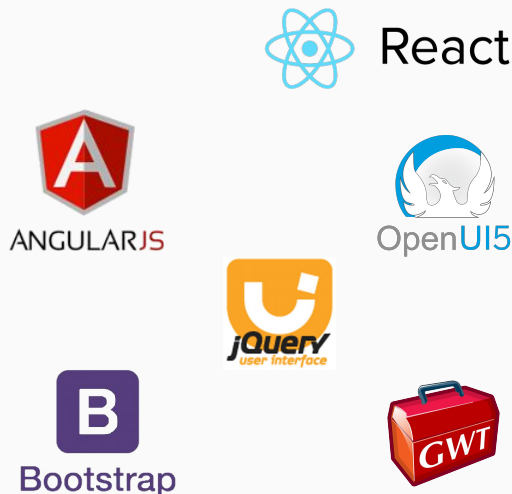


Frameworks

En la actualidad existen **muchos** frameworks de desarrollo web, cada uno con sus características, ventajas y desventajas.

Algunos de ellos son:

- AngularJS (Google)
- ReactJS (Facebook)
- Bootstrap (Twitter)
- OpenUI5 (SAP)
- Google Web Toolkit
- muchos muchos más...



Frameworks

ANGULAR JS

- Mantenido por Google
- Permite desarrollar SPA
- Propone una arquitectura MVC
- Data binding
- Provee una extensión al lenguaje HTML mediante *directivas*.
- Routing
- Hay una gran comunidad alrededor



ANGULARJS

Frameworks

BOOTSTRAP

- No es un framework a nivel arquitectura, sino orientado al diseño
- Mantenido por Twitter
- Incluye un set de componentes UI reutilizables y configurables
- Implementa responsive design mediante Grid layout



Bootstrap



¿Preguntas?