



Desarrollo en HTML, CSS y Javascript de Apps Web, Android, IOS, FirefoxOS, ...



Objetivo del curso

Curso de **diseño y programación de aplicaciones Web** multi-dispositivo

(PC, tableta, teléfono, ...)

para la plataforma Web

(**HTML5, CSS3 y JavaScript5**)

y de como empaquetarlas

para su publicación en las **tiendas**

de Android, iOS,



Diseño de servicios en la nube, utilizando JavaScript, para acceso móvil y multi-dispositivo con HTML5*



◆ Programa de especialización con 5 cursos MOOC

- Desarrollo de Aplicaciones en HTML5, CSS y Javascript, incluyendo Dispositivos Móviles Firefox O.S.
- Desarrollo avanzado de Aplicaciones HTML5 y Firefox O.S.*, incluyendo técnicas de ingeniería de software
- Desarrollo de aplicaciones HTML5 multi-terminal (Android, iOS, ...)*
- Desarrollo de servicios en la nube con HTML5, Javascript y node.js*
- Desarrollo avanzado de servicios en la nube con Javascript y node.js*



***Nota:** Los otros 4 cursos empezarán en MiriadaX en 2014 y 2015.

Estructura del curso

- ◆ El curso consta de 5 tramos de 2 modulos
 - Diseñado para realizarse en 5 semanas (1 semana por tramo)
 - ◆ Además se deja una semana adicional por tramo
 - ◆ tiempo máximo de realización ~12 semanas (2 semanas por tramo + ..)
- ◆ Ejercicios P2P de **entrega obligatoria**
 - Al final de cada módulo
 - además hay tests obligatorios y mas ejercicios P2P opcionales
- ◆ Apertura y cierre de tramos
 - Tramo 1: comienzo del curso y cierra aprox. final semana 4
 - Tramo 2: comienzo semana 2 y cierra aprox. final semana 6
 - Tramo 3: comienzo semana 3 y cierra aprox. final semana 8
 - Tramo 4: comienzo semana 4 y cierra aprox. final semana 10
 - Tramo 5: comienzo semana 5 y cierra aprox. final semana 12

- ◆ **Modulo 1:** Introducción a los elementos básicos de HTML y CSS, así como al uso de las herramientas del curso.
- ◆ **Modulo 2:** Estructuración de interfaces en HTML y CSS adaptados a un entorno multi-pantalla adaptado a móviles y tabletas.
- ◆ **Modulo 3:** Introducción a JavaScript: programas y sentencias, expresiones con números y strings, variables, funciones, objetos, propiedades y métodos
- ◆ **Modulo 4:** Descripción detallada de strings, números y booleanos. Más sobre objetos DOM (Document Object Model) y primeros pasos con eventos e interacción con el usuario.
- ◆ **Modulo 5:** Bucles y arrays, funciones como objetos, ámbitos de visibilidad y cierres. Introducción al uso de la librería jQuery para acceder a DOM y para atender eventos.
- ◆ **Modulo 6:** jQuery UI y patrones de interacción con el usuario. Introducción a la librería Bootstrap para diseño adaptado a móviles y tabletas.
- ◆ **Modulo 7:** Más sobre arrays, objetos, propiedades, métodos, tipos y clases. Repaso de los conceptos más importantes basado en un ejemplo de presentación de citas.
- ◆ **Modulo 8:** LocalStorage, iFrames, origin policy y JSON.
- ◆ **Modulo 9:** Introducción a los gráficos y animaciones con SVG (Scalable Vector Graphics) y algunos ejemplos de geolocalización y google maps.
- ◆ **Modulo 10:** Empaquetar aplicaciones Web para Android, iOS o FirefoxOS.



Módulos

Actividades de un Módulo

◆ **Tarea 0:** Descargar transparencias y ejemplos del módulo

- Fichero ZIP para descargar con
 - ◆ Transparencias en formato PDF
 - ◆ Directorio con ejemplos presentados en el módulo
 - para realizar ejercicios modificando los descargados

◆ **Tareas de Aprendizaje (varias):**

- un **video** o **screencast** del tema (3 y 15 minutos)
 - ◆ evaluado (no siempre) con un **test** o un **ejercicio P2P opcional**

◆ **Tarea final:** Ejercicio P2P final de cada modulo

Equipos y servicios a utilizar

- ◆ Un PC o portatil de trabajo (necesario)
 - con S.O. Windows de Microsoft
 - con S.O. UNIX (Ubuntu, ..) de software libre
 - Ordenador MAC con sistema operativo con OS X
- ◆ Móvil o tableta (conveniente para probar, pero no necesario)
 - Android, iPhone, iPad, FirefoxOS, .. pero no es necesario
 - ◆ Se puede utilizar un simulador de FirefoxOS para probar Apps
- ◆ Cuenta en Neocities (es gratis)
 - para publicar Páginas y Apps Web en la nube
 - ◆ <https://neocities.org/>

Herramientas a utilizar

◆ Simulador/Editor interactivo

- Editor y visualizador de HTML5, CSS y JS con ejemplos cargados
 - ◆ <http://vishub.org/excursions/2209.full>

◆ Sublime Text 2

- Editor de HTML, CCS y JavaScript con ayudas visuales
 - ◆ <http://www.sublimetext.com>

◆ Creadores (wysiwyg) de páginas Web (opcional)

- Dreamweaver: Creador de páginas Web muy potente e intuitivo.
 - ◆ <http://www.adobe.com/es/products/dreamweaver.html>
- Bluegriffon: editor wysiwyg gratuito y de código abierto
 - ◆ <http://www.bluegriffon.org>

◆ Navegador (Chrome, Firefox, ...) y su entorno de desarrollo

- <https://developer.chrome.com/devtools>
- <https://www.mozilla.org/es-ES/firefox/new/>

Simulador en ViSH

Guardar modificaciones en memoria local del navegador

Descargar ejemplo con modificaciones a un fichero

Subir código de un fichero al area de edición

Inicializar

Seleccionar módulo

Seleccionar tema

Seleccionar ejemplo

Visualizar modificaciones

Area de visualización de la aplicación

Area de edición del código fuente de los ejemplo

The screenshot shows the ViSH web editor interface. At the top, there is a navigation bar with a dropdown menu showing 'mod_01'. Below it, there are tabs for '01_mi_primera_pagina', '02_Estructura_HTML', and '03_herencia_CSS'. A toolbar contains icons for 'Inicializar', 'Guardar', 'Subir código', 'Descargar', and 'Visualizar'. The main area is split into two panels. The left panel, titled 'Ejemplo: Primera pagina con imagen', contains HTML code with line numbers 1 through 19. The right panel, titled 'Resultado:', shows the rendered output: a heading 'Titular nivel 1', a paragraph 'Párrafo de texto. **negrita** y *cursiva*', a bulleted list with three items, and an image of a man's portrait.


```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Párrafo con imágenes</title>
6 </head>
7
8 <body>
9 <h1>Titular nivel 1 </h1>
10 <p>Párrafo de texto. <strong>negrita</strong> y <em>cursiva</em> </p>
11 <ul>
12 <li>Elemento de la lista 1</li>
13 <li>Elemento de la lista 2</li>
14 <li>Elemento de la lista 3</li>
15 </ul>
16 
17 </body>
18 </html>
19
```

Resultado:

Titular nivel 1

Párrafo de texto. **negrita** y *cursiva*

- Elemento de la lista 1
- Elemento de la lista 2
- Elemento de la lista 3



<http://vishub.org/excursions/2209.full>



Final del tema
Muchas gracias!



Introducción a Internet y a la Web

Cientes, servidores y la nube



◆ **Cientes:**

- Dan acceso a información y servicios en Internet

◆ **Servidores:**

- Alojan la información y los servicios

◆ **La nube:** conjunto de terminales y servidores

- interconectados con aplicaciones y protocolos de Internet

◆ **TCP/IP:** protocolos de interconexión de redes de fibra, cable, WIFI, ...

- sobre los que se implementan las aplicaciones de Internet y sus protocolos

Cientes, navegadores y tiendas



◆ **Cientes** de acceso a Internet más importantes

- PCs, portátiles, tabletas, teléfonos inteligentes

◆ **Navegador (browser)** cliente Web de acceso a servidores

- Utilizando: **URL, HTTP, HTML, CSS y JS**
 - ◆ p.e. Chrome, Firefox, Internet Explorer, Opera, Safari, ...



◆ **Tiendas de aplicaciones**

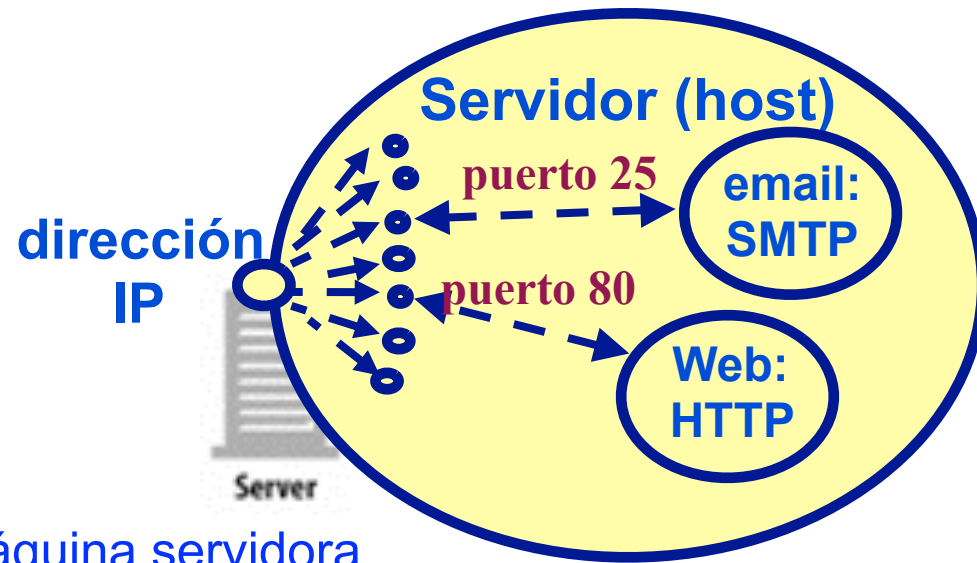
- Instalan aplicaciones en móviles y tabletas
 - ◆ Las aplicaciones usan las normas de la Web (URL, HTTP,)

Máquina servidora (host)



- ◆ Contiene información y servicios
- ◆ Una máquina servidora tiene una dirección “**conocida**” en Internet
 - Dirección simbólica (de dominio o **DNS**): **upm.es**, **google.com**, ...
 - ◆ Cada dirección de dominio tiene una dirección IP (binaria) asociada
- ◆ Hay 2 tipos de direcciones IP: IPv4 e IPv6
 - **IPv4**: versión 4 del protocolo IP con dirección de **32 bits** o **4 octetos**
 - ◆ ejemplo: **192.9.0.144**, **127.0.0.1** (localhost - mi máquina), ...
 - **IPv6**: versión 6 del protocolo IP (última) con dirección de **128 bits**
 - ◆ ejemplo: **2001:db8:85a3::8a2e:370:7334**,

Servidores y puertos



◆ Puerto

- Dirección de 16 bits dentro de la máquina servidora
 - ◆ Es donde se instala el **programa servidor**
- El **programa servidor** es lo que normalmente denominamos **servidor**
 - ◆ Cliente y servidor se comunican a través de un protocolo: HTTP, SMTP,
 - Utilizando el interfaz de sockets TCP/IP para comunicar entre ambos

◆ Los servicios tienen un protocolo y un puerto por defecto

- **Web:** protocolo HTTP (puerto 80), HTTPS (443)
- **Email:** protocolo SMTP (puerto 25), POP3 (110), IMAP143)
- **Shell segura:** protocolo SSH (puerto 22)

◆ Si un servidor no está en el puerto por defecto

- Su dirección debe incluir el puerto, p.e. dit.upm.es:8080, 192.9.0.144:8080

URL



◆ URL (Uniform Resource Locator)

- Dirección de un recurso en un servidor en Internet

◆ Internet soporta muchos tipos de servicios diferentes

- Cada tipo de servicio utiliza un URL y protocolo diferentes

◆ Algunos ejemplos de tipos de URLs

- **URL Web:** utiliza HTTP para acceder a recursos, incluye
 - ◆ Protocolo, servidor y recurso (camino): <http://google.com/picture.png>
- **URL de correo (email):** identifica el buzón de usuario, incluye
 - ◆ protocolo, buzón de usuario y servidor: mailto:pepe_garcia@gmail.com

HTTP (HiperText Transfer Protocol)

◆ Protocolo del Web

- Procesa recursos identificados por un URL en un servidor remoto

◆ Métodos o comandos principales de HTTP

- **GET:** trae al cliente (lee) un recurso identificado por un **URL**
- **POST:** crea un recurso identificado por un URL
- **PUT:** actualiza un recurso identificado por un URL
- **DELETE:** borra un recurso identificado por un URL
- (hay mas comandos)



Aplicación Web

- ◆ Aplicaciones ejecutables en un navegador creadas con
 - **HTML, CSS y JavaScript**



- ◆ **HTML**

- Lenguaje de marcado de páginas Web
 - ◆ define la estructura del contenido de una página Web
- En WebApps define la interfaz de la aplicación con el usuario

- ◆ **CSS**

- Define el estilo visual de un una página o aplicación Web (HTML)

- ◆ **JavaScript**

- Lenguaje de programación de aplicaciones de cliente

Aplicación Web: HTML, CSS y JavaScript

◆ HTML

- Lenguaje de marcado

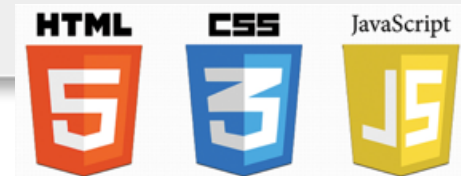
◆ CSS

- Estilo la visualización

◆ JavaScript

- Lenguaje de programación

```
02-date_CSS.htm UNREGISTERED
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">
  <style type="text/css">
    body {color: blue;}
  </style>
</head>
<body>
  <h3>Fecha y hora</h3>
  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```



WebApps o aplicaciones de cliente

◆ Aplicaciones que **residen en un servidor**

- pero se **ejecutan en un cliente**

- ◆ Se identifican con un **URL**: `http://upm.es/apps/webapp.html`

- El cliente trae la aplicación del servidor con el **protocolo HTTP (GET)**

◆ Las apps se construyen con las tecnologías de la Web

- URLs, HTTP, HTML, CSS y JavaScript

1) Cliente solicita WebApp identificada con URL

2) Script se ejecuta al cargar la página Web en el navegador:



Client

Solicitud HTTP GET asociada a un URL

Respuesta HTTP: página Web con script



Server

```
02-date_CSS.htm UNREGISTERED
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">
  <style type="text/css">
    body {color: blue;}
  </style>
</head>
<body>
  <h3>Fecha y hora</h3>
  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```



Servidor sirve fichero identificado por URL



Final del tema
Muchas gracias!

