

# DISEÑO DE INTERFACES WEB

## U.D. 1: Planificación de interfaces gráficas



# Elementos de diseño: Percepción visual

- **Sitio Web:** Conjunto de páginas interconectadas entre sí bajo un dominio, alojadas en un servidor Web y que se hacen accesibles para los usuarios a través de un navegador Web.
- **Homepage:** Página principal de un sitio Web desde la que se puede acceder a todos los contenidos ofrecidos por el sitio.
- Desde el punto de vista del diseño, todas las páginas que componen el sitio deben cumplir unos criterios de **homogeneidad y consistencia** ya que el diseño Web es una marca comercial o personal, por lo que deben siempre identificarse con esa marca.
- El objetivo principal es **comunicar**, ya sea con fines comerciales o simplemente informativos, por lo que el diseño del sitio Web debe ser **atractivo y funcional**.

# Elementos de diseño: Percepción visual

- Una de las **primeras impresiones** que causará la interfaz de usuario será **visual**, por lo que dependiendo de cómo sea ésta influirá de una determinada forma en el usuario.
- **Sentido gráfico:** El diseñador del sitio debe buscar un equilibrio entre los elementos que constituyen la interfaz. Por ejemplo, un sitio con demasiados elementos animados distraen, enmascarando el mensaje de la comunicación.
- **Área de diseño:** Tamaño que se asignará al espacio del que se dispone para la composición gráfica.
- **Composición gráfica:** Se compondrá principalmente de texto e imágenes. Por ejemplo, no es lo mismo diseñar para un cartel de conciertos de música clásica que de rock.

# Elementos de diseño: Percepción visual

- El diseño de sitios Web debe tener presentes:
  - Color, fuentes, iconos.
  - Navegación.
  - Interacción.
  - Sonido.
  - Diseño y maquetación.
  - Propósito.
  - Etc.

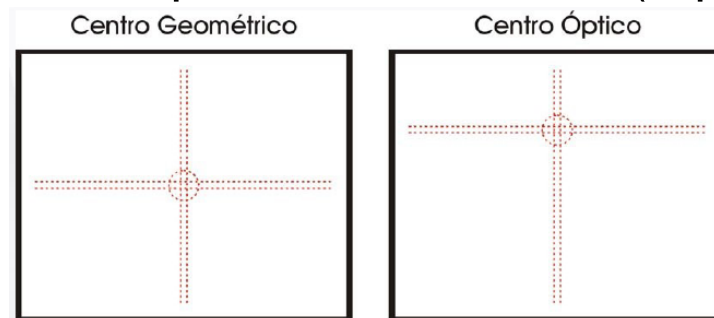


# Fundamentos de la composición

- **Composición gráfica:** Ordenar todos los elementos (texto, imágenes, etc.) destinado a impactar visualmente en el público receptor del mensaje. Existen unas pautas relacionadas con la **percepción**:
  - **Componentes psicosomáticos del sistema nervioso:** Mediante la vista se recoge información visual (se percibe distintas formas, ubicaciones, longitudes de onda de un color, etc.) que luego el cerebro interpreta como contornos, texturas, dimensiones, etc., dotándolas de un significado gráfico.
  - **Componentes de tipo cultural:** Influyen en la interpretación que hacemos de los estímulos desde un punto de vista cultural. Por ejemplo, el luto se relaciona en occidente con el color negro pero en oriente con el blanco.
  - **Experiencias compartidas con el entorno:** Existen dualidades que se van aprendiendo desde la infancia que también serán utilizadas como patrones con los que interpretar la realidad. Por ejemplo, hierba/verde, hielo/frío, etc.

# Fundamentos de la composición

- Hay más factores relacionados con la disposición de los elementos para conseguir una composición adecuada (equilibrada).



- **Equilibrio visual. Tipos:**
  - **Formal:** Responde a criterios geométricos.
  - **Informal:** Responde a criterios creativos y artísticos.
- En relación a los componentes, el diseñador debe tener en cuenta:
  - **Peso visual:** Formas pequeñas poseen menos peso visual que las grandes, si son regulares aumenta su peso, también cuanto más luminosos.
  - **Tamaño:** Para el mismo color, mayor peso el de mayor tamaño.
  - **Posición:** En occidente, los situados en la parte superior izquierda tienen más protagonismo.

# Fundamentos de la composición

- **Tensión compositiva:** Opuesto al equilibrio desde el punto de vista estructural. Finalidad captar y dirigir la mirada del observador. Técnicas:
  - **Sugestiva:** Dirigir intencionadamente la atención a un punto utilizando elementos de apoyo. Por ejemplo, imágenes de personas que miran o líneas que apuntan hacia un punto determinado.
  - **Rítmica:** Basada en la tendencia innata del ojo a completar secuencias de elementos (números, formas, figuras geométricas o colores), agrupando aquellos que poseen formas semejantes.



# Color, tipografía, iconos

- Elementos más destacados dentro de las composiciones para diseñar sitios Web.
- **Colores:** En entornos gráficos digitales se forman a partir de tres básicos, rojo #FF0000, verde #00FF00 y azul #0000FF.
  - La intensidad se expresa como un número hexadecimal respecto de la de los básicos. Por ejemplo, amarillo #FFFF00, mezcla de rojo y verde.
  - Para hacer un color más oscuro, se reduce la intensidad del componente. Por ejemplo, rojo más oscuro #990000.
  - Elegir una combinación de colores apropiada es una tarea difícil, existen herramientas software que ayudan a la hora de crear combinaciones armónicas: <https://aulacm.com/gama-colores-diseno-grafico-web/>



# Color, tipografía, iconos

- **Tipografía:** Transmitir información mediante texto es lo más común.
  - **Fuente:** Conjunto de caracteres con un estilo o modelo gráfico particular, sinónimo de tipo de letra.
  - Las más comunes suelen ser las Sans Serif, entre ellas Verdana, Arial y Helvetica.
  - No es aconsejable utilizar muchos tipos de fuentes.

Last Words  
This is Type-Ra  
Underwood 1913  
39 Smooth Font  
King 668  
F25 executive  
TELEGRAPHUM  
Another Typewriter  
Stalker font  
XBAND ROUGH

# Color, tipografía, iconos

- **Iconos:** Imágenes gráficas generalmente pequeñas y suelen ser metáforas de información o acciones que el usuario puede consultar y hacer en un sitio Web. Evitan leer ya que el usuario está acostumbrado a ellos, suelen estar **estandarizados**.
  - Nunca son totalmente claros e inequívocos, existe el riesgo de que los malinterprete.

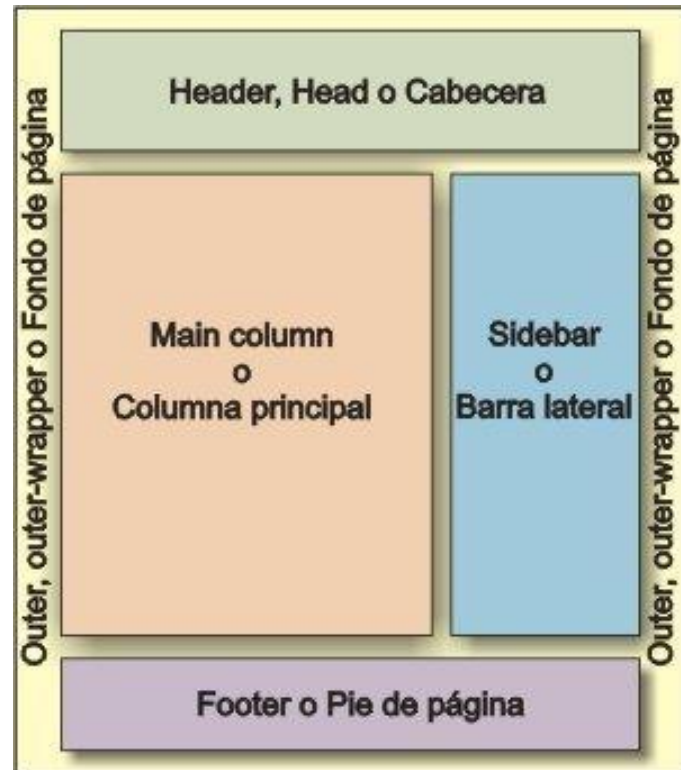


# Ejercicios

- Ejercicio: Observa los siguientes sitios Web, ¿qué piensas y sientes al verlos? Indica cuatro diferencias entre ellos desde el punto de vista de la composición gráfica y del equilibrio visual.
  - <https://www.art.yale.edu/>
  - <https://www.amazon.es>
- Ejercicio: Desde el punto de vista de los colores, fuentes e iconos, compara estos dos sitios Web. Indica al menos tres aspectos positivos y tres negativos de cada uno.
  - <https://www.lingscars.com/>
  - <https://www.avis.es>

# Componentes de una interfaz Web

- Las partes básicas que forman un sitio Web, aunque siempre opcionalmente dependiendo de lo que se quiera mostrar en él, son:
  - Cabecera
  - Sistema de navegación
  - Cuerpo de la página
  - Pie de página
  - Espacios en blanco



# Componentes de una interfaz Web

- **Cabecera:** Zona de la interfaz Web situada en la parte superior de la misma, en la que se ubica generalmente el logotipo con texto, acceso al formulario de autenticación (login) o al de registro, etc.
  - Su objetivo es identificar y homogeneizar todas las páginas pertenecientes al sitio Web. Crear una separación visual con el contenido central.
  - No es obligatoria, aunque muy habitual.



- <https://cpilosenlaces.com/>

# Componentes de una interfaz Web

- **Sistemas de navegación:** Permiten la navegación por las diferentes secciones y páginas que componen el sitio Web. Generalmente se presentan como menús desplegables, de árbol, pestañas, etc.
  - **“estás aquí”:** Formato de menú que permite ubicar al usuario en el total del sitio.



# Componentes de una interfaz Web

- **Cuerpo de la página:** Parte de la página donde se presenta toda la información referente a los contenidos que se pretenden comunicar. Suele ser el objetivo del sitio.
  - Es habitual un **título** en la parte superior reforzado mediante un menú de navegación tipo “estás aquí” de tamaño superior al resto de contenidos.
  - Es importante que todos los elementos gráficos presenten un aspecto similar, respetando el estilo de todo el sitio.

## ¿Qué hacemos?



El **Centro Público Integrado de Formación Profesional Los Enlaces** comenzó a funcionar como instituto en el curso 1985-86. Su puesta en marcha responde a la preocupación de la Administración Educativa por expandir este tipo de enseñanzas en Zaragoza que, además, coincide con la apertura de otros centros de Formación Profesional en la ciudad.

Después de unos años como I.E.S, tiempo durante el cual se imparten y se comparten las enseñanzas de E.S.O, Bachillerato y Formación Profesional, el centro se transforma, definitivamente, en un **Centro Integrado de Formación Profesional** el curso 2010-2011.

# Componentes de una interfaz Web

- **Pie de página:** Parte situada en el inferior, bajo el cuerpo.
  - Uso común para mostrar enlaces a servicios muy particulares del sitio Web como formulario de contacto, condiciones de uso, política de privacidad, responsable directo, uso de cookies, etc. También incluye los menús para no tener que buscar el menú principal. Contenidos centrados en pantalla.

## RECONOCIMIENTOS



 **Los Enlaces**  
centro público integrado  
de formación profesional

📍 C/Jarque de Moncayo nº 10.  
50012 ZARAGOZA.

☎ 976 300 804

✉ [atencionalpublico@cpiosenlaces.com](mailto:atencionalpublico@cpiosenlaces.com)

### SECCIONES

Centro >  
Secretaría >  
Oferta formativa >  
Servicios >  
Alumnado >  
Innovación >

### ÚLTIMAS NOTICIAS

Claustro de inicio de curso para profesorado  
con destino confirmado en el centro  
30 de agosto de 2023  
Instrucciones depósito proyecto septiembre  
2023  
29 de agosto de 2023  
Curso de especialización videojuegos y RV:  
listas definitivas y matrícula  
13 de julio de 2023



# Componentes de una interfaz Web

- **Espacios en blanco:** Zonas donde no hay ningún otro elemento gráfico.
  - Entre sus objetivos está el compensar el peso visual del resto de elementos, crear márgenes o separación entre ellos, encuadrándolos de forma adecuada y marcar los límites que estructuran la composición, haciendo la interfaz más equilibrada y limpia. Establece la rejilla base de la composición.



# Lenguajes de marcas

- **Lenguaje de marcado:** Forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. Es un tipo de lenguaje de programación (aunque no tiene variables ni funciones aritméticas) que combina texto con información extra acerca de dicho texto.
  - Las marcas aportan información sobre la estructura y la presentación del texto.
  - Ejemplos: HTML, XML, XHTML, etc.
- **HTML:** Permite describir contenidos de una página Web de forma textual y estructurada para que un navegador Web conectado a Internet los visualice.

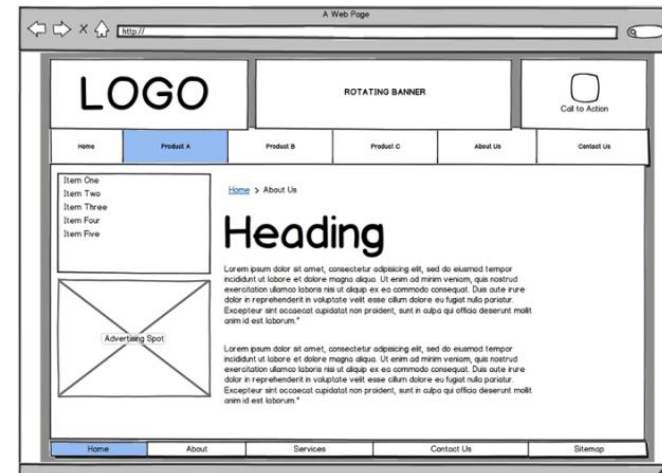
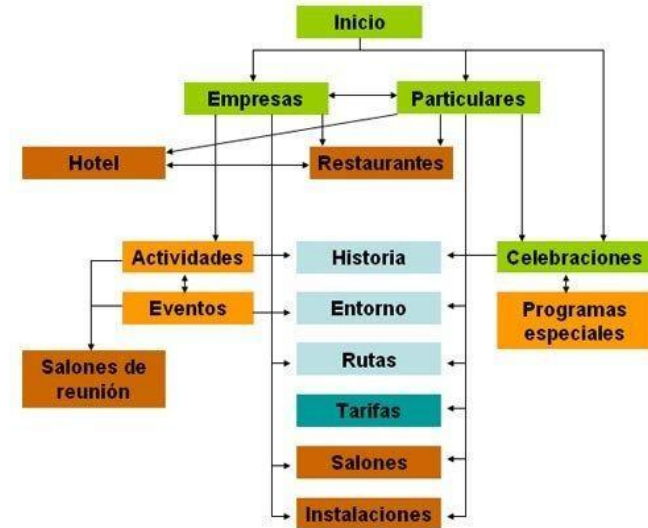
# Maquetación Web.

## Elementos de ordenación

- **Maquetación Web:** Distribución de los elementos (texto, imágenes, etc.) que conforman una página Web. La ventaja es tener separado el contenido de la presentación, mejorando el mantenimiento del código, tiempo de desarrollo y de carga.
- **Capas** (layout / DIV): Contenedores donde se colocan los elementos. Son bloques con contenido HTML que pueden posicionarse de manera dinámica y anidarse (unas dentro de otras). Se necesita del lenguaje CSS.
- **Tablas y Marcos:** Mantenimiento complejo (en desuso).

# Mapa de navegación y Prototipo

- **Mapa de navegación:** Esquema con las secciones del sitio Web y la relación entre los bloques de contenidos.
- **Prototipo Web:** Borrador o modelo inicial a partir del cual se empieza a concebir y desarrollar la idea original del diseño de un sitio Web.
  - Considera los elementos que conforman la interfaz, los que serán comunes en las distintas páginas, cómo se organiza el mapa de navegación, extensión del sitio, aspectos técnicos, de usabilidad, accesibilidad, etc.



# Guía de estilo

- Documento que **define las pautas** y normas de calidad que debe seguir una interfaz Web. Garantiza la coherencia, homogeneidad y consistencia del sitio ya que establece las pautas que todos los miembros de un equipo de trabajo deben seguir.
  - No existe una estructura única.
  - Debe responder a: ¿Qué colores tendrá la Web? ¿Qué fuentes se usarán? ¿Qué formato de fuente se usará para los títulos, subtítulos, encabezados y el texto principal? ¿Cuál será la estructura? ¿Habrá encabezado, pie o menús? ¿Habrá un menú o varios? ¿Qué imágenes se mostrarán? ¿Dónde se colocarán? ¿Habrá logotipo?
  - Ejemplo: [https://es.wikipedia.org/wiki/Wikipedia:Manual\\_de\\_estilo](https://es.wikipedia.org/wiki/Wikipedia:Manual_de_estilo)

# Aplicaciones para el desarrollo Web

- En la actualidad existen muchas **herramientas software** que facilitan la planificación, diseño, desarrollo y mantenimiento de sitios Web. Tipos:
  - Generales: Navegadores, tratamiento de imagen y vídeo, planificación, transferencia de archivos, etc.
  - De diseño (prototipos): Balsamiq, NinjaMock, Canva, PowerPoint.
  - Multimedia: Silverlight, OpenSilver.
  - Programación: Notepad++, UltraEdit, Visual Studio Code.
  - Editores y validadores de HTML/CSS: W3C, W3Schools, Fiddle, CodePen.
  - Gestores de contenidos: WordPress, Wix, Joomla, Shopify.

# Aplicaciones para el desarrollo Web

- Las numerosas versiones del lenguaje HTML y las malas prácticas de algunos desarrolladores Web hacen difícil encontrar páginas bien codificadas, es decir, aquellas que cumplen los estándares del W3C.
- Los navegadores suelen ignorar errores en el código para mostrar la página Web.
- ¿Por qué es importante **validar el código** de mi página Web?
  - Mantenimiento: Facilita la reutilización de código.
  - Buscadores: Mejora la posición de la página en buscadores.
  - Accesibilidad: Facilita el funcionamiento correcto en navegadores.

# Generación de documentos y sitios Web

- **Gestor de contenidos (CMS):** Interfaz que controla una base de datos donde se aloja el contenido del sitio.
  - Permite manejar de forma independiente el contenido y el diseño.
  - Alejan aspectos técnicos, un administrador puede crear contenidos sabiendo solo manejar un procesador de texto.
  - Ejemplos: WordPress, Wix, Joomla, Shopify.





# Plantillas de diseño Web

- Son sitios Web prediseñados, ya disponen de una estructura definida y solo hay que incorporar los contenidos particulares sin preocuparse del aspecto.
  - Permiten desarrollar el sitio de forma ágil y rápida.
  - Aportan la componente visual y llevan asociadas código HTML y hojas de estilo.
  - Adecuadas para sitios sin estructura compleja cuya función principal solo sea mostrar información.
  - Ejemplo (WordPress): <https://www.hostinger.es/tutoriales/mejores-plantillas-wordpress>

# Interacción persona-ordenador

- **IPO:** Disciplina que estudia el intercambio de información entre personas y ordenadores.
  - Su objetivo es que este intercambio sea más eficiente, minimizando errores e incrementando la satisfacción.
  - Ofrece un marco empírico con todo lo referente a calidad de uso e interacción de interfaces de usuario.
  - Su origen está en la Psicología Cognitiva, rama que estudia la percepción, memoria, aprendizaje, procesamiento de la información, resolución de problemas, etc.



# Ejercicios

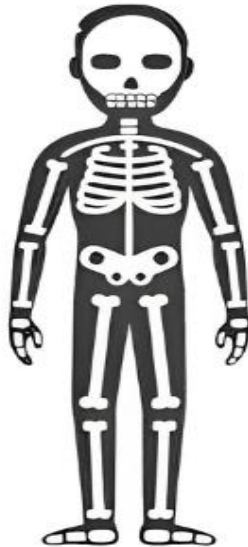
- Realiza todos los ejercicios del bloque 'Ejercicios 1.1' (entrega).

# Capas de una aplicación Web

**HTML**



**HTML** the Skeleton



**CSS**



**CSS** the Skin



**JS**



**Javascript** the Brain



# Capas de una aplicación Web

- **Capa de estructura de la web.** Indica qué elementos tiene una web y cómo se relacionan semánticamente entre ellos: secciones, cabeceras, pie de página, menú de navegación, listas, tablas... Se escribe usualmente en lenguaje HTML o alguno de sus derivados
- **Capa de presentación de los elementos de la web.** Indica cómo se tienen que mostrar los elementos descritos en la estructura de la web: colores de los textos, fondo de las secciones, fuentes tipográficas, tamaño de los iconos... Se describe en CSS
- **Capa de comportamiento de la web.** Gestiona los cambios de una web producidos por interacciones del usuario o porque llegan nuevas secciones desde el servidor web: qué sucede cuando el usuario hace clic sobre un botón, dónde colocar las noticias que llegan del servidor o cómo enviar un texto que ha introducido el usuario a un servidor de correo electrónico. Se programa en JavaScript y utiliza lenguajes como XML y JSON para comunicarse con el servidor

# Lenguajes y tecnologías de programación en entorno cliente

- Los lenguajes de programación del entorno de cliente son aquellos que se ejecutan en el navegador Web.
  - Lenguajes principales:
    - HTML.
    - DHTML.
    - XML.
    - XHTML.
  - Lenguajes de scripting:
    - JavaScript.
    - VBScript.
  - Otros lenguajes:
    - ActionScript.
    - AJAX.

# Lenguajes y tecnologías de programación en entorno cliente

- Lenguajes básicos del desarrollo front-end:

**HTML**



**Estructura**

**CSS**



**Presentación**

**JavaScript**



**Comportamiento**

- El consorcio W3C (World Wide Web Consortium) determina las especificaciones y estándares CSS y HTML.

# Lenguajes y tecnologías de programación en entorno cliente

- HTML y derivados (I):
  - HTML: ***Hyper Text Markup Language*** (lenguaje de marcado de hipertexto) es el lenguaje de marcas de texto más utilizado en la *World Wide Web*.
  - Se basa en la utilización de un sistema de etiquetas cerrado aplicado a un documento de texto.
  - No necesita ser compilado, sino que es interpretado (ejecutado a medida que se avanza por el documento HTML).
  - Hipervínculo: Enlace de una página Web o un archivo a otra página Web u otro archivo.



# Lenguajes y tecnologías de programación en entorno cliente

- HTML y derivados (II):
  - **XML**: Lenguaje de etiquetado extensible cuyo objetivo principal es describir datos para su transferencia eficiente y no mostrarlos, como es el caso de HTML.
  - **XHTML**: Adaptación de HTML al lenguaje XML.
  - HTML Dinámico (**DHTML**): Integración de HTML con lenguajes de *scripting* (JavaScript), hojas de estilo personalizadas (CSS) y la identificación de los contenidos de una página Web en formato de árbol (DOM).

# Lenguajes y tecnologías de programación en entorno cliente

- **CSS** (*Cascade Style Sheets*): Sirve para separar el formato que se quiere dar a la página Web de la estructura de la página Web y las demás instrucciones.
- **JavaScript**: Lenguaje de programación de *scripting* (interpretado) y, normalmente, embebido en un documento HTML.
- **AJAX** (*Asynchronous JavaScript And XML*): Conjunto de técnicas y métodos de desarrollo Web para la creación aplicaciones Web interactivas y asíncronas.

# Preprocesadores de CSS

- Herramientas que permiten escribir pseudocódigo CSS que luego será compilado y convertirse en CSS nativo.
- Lenguajes de hojas de estilos (extensiones de CSS) que permiten añadir nuevas funcionalidades para hacer el trabajo de diseño y maquetación del sitio Web más sencillo.



# Librerías y Frameworks de CSS

- Estilos ya preparados para usar. Permiten el desarrollo de interfaces Web adaptativas de forma más fácil y rápida.



# Herramientas de edición

- Resulta esencial utilizar un **editor de texto plano** (aquellos que no tienen formato) para la programación web
- Cualquier herramienta básica (como el “Bloc de notas”) sirve para programar en HTML, CSS y JavaScript
- Sin embargo, existen herramientas que facilitan la programación web: “formateando el código”, autocompletando texto, detectando errores...

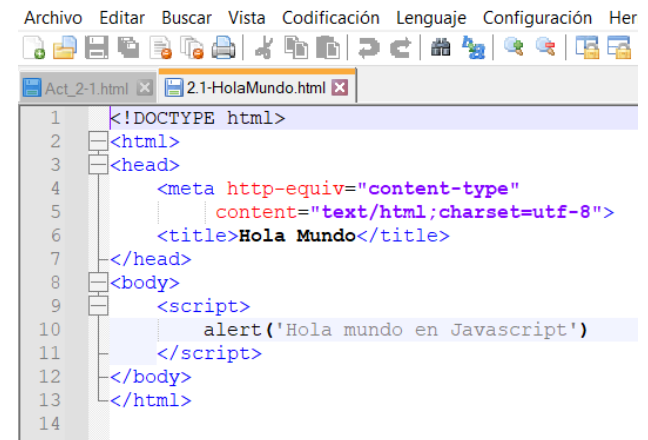
```
<!DOCTYPE html>
<html lang="es-ES">
<head>
<meta charset="UTF-8">
<title> Test </title>
<link rel="icon" href="https://www.universidadvlu.com/media/layout/favicon.PNG">
</head>
<body>
<header>
<h1> Integración de servicios telemáticos </h1>
<h2> Práctica número 1 </h2>
</header>
<nav class="principal">
<ul>
<li><a href="seccion1.html" target="_self">Sección 1</a></li>
<li><a href="seccion2.html" target="_self">Sección 2</a></li>
</ul>
</nav>
<hr>
</body>
</html>
```

```
1 <!DOCTYPE html>
2 <html lang="es-ES">
3 <head>
4   <meta charset="UTF-8">
5   <title> Test </title>
6   <link rel="icon" href="https://www.universidadvlu.com/media/layout/favicon.PNG">
7 </head>
8 <body>
9   <header>
10    <h1> Máster Universitario en Desarrollo de Aplicaciones y Servicios Web </h1>
11    <h2> Herramientas para el desarrollo web </h2>
12  </header>
13  <nav class="principal">
14    <ul>
15      <li><a href="seccion1.html" target="_self">Sección 1</a></li>
16      <li><a href="seccion2.html" target="_self">Sección 2</a></li>
17    </ul>
18  </nav>
19  <hr>
20 </body>
21 </html>
```

# Herramientas de edición

- **Notepad++:** Editor de texto plano y de código fuente libre. Coloreado y envoltura de sintaxis: si se escribe en un lenguaje de programación o marcado, es capaz de resaltar las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.

- <https://notepad-plus-plus.org/downloads/>

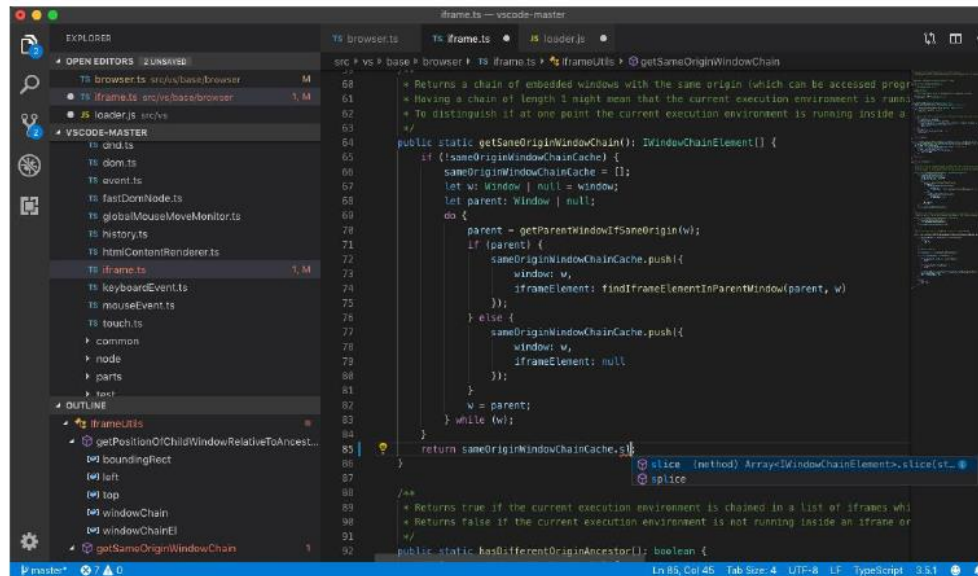


- **UltraEdit:** Requiere licencia.
- <https://www.ultraedit.com/downloads/>

# Herramientas de edición

## Visual Studio Code

- Seguramente, el editor de texto más utilizado para la programación web
- Disponible en: <https://code.visualstudio.com/>



# Herramientas de edición

## Sublime Text

- <https://www.sublimetext.com/>

```
1 <!doctype html>
2 <html>
3 <head>
4   <!--Title-->
5   <title></title>
6
7   <!--Meta-->
8   <meta http-equiv="X-UA-Compatible" content="IE=edge">
9   <meta name="revisit-after" content="30 days">
10  <meta name="distribution" content="web">
11  <meta charset="UTF-8">
12
13  <meta name="description" content="">
14  <meta name="keywords" content="">
15  <meta name="author" content="Twipply">
16
17  <!--Stylesheets-->
18  <link href="/css/styleset.css" type="text/css" rel="stylesheet">
19  <link href="https://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css" rel="stylesheet">
20  <link href="https://netdna.bootstrapcdn.com/bootstrap/2.3.1/css/bootstrap.min.css" rel="stylesheet">
21
22  <!--Icons-->
23  <link rel="shortcut icon" href="">
24
25  <!--JavaScript-->
26  <script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
27  <script src="https://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
28  <script src="https://netdna.bootstrapcdn.com/bootstrap/3.1.0/js/bootstrap.min.js"></script>
29
30 </head>
31 <body>
32   <div class="container">
33
34 </div>
35 </body>
36 </html>
```



## Atom

- <https://atom.io/>

```
Project
├── real-time
│   ├── .git
│   └── lib
│       ├── buffer-binding.js
│       ├── editor-binding.js
│       ├── ghost-portal-binding.js
│       ├── join-portal-dialog.js
│       ├── normalize-url.js
│       └── real-time-package.js
├── node_modules
├── script
├── styles
├── test
├── .gitignore
├── .travis.yml
├── index.js
├── package-lock.json
├── package.json
└── README.md

real-time-package.js
1  const {CompositeDisposable} = require('atom')
2  const {allowUnsafeNewFunction} = require('loophole')
3
4  let Client
5  allowUnsafeNewFunction(() => { Client =
6
7  const BufferBinding = require('./buffer-binding')
8  const EditorBinding = require('./editor-binding')
9
10 module.exports =
11 class RealTimePackage {
12   constructor (options) {
13     cons
14
```



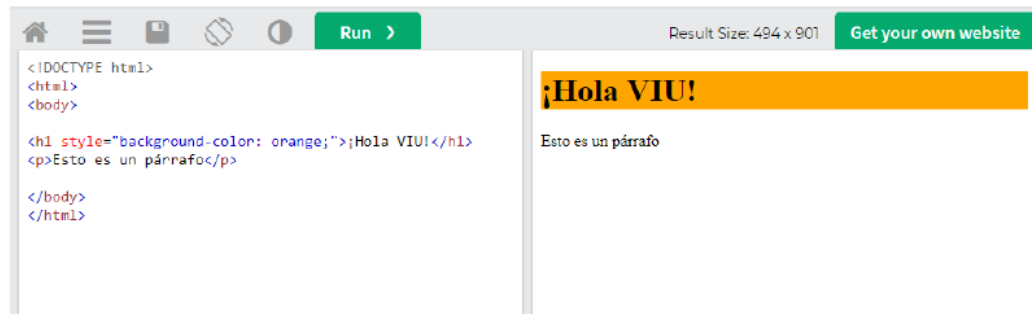


# Herramientas online de edición

- Permite editar código HTML, CSS y JavaScript (entre otros) desde un navegador

## W3Schools

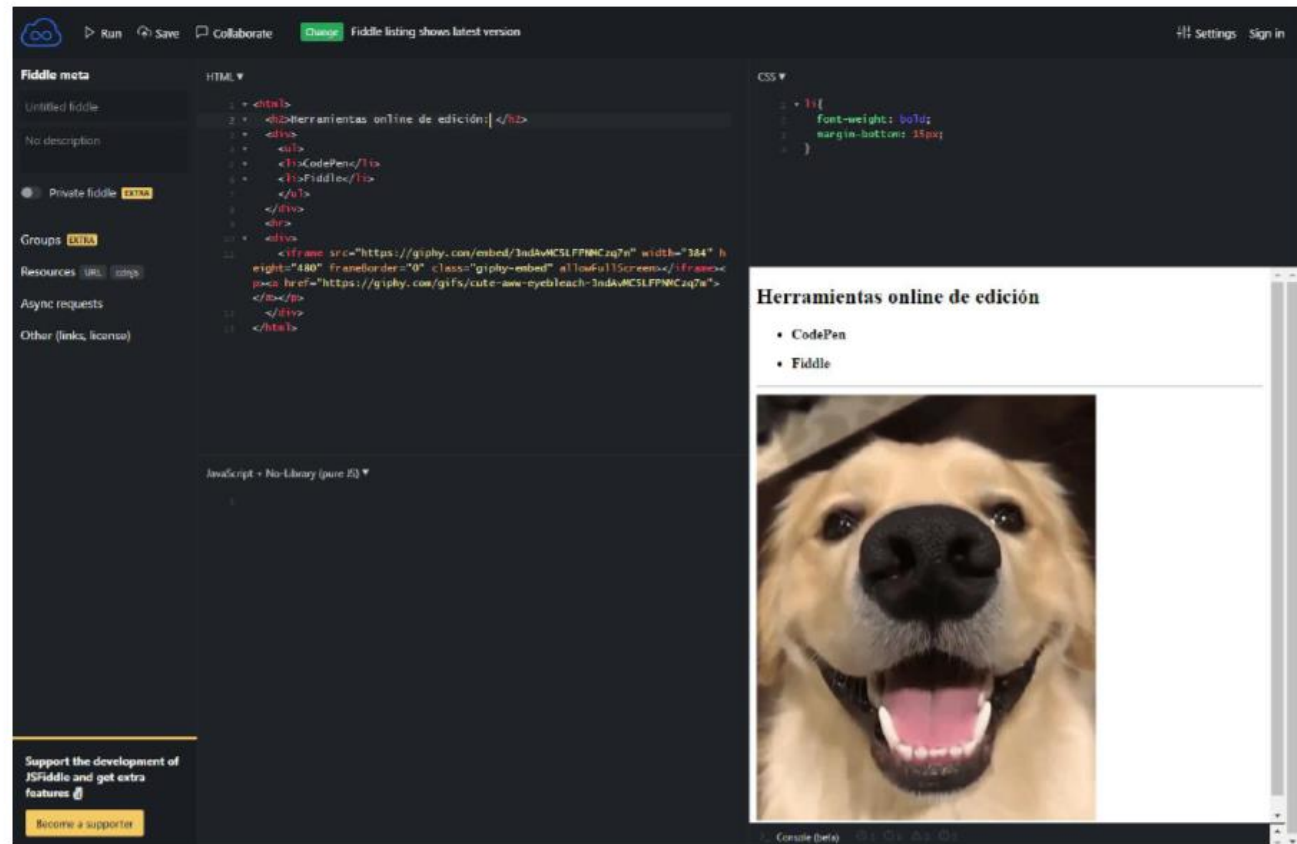
- <https://www.w3schools.com>
- Página de referencia para la programación web
- Ofrece múltiples ejemplos y una aplicación para probar código



# Herramientas online de edición

## Fiddle

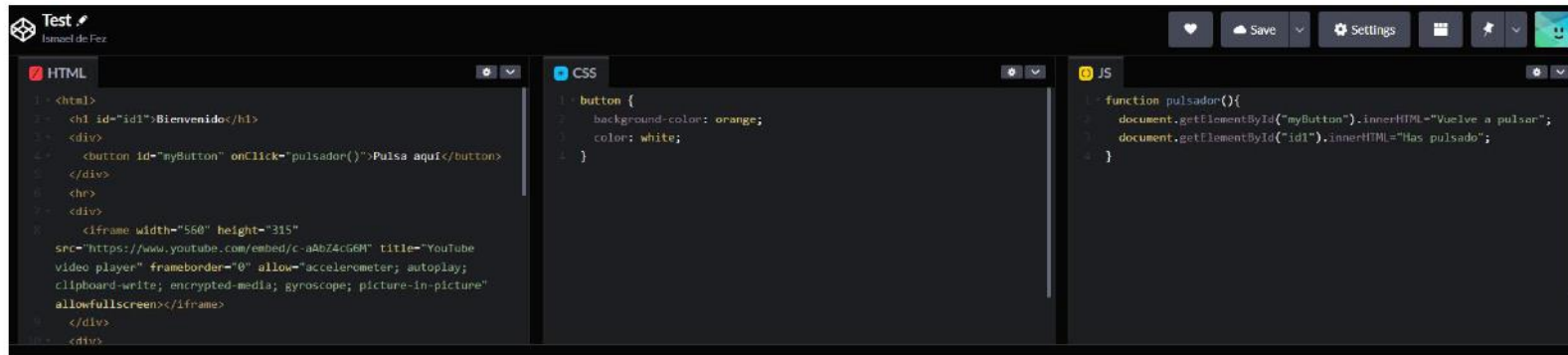
- <https://jsfiddle.net/>



# Herramientas online de edición

## CodePen

- <https://codepen.io/>



# Herramientas de depuración en navegadores

- Es posible ver el código de una página web a través de los diferentes navegadores (e.g. *view-source:https://www.amazon.es/*)
- Sin embargo, gran parte del código está “enmascarado” por la utilización de librerías y de elementos minimizados y ofuscados

```
(function(d,h,N){function H(a){return a&&a.replace&&a.replace(/^\s+|\s$/g,"")}function u(a){return"undefined"===typeof a}function c){return c+b}}),d.ue_sid=b),c&&a.tag("page-source:"+c),d.ue_fpf=w}function P(){var a={};return function(b){b&&(a[b]=1);b=[];for(va function(a){f?(e(a),k(a)):(k(a),e(a));b[c]&&(b[c].isUeh=1)}function S(m,b,c,q){function p(b,c){var d=[b],g=0,f={},k,h;c?(d.push("m a.iel.push(e),e.src=b,a.count&&a.count("postbackImageSize",b.length));if(w){var m=h.encodeURIComponent;m&&b&&(e=new Image,b=""d.ue (d=a.ssw(a.oid),d.e|u(d.val))|(a.isNRBF=1<d.val?0:1)),u(a.isNRBF)|b+=&"&nrbf="+a.isNRBF)),a.isBFT&&!a.isNRBF&&(b+=&"&bft="+a.isBFT if(!("ld"!=m&&"ul"!=m|b&&b!=s))){if("ld"==m){try{h[K]&&h[K].isUeh&&(h[K]=null)}catch(I){}if(h.chrome)for(t=0;t<L.length;t++)T(G,L[t co_:f.connectStart,_co:f.connectEnd,sc_:f.secureConnectionStart,rq_:f.requestStart,rs_:f.responseStart,_rs:f.responseEnd,dl_:f.domL b!=s|ca(b);(c=d.ue_mbl)&&c.cnt&&!l&&(g=c.cnt());l?e("wb",b,2):"ld"==m&&(a.ld=H(s));for(r in a.sc)if(l==e("wb",r))break;if(l){if "ld"!=m|b|a.markers|a.markers||(a.markers={},C(a.markers,e("t",b))),e("t",b,{));a.tag&&a.tag().length&&(g+=&"&csmtags="+a.tag().join("|"), else if(c[W])c[W]("on"+a,b)}function T(a,b,c){c=c|h;if(c[X])c[X](a,b,!1);else if(c[Y])c[Y]("on"+a,b)}function Z(){function a(){d.o {requestId:e,transitionType:"soft"}},p("mark","transitionStart",b));a.tag("ajax-transition");d.ueinit=(d.ueinit||0)+1;var a=d.ue=d. fcp:"firstContentfulPaint",bb:"bodyBegin",be:"bodyEnd",ld:"loaded"},E=h.Date,B=h.performance||h.webkitPerformance,f=(B||{}).timing,
```

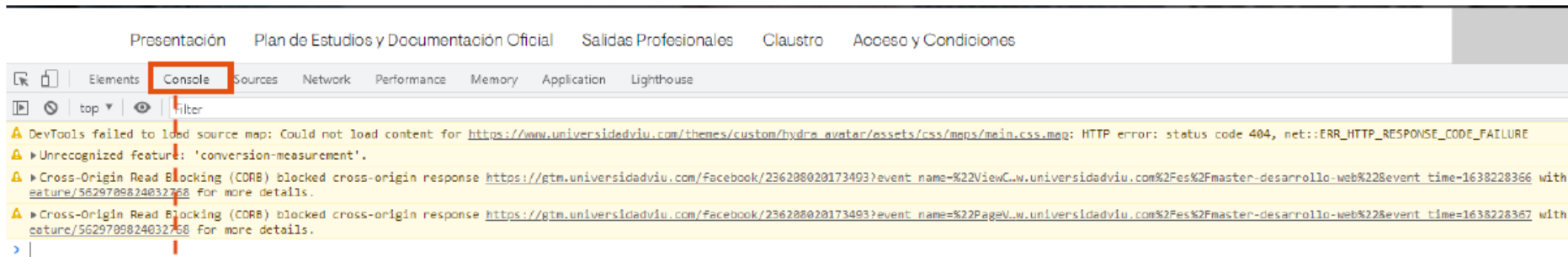
- Los navegadores disponen de “Herramientas para desarrolladores” que ayudan a realizar el desarrollo front-end

# Herramientas de depuración en navegadores

- Las funcionalidades básicas que ofrecen los distintos navegadores son prácticamente las mismas, y difieren en alguna funcionalidad adicional



# Herramientas de depuración en navegadores



- ▶ La opción de consola es muy importante cuando desarrollamos porque podemos ver errores que surgen y los mensajes que mostramos de prueba