

**PRÁCTICA 2 de HLC**  
**HTML5, CSS3 y JavaScript / Frameworks**  
**Visual Studio Code / NodeJS**

<b>Fecha entrega</b>	Lunes, 24 de enero.	<b>Fecha defensa</b>	Viernes, 28 de enero.
<b>Tipo</b>	Parejas	<b>Asunto e-mail</b>	HLC-P02
<b>Formato fichero</b>	Enlace a git		

1.- Desarrolla una aplicación Web utilizando HTML5, CSS3 y JavaScript.

1.1.- Temáticas a explicar:

1.1.1.- JavaScript

1.1.2.- (Voluntario) NodeJS en VSC. El framework Express.

1.1.3.- Framework JS a elegir uno Front-End y otro Back-End:

Front End: React JS, Vue Js, Angular JS, Ember JS ...

Back End: Express JS, Next.js

1.1.4.- Bootstrap: Responsive con HTML, CSS y JS (Ver apartado 5).

1.1.5.- jQuery: JS (Ver apartado 5).

1.2.- Aparecerán los nombre de todos los miembros del grupo en la página inicial y el nombre o nombres de los autores de cada sección o apartado.

2.- Requisitos HTML5.

2.1.- Obligatoria mente se implementará la estructura mediante etiquetas semánticas: header, nav, section, article, aside, footer. No abusar de div y evitar iframe.

2.2.- "Responsividad" al menos hasta un 50%.

2.3.- Formularios: input, type="text", type="radio", type="checkbox", type="hidden"; type="button", textarea, select, option (especifica una de las opciones seleccionables. Si tiene definido el atributo default, será la por defecto nada más cargar el formulario. Si tiene definido el atributo , el valor de éste será el enviado en los datos del formulario aunque se muestre otro texto), button, submit.

2.4.- Control de campos vacíos y de su valor con HTML5 (input pattern) y JavaScript

2.5.- Icono en las pestañas (shortcut icon)

3.- Requisitos CSS3:

3.1.- Se usaran ficheros externos: ficheros CSS referenciados (link rel y href). NO se pueden usar estilos de línea (style), ni embebidos (en la cabecera / head).

3.2.- Aplicar modos de referencia por: Etiquetas, Id, Class, etc.

3.3.- Usar estilos o definir reglas:

- border-radius, opacity margin, padding, Box-shadow y Text-shadow

- font (usar google fonts), text-align, width, background, border, display, list-style.

- Linear-gradient

<http://www.colorzilla.com/gradient-editor/>

Compatibilidad con los diferentes navegadores.

- display: block e inline-block (no en responsive).

- float. Clear

- Transform (scale, rotate, translate).

- Transition y hover.

- Responsive y Mobile First: media queries

@media screen and (max-width: 300px) {

nav { display: none; }

article {

width: 100%;

height: 100px;

}

}

Responsive por tamaño y por densidad de pixel.

4.- JavaScript

Funciones: alert(), console.log()

Uso de objetos (Browser BOM): Window Navigator Screen History Location

Uso de objetos (HTML DOM): Document, Elements, Attributes, Events, Style.

Eventos: click, load, etc.

Manipulación del DOM: getElementById, etc.

```
< a href="#" id="mienlace" style="background-color: orange;">Enlace demo< /a >  
< script >  
document.getElementById("mienlace").style.backgroundColor="orange";  
< /script >
```

Manejadores de eventos: onclick, onMouseOver, onMouseOut, de teclado y de ventana (onload).  
Manejadores de eventos con addEventListener().

Selectores o referencias: getElementByTagName, getElementById, getElementByClassName, querySelector, querySelectorAll().

- Ejemplos básicos: mostrar o ocultar información (texto) y modificar los estilos.
- Ejemplo de validar y manipular un formulario.
- Ejemplos mas complejos o creativos copiados de Internet.

#### 5.- Paginas Web que utilicen frameworks:

- 5.1.- Una con Bootstrap (HTML, CSS y JS / responsive).
- 5.2.- Otra con jQuery (JS).

#### 6.- Varios:

- 6.1.- Control de video para los tipos de ficheros más usados: webm y mp4.
- 6.2.- Tamaño de imágenes según resolución (srcset)

#### 7.- Se valora la estética, la funcionalidad y la creatividad.

- 7.1.- Menú despegable y reponsive (sidenav):  
<http://webgenio.com/2014/05/22/100-increibles-menus-css/>
- 7.2.- Slideshow,  
[https://www.w3schools.com/howto/howto\\_js\\_slideshow.asp](https://www.w3schools.com/howto/howto_js_slideshow.asp)
- 7.3.- Modal images. Responsive Image Grid, etc.
- 7.4.- Cualquier otra animación o efecto.

#### 8.- Los ficheros deben ser validados en "<https://validator.w3.org/>"

#### 8.- Debe usarse repositorios git.

#### Ayuda:

- [https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp) Navegadores y etiquetas.
- <http://caniuse.com> Navegadores y etiquetas reconocibles.
- <https://www.browserstack.com> Testear navegadores.

#### Developer Tools:

- <https://developer.mozilla.org/en-US/docs/Tools>
- <https://developers.google.com/web/tools/chrome-devtools>

#### JavaScript:

##### Openwebinars:

- Curso de JavaScript para principiantes (muy básico).
- Curso de JavaScript intermedio

##### Curso de código facilito:

- <https://codigofacilito.com/cursos/javascript>
- [https://www.youtube.com/watch?v=zJthvhqTUbW&feature=emb\\_rel\\_end](https://www.youtube.com/watch?v=zJthvhqTUbW&feature=emb_rel_end)

##### Codepen: editor online

- <https://codepen.io/pen/>

##### Probando ejemplos de W3school

- [http://www.w3schools.com/js/js\\_htmlDOM.asp](http://www.w3schools.com/js/js_htmlDOM.asp)
- [http://www.w3schools.com/jsref/jsref\\_regexp\\_test.asp](http://www.w3schools.com/jsref/jsref_regexp_test.asp)
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

#### Teoria:

- <https://desarrolloweb.com/articulos/que-es-el-dom.html>
- <http://html.conclase.net/w3c/dom1-es/introduction.html>
- <https://www.w3.org/DOM/DOMTR>