



Universitat
Oberta
de Catalunya

M4.257 - Herramientas HTML y CSS

PEC 2: web y recursos multimedia

Aníbal Santos Gómez

ÍNDICE: DOCUMENTACIÓN.

- 1. Introducción y objetivos.**
- 2. Patrones de diseño.**
- 3. Modificaciones y cambios realizados.**
- 4. Recursos audiovisuales utilizados y tratamiento.**
- 5. Animaciones y efectos visuales.**
- 6. Publicación.**

1. Introducción y objetivos.

A. *Introducción.*

Para el desarrollo de una web moderna, necesitamos herramientas modernas que nos permitan fácilmente el desarrollo de la misma.

El principal objetivo de este desarrollo es centrarnos en la integración de elementos visuales tales como imagen y video así como el básico texto semántico que dan lugar a la vida propiamente dicha de nuestra web.

Para ello se eligen un conglomerado de herramientas que nos facilitarán, el desarrollo en local, el pase a producción, el control de versiones, la maquetación, adaptación a diferentes navegadores incluyendo versiones antiguas, y otra serie de herramientas que se justifican continuación:

- Editor de código: **Visual Studio Code**.

Se utilizará **VS Code** como editor de código, ya que nos permite instalar plugins que nos permiten utilizar snippets para agilizar nuestra forma de programar. Además es totalmente personalizable en función del proyecto que se vaya a realizar, en cuestión de herramientas. Actualmente se han instalado plugins como auto close tag, rename tag, css formatter, highlight matching tag, JS snippets, Vetur, Vue VS Code extension, o Prettier.

- Control de versiones y repositorio: **Git y Github**.

Para el control de versiones utilizaremos **Git** y para almacenar nuestro repositorio en la nube utilizaremos el servicio que nos brinde **Github**.

- Gestor de paquetes: **Npm**.

Como gestor de paquetes de Node.js utilizaremos **Npm**, que contiene un sin fin de dependencias desarrolladas por otros programadores que nos facilitaran la configuración de los diversos paquetes que utilizaremos.

- Module bundler: **Parcel**.

Para empaquetar todo nuestro proyecto utilizaremos **Parcel** con una serie de plugins para poder configurar **Babel** con **Vue**.

- Preprocesadores de código: **Babel y PostCSS**

Utilizaremos **Babel** como polyfill para convertir nuestro código moderno en ES6 a código JavaScript que pueda ser interpretado por navegadores antiguos, y **PostCSS** junto con **Autoprefixer** para convertir nuestro CSS moderno en CSS interpretable por navegadores antiguos.

- Framework frontend: **VueJS**.

Aunque los requisitos de esta práctica no exigen la utilización de un framework front-end. Se elige Vue JS para agilizar y ordenar la estructura de datos que van a ser renderizados y así facilitar el manejo del código HTML y CSS. Además configuraremos el enrutado dinámico para que las estructuras sean homogéneas, el tiempo de carga entre página y página no exista y otro tipo de características que se irán desarrollando a lo largo del resto del semestre.

- Framework UI: **Bulma**.

Bulma como framework UI, nos facilita el maquetar algunas partes del proyecto, por lo tanto tiempo ganado en maquetación responsiva, es tiempo invertido en programación y refactorización. Además utilizaremos **FontAwesome** para utilizar algunos de los iconos vectorizados.

- Publicación: **Netlify**.

Por último para el despliegue se utilizará **Netlify**, que nos permite vincular nuestra cuenta de **Github**, y el repositorio en cuestión, y ejecutar la compilación en nuestro servidor en cuestión de un comando.

B. Objetivos.

- Realizar modificaciones pertinentes respecto al proyecto ya establecido con las herramientas anteriormente mencionadas y utilizadas.
- Elegir un patrón de diseño visual moderno y correspondiente al mercado actual.
- Crear una aplicación web responsive, visible en todos los dispositivos y navegadores.
- Mejorar en lo posible maquetación y recursos gráficos que hagan más atractiva la web al usuario, es decir, mejorar la experiencia de usuario.

2. Patrones de diseño.

Los patrones de diseño web adaptables evolucionan rápidamente, pero existen varios patrones establecidos que funcionan bien en los diferentes equipos de escritorio y dispositivos móviles.

El responsive design es una configuración en la que el servidor siempre envía el mismo código HTML a todos los dispositivos y se usa CSS para modificar el procesamiento de la página en el dispositivo.

Es una técnica de diseño y desarrollo web que mediante el uso de estructuras flexibles (contenedores flexibles, imágenes y video flexibles) y junto con Media Queries especificados en CSS, logran adaptar un sitio web al entorno del dispositivo en el que se encuentre. Con ésta técnica se consigue que el contenido del diseño de un sitio se vea bien y pueda ser legible para los usuarios sobre el dispositivo con el que éste interactúe.

2.1. Características más importantes del diseño responsive

- **Adapta la web al ancho del dispositivo.** El diseño normalmente pasa de ser un diseño horizontal a un diseño vertical, adaptándose al ancho del dispositivo. El usuario no tiene que desplazarse horizontalmente.
- **Reorganiza los distintos elementos de la web.** Los contenidos de la web se disponen de tal forma que puedan verse correctamente en todas las pantallas. Se utilizan botones de mayor tamaño, para facilitar la interacción táctil, se reajustan el tamaño y la separación de la letra, para que sea más legible y se usan imágenes adaptables para cada dispositivo.
- **Simplifica la web.** Se priorizan los elementos, solo se utilizan los necesarios eliminando objetos gráficos estéticos y que sólo tienen sentido en la versión de escritorio. También se eliminarán los elementos que tardan mucho en cargarse.
- **Cambio de apariencia de algunos elementos.** Como por ejemplo el menú de navegación, que pasa a ser desplegable, lo que ahorra espacio de pantalla.

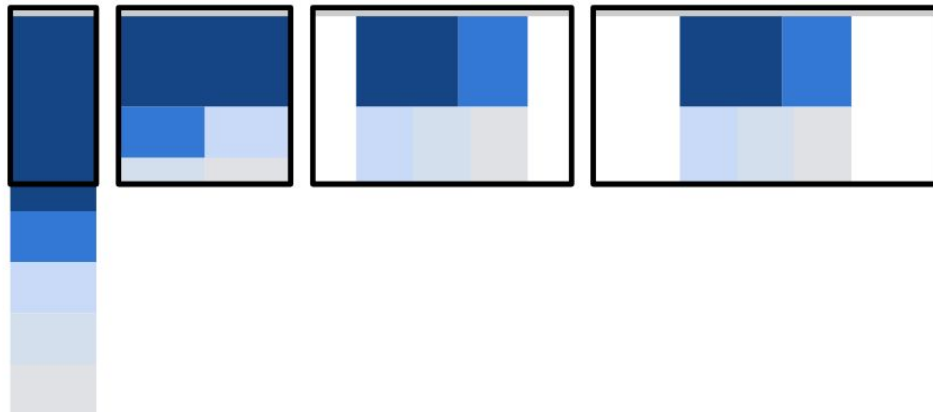
- **Mobile first.** Es una forma de mejora del diseño responsive progresivo, que se enfoca en la priorización del diseño y el desarrollo para dispositivos móviles por encima del diseño y desarrollo para pantallas de escritorio.

Básicamente este tipo de diseño prioriza la construcción del diseño primero en dispositivos pequeños para abordar después los intermedios o más grandes. Es decir, primero dispositivos móviles y después los de escritorio.

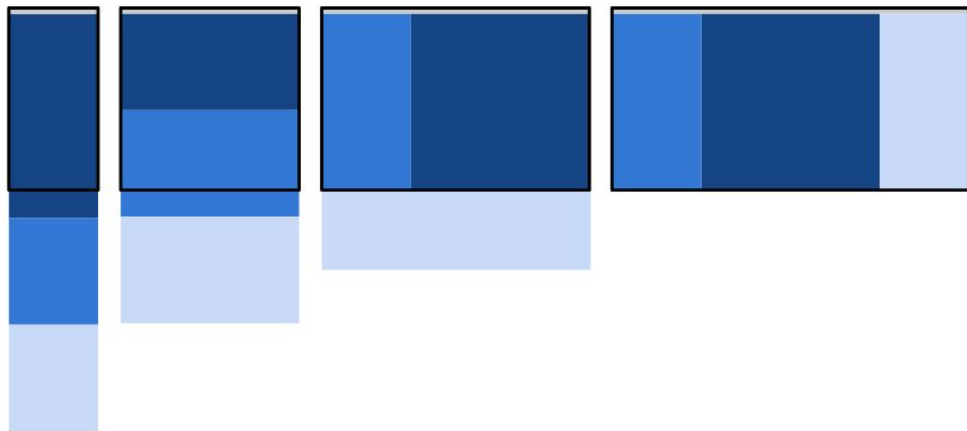
2.2.Técnicas o patrones de diseño responsive más conocidas:

- **Mostly fluid** consiste en una cuadrícula fluida. En las pantallas grandes o medianas se mantiene el mismo tamaño y simplemente se ajustan los márgenes en las más anchas.

En las pantallas más pequeñas, la cuadrícula fluida genera el reprocesamiento del contenido principal, mientras que las columnas se apilan verticalmente.



- **Drop Column**, en el caso de los diseños con varias columnas de ancho completo, durante el proceso de colocación de columnas éstas únicamente se colocan de forma vertical debido a que el ancho de la ventana es demasiado reducido para el contenido. En un momento dado, todas las columnas se apilan verticalmente.



- **Layout shifter** es el más adaptable, ya que posee varios puntos de interrupción en diferentes anchos de pantalla.

Debido a las diferencias significativas entre cada punto de interrupción principal.

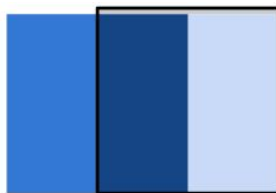


- **Tiny tweaks** permite realizar pequeños cambios en el diseño, como ajustar el tamaño de la fuente, cambiar el tamaño de las imágenes o desplazar el contenido de maneras muy poco significativas.

Funciona correctamente en diseños con una sola columna, como los sitios web lineales de una sola página y los artículos con mucho texto.



- **Off canvas** coloca contenido menos usado (tal vez menús de navegación o de apps) fuera de la pantalla y solo lo muestra cuando el tamaño de la pantalla es suficientemente grande. En las pantallas más pequeñas, el acceso al contenido es posible con solo a un click.



En nuestro caso, partiendo de un modelo de construcción mobile first, hemos realizado un diseño híbrido de entre los anteriormente comentados.

Por una parte utilizamos el patronaje de **drop column**, que coincide en la parte de mobile y desktop en una estructura genérica.

Para colocar ciertos elementos que exceden visualmente este patrón, utilizamos **mostly fluid**, para manejar el contenido y moldearlo a las necesidades responsive de dispositivos intermedios.

Esta reflexión se lleva a cabo debido a que estamos ante una página de documentación, en la cual crearemos una portada llamativa, y páginas categóricas que enmarcan nuestro contenido de una manera más uniforme, facilitando la lectura del mismo.

2.3. Material Design.

Siguiendo esta línea de diseño y patronaje, nos hemos guiado por la línea de diseño que nos facilita Material Design de Google.

Material Design es un lenguaje visual que sintetiza los principios clásicos del buen diseño con la innovación de la tecnología y la ciencia. Los **objetivos** de Material son:

- **Crear.**

Crear un lenguaje visual que sintetice los principios clásicos del buen diseño con la innovación y posibilidad de la tecnología y la ciencia.

- **Unificar.**

Desarrollar un único sistema subyacente que unifique la experiencia del usuario en todas las plataformas, dispositivos y métodos de entrada.

- **Customizar.**

Ampliar el lenguaje visual de Material y proporcionar una base flexible para la innovación y la expresión de la marca.

Para llegar a este objetivo, hemos planteado de forma genérica una paleta de colores que nos da uniformidad en el diseño y tipografías concretas, así como tamaños específicos para cada tipo de contenido, con el fin de seguir un estándar que podrá saltarse siempre y cuando haya una justificación notoria.

- Paleta de colores:

1. Color primario: #654ea3
2. Color primario claro: #957bd5
3. Color primario oscuro: #362574
4. Color secundario: #ffffff
5. Color primario texto: #ffffff
6. Color secundario texto: #362574

- Tipografías utilizada:

- a. TtCommons-Regular , sin serifa.
- b. Source Sans Pro, sin serifa.

3. Modificaciones y cambios realizados.

3.1. Cabecera y barra de navegación.

En la cabecera y barra de navegación hemos añadido el título descriptivo de la web “The roadmap”, así como un icono en formato .svg, representativo del sitio. El logotipo será utilizado además en otros puntos de la página para crear imagen de marca.

Por otro lado separando el título y el logotipo en primera instancia, porque empezamos a leer de izquierda a derecha, hemos situado a la derecha final de la barra de navegación cada una de las rutas o enlaces a cada página de categoría; que cuando accedemos desde un dispositivo móvil o con un ancho por pantalla inferior a 768 píxeles, será un icono de hamburguesa que nos colapsará las rutas en un desplegable para favorecer la visibilidad y ordenación del contenido.

Por último resaltar que hemos fijado la barra de navegación y nos acompañará durante nuestra lectura para mejorar la experiencia y poder cambiar de ruta sin tener que subir al principio de la página.

 **The frontend roadmap**

BASICS TOOLING FRAMEWORKS INFLUENCE

 **The frontend roadmap**





3.2. Pié de página.

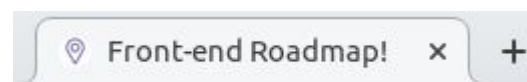
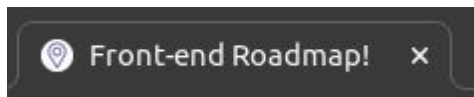
El pié de página se compone ahora de dos columnas donde hemos colocado el título, unos enlaces a redes sociales de contacto del autor, en la primera columna; y en la segunda un clip path del logo que se encaja en la paleta de colores que compone nuestra página.

Aparece además un botón flotante que se posiciona por encima del footer para llevarnos a la parte superior sin tener que realizar ningún esfuerzo de scroll.

3.3. Favicon.

Un favicon es un pequeño icono, que sirve como marca para su sitio web. Su principal propósito es ayudar a los visitantes a localizar su página más fácilmente cuando tienen varias pestañas abiertas. No deben ser confundidos con logos, pero a veces son lo mismo.

Hemos introducido nuestro logo en este caso como favicon, ya que es sencillo y fácil de recordar. Para ello hemos generado una serie de imágenes en .png, ico y las hemos registrado en un xml y un json para el manifiesto web, que posteriormente hemos insertado en la cabecera. Así tendremos un favicon no solo en un navegador de escritorio, también móvil android o ios y para navegadores dark theme y light theme.



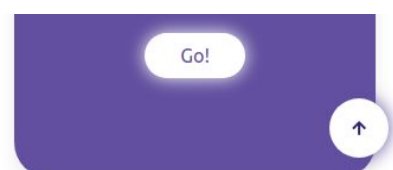
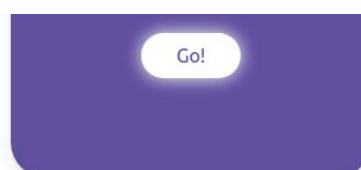
3.4. Nueva sección.

Hemos creado una nueva sección titulada Influences para darle algo más de contenido a nuestro sitio. En esta sección hemos añadido datos biográficos, así como fotografías de los personajes que en ella están contenidos.

Posteriormente se mostrarán las capturas de pantalla correspondientes, a la hora de ejemplificar la dirección de arte del sitio y el porqué de la misma.

3.5. Botón al principio.

Por último y para mejorar la interactividad del sitio hemos creado un botón que vuelve al principio de cada página, flotante y que solo aparecerá cuando hayamos avanzado en la navegación de la página.



4. Recursos audiovisuales utilizados y tratamiento.

4.1. Imágenes y su formato.

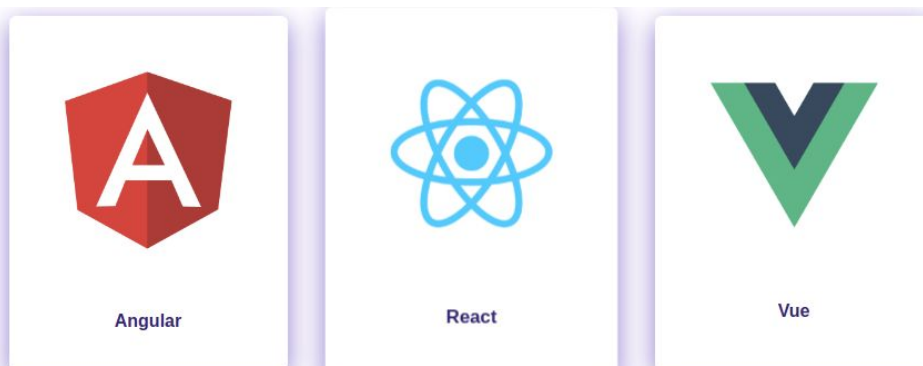
Seleccionar las imágenes de nuestro sitio nos ayudará darle vida a la página, otorgando un significado característico a cada una de las secciones, ya sea de portada, sección, detalle o biografía.

Los formatos elegidos han sido:

- Svg: este formato es el elegido para los logotipos y para las imágenes de portada que requieren una redimensión y adaptación correcta a diversos dispositivos así como su modificación de colores.



- Jpg: este formato es seleccionado para las fotografías de la galería fotográfica. Son imágenes con gran resolución y con un peso ligero.



- Png: es el elegido para los logotipos genéricos de cada sección y detalle, ya que es más uniforme a la hora de encontrar los recursos gráficos y no necesitaremos una resolución demasiado amplia y tampoco un fondo en concreto.



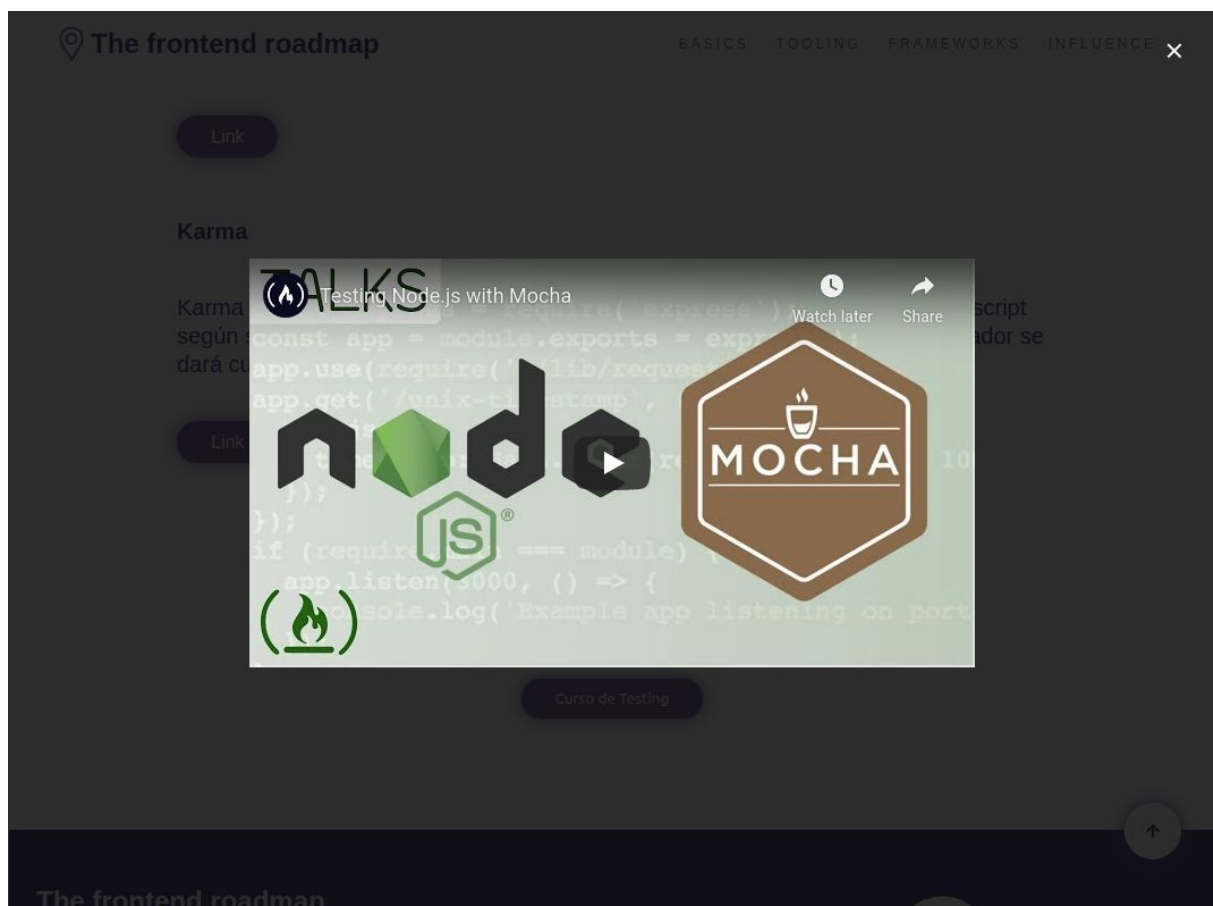
4.2. Video.

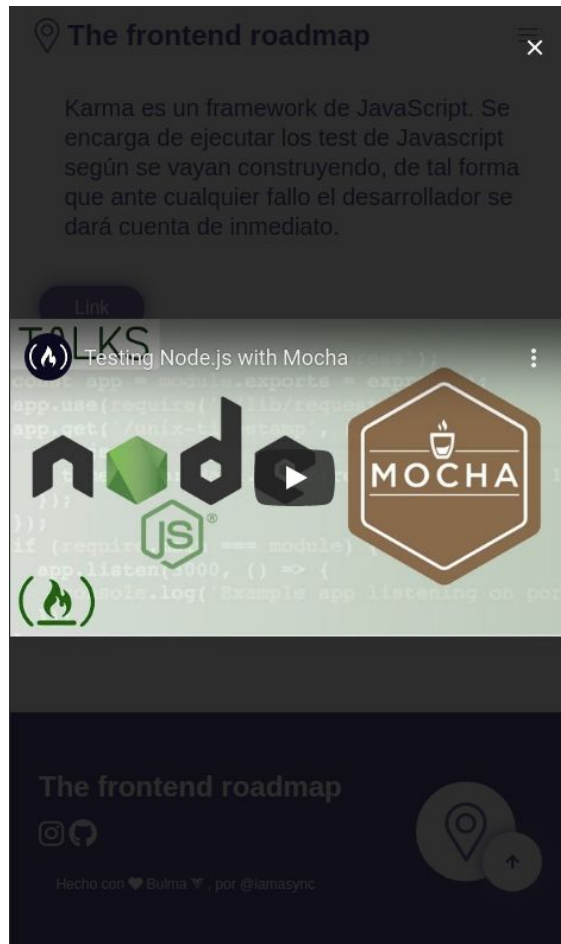
Hemos decidido incluir contenido de video dentro de nuestra página de documentación, en este caso hemos seleccionado el formato de etiqueta de iframe, para no tener que cargar directamente en nuestra aplicación los recursos videográficos y así evitar un tiempo de carga muy excesivo. Por lo que el material contenido se encuentra alojado en YouTube.

El video aporta en este caso un plus de “entretenimiento” o navegación añadida, y hace que el usuario permanezca más tiempo en nuestro sitio.

Por otra parte el video se encuentra localizado dentro de un modal, que se activará una vez hagamos click sobre el botón correspondiente. Hemos tratado el video de una manera responsive y se adapta a todos los dispositivos manteniendo el aspecto por ratio.

Curso de Testing





4.3. Adaptación de los recursos visuales según su utilidad, dirección de arte.

A continuación se justifican las disposiciones de los elementos o recursos visuales que se utilizan en la aplicación.

En primer lugar la **página de portada** deberá contener elementos de imagen adaptativos por columnas en resolución desktop. Pasando a ser una única columna total en dispositivos más pequeños, ya sean móviles o tablets.



Crea proyectos y aplicaciones modernas

Cada año aparecen herramientas que nos facilitan el desarrollo de aplicaciones web, adaptadas a las últimas tendencias y respondiendo a la demanda creciente del usuario medio. ¿Serás el último en utilizarlas?

No te pierdas en este mapa extenso

Aprender herramientas tecnológicas puede ser abrumador. Sobre todo en el amplio abanico del desarrollo web. Aquí verás una selección para que no te sientas perdido y siempre tengas ganas de experimentar y ampliar tus conocimientos



adaptadas a las últimas tendencias y respondiendo a la demanda creciente del usuario medio. ¿Serás el último en utilizarlas?



No te pierdas en este mapa extenso

Aprender herramientas tecnológicas puede ser abrumador. Sobre todo en el amplio abanico del desarrollo web. Aquí verás una selección para que no te sientas perdido y siempre tengas ganas de experimentar y ampliar tus conocimientos

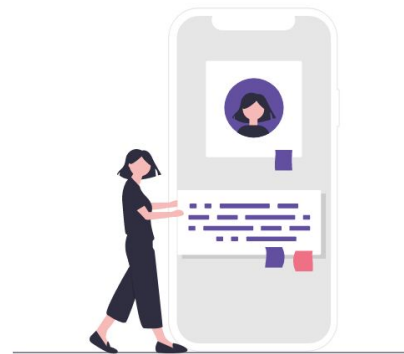


La imagen inicial de la cabecera en desktop se mantiene mientras que en dispositivos móviles desaparece para dar importancia al texto introductorio y no restarle importancia, además de que no encaja en ninguna composición visualmente estética dentro del patrón de móvil.

Un pequeño cohete

Conoce las principales herramientas para desarrollar aplicaciones web a la velocidad de la luz.

Empezar



Un pequeño cohete

Conoce las principales herramientas para desarrollar aplicaciones web a la velocidad de la luz.

Empezar

Un pequeño cohete

Conoce las principales herramientas para desarrollar aplicaciones web a la velocidad de la luz.

Empezar

En cada **página de categoría** genérica hemos añadido imágenes referentes a cada página de detalle dentro de las tarjetas, en este caso son archivos png sin fondo, en resolución desktop, aparecerán apaisadas las opciones, es decir en horizontal. Pero según el ancho de otras pantallas de dispositivos inferiores se irán agrupando por columnas o filas, según el caso.

¿Qué es CSS?

Es el lenguaje utilizado para describir la presentación de documentos HTML

¿Qué es Javascript?

Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

¿Qué es Git?

Es un software de control de versiones diseñado por Linus Torvalds



HTML



CSS



Javascript



Git



HTML



CSS



Javascript



Git



HTML



CSS

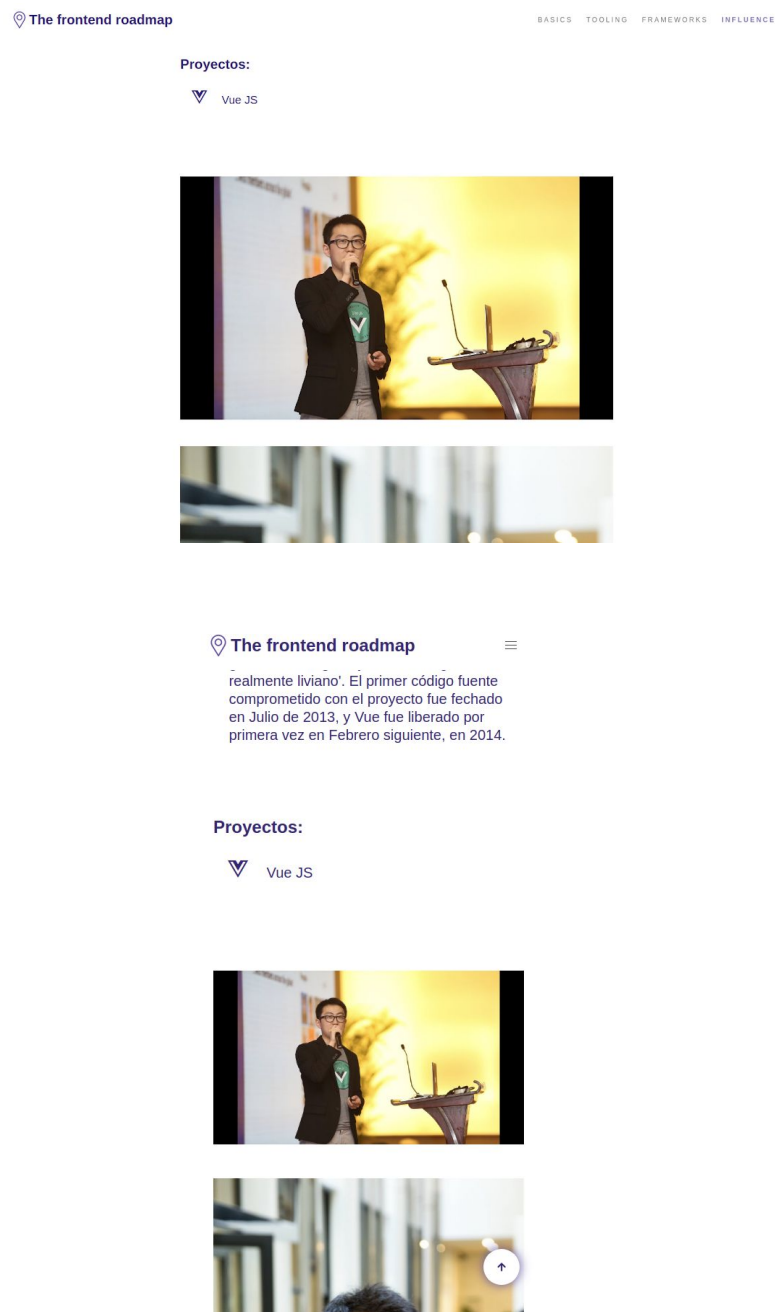


Javascript

Hemos creado además una **página de Influence**, para explicar datos biográficos y añadir una pequeña galería de imágenes por cada autor. En primer lugar deberemos mantener una imagen ilustrativa o avatar (hecha con clip path), donde figura el autor del que se hablará. Posteriormente al acabar la biografía mostraremos una serie de fotografías del mismo.

Se ha mantenido un diseño en bloque donde las fotografías se adaptan al ancho de la pantalla con un máximo en desktop, mientras que en móvil ocuparán el 100% del ancho de la pantalla.

Esta solución se adopta ya que vamos a utilizar imágenes representativas de cada autor donde es preferible mantener la atención una por una, a modo de flujo de contenido.



5. Animaciones y efectos visuales.

A continuación se mencionan las animaciones y efectos visuales utilizados para hacer la página más atractiva para el usuario. Estos recursos han sido creados para facilitar la navegación entre páginas, y la mejora en la usabilidad de las mismas.

5.1. Navegación entre rutas.

Para la navegación entre rutas o diferentes páginas hemos utilizado unas transiciones con un efecto fade-out donde modificamos la opacidad y un tiempo de carga de 0.35 segundos. Estas son las clases utilizadas:

```
/*  
Transitions  
*/  
  
.fade-enter,  
.fade-leave-to {  
  opacity: 0;  
  transform: translate(2em);  
}  
  
.fade-enter-active,  
.fade-leave-active {  
  transition: all 0.35s ease;  
}
```

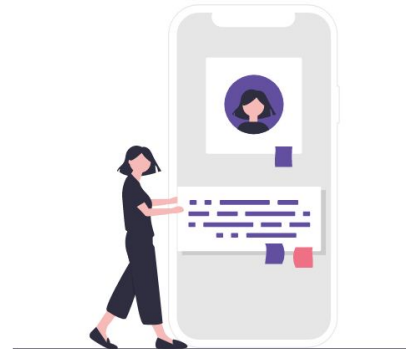
5.2. Scroll desplazativo.

Para favorecer la navegación hemos utilizado un scroll desplazativo en la página de portada situado al hacer click en el botón de “Empezar”, que nos llevará a las tarjetas que posteriormente nos introducen en cada página de categoría.

Un pequeño cohete

Conoce las principales herramientas para desarrollar aplicaciones web a la velocidad de la luz.

Empezar



Basics

Aprende HTML, CSS, Javascript y Git, para organizar tu contenido web, servirlo, maquetarlo, programarlo. Haz que cada página sea única con unas líneas de código.

Go!



Tooling

Empaqueta tus aplicaciones web. Convierte todos tus proyectos en un cohete que llegue rápidamente a todo el mundo y con la mitad de esfuerzo.

Go!



Frameworks

Elimina todo ese esfuerzo y líneas que no te hacen falta y te impiden organizar tus proyectos con claridad. Aprende Angular, React o Vue.

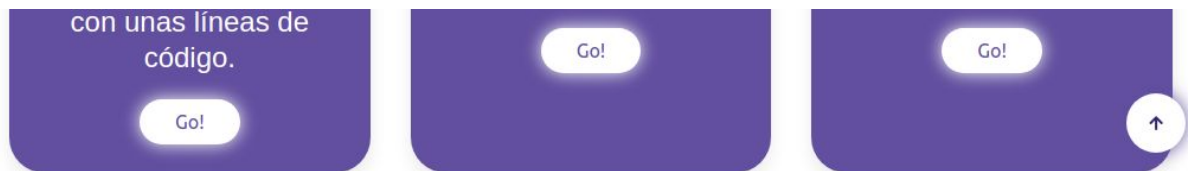
Go!



Este efecto lo hemos creado mediante javascript a través de scroll behavior “smooth” con la siguiente función:

```
scrollToCards() {  
  const element = document.getElementById("card-id");  
  element.scrollIntoView({  
    behavior: "smooth"  
  });  
};
```

Además hemos creado un botón para volver al principio de la página que aparecerá únicamente cuando ya hayamos hecho scroll en la página:



Para realizar este botón, hemos creado un componente específico en vue donde manipulamos el scroll behavior con las siguientes funciones:

```
handleScroll: function() {  
  if (this.scrollTimer) return;  
  this.scrollTimer = setTimeout(() => {  
    this.scrollToY = window.scrollToY;  
    clearTimeout(this.scrollTimer);  
    this.scrollTimer = 0;  
  }, 100);  
},  
goTop() {  
  window.scrollTo({  
    top: 0,  
    behavior: "smooth"  
  });  
}
```

Donde en handleScroll controlamos el tiempo de scroll y la coordenada de scroll donde aparecerá el botón, y con la función goTop seteamos la posición inicial donde queremos que vaya.

5.3. Botones.

Por último hemos creado unos efectos de animación al hacer hover en cada botón genérico de la página, además les hemos añadido una sombra acorde. Cuando hacemos hover existe una aclaración del botón:

Empezar

Empezar

Para crear este efecto hemos utilizado las siguientes clases:

```
.btn {  
  margin-top: 20px;  
  border-radius: 50px;  
  border: none;  
  padding: 10px 30px;  
  color: #ffffff;  
  box-shadow: 0px 0px 15px 0px rgba(101, 78, 163);  
  transition: opacity 1s ease;  
  opacity: 1;  
  cursor: pointer;  
  border: none;  
  outline: none;  
}
```

```
.btn:hover {  
  opacity: 0.8;  
}
```

Además hemos incluido una animación de zoom en las tarjetas de cada página de categoría que nos llevan a la parte de detalle:



HTML



CSS



Javascript



Git



Las clases utilizadas para generar esta animación son las siguientes:

```
.block {  
  border-radius: 6px;  
  display: block;  
  padding: 50px;  
  margin: 20px;  
  width: 251px;  
  height: 320px;  
  transition: transform 0.2s;  
}
```

```
.block:hover {  
  transform: scale(  
    1.05  
  );  
}
```

6. Publicación.

Finalmente, después de realizar todo el proceso de desarrollo y testeo correspondiente, realizamos el push para publicar en master nuestros últimos cambios. Estos serán compilados y publicados en Netlify en la siguiente dirección:

<https://anibalsantos.netlify.com>

El repositorio público se encuentra alojado en la siguiente dirección de Github:

<https://github.com/ansango/PEC-1-Desarrollo-de-una-web>