

Informe del Proyecto: Portfolio

1. Introducción

Este informe detalla el diseño, la estructura y la implementación de una web personal que incluye dos secciones principales: un portfolio y un formulario de registro. El proyecto utiliza tecnologías web modernas como HTML5, CSS (mediante Grid y Flexbox) y JavaScript (ES6). Además, se ha llevado un control de versiones con Git para gestionar el desarrollo y las actualizaciones del proyecto.

2. Descripción Detallada del Diseño y la Estructura de la Web

La web se ha estructurado siguiendo las buenas prácticas de HTML5, dividiendo el contenido en áreas semánticas bien definidas:

Header: Contiene el título del portfolio, un menú de navegación interactivo y responsivo con efectos hover que resaltan cada opción mediante transiciones.

Sección Principal (Section y Main):

Sección inicial: Presenta un artículo con una imagen destacada y un mapa incrustado para indicar la ubicación (por ejemplo, un pueblo de Almería).

Contenido principal: Se organiza en varias secciones, como “Sobre mí”, “Estudios”, “Experiencia Laboral”, “Habilidades e idiomas” y “Hobbies”. Cada sección muestra información relevante mediante tablas, textos y elementos multimedia.

Sidebar: Una barra lateral que ofrece información de contacto o contenido complementario.

Footer: Contiene enlaces a redes sociales y se ha diseñado para mantener la coherencia visual con la paleta de colores seleccionada.

El diseño es responsivo, utilizando **CSS Grid** para la distribución global de la página y **Flexbox** para la alineación y distribución de elementos en el menú y otras secciones, lo que garantiza una experiencia óptima en dispositivos de escritorio, tabletas y móviles.

3. Tecnologías Utilizadas

HTML5

Estructura Semántica: Se han utilizado elementos semánticos como `<header>`, `<section>`, `<article>`, `<main>`, `<aside>` y `<footer>` para organizar el contenido de manera clara y accesible.

Formularios y Multimedia: Implementación de formularios de registro y elementos multimedia (imágenes, iframes) para enriquecer la interacción del usuario.

CSS (Grid y Flexbox)

- **Grid Layout:** Se emplea para estructurar la página en áreas (cabecera, sección, contenido principal y pie de página), permitiendo un diseño modular y adaptable.
- **Flexbox:** Se utiliza en el menú de navegación y en listas para asegurar una distribución homogénea y centrada de los elementos, facilitando la adaptación a diferentes tamaños de pantalla.
- **Transiciones y Animaciones:** Se han implementado animaciones suaves (por ejemplo, la clase `.fade-in` para el carrusel) que mejoran la experiencia de usuario sin sobrecargar el diseño.

JavaScript (ES6)

Programación Orientada a Objetos: Se utiliza la sintaxis de ES6 para declarar clases y métodos, lo que facilita la modularidad y el mantenimiento del código.

Ejemplo: La clase Carousel gestiona el carrusel de imágenes mediante métodos como `updateImage()`, `nextImage()`, `prevImage()` e `initEvents()`, permitiendo un manejo dinámico y atractivo de las imágenes.

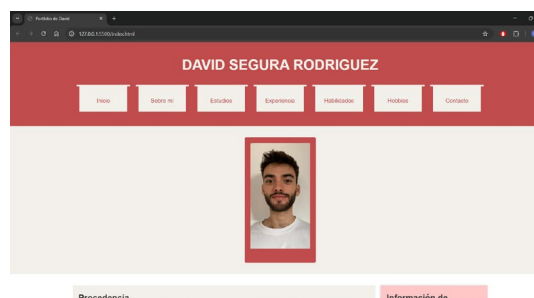
Eventos y DOM: Se gestionan eventos (`click`, `mouseover`, `mouseout`) para implementar interacciones en tiempo real, como la navegación por el carrusel y efectos visuales.

Git

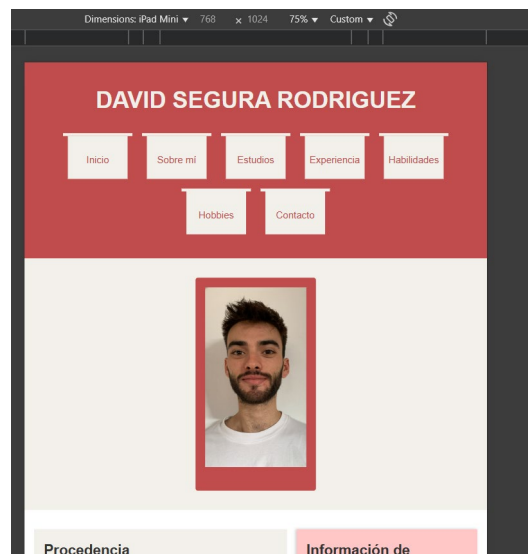
Control de Versiones: Git se ha empleado para llevar un registro detallado de los cambios, colaborar en equipo y mantener una línea de desarrollo clara. Esto facilita la implementación de nuevas características, la resolución de conflictos y la revisión del historial de cambios.

4. Capturas de Pantalla de la Web en Diferentes Dispositivos

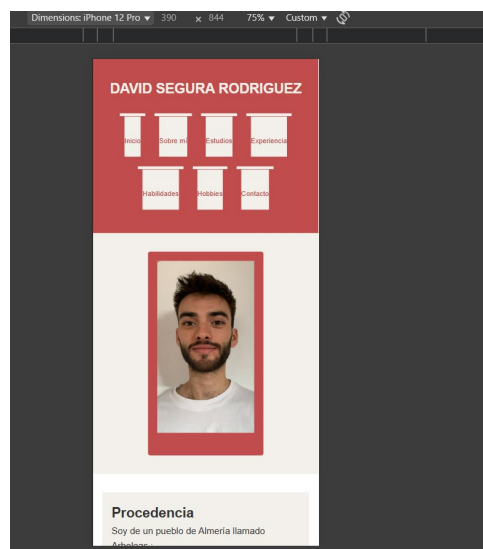
Escritorio:



Tablet:



Móvil:



5. Implementación de JavaScript: Declaración de Objetos y Métodos

La funcionalidad interactiva del carrusel se implementa mediante una clase en JavaScript:

Clase Carousel:

➤ Propiedades:

- images: Arreglo con las rutas de las imágenes.
- currentIndex: Índice que determina qué imagen se muestra.

- `carouselImg`: Referencia al elemento `` donde se visualizan las imágenes.

➤ **Métodos:**

- `updateImage()`: Actualiza la imagen mostrada y reinicia la animación.
- `nextImage()`: Avanza al siguiente índice del arreglo, utilizando la modularidad para reciclar el contenido.
- `prevImage()`: Retrocede al índice anterior.
- `initEvents()`: Configura los eventos para la navegación del carrusel (click en botones y efectos de zoom al pasar el mouse).

ES6 y Modernidad: La implementación utiliza funciones flecha para mantener el contexto del objeto y facilita la comprensión y el mantenimiento del código.

6. Lista de Mejoras o Futuras Implementaciones

- **Optimización de Imágenes:** Implementar carga diferida (lazy loading) para mejorar el rendimiento en dispositivos móviles.
- **Validación Avanzada de Formularios:** Incorporar validaciones en tiempo real y mensajes de error claros para mejorar la experiencia en el formulario de registro.
- **Accesibilidad:** Mejorar atributos ARIA y optimizar el contraste de colores para usuarios con discapacidades.
- **Animaciones Adicionales:** Explorar nuevas transiciones y microinteracciones para enriquecer la experiencia del usuario.
- **Integración con API:** Conectar el carrusel o secciones de la web con servicios externos (por ejemplo, redes sociales o bases de datos) para contenido dinámico.
- **Optimización SEO:** Revisar y optimizar metaetiquetas, descripciones y estructura semántica para mejorar la visibilidad en buscadores.
- **Soporte Multilenguaje:** Añadir opciones de traducción para alcanzar un público más amplio.

7. Uso de la Paleta de Colores y su Propósito en la Experiencia de Usuario

La selección de la paleta de colores juega un papel fundamental en la identidad y usabilidad del sitio:

Color #c04c4c:

Se utiliza en el fondo del header, botones y bordes. Este color evoca sensaciones de energía y pasión, ayudando a atraer la atención hacia elementos importantes.

Color #f2f0ea:

Predomina en el fondo de secciones y formularios, proporcionando un contraste claro y facilitando la lectura.

Color #020103:

Utilizado en los bordes y algunos textos, crea un contraste fuerte con el fondo claro, mejorando la legibilidad y la estructura visual.

Impacto en la Experiencia: La combinación de estos colores asegura una interfaz equilibrada, donde la información es fácilmente legible y los elementos interactivos se destacan adecuadamente, promoviendo una experiencia de usuario agradable y coherente.

8. Conclusiones

El proyecto presenta una implementación moderna y bien estructurada de un portfolio y un formulario de registro. Gracias al uso de HTML5, CSS Grid y Flexbox, y JavaScript con ES6, se ha logrado una web responsiva, interactiva y visualmente atractiva. La integración de Git facilita el control y la evolución del proyecto. Además, la paleta de colores seleccionada contribuye a una experiencia de usuario coherente y agradable, aunque siempre existen áreas de mejora que se pueden abordar en futuras versiones para optimizar la accesibilidad, el rendimiento y la integración con servicios externos.