1º ASIR

[Adrián Requena González](mailto:areqgon1008@g.educaand.es)

**PORTFOLIO PERSONAL**

línea horizontal

# 

# 

# 

# 

# 

# 

## 1. Introducción

El Portfolio Web Personal es un proyecto final diseñado para mostrar el currículum profesional y las habilidades del desarrollador de forma atractiva, funcional y moderna. Este proyecto integra conceptos fundamentales de desarrollo web, combinando tecnologías de front-end como HTML5, CSS3 y JavaScript (ES6) para lograr un sitio dinámico, interactivo y responsive. Además, se ha puesto especial énfasis en la calidad del código, la organización mediante Git y la optimización para diferentes dispositivos y navegadores.

## 2. Objetivos del Proyecto

* **Mostrar información profesional**: Presentar datos personales, habilidades y experiencia laboral de manera clara y visual.
* **Interactividad y dinamismo**: Implementar funcionalidades dinámicas con JavaScript, como el carrusel de imágenes y efectos interactivos en la navegación.
* **Diseño responsive**: Garantizar que la web se adapte a diferentes resoluciones de pantalla, desde dispositivos móviles hasta escritorios.
* **Calidad de código y mantenibilidad**: Utilizar buenas prácticas de desarrollo, control de versiones y organización del repositorio.
* **Accesibilidad y usabilidad**: Asegurar una experiencia óptima para todos los usuarios, incluyendo aquellos con necesidades especiales.

## 3. Tecnologías Utilizadas

* **HTML5**: Proporciona la estructura semántica del sitio, permitiendo definir secciones como encabezados, navegación, contenido principal y pie de página.
* **CSS3**: Se encarga de la presentación visual del contenido. Se han implementado técnicas avanzadas como Grid y Flexbox para estructurar el layout, junto con variables CSS para mantener la coherencia de la paleta de colores y transiciones suaves.
* **JavaScript (ES6)**: Añade dinamismo e interactividad. Se han utilizado funciones y objetos para manipular el DOM, gestionar eventos (click, hover, submit) y desarrollar un carrusel de imágenes automático y manual.
* **Git**: Herramienta esencial para el control de versiones, permitiendo el trabajo en ramas, la realización de commits atómicos y la revisión del código mediante Pull Requests.

## 4. Estructura y Diseño del Proyecto

### 4.1. Estructura de Archivos

El proyecto se organiza en varias carpetas y archivos para facilitar su mantenimiento:

* **index.html**: Página principal que muestra la información personal, habilidades, experiencia y un carrusel de imágenes.
* **contacto.html**: Página dedicada a la sección de contacto, con un formulario interactivo.
* **styles.css**: Archivo que contiene todos los estilos CSS, incluyendo definiciones de variables, reglas para Grid y Flexbox, media queries para la responsividad y animaciones.
* **script.js**: Archivo JavaScript que implementa la funcionalidad dinámica, gestiona eventos y controla el carrusel de imágenes.
* **IMG/**: Directorio que almacena todas las imágenes utilizadas en el proyecto.
* **README.md**: Archivo de documentación general, donde se describen los objetivos, tecnologías y pasos de instalación o despliegue.

### 

### 4.2. Diseño y Distribución

El diseño se basa en una estructura clara y ordenada que facilita la navegación y mejora la experiencia de usuario:

* **Encabezado (Header)**: Incluye el título del portfolio y un menú de navegación que utiliza Flexbox para distribuir los elementos de manera fluida.
* **Sección Principal (Main)**:  
  + **Sobre Mí**: Presenta una breve biografía, foto de perfil y descripción del autor.
  + **Habilidades**: Lista las competencias técnicas mediante una lista con íconos para mayor impacto visual.
  + **Experiencia**: Se muestra en forma de tabla, detallando la experiencia laboral con columnas de empresa, puesto y año.
  + **Carrusel de Imágenes**: Un módulo interactivo que muestra imágenes relacionadas con el desarrollo y la formación del profesional, implementado con JavaScript.
* **Pie de Página (Footer)**: Contiene información sobre derechos de autor y se muestra con un diseño consistente y animado para integrarse con el resto del sitio.
* **Página de Contacto**: Similar en estructura al index, pero centrada en el formulario de contacto, permitiendo a los usuarios enviar consultas o comentarios.

## 

## 

## 

## 5. Implementación de Funcionalidades Dinámicas con JS

### 5.1. Declaración de Objetos y Organización del Código

El proyecto utiliza JavaScript moderno (ES6) para organizar funcionalidades en bloques lógicos, facilitando la reutilización y mantenibilidad del código. Aunque la declaración de objetos en el ejemplo es sencilla, se ha puesto especial atención en modularizar la funcionalidad:

* **Objetos para manipular el DOM**: Se han definido métodos para controlar la animación del carrusel y la respuesta a eventos del usuario (por ejemplo, el escalado de enlaces en el menú).

### 5.2. Carrusel de Imágenes

El carrusel es uno de los elementos interactivos clave del proyecto:

* **Funcionamiento**:  
  + Se seleccionan todas las imágenes del carrusel mediante document.querySelectorAll.
  + Se define una variable de índice que controla qué imagen se muestra.
  + Los botones "prev" y "next" permiten desplazar manualmente el carrusel. Al hacer clic, se actualiza el índice y se aplica una transformación CSS (translateX) para mover la galería.
  + Un setInterval se utiliza para la transición automática, renovando el índice cada 5 segundos.
* **Interactividad**:  
  + Se añaden eventos de clic a los botones para cambiar la imagen visible.
  + Las transiciones suaves y la actualización de la posición del carrusel aseguran una experiencia visual agradable.

### 5.3. Gestión de Eventos

Además del carrusel, el JavaScript gestiona otros eventos para mejorar la interactividad:

* **Navegación**:
  + Se implementan eventos mouseover y mouseout sobre los enlaces del menú, escalando los elementos para dar feedback visual.
* **Formulario de Contacto**:
  + Se intercepta el evento submit del formulario para evitar el envío real y mostrar un mensaje de confirmación, simulando el envío de datos.

## 6. Diseño Responsive con CSS Grid y Flexbox

### 6.1. Uso de CSS Grid

* **Estructuración de la Página**:
  + Se ha empleado CSS Grid para definir la disposición general del contenido en el elemento <main>, permitiendo que las secciones se organicen en una o varias columnas según el ancho de la pantalla.
  + Las media queries permiten cambiar la cantidad de columnas: una en dispositivos móviles, dos en tablets y tres en pantallas de escritorio.

### 6.2. Uso de Flexbox

* **Menú de Navegación**:
  + Flexbox se utiliza para alinear y distribuir los elementos del menú de forma uniforme y fluida, garantizando que la navegación sea cómoda en cualquier dispositivo.

### 

### 

### 

### 6.3. Media Queries y Adaptabilidad

* **Responsividad**:
  + Se han definido media queries específicas para adaptar la visualización de secciones, imágenes y el carrusel, asegurando que el contenido se muestre de forma correcta en dispositivos móviles, tablets y escritorio.
  + La utilización de unidades relativas y porcentajes en lugar de medidas fijas contribuye a la escalabilidad del diseño.

## 7. Mejoras de Accesibilidad y Usabilidad

* **Textos Alternativos**:
  + Todas las imágenes incluyen descripciones alternativas (alt) que facilitan la comprensión del contenido a usuarios con discapacidades visuales y mejoran la indexación en buscadores.
* **Contraste de Colores**:
  + Se ha verificado que la paleta de colores utilizada cumpla con los estándares de contraste para asegurar la legibilidad de textos y elementos gráficos.
* **Interactividad Adaptada**:
  + Se han añadido animaciones y transiciones que no solo mejoran la experiencia visual, sino que también brindan feedback inmediato al usuario en interacciones como clics y cambios de estado.

## 

## 

## 

## 

## 8. Uso Avanzado de CSS y Paleta de Colores

### 8.1. Variables y Consistencia de Diseño

* **Variables CSS**:
  + Se han definido variables (por ejemplo, --color-primario, --transicion-rapida) para mantener una coherencia en el uso de colores y tiempos de transición en toda la web.
* **Paleta de Colores**:
  + **Color primario (#2c3e50)**: Utilizado en encabezados, pie de página y elementos destacados.
  + **Color secundario (#3498db)**: Empleado en botones, enlaces y efectos hover.
  + **Color de fondo (#ecf0f1)** y **color de texto (#333)**: Aseguran una lectura clara y un contraste adecuado.

### 8.2. Animaciones y Transiciones

* **Animaciones**:
  + Se han implementado animaciones clave, como fadeIn y moverDesdeArriba, para suavizar la aparición de elementos y mejorar la experiencia del usuario.
* **Transiciones Suaves**:
  + Los botones y enlaces utilizan transiciones para cambios de color y escalado, haciendo que las interacciones sean visualmente agradables y responsivas.

## 

## 

## 9. Gestión del Proyecto con Git

### 9.1. Ramas y Commits

* **Organización de Ramas**:
  + Se han creado ramas específicas para el desarrollo de nuevas funcionalidades y mejoras en el diseño, facilitando la integración y revisión mediante Pull Requests.
* **Commits Detallados**:
  + Cada commit se ha realizado de forma atómica, con mensajes claros que indican los cambios realizados, lo que facilita la trazabilidad y el mantenimiento del código.

### 9.2. Organización del Repositorio

* **Estructura de Carpetas**:
  + El repositorio está organizado en carpetas bien definidas (por ejemplo, IMG/, css/, js/) y se incluye un archivo README.md actualizado con la información del proyecto.
* **Revisión de Código**:
  + Se han realizado revisiones mediante Pull Requests para garantizar la calidad y coherencia del código antes de fusionar los cambios en la rama principal.

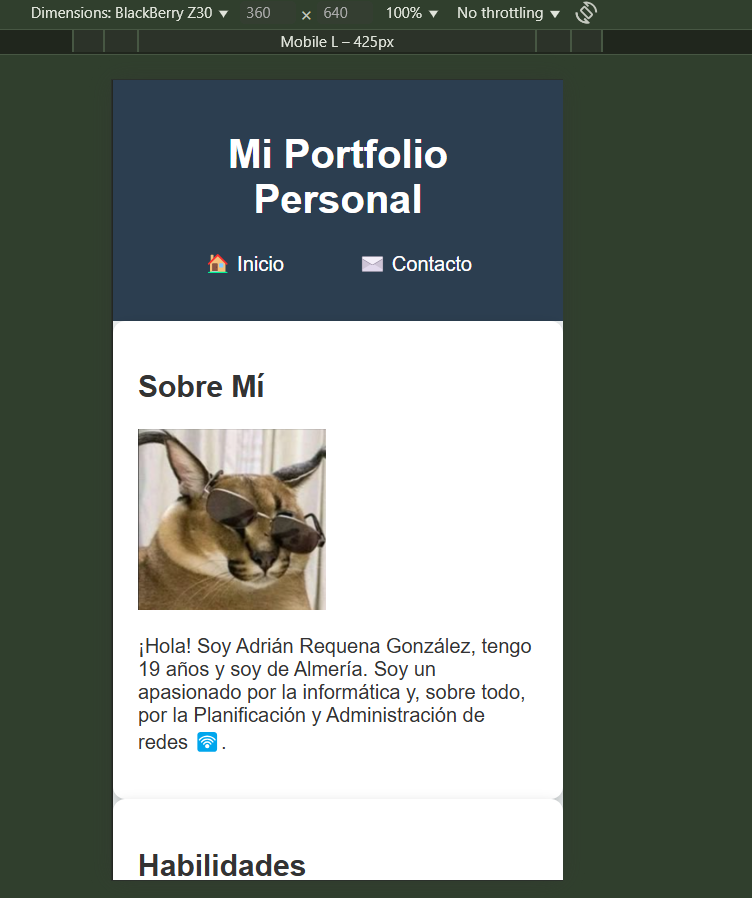
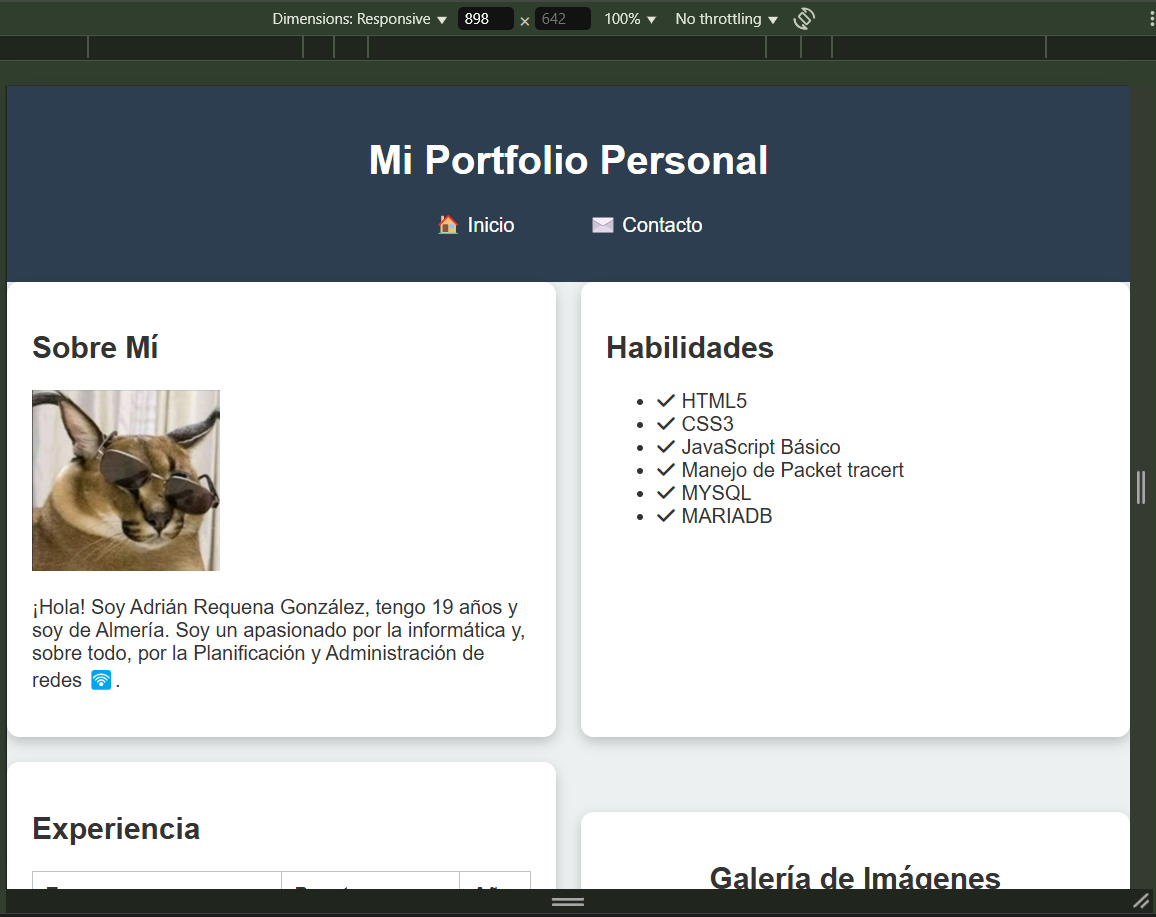
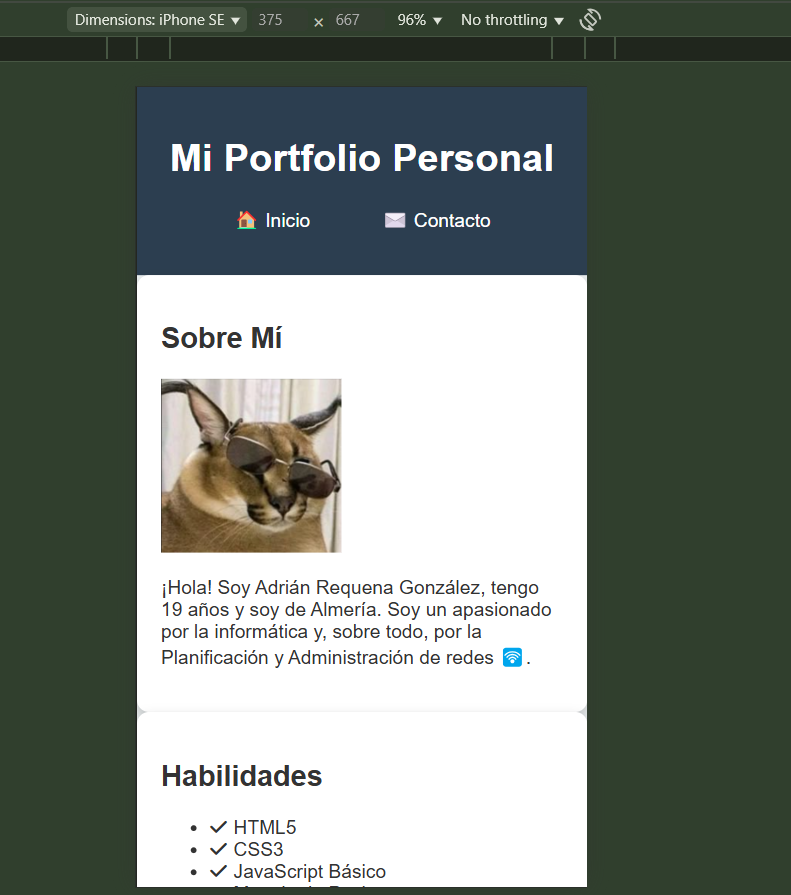
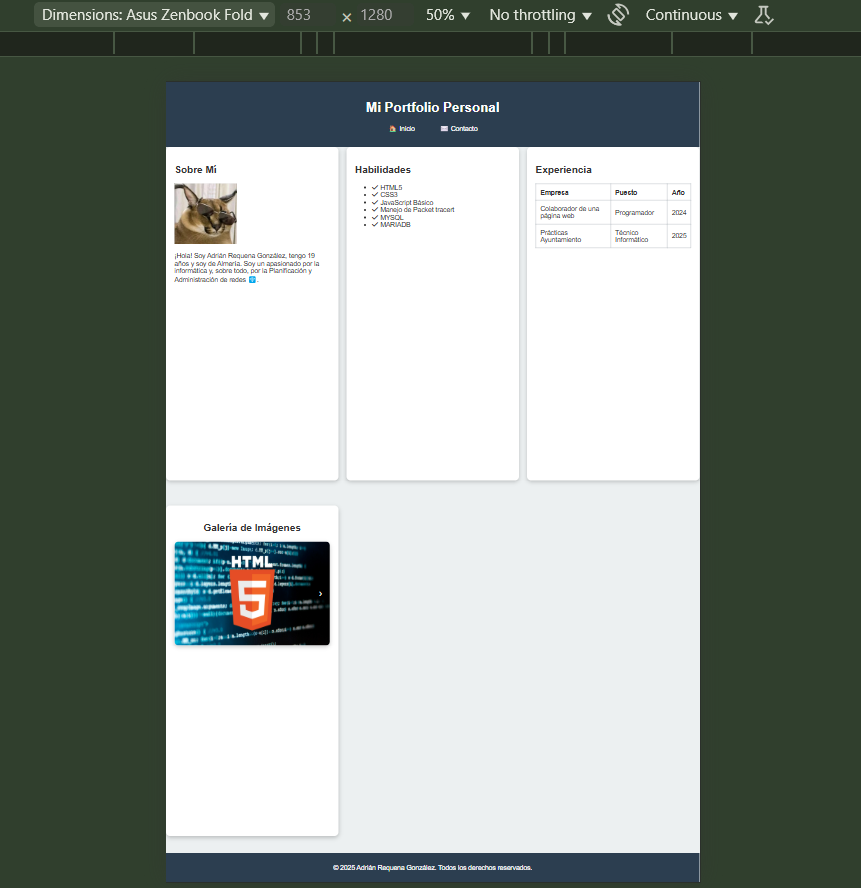
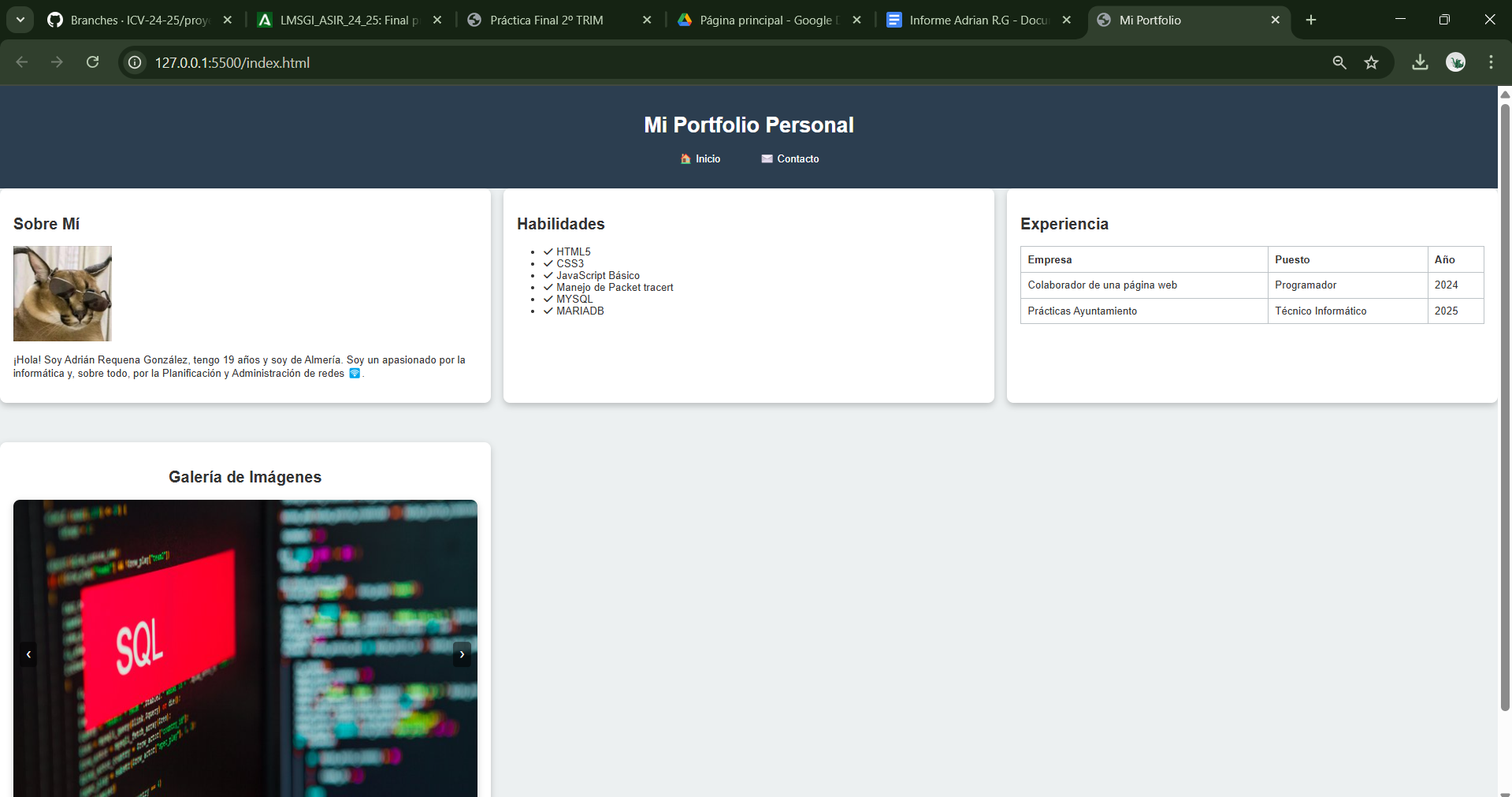
## 

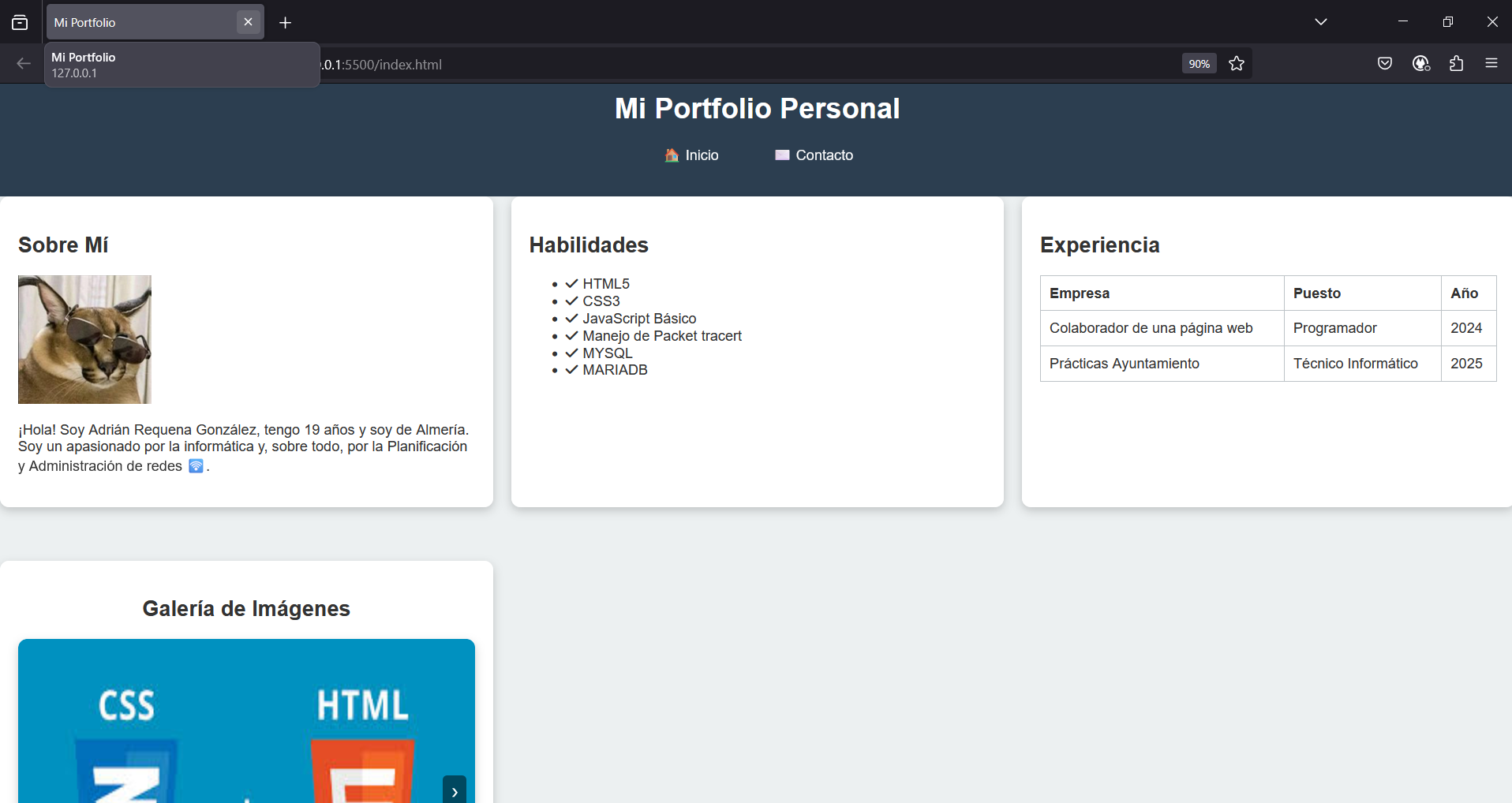
## 

## 

## 

## 10. Capturas de Pantalla y Pruebas de Usabilidad

* **Capturas en Diferentes Dispositivos**:
  + Se recomienda incluir capturas de pantalla que muestren el portfolio en dispositivos móviles, tablets y escritorio para evidenciar la responsividad.
  + 
  + 
  + 
  + 
* **Pruebas en Múltiples Navegadores**:
  + Se ha testeado la web en navegadores modernos (Chrome, Firefox, Safari) para asegurar compatibilidad y rendimiento.
* CHROME
  + 
* FIREFOX



* **Feedback de Usuarios**:
  + Se consideró la retroalimentación de usuarios para mejorar aspectos de usabilidad y accesibilidad.

## 12. Conclusión

El Portfolio Web Personal cumple con los objetivos del proyecto final, integrando un diseño atractivo, funcionalidad dinámica y adaptabilidad a múltiples dispositivos. La implementación de JavaScript (ES6) para interactividad, combinada con CSS Grid y Flexbox para un diseño responsive, demuestra un enfoque moderno y profesional. El uso de Git para el control de versiones y la organización del código garantiza un desarrollo estructurado y escalable. En resumen, este proyecto es una muestra sólida del perfil profesional del desarrollador y un paso importante en mi carrera en el ámbito de la informática.