

Universidad Instituto Tecnológico de las Américas (ITLA)



Presentado Por:

Erick Josue Oran

Matricula:

2020-9304

Presentado a:

Kelyn Tejada

Proyecto:

Venta de Vehiculos

Fecha:

3/12/2024

Índice

1. Portada

- Universidad Instituto Tecnológico de las Américas (ITLA)
- Presentado por: Erick Josue Oran
- Matrícula: 2020-9304
- Presentado a: Kelyn Tejada
- Proyecto: Venta de Vehículos
- Fecha: 5/12/2024

2. Introducción

- Objetivo del proyecto
- Tecnologías utilizadas
- Metodología Scrum

3. Proyecto Final - Programación III

- Nombre del proyecto: Venta de Vehículos
- Tecnologías para aplicar
- Objetivo del proyecto
- Alcance del proyecto
 - Incluye
 - Excluye

4. Cronograma del Proyecto

- Actividades, fechas de inicio y fin, responsable

5. Definición del Primer Release

- Sistema en la primera versión
- Requerimientos funcionales
- Requerimientos no funcionales

6. Metodología Scrum

- Análisis de requisitos y planificación del proyecto
- Diseño del frontend
- Interactividad con JavaScript
- Pruebas y ajustes
- Despliegue en GitHub Pages

7. Definición del Equipo de Trabajo

- Scrum Master
- Product Owner (PO)
- Equipo de desarrollo

8. Herramientas Utilizadas

- Gestión de proyectos y tareas
- Control de versiones
- Documentación
- Despliegue

9. Definir las Épicas

- Epic 1: Diseño del sitio web
- Epic 2: Funcionalidad de interacción con el usuario
- Epic 3: Optimización y pruebas
- Epic 4: Despliegue y finalización del proyecto

10. Ceremonias de Scrum

- Sprint Planning
- Daily Standup
- Sprint Review
- Sprint Retrospective

11. Historias de Usuario

- Historia de Usuario 1: Visualización de vehículos
- Historia de Usuario 2: Diseño responsivo
- Historia de Usuario 3: Modal de detalles
- Historia de Usuario 4: Optimización de imágenes
- Historia de Usuario 5: Diseño atractivo
- Historia de Usuario 6: Tiempo de carga
- Historia de Usuario 7: Modal funcional
- Historia de Usuario 8: Accesibilidad
- Historia de Usuario 9: Documentación del código
- Historia de Usuario 10: Compatibilidad con navegadores

12. Plan de Pruebas

- Requerimientos funcionales y no funcionales
- Criterios de aceptación y rechazo de pruebas
- Herramientas de pruebas

13. Cronograma de Ejecución de Pruebas

- Fase
- Actividad
- Herramienta
- Plazo
- Responsable

Introducción

El proyecto Venta de Vehículos tiene como objetivo desarrollar un sistema web que permita a los usuarios visualizar una galería de vehículos disponibles para la venta. La página estará diseñada con un enfoque moderno, intuitivo y amigable, utilizando tecnologías web estándar como HTML5, CSS y JavaScript. Este sistema tiene como prioridad la presentación visual de los vehículos, optimizando la experiencia del usuario mediante funcionalidades como un botón de "Ver detalles", que desplegará información adicional a través de un modal, y formularios básicos de contacto para solicitar más información.

El proyecto seguirá un enfoque ágil basado en Scrum, lo que asegura una planificación eficiente, revisión constante y entregables que cumplan con los objetivos planteados. Además, se utilizarán herramientas de control de versiones y despliegue como GitHub y GitHub Pages, garantizando que el sistema esté disponible en línea de manera gratuita.

PROYECTO FINAL- PROGRAMACIÓN III

Nombre del proyecto de software:

Venta de Vehículos

tecnología para aplicar:

Frontend: HTML5, CSS, JavaScript.

Control de versiones: GitHub.

Editor de código: Visual Studio Code.

Herramienta de despliegue: GitHub Pages.

Objetivo del proyecto:

El objetivo es presentar una solución práctica y escalable que pueda ser utilizada como base para proyectos más avanzados en la venta de vehículos, integrando principios de diseño y programación web moderna.

Alcance del proyecto:

Incluye:

- Página principal con un listado de vehículos (imagen, nombre, año, precio).
- Botón Ver detalles que simula una funcionalidad mediante JavaScript para mostrar más información en una ventana modal.
- Función de para enviar la información de contacto
- Función para Solicitar más información
- Diseño responsivo para dispositivos móviles, tablets y escritorios.

Excluye:

- Funcionalidades dinámicas como backend, bases de datos o autenticación.
- Carga y almacenamiento dinámico de datos.

Cronograma del proyecto:

Actividad	Fecha de inicio	Fecha de fin	Responsable
Análisis de requerimientos	27/11/2024	1/12/2024	[erick]
Diseño de estructura HTML	28/11/2024	1/12/2024	[erick]
Estilo y diseño con CSS	28/11/2024	1/12/2024	[erick]
Implementación de interacción con JS	28/11/2024	1/12/2024	[erick]
Pruebas y ajustes finales	29/11/2024	1/12/2024	[erick]

Definición del primer Release:

Versión 1.0:

El sistema podrá:

- Mostrar una galería de vehículos con imágenes, nombres, años y precios.
- Incluir botones de acción (Ver detalles) que abran un modal con más información.

Requerimientos funcionales:

- La página debe ser totalmente funcional en navegadores modernos.
- Las imágenes de los vehículos deben estar optimizadas para una carga rápida.

Requerimientos no funcionales:

- Diseño minimalista y atractivo visualmente.
- Interacción básica implementada con JavaScript sin dependencias externas.
- Implementar un diseño responsivo.

Definición del primer Release:

Sistema en la primera versión:

La primera versión del sistema, con un enfoque estático, se centrará en mostrar una página principal con información básica de los vehículos disponibles para la venta. Los usuarios podrán ver imágenes, nombres, años, y precios de los vehículos, con un botón de "Ver detalles" que abrirá un modal con información adicional.

Requerimientos funcionales:

- **Visualización de vehículos:** La página mostrará un listado de vehículos con imágenes, nombre del modelo, año y precio.

- **Interacción con los usuarios:** Los usuarios podrán hacer clic en el botón "Ver detalles" de cada vehículo para abrir un modal con más detalles (como especificaciones adicionales).
- **Carga rápida:** Las imágenes de los vehículos deben estar optimizadas para garantizar tiempos de carga rápidos.
- **Funcionalidad del modal:** El modal se mostrará correctamente con los detalles del vehículo seleccionado.
- **Diseño responsivo:** La página será accesible y fácil de navegar en dispositivos de escritorio, tabletas y móviles.

Requerimientos no funcionales:

- **Usabilidad:** La interfaz debe ser clara, intuitiva y fácil de navegar.
- **Accesibilidad:** El sitio debe cumplir con los estándares básicos de accesibilidad (contrastes adecuados, textos alternativos para imágenes, etc.).
- **Seguridad básica:** Aunque no se usen datos sensibles, el sitio debe tener prácticas de seguridad como protección contra inyecciones de código o vulnerabilidades de front-end.
- **Escalabilidad mínima:** Si en el futuro se añaden más vehículos, la estructura debe permitir una fácil expansión del contenido.

Metodología Scrum

Análisis de requisitos y planificación del proyecto:

- Establecer la estructura básica del proyecto.
- Identificar las tecnologías a usar.

Diseño del frontend:

- Estructura de la página principal (HTML).
- Estilización de la página (CSS).

Interactividad con JavaScript:

- Implementación de la funcionalidad del modal.
- Creación de la lógica para mostrar los detalles del vehículo.

Pruebas y ajustes:

- Pruebas de usabilidad y accesibilidad.
- Optimización de imágenes y tiempos de carga.

Despliegue en GitHub Pages:

- Subir los archivos a un repositorio de GitHub.

- Configuración para que el proyecto sea accesible en línea mediante GitHub Pages.

Definir el equipo de trabajo:

Scrum Master:

- Responsable de garantizar que el equipo siga los procesos de Scrum y elimine obstáculos.

Product Owner (PO):

- Encargado de definir y priorizar las características del proyecto.

Equipo de desarrollo:

- responsable del diseño, desarrollo, pruebas e implementación del proyecto.

Herramientas que usarían:

Gestión de proyectos y tareas:

- Trello o Jira para gestionar tareas y sprints. Estas plataformas permiten crear tableros donde se asignan tareas y se realiza un seguimiento del progreso.

Control de versiones:

- Git con GitHub para versionar el código y trabajar de manera colaborativa si fuera necesario.

Documentación:

- Google Docs para la documentación del proyecto (como el cronograma, definición de requerimientos, etc.).

Despliegue:

- GitHub Pages para publicar el proyecto de forma gratuita en línea.

Definir las épicas:

Epic 1: Diseño del sitio web

- Crear la estructura básica de la página.
- Aplicar diseño responsivo.

Epic 2: Funcionalidad de interacción con el usuario

- Implementar los botones "Ver detalles".
- Desarrollar el modal para mostrar más detalles.

Epic 3: Optimización y pruebas

- Optimizar imágenes.
- Realizar pruebas de funcionalidad y accesibilidad.

Epic 4: Despliegue y finalización del proyecto

- Subir el proyecto a GitHub.
- Configurar GitHub Pages para mostrar el proyecto en línea.

ceremonias de Scrum:

Sprint Planning:

- Fecha sugerida: Primer día de cada semana.

Daily Standup:

- Fecha sugerida: Diario, al inicio o final del día.

Sprint Review:

- Fecha sugerida: Al final de cada semana.

Sprint Retrospective:

- Fecha sugerida: Al final de cada sprint.

Historias de usuario:

Historia de Usuario 1: Como usuario, quiero ver una lista de vehículos disponibles para la venta.

- **Criterios de aceptación:** La página muestra al menos tres vehículos con imagen, nombre, año y precio.
- **Puntos de historia:** 3

Historia de Usuario 2: Como usuario, quiero que la página se vea bien en dispositivos móviles.

- **Criterios de aceptación:** El diseño debe ser responsivo y adaptarse a diferentes tamaños de pantalla.
- **Puntos de historia:** 5

Historia de Usuario 3: Como usuario, quiero ver más detalles de un vehículo cuando haga clic en "Ver detalles".

- **Criterios de aceptación:** Se abre un modal con más información del vehículo.
- **Puntos de historia:** 3

Historia de Usuario 4: Como usuario, quiero que las imágenes de los vehículos se carguen rápidamente.

- **Criterios de aceptación:** Las imágenes deben estar optimizadas para una carga rápida sin pérdida significativa de calidad.
- **Puntos de historia:** 2

Historia de Usuario 5: Como usuario, quiero que la página tenga un diseño limpio y atractivo.

- **Criterios de aceptación:** La página debe ser visualmente atractiva, con una disposición ordenada y sin elementos desordenados.
- **Puntos de historia:** 4

Historia de Usuario 6: Como usuario, quiero que el sitio cargue rápidamente.

- **Criterios de aceptación:** El sitio debe cargar en menos de 2 segundos en una conexión estándar.
- **Puntos de historia:** 4

Historia de Usuario 7: Como usuario, quiero que el modal sea fácil de cerrar y mostrar correctamente.

- **Criterios de aceptación:** El modal debe ser cerrado con un botón de "Cerrar" y debe mostrar la información del vehículo correctamente.
- **Puntos de historia:** 3

Historia de Usuario 8: Como usuario, quiero que los textos sean legibles y los colores sean accesibles.

- **Criterios de aceptación:** Los contrastes de color deben cumplir con las pautas de accesibilidad.
- **Puntos de historia:** 3

Historia de Usuario 9: Como desarrollador, quiero que el código esté bien documentado.

- **Criterios de aceptación:** El código debe contener comentarios claros que expliquen su propósito.
- **Puntos de historia:** 2

Historia de Usuario 10: Como usuario, quiero que la página funcione correctamente en los navegadores más comunes.

- **Criterios de aceptación:** La página debe funcionar en Chrome, Firefox y Safari sin errores.
- **Puntos de historia:** 3

Plan de Pruebas para el Proyecto de Venta de Vehículos

Lista de requerimientos funcionales y no funcionales:

Requerimientos Funcionales:

Visualización de vehículos:

- Los vehículos deben ser listados con nombre, año, precio e imagen.

Interactividad con el usuario:

- Al hacer clic en "Ver detalles", se debe abrir un modal con información adicional sobre el vehículo.

Funcionalidad del modal:

- El modal debe abrirse correctamente y debe cerrarse al hacer clic en el botón "Cerrar".

Optimización de imágenes:

- Las imágenes de los vehículos deben cargarse rápidamente y verse nítidas.

Requerimientos No Funcionales:

Usabilidad:

- La interfaz debe ser clara, intuitiva y fácil de navegar.

Accesibilidad:

- Los contrastes de color deben ser apropiados y las imágenes deben tener textos alternativos.

Performance:

- Las páginas deben cargar en menos de 2 segundos.

Seguridad:

- El sitio debe ser protegido contra vulnerabilidades de front-end comunes.

Criterios de aceptación y rechazo de pruebas:

Criterios de Aceptación:

Visualización de vehículos: La lista de vehículos debe mostrar al menos tres vehículos con sus detalles visibles (nombre, año, precio, imagen).

Modal de detalles: Al hacer clic en "Ver detalles", el modal debe mostrar información detallada del vehículo seleccionado.

Optimización de imágenes: Las imágenes deben cargar en menos de 2 segundos y verse claras en todas las resoluciones.

Accesibilidad: El contraste entre el fondo y el texto debe ser de al menos 4.5:1 y todas las imágenes deben tener un texto alternativo descriptivo.

Criterios de Rechazo:

Visualización de vehículos: Si algún vehículo no se carga correctamente o no se muestra con los datos requeridos, la prueba será rechazada.

Modal de detalles: Si el modal no se abre correctamente o no muestra los datos correctos del vehículo, la prueba será rechazada.

Optimización de imágenes: Si las imágenes tardan más de 2 segundos en cargar o tienen mala calidad, la prueba será rechazada.

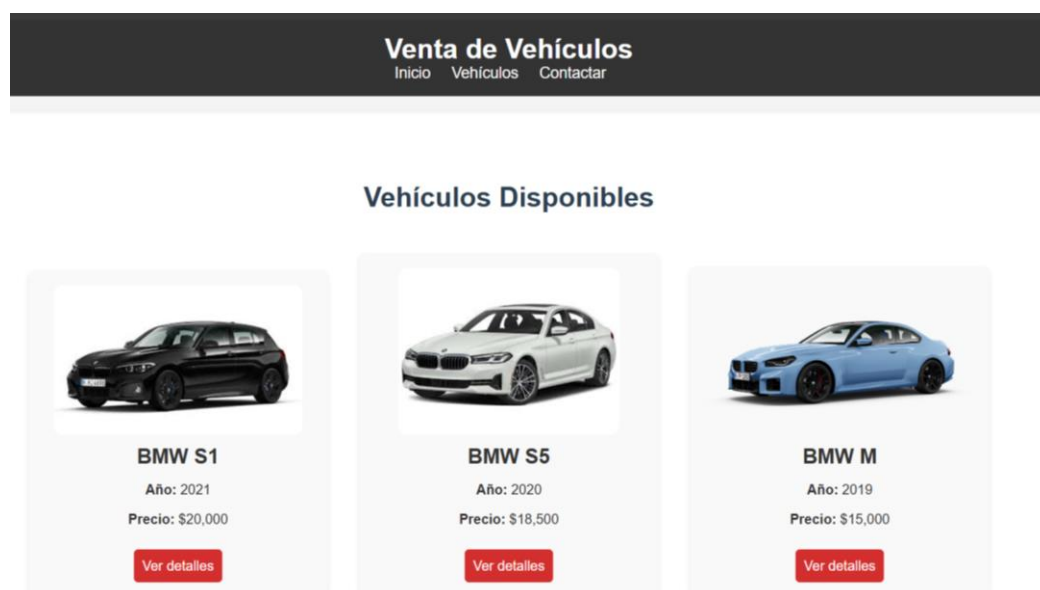
Accesibilidad: Si el contraste de color no es suficiente o las imágenes no tienen texto alternativo, la prueba será rechazada.

Herramientas de Pruebas

Herramientas utilizadas:

Prueba de carga del sitio web (Caso de prueba 1)

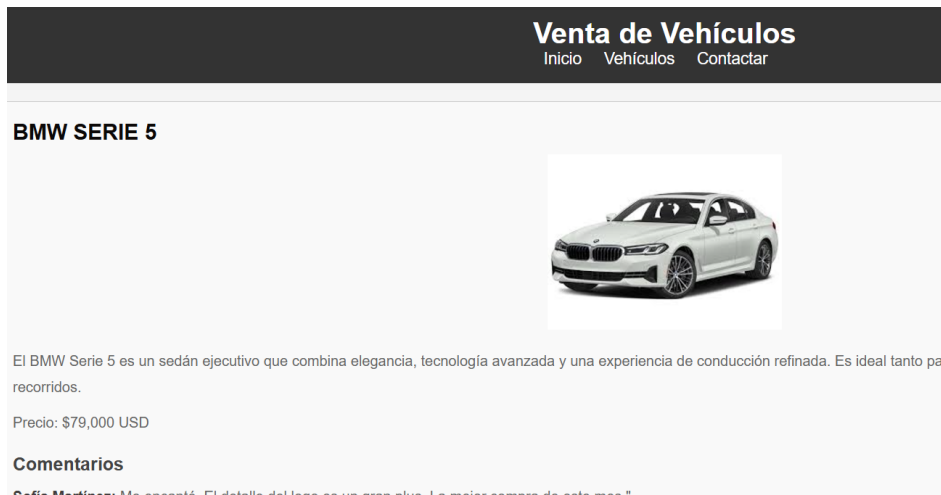
Objetivo: Verificar que el sitio cargue correctamente sin errores.



Resultado: El sitio se carga sin errores, y todos los elementos visuales están presentes.

Prueba de funcionalidad del botón (Caso de prueba 2)

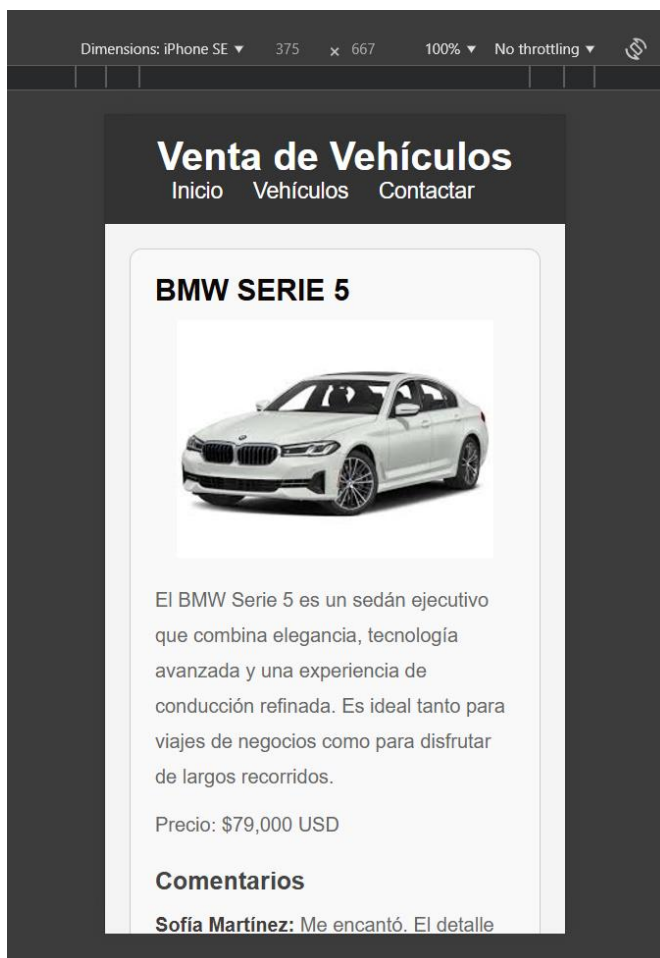
Objetivo: Verificar que el botón "Ver detalles" funcione correctamente.



Resultado: El modal o página se abre correctamente y muestra la información del producto.

Prueba de diseño responsivo (Caso de prueba 3)

Objetivo: Verificar que el diseño sea responsivo en dispositivos móviles.



Resultado: El diseño se adapta correctamente a diferentes tamaños de pantalla (por ejemplo, móviles y tabletas).

Cronograma de Ejecución de Pruebas

Fase	Actividad	Herramienta	Plazo	Responsable
Configuración	Configuración del entorno de pruebas	Google chrome	1 día	erick
Pruebas manuales	Pruebas de carga del sitio web	Google chrome	1 día	erick
Pruebas de funcionalidad	Validar funcionalidad de botones	Google chrome	2 días	erick
Pruebas de responsividad	Verificar diseño en dispositivos móviles	Google chrome	1 día	erick
Validación final	Corrección de errores y ajustes	Google chrome	2 días	erick

Plantillas para Casos de Pruebas**Formato Estándar:**

Caso de Prueba	Descripción	Pasos	Resultados Esperados	Resultados Obtenidos	Estado
CP01	Validar carga correcta del sitio web	1. Abrir URL del sitio.	El sitio carga en menos de 3 segundos.	El sitio logra cargar en el tiempo estimado	Verificado
CP02	Verificar diseño responsivo en pantallas móviles	1. Cambiar resolución del navegador.	Todos los elementos se ajustan.	se ajustan correctamente	Verificado
CP03	Comprobar funcionalidad del botón "Ver detalles"	1. Hacer clic en el botón correspondiente.	Se despliega el modal con información.	Los botones funcionan correctamente	Verificado
CP04	Validar tiempos de respuesta para la navegación entre secciones	1. Navegar entre páginas.	Respuesta menor a 2 segundos por sección.	Cumple con los 2 segundos de sección	Verificado

Equipos de Pruebas y Responsabilidades

Equipo	Responsabilidades
erick	Realizar pruebas manuales y documentar hallazgos.
erick	Diseñar y ejecutar scripts automatizados.
erick	Resolver los errores reportados por el equipo QA.
erick	Verificar accesibilidad y diseño responsivo.

Plan de Automatización de Pruebas

Estrategias:

- Cobertura de pruebas clave: Automatizar los flujos críticos como navegación, carga de la galería y funcionalidad del botón "Ver detalles".
- Ejecución periódica: Integrar pruebas en un pipeline CI/CD para verificar la estabilidad en cada despliegue.

Herramientas:

- GitHub Actions: Configurar pipelines para ejecutar pruebas automatizadas tras cada commit.

Plan:

- Diseñar scripts para los casos de pruebas automatizables.
- Configurar el entorno de Jenkins o GitHub Actions para integrarlos.
- Realizar pruebas iniciales y ajustar según resultados.
- Ejecutar pruebas automatizadas en cada despliegue y corregir errores detectados.

Conclusión

El desarrollo del proyecto Venta de Vehículos permitió aplicar conocimientos fundamentales de diseño y programación web, destacando la importancia de una estructura bien organizada y de una interfaz que priorice la usabilidad y la accesibilidad. A través del uso de tecnologías como HTML, CSS y JavaScript, se logró crear un sistema estático funcional y atractivo que cumple con los requerimientos planteados. El enfoque ágil basado en Scrum aseguró que las tareas fueran distribuidas y completadas eficientemente, mientras que el despliegue en GitHub Pages facilitó la publicación del proyecto en línea.