

# Explicación Técnica del 2do Parcial Rectologías Web

---

## 1. Descripción General del Proyecto

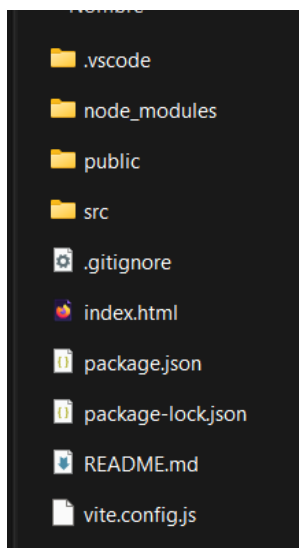
Este proyecto consiste en un portal web desarrollado utilizando **Vue.js 3** junto con **Vite** como herramienta de construcción. El objetivo principal del portal es **consumir la API pública de Star Wars (SWAPI)** para mostrar información sobre personajes en una interfaz organizada y responsiva.

El sitio permite navegar de forma sencilla por los datos consumidos, mostrando **nombre, altura, género y año de nacimiento** de cada personaje. Además, el sitio sigue una estructura basada en componentes, lo que permite su **reutilización y mantenimiento eficiente**.

---

## 2. Estructura del Proyecto

El proyecto está organizado de la siguiente manera:



## 3. Lógica del Código

La lógica del proyecto se centra principalmente en el componente `PeopleList.vue`, donde se hace la solicitud HTTP a la API de **SWAPI**.

Se utiliza `fetch` para obtener los datos desde:

<https://swapi.dev/api/people/>

Los resultados se almacenan en un arreglo local (people). Luego se recorre este arreglo con v-for para renderizar un componente PersonCard por cada personaje. Cada tarjeta muestra información básica del personaje.

---

## 4. Manejo de Componentes

El proyecto utiliza **componentes Vue** para modularizar la interfaz. Los componentes principales son:

- **Navbar.vue**: Una barra de navegación simple con el título de la aplicación.
- **Footer.vue**: Un pie de página sencillo con derechos de autor.
- **PersonCard.vue**: Una tarjeta que muestra datos del personaje (nombre, altura, género y nacimiento).

Esta estructura permite escalar el proyecto fácilmente y mantener el código limpio y reutilizable.

---

## 5. Uso de Vue Router

Vue Router está configurado en `src/router/index.js`. Aunque solo se utiliza una ruta en esta versión (/), el enrutador está preparado para manejar múltiples vistas en el futuro (como detalles de personajes, planetas, naves, etc.).

Esto permite que la aplicación crezca de manera ordenada y con una navegación más rica.

---

## 6. Estilos y Diseño

Los estilos del portal se realizaron con **CSS puro**, sin usar frameworks como Tailwind o Bootstrap.

Se utilizaron herramientas modernas como **Flexbox y Grid CSS** para lograr una interfaz visual ordenada y adaptable. El diseño es **responsive**, lo que permite visualizarlo bien desde dispositivos móviles hasta pantallas de escritorio.

---

## 7. Lógica de Carga y Manejo de Errores

Dentro de `PeopleList.vue` se implementó una lógica básica de carga y error:

- Mientras los datos se están solicitando, se muestra el mensaje **"Cargando personajes..."**.
- Si ocurre un error en la petición a la API, se muestra un mensaje de error visible para el usuario.

Esto mejora la experiencia de usuario y hace el sistema más robusto ante errores externos como fallas de red o problemas de la API.

## 8. Conclusión

Este proyecto cumple con todos los requerimientos de la segunda evaluación de Tecnologías Web I:

- Consumo de una API externa (SWAPI)
- Uso de Vue 3 y Vite
- Componentes reutilizables
- Estilos hechos con CSS puro
- Navegación mediante Vue Router
- Manejo de carga y errores
- Documentación técnica clara

La arquitectura del proyecto también permite **escalar fácilmente** a otros recursos de la API (planetas, naves, películas) o agregar detalles individuales por personaje.

## Información del Estudiante

**Nombre completo:** Guachalla Quispe Andrea Camila

**Carrera:** Ingeniería Multimedia e Interactividad Digital

**Materia:** Tecnologías Web I

**Evaluación:** Segunda Evaluación (Segunda Opción)

**Gestión:** 1-2025

**Universidad:** Universidad Católica Boliviana “San Pablo”