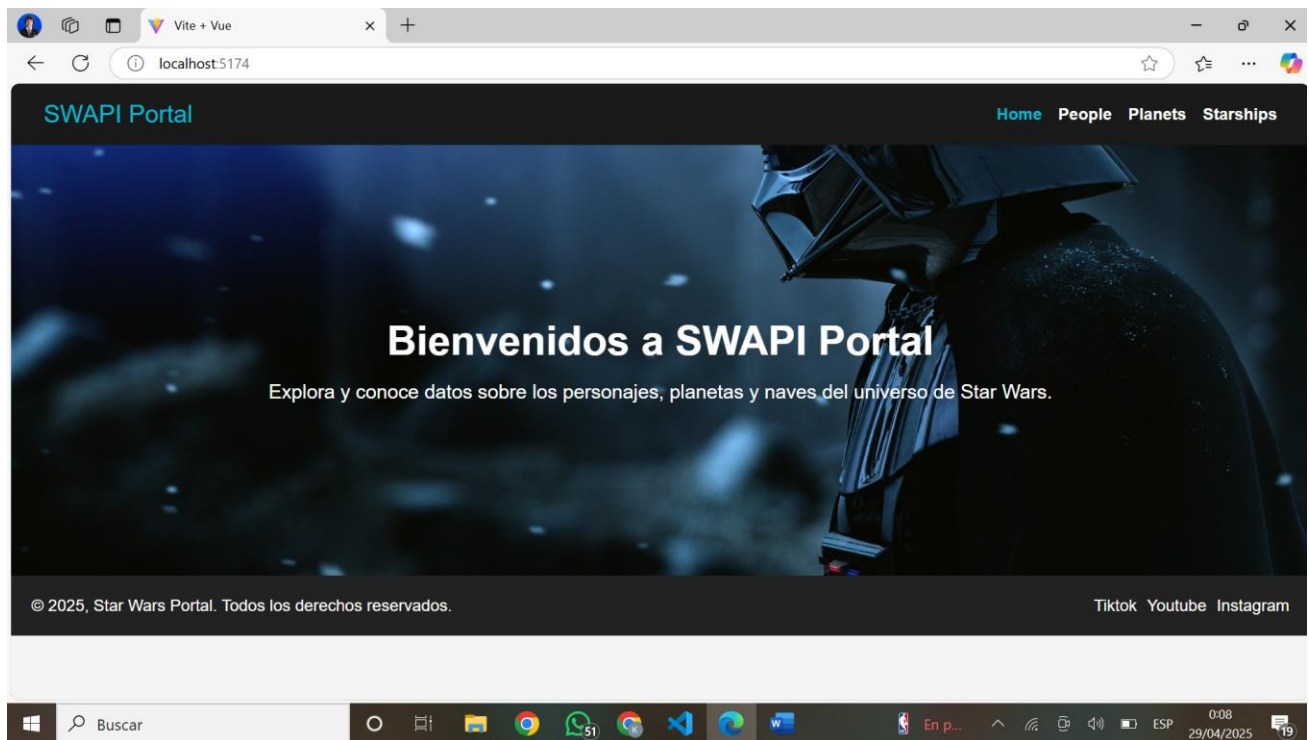
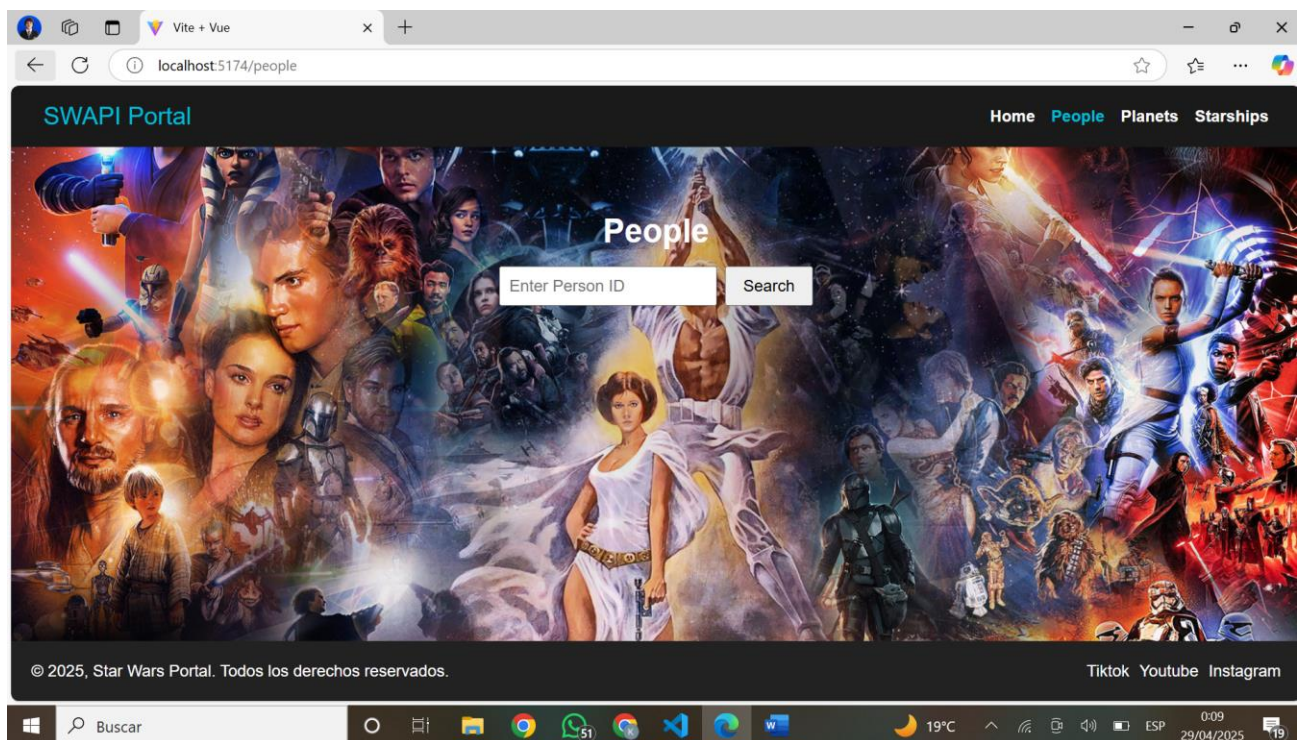
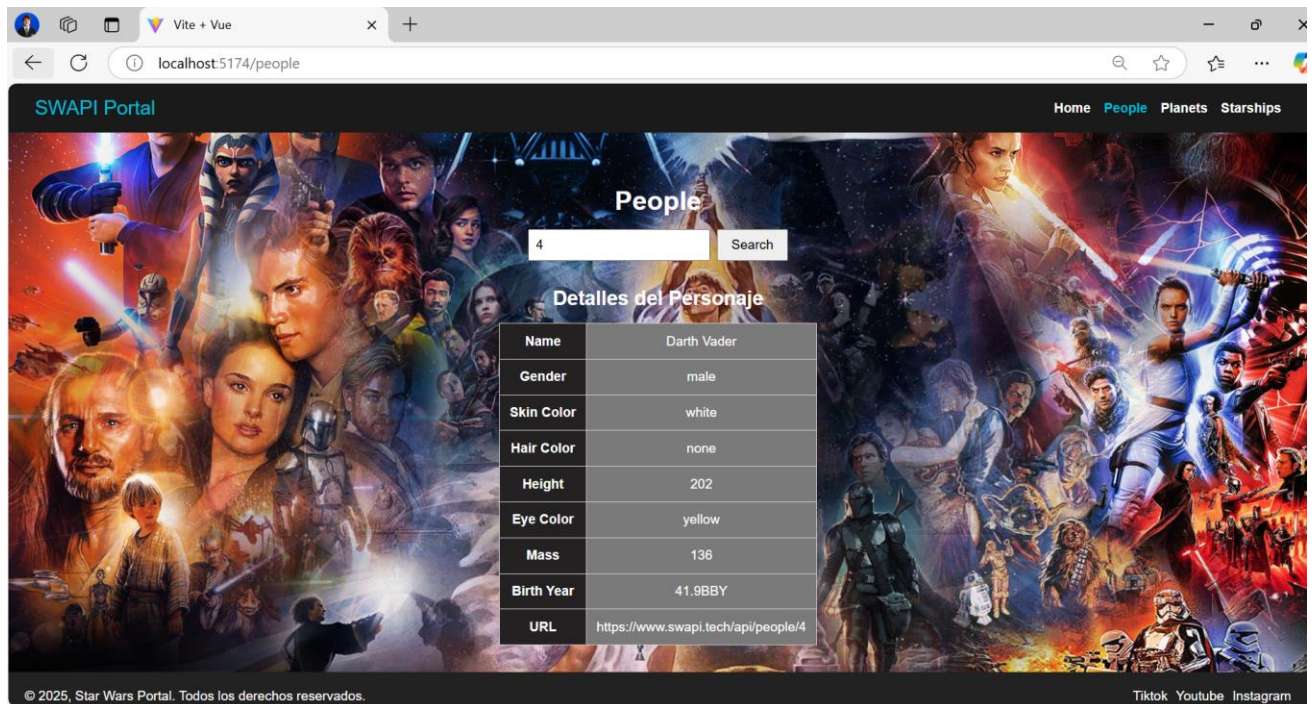


1. Home Page

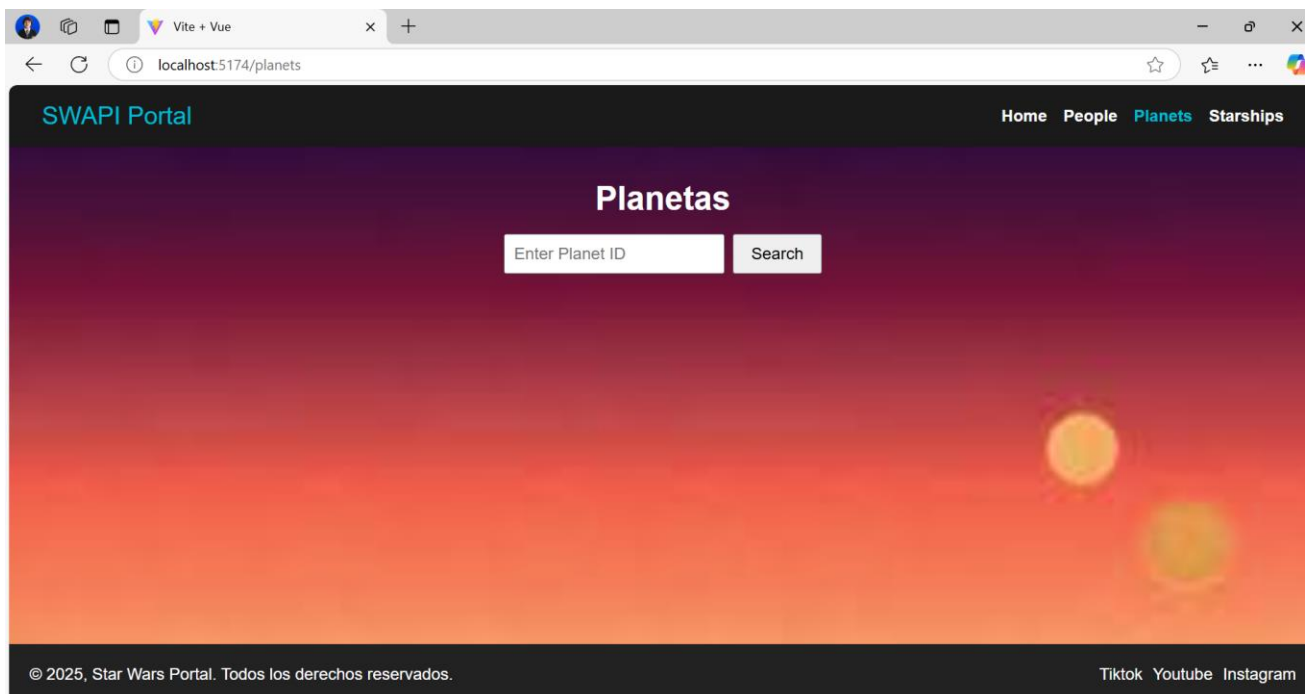


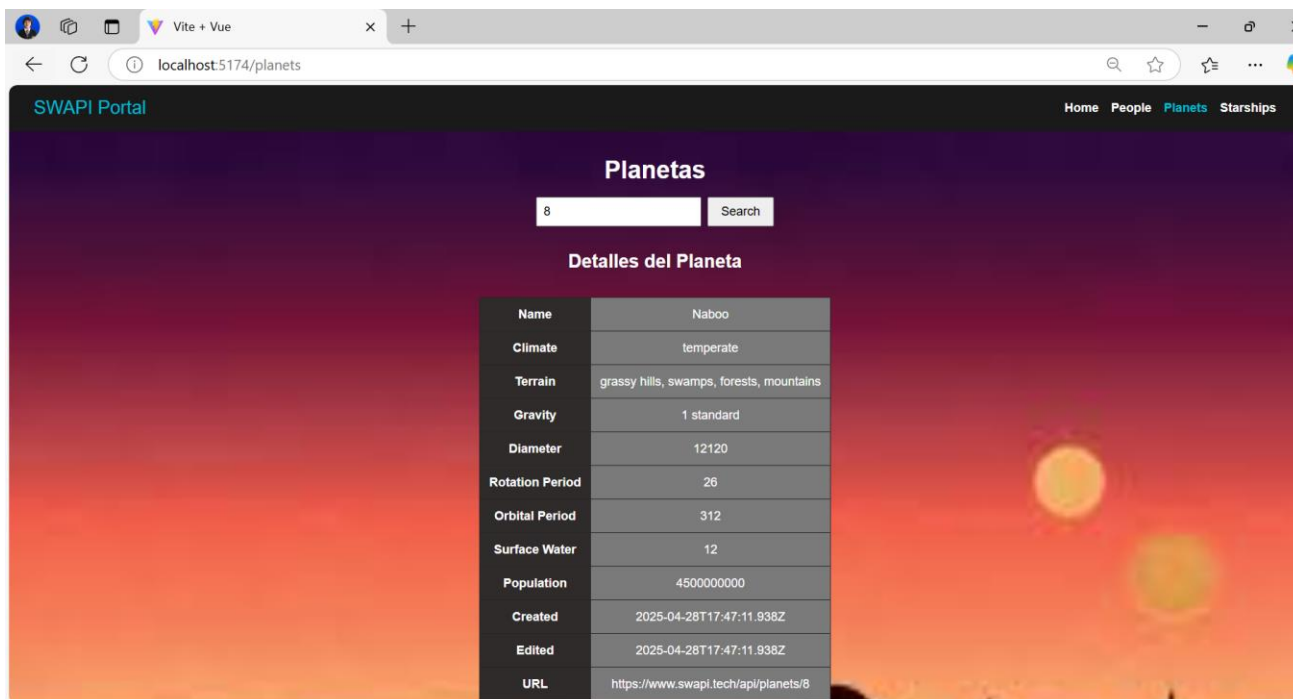
2. People Page



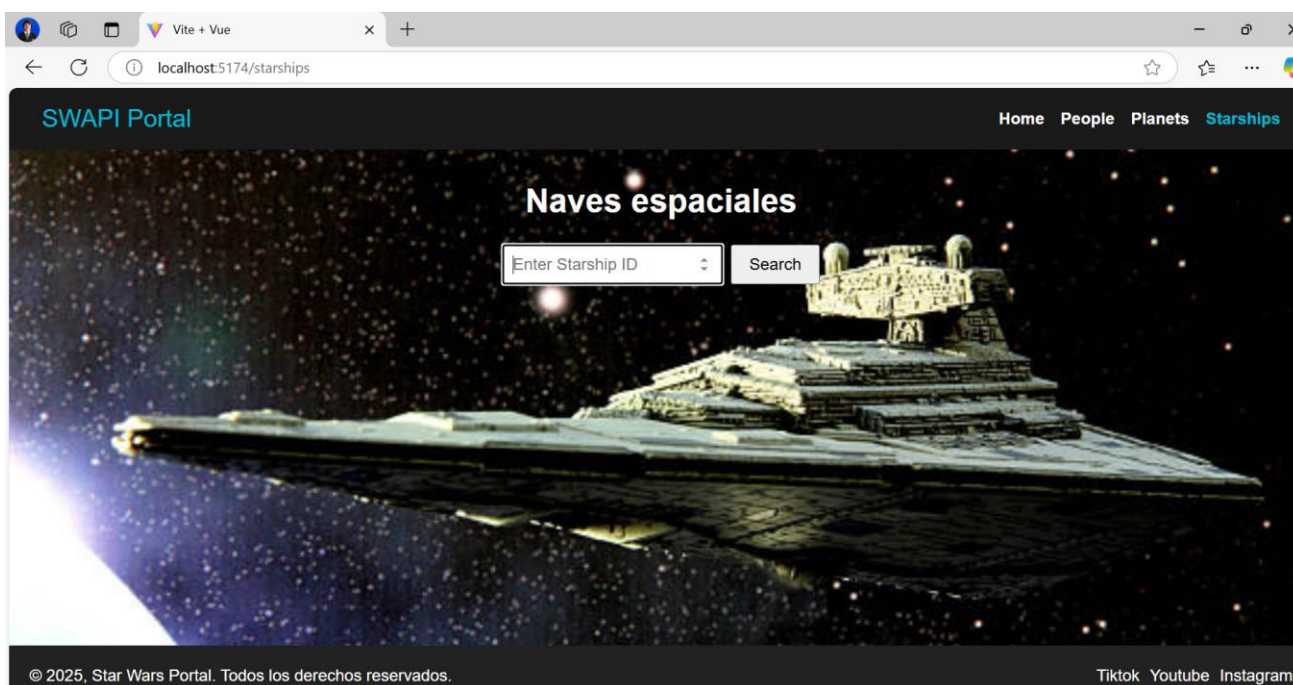


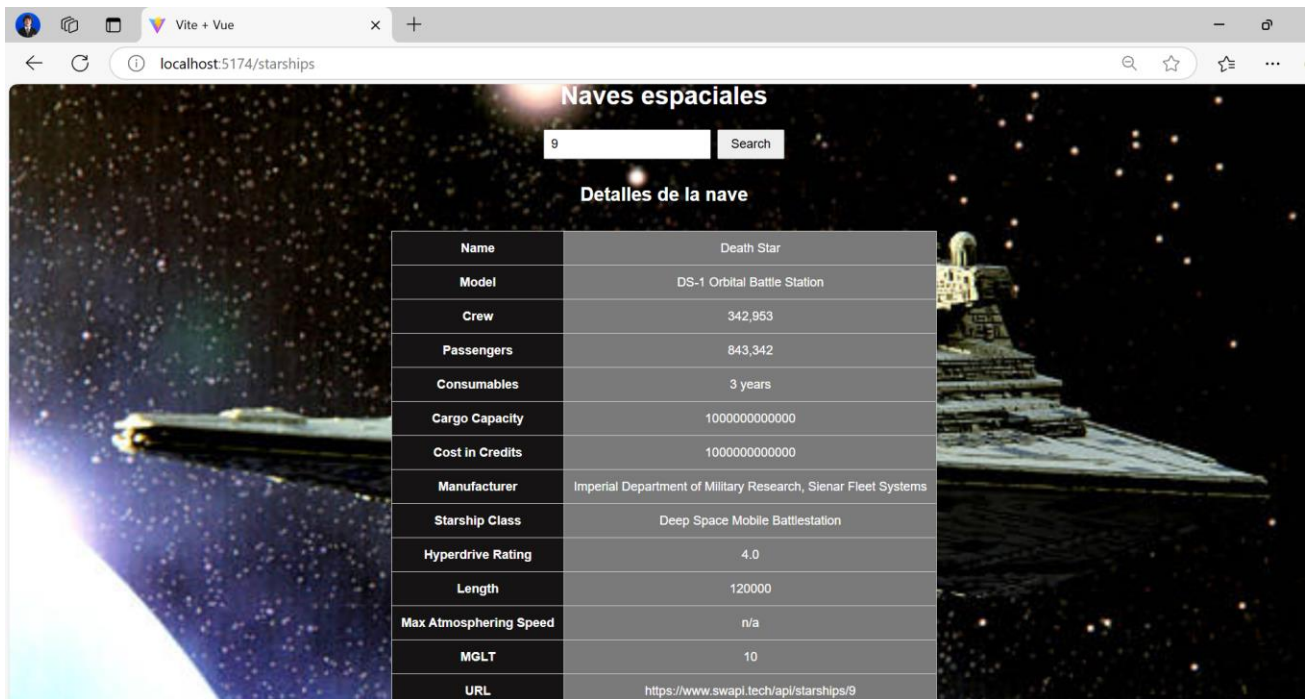
3. Planets page





4. Starships page





Explicación Técnica del Código, Componentes y API Consumida

Estructura General del Proyecto

El proyecto está desarrollado con Vue 3 y se organiza mediante Single File Components (SFC), lo que facilita una separación clara entre la estructura (<template>), lógica (<script setup>) y estilos (<style scoped>). Está compuesto por vistas independientes para People, Starships y Planets, además de componentes reutilizables.

Componentes Principales

1. PeoplePage.vue, StarshipsPage.vue, PlanetsPage.vue

Cada uno de estos archivos representa una **vista dedicada** con funcionalidad casi idéntica, pero adaptada al recurso correspondiente de la API.

Funcionalidades compartidas:

- Capturan un **ID ingresado por el usuario**.
- Realizan una **petición HTTP** a la **Star Wars API (SWAPI)** para obtener los datos.
- Muestran un componente de **carga (Loading.vue)** mientras se espera la respuesta.
- Muestran un componente de **error (Error.vue)** si ocurre un fallo en la petición.
- Renderizan una **tabla con los detalles del recurso** recuperado (personaje, nave o planeta).
- Están envueltos entre los componentes **Navbar** y **Footer**.

Componentes Reutilizables

Navbar.vue

Una barra de navegación fija que permite al usuario navegar entre las vistas: People, Starships y Planets.

Footer.vue

Un pie de página que se mantiene en la parte inferior gracias a la estructura flex de people-page, starships-page, y planets-page.

Loading.vue

Muestra un spinner o texto mientras se espera una respuesta de la API.

Error.vue

Recibe un prop message y muestra un mensaje de error si la búsqueda falla.

API Consumida: [SWAPI - Star Wars API](https://www.swapi.tech/api/)

El proyecto utiliza **Axios** para hacer solicitudes HTTP a la API REST de Star Wars (<https://www.swapi.tech/api/>).

Endpoints usados:

- /people/:id → Información de personajes.
- /starships/:id → Información de naves.
- /planets/:id → Información de planetas.

Ejemplo de uso con Axios en searchPerson():

```
js
CopiarEditar
const response = await api.get(`/people/${searchId.value}`)
```

La respuesta JSON tiene esta forma:

```
json
CopiarEditar
{
  "message": "ok",
  "result": {
    "properties": {
      "name": "Luke Skywalker",
      "height": "172",
```

```
...  
}  
}  
}
```

Los datos extraídos se muestran directamente en la tabla dentro del <template> de cada vista.