**Universidad Técnica Nacional**

Sede Regional, San Carlos

Ingeniería en Software

Calidad del Software

**Avance N°1 SWAPI Testing**

Alumnos,

Jordan Josué Herrera Carvajal.

Roney Ruiz Rojas.

Fabian Zamora Araya.

Fecha de entrega,

25/07/2025

II Cuatrimestre, 2025

Logotipo

El contenido generado por IA puede ser incorrecto.

**Tabla de contenido**

[**Introducción** 3](#_Toc204278377)

[**Desarrollo** 4](#_Toc204278378)

[**API Seleccionada:** 5](#_Toc204278379)

[**URL Oficial:** 5](#_Toc204278380)

[**Herramienta:** 5](#_Toc204278381)

[**Endpoint #1:** 5](#_Toc204278382)

[**URL:** 5](#_Toc204278383)

[**Método HTTP:** 5](#_Toc204278384)

[**Descripción:** 6](#_Toc204278385)

[**Parámetros:** 6](#_Toc204278386)

[**Endpoint #2:** 6](#_Toc204278387)

[**URL:** 6](#_Toc204278388)

[**Método HTTP:** 7](#_Toc204278389)

[**Descripción:** 7](#_Toc204278390)

[**Parámetros:** 7](#_Toc204278391)

[**Endpoint #3:** 8](#_Toc204278392)

[**URL:** 8](#_Toc204278393)

[**Método HTTP:** 8](#_Toc204278394)

[**Descripción:** 8](#_Toc204278395)

[**Parámetros:** 8](#_Toc204278396)

[**Endpoint #4:** 9](#_Toc204278397)

[**URL:** 9](#_Toc204278398)

[**Método HTTP:** 9](#_Toc204278399)

[**Descripción:** 9](#_Toc204278400)

[**Parámetros:** 9](#_Toc204278401)

[**Endpoint #5:** 10](#_Toc204278402)

[**URL:** 10](#_Toc204278403)

[**Método HTTP:** 10](#_Toc204278404)

[**Descripción:** 10](#_Toc204278405)

[**Parámetros:** 10](#_Toc204278406)

[**Endpoint #6:** 11](#_Toc204278407)

[**URL:** 11](#_Toc204278408)

[**Método HTTP:** 11](#_Toc204278409)

[**Descripción:** 11](#_Toc204278410)

[**Parámetros:** 11](#_Toc204278411)

[**Pruebas Funcionales** 12](#_Toc204278412)

[**Descripción:** 12](#_Toc204278413)

[**Importancia:** 12](#_Toc204278414)

[**Pruebas de Validación de Errores** 12](#_Toc204278415)

[**Descripción:** 12](#_Toc204278416)

[**Importancia:** 12](#_Toc204278417)

[**Pruebas de Contrato** 12](#_Toc204278418)

[**Descripcion:** 12](#_Toc204278419)

[**Importancia:** 12](#_Toc204278420)

# **Introducción**

En este proyecto, nos interesamos en la Star Wars API (SWAPI), donde prácticamente es una interfaz abierta que nos permite acceder a información de personajes, naves, planetas y mucho más del Universo paralelo de Star Wars.

Para esto, utilizaremos el apoyo de la herramienta de Postman, ya que es poderosa y amigable que nos permite interactuar fácilmente con servicios web.

La idea es adquirir conocimiento de como funcionan las peticiones HTTP, explorar respuestas en formato JSON y entender cómo los sistemas modernos a través de APIS.

# **Desarrollo**

## **API Seleccionada:**

* SWAPI

## **URL Oficial:**

* <https://swapi.info>

## **Herramienta:**

* Postman

# **Endpoint #1:**

**URL:** https://swapi.info/api/films/:id

**Método HTTP:** GET

**Descripción:** Obtiene información sobre una película específica por su ID.

**Parámetros:**

ID: Número entero que identifica a la película (ej. 4 =A New Hope).

Respuesta y Esquema: 200 OK



# **Endpoint #2:**

**URL:** <https://swapi.info/people/:id>

**Método HTTP:** GET

**Descripción:** Obtiene información sobre un personaje específico por su ID.

**Parámetros:**

ID: Número entero que identifica al personaje (ej. 1 = Luke Skywalker).

Respuesta y Esquema: 200 OK

A screenshot of a computer screen

AI-generated content may be incorrect.

# **Endpoint #3:**

**URL:**  [https://swapi.info/api/planets/:id](https://swapi.info/people/:id)

**Método HTTP:** GET

**Descripción:** Obtiene información sobre un planeta específico por su ID.

## **Parámetros:**

ID: Número entero que identifica al planeta (ej. 23 = Tatooine).

Respuesta y Esquema: 200 OK



# **Endpoint #4:**

**URL:**  [https://swapi.info/api/species/:id](https://swapi.info/people/:id)

**Método HTTP:** GET

**Descripción:** Obtiene información sobre una especie específico por su ID.

## **Parámetros:**

ID: Número entero que identifica a la especie (ej. 180 = Human).

Respuesta y Esquema: 200 OK



# **Endpoint #5:**

**URL:**  [https://swapi.info/api/vehicles/:id](https://swapi.info/people/:id)

**Método HTTP:** GET

**Descripción:** Obtiene información sobre un vehículo específico por su ID.

## **Parámetros:**

ID: Modelo que identifica el vehículo (ej. Digger Crawler = Sand Crawler).

Respuesta y Esquema: 200 OK



# **Endpoint #6:**

**URL:**  [https://swapi.info/api/starships/:id](https://swapi.info/people/:id)

**Método HTTP:** GET

**Descripción:** Obtiene información sobre una nave especial específica por su ID.

## **Parámetros:**

ID: Modelo que identifica la nave especial (ej. CR90 corvette = CR90 corvette).

Respuesta y Esquema: 200 OK



# **Pruebas Funcionales**

**Descripción:**

Estas pruebas verifican que cada Endpoint de la API funcione correctamente bajo condiciones normales, es decir, cuando se envían solicitudes válidas con parámetros correctos.

**Importancia:**

Sin funcionalidad básica correcta, cualquier integración con la API fallará. Estas pruebas actúan como el primer filtro de calidad, asegurando que los consumidores puedan obtener los datos esperados.

# **Pruebas de Validación de Errores**

**Descripción:**

Este tipo de pruebas evalúa cómo responde la API ante entradas incorrectas, inválidas o fuera de rango, como IDs inexistentes, formatos incorrectos o parámetros no permitidos.

**Importancia:**

Una API robusta debe manejar malas entradas con elegancia, evitando caídas del sistema o respuestas ambiguas.

# **Pruebas de Contrato**

**Descripcion:**

Las pruebas de contrato verifican que la estructura, los tipos de datos y los campos de la respuesta de la API sean consistentes con lo definido en su documentación, incluso si el código de estado es 200.

**Importancia:**

Cambios no documentados en la estructura de la respuesta pueden romper aplicaciones que dependen de la API, incluso si esta "parece" funcionar.