**Requisitos:**

Java 11 versión 11.0.27

Maven versión 3.9.10

IDEA utilizado IntelliJ

Postman

**Ejecución del proyecto:**

Para ejecutar el proyecto se puede hacer desde el IDEA de IntelliJ o a través de líneas de comandos Maven:

Opción 1 desde IntelliJ:  
1. Abre el proyecto desde IntelliJ IDEA

2. asegúrese de tener JDK 11.0.27 instalado  
3. Diríjase al runner que desea ejecutar, están en la dirección: api-tests\ src test\\java\runners…  
4. Ejecute el runner que desee  
5. El reporte HTML lo encuentra en api-tests\target\karate-reports

Opción 2 con línea de comandos Maven:

mvn test -Dtest=ProductsRunner

mvn test -Dtest=CartRunner

mvn test -Dtest=UsersRunner

mvn test -Dtest=AuthRunner

**DETALLES DEL PROYECTO:**

**Módulo de productos:**  
casos positivos

Obtener todos los productos

Verifica que se retorne el listado de los productos y su estructura retornado por el endpoint.

Obtener detalle de un producto existente

Consulta la información correspondiente al id 1

Actualizar producto 1  
realiza la actualización de los datos del producto correspondiente al id 1

Actualizar producto 2  
igual que el producto anterior pero con el id 2

Casos negativos

**Consultar producto inexistente**  
se recibe un objeto vacío para el id del producto 99999 para confirmar un caso negativo

**Actualizar producto con datos inválidos**  
el api que se usa no tiene validaciones reales, por eso se ve 200 ok pero los datos son incorrectos, pero es un caso negativo, en un entorno real se esperaría un 400.

**Módulo de usuarios:**  
casos positivos:

**Crear nuevo usuario correctamente**  
se hace una solicitud post para enviar los datos de un usuario

**Actualizar un usuario existente**  
se envía una solicitud put para actualizar los datos del usuario

**Eliminar un usuario existente**  
se envía la solicitud delete para eliminar un usuario

Casos negativos:

**Crear usuario con datos incompletos**  
se envía una solicitud post para crear un usuario pero este con datos incompletos, se imprime la respuesta para validar comportamiento inesperado.

**Actualizar usuario inexistente**  
se envía solicitud put para actualizar datos de manera errónea, se imprime respuesta para observar el resultado.

**Eliminar usuario inexistente**  
se envía una solicitud delete para eliminar un usuario de manera errada y se valida al recibir un objeto vacío.

***Modulo del cart:***

Casos positivos:

**Obtener carrito del usuario 2**

Se envía una solicitud get para obtener la información del carrito 2, verifica que el primer elemento del array tenga id 2.

**Eliminar carrito existente**

Se envía una solicitud delete para eliminar un carrito diferente, confirma que el carrito con el id 5 fue eliminado correctamente.

Casos negativos:

**Obtener carrito de usuario inexistente**

Se envía una solicitud get para obtener la información de un carrito, verifica que el resultado sea un array vacío.

**Eliminar carrito inexistente**

Se envía una solicitud delete de un carrito diferente, verifica que la respuesta sea un objeto vacío ({}), indicando que no existe.

**Módulo de autenticación:**

Casos positivos:

**Autenticación exitosa**

Se envía una solicitud post para el logueo de usuario y contraseña, verifica que el servidor responda con el código de estado **200** y cuerpo que contiene un **token** de acceso.

Casos negativos:

**Autenticación fallida**

Igual que la solicitud anterior pero con datos incorrectos verifica que el servidor responda con el código de estado **401 y** autenticación denegada