Se espera como objetivo de este laboratorio que los alumnos obtengan una especificación OpenAPI válida y bien documentada que describa una API RESTful para gestionar un sistema de facturación y stock.

La especificación debe incluir detalles como los endpoints, métodos HTTP, parámetros de entrada y salida, códigos de respuesta, y ejemplos de datos de entrada y salida para cada endpoint.

Además, se espera que los alumnos utilicen las herramientas integradas de Swagger Editor para validar la especificación y realizar pruebas preliminares antes de exportarla para su uso posterior.

Detalles:
Realizar una especificación OpenAPI en formato YAML para diseñar y documentar una API RESTful para gestionar facturación y stock utilizando Swagger Editor.

En este ejemplo:
Se define una especificación OpenAPI versión 3.0.0.
Se especifica información general sobre la API, como el título, descripción y versión.
Se definen los endpoints los endpoints: /facturas, /facturas/{id} y /productos con sus correspondientes operaciones GET para obtener datos de facturas y productos.
Se especifican los esquemas de datos utilizando el componente schemas, incluyendo los esquemas de datos para las facturas y los productos.
Se documentan las respuestas esperadas para cada endpoint, como 200 (OK) para respuestas exitosas y por ejemplo 404 (Not Found) para facturas no encontradas.

Pasos:

1- ***Acceder a:*** https://editor.swagger.io/

2- Usar el Yaml base (dentro del swagger editor):

Puedes copiar y pegar este código en Swagger Editor para ver y probar la especificación.

```
openapi: 3.0.0
info:
  title: API de Sistema de Facturación y Stock
  description: API para gestionar facturas, productos, clientes, proveedores, inventario
y reportes.
  version: 1.0.0
servers:
  - url: localhost:8080/par2024

paths:
  /facturas:
    get:
      summary: Obtener todas las facturas
      responses:
        '200':
          description: Lista de facturas obtenida correctamente
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Factura'
  /facturas/{id}:
    get:
      summary: Obtener una factura por su ID
      parameters:
        - name: id
          in: path
```

```
          required: true
          description: ID único de la factura
          schema:
            type: integer
            format: int64
      responses:
        '200':
          description: Factura obtenida correctamente
        '404':
          description: Factura no encontrada
  /productos:
    get:
      summary: Obtener todos los productos en stock
      responses:
        '200':
          description: Lista de productos obtenida correctamente
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Producto'

components:
  schemas:
    Factura:
      type: object
      properties:
        id:
          type: integer
          format: int64
          description: ID único de la factura
        numero:
          type: string
          description: Número de la factura
        cliente:
          type: string
          description: Nombre del cliente
        total:
          type: number
          format: double
          description: Total de la factura
      required:
        - numero
        - cliente
        - total
        # Definición de otras propiedades de la factura
    Producto:
      type: object
      properties:
        id:
          type: integer
          format: int64
          description: ID único del producto
        nombre:
          type: string
          description: Nombre del producto
        precio:
          type: number
          format: double
          description: Precio del producto
        descripcion:
          type: string
          description: Descripción del producto
      required:
        - nombre
        - precio
```

Nota:
Como se puede apreciar el código base, define los siguientes endpoints API:

Facturas:
```
GET /facturas: Obtener todas las facturas.
GET /facturas/{id}: Obtener una factura por su ID.
```

Productos:
```
GET /productos: Obtener todos los productos en stock.
```

3- ***Utiliza el código base y define los siguientes endpoints API que podría tener su sistema adicionalmente, aquellas que considere necesarios.***

Facturas:
```
POST /facturas: Crear una nueva factura.
PUT /facturas/{id}: Actualizar una factura existente por su ID.
DELETE /facturas/{id}: Eliminar una factura existente por su ID.
```

Productos:
```
GET /productos/{id}: Obtener un producto por su ID.
POST /productos: Crear un nuevo producto en el stock.
PUT /productos/{id}: Actualizar un producto existente por su ID.
DELETE /productos/{id}: Eliminar un producto existente por su ID.
```

Clientes:
```
GET /clientes: Obtener todos los clientes registrados.
GET /clientes/{id}: Obtener un cliente por su ID.
POST /clientes: Crear un nuevo cliente.
PUT /clientes/{id}: Actualizar los datos de un cliente existente por su ID.
DELETE /clientes/{id}: Eliminar un cliente existente por su ID.
```

Proveedores:
```
GET /proveedores: Obtener todos los proveedores registrados.
GET /proveedores/{id}: Obtener un proveedor por su ID.
POST /proveedores: Crear un nuevo proveedor.
PUT /proveedores/{id}: Actualizar los datos de un proveedor existente por su
ID.
DELETE /proveedores/{id}: Eliminar un proveedor existente por su ID.
```

Importante:

Dentro del repositorio creado en GitHub, con tu usuario y la descripción: **par2025**
Dentro de **laboratorios/Lab02**
Levantar el archivo **lab02.yaml**, con las especificaciones dadas.
Permitir el acceso a: cnpalacios@pol.una.py.
Los códigos que no pueden ser probado, no serán considerados.

Entregas:

Las entregas de laboratorio, deberá realizarse a través de la plataforma EDUCA, mediante un Documento (formato independiente), donde se presenten las evidencias de las implementaciones, los comentarios que consideren apropiado y el enlace al repositorio para la evaluación de sus códigos resultantes.
Nota: Pueden utilizar **Plantilla de entrega - Lab02.docx** como documento base para la entrega de la actividad.