

# Open API

# Concepto

OpenAPI es una especificación para describir y documentar API RESTful.

Originalmente desarrollada por SmartBear como el proyecto Swagger, la especificación fue donada a la Linux Foundation y renombrada como [OpenAPI Specification](#) (OAS).

Permite a los desarrolladores definir la estructura y los comportamientos de sus API en un formato estandarizado, lo que facilita la colaboración, el desarrollo, y el mantenimiento de las API.



## Características

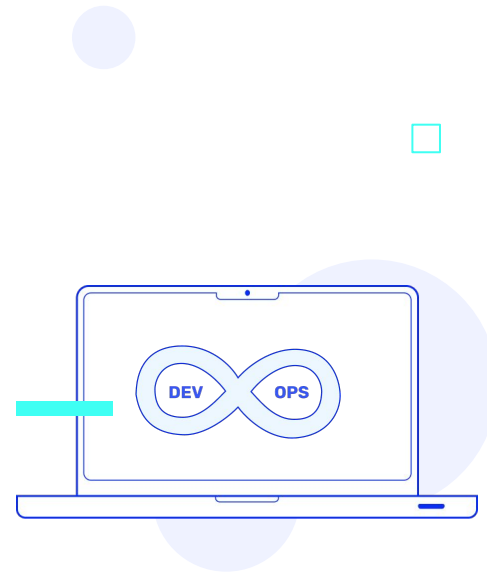
- **Descripción de API RESTful:** la especificación permite describir los *endpoints* (URL), los métodos HTTP (GET, POST, PUT, DELETE, etc.), los parámetros de entrada y salida, los tipos de respuesta, y los códigos de estado HTTP.
- **Formato Estándar:** OpenAPI utiliza un formato estándar (JSON o YAML) para describir una API. Este formato es legible tanto para los humanos como para las máquinas.
- **Documentación interactiva:** herramientas como Swagger UI y ReDoc pueden generar documentación interactiva directamente a partir de un documento OpenAPI, permitiendo a los usuarios explorar y probar la API desde su navegador.



- **Generación de código:** permite la generación de código para clientes y servidores en varios lenguajes de programación. Herramientas como Swagger Codegen y OpenAPI Generator pueden generar stubs de servidor, SDK de cliente, y documentación.
- **Validación y testing:** la especificación facilita la validación de los *endpoints* de una API y las pruebas automatizadas, asegurando que la API se comporte conforme a su definición.
- **Versionado y evolución:** OpenAPI soporta el versionado de API, permitiendo a los desarrolladores describir diferentes versiones de una API y gestionar su evolución de manera ordenada.



- **Integración con DevOps:** OpenAPI puede integrarse en pipelines de CI/CD para automatizar la generación de documentación, la validación de contratos de API y la generación de código, mejorando el flujo de trabajo de DevOps.
- **Compatibilidad con *middleware*:** muchas plataformas de middleware y gateways API (como Kong, Apigee, y AWS API Gateway) pueden consumir documentos OpenAPI para configurar automáticamente rutas, políticas de seguridad, y transformaciones de datos.



## Especificación de una API utilizando OpenAPI 3.0

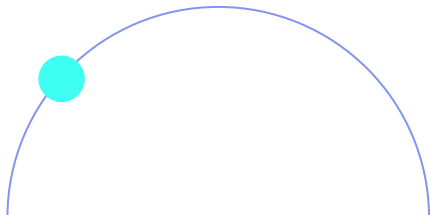
```
openapi: 3.0.0
info:
  title: API de Ejemplo
  description: Documentación de la API de ejemplo
  version: 1.0.0
servers:
  - url: http://localhost:8080
paths:
  /productos:
    get:
      summary: Obtener lista de productos
      responses:
        '200':
          description: Lista de productos
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Producto'
```



```
post:
  summary: Crear un nuevo producto
  requestBody:
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Producto'
  responses:
    '201':
      description: Producto creado
components:
  schemas:
    Producto:
      type: object
      properties:
        id:
          type: integer
          format: int64
        nombre:
          type: string
        precio:
          type: number
          format: double
        descripcion:
          type: string
        url_foto:
          type: string
```

## Explicación del código

- **openapi: 3.0.0:** especifica la versión de OpenAPI que se está utilizando (en este caso, la versión 3.0.0).
- **info:** contiene metadatos sobre la API, como el título, la descripción y la versión.
- **servers:** define la URL base del servidor donde la API está disponible.
- **paths:** define los *endpoints* de la API. En este caso, /productos tiene dos operaciones:
  - **GET:** para obtener la lista de productos.
  - **POST:** para crear un nuevo producto.
- **components:** define los esquemas de los objetos, en este caso, el esquema Producto con propiedades como id, nombre, precio, descripción, y url\_foto.





## Conclusión

OpenAPI es una poderosa herramienta para definir, documentar y gestionar APIs RESTful.

Sus características estándar y su ecosistema de herramientas asociadas hacen que el desarrollo y mantenimiento de APIs sean más eficientes y efectivos, promoviendo una mejor colaboración entre desarrolladores y facilitando la integración con diversas plataformas y servicios.

