

# Guía para Crear un Archivo OpenAPI (.yaml)

---

## 1. ¿Qué es un archivo OpenAPI?

Un archivo OpenAPI (anteriormente Swagger) es un documento estructurado en formato YAML o JSON que describe de forma estandarizada una API REST. Permite definir endpoints, operaciones, parámetros, respuestas, esquemas de datos y más. Es utilizado para generar documentación, validaciones, pruebas automáticas y SDKs.

## 2. Estructura Básica del Archivo

*Toda especificación OpenAPI comienza con una cabecera básica:*

---

```
openapi: 3.0.3
info:
  title: Nombre de tu API
  version: 1.0.0
  description: Descripción breve de la API
servers:
  - url: http://localhost:3000/api/v1
```

## 3. Definición de Rutas (paths)

*Cada ruta (endpoint) se define en la sección `paths`. Por ejemplo, para definir `/book` con métodos GET y POST:*

---

```
paths:
  /book:
    get:
      summary: Obtener todos los libros
    responses:
```

```
'200':
  description: Lista de libros
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Books'
post:
  summary: Añadir un nuevo libro
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Book'
responses:
  '200':
    description: Libro creado
```

## 4. Parámetros en la URL

*Para rutas como `/book/{bookId}`, se define un parámetro de ruta en `components.parameters` y se referencia en `paths`.*

---

```
parameters:
  bookId:
    name: bookId
    in: path
    required: true
    schema:
      type: string
```

## 5. Componentes (schemas)

*La sección `components.schemas` define los modelos de datos reutilizables. Ejemplo de un esquema para un libro:*

---

```
components:
  schemas:
    Book:
      type: object
      properties:
        _id:
          type: string
        title:
          type: string
        author:
          type: string
```

## 6. Uso de Referencias (\$ref)

Para evitar duplicación, se usa `\$ref` para reutilizar estructuras definidas en `components`, como:

```
$ref: '#/components/schemas/Book'
```

## 7. Buenas Prácticas

- Usa OpenAPI 3.0.3 o superior.
- Define claramente las respuestas con códigos HTTP.
- Utiliza `\$ref` para mantener el archivo limpio.
- Documenta todos los endpoints y parámetros.
- Valida el archivo con herramientas como Swagger Editor.