

# API TESTING

- [INTRODUCCIÓN A API-TESTING](#)

# INTRODUCCIÓN A API- TESTING

*Overskull*

Introducción a la automatización

05/02/2026

Versión 0.1

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito a overskull.

**ÍNDICE**

## 1.INTRODUCCIÓN

### 1.1 Propósito

### 1.2 Objetivo

## 2.CONCEPTOS GENERALES DE API

### 2.1 ¿Qué es una API?

### 2.2 Métodos HTTP más utilizados

### 2.3 Estructura de una petición

## 3.IMPORTANCIA DEL API TESTING

## 4.TIPOS DE PRUEBAS EN APIs

### 4.1 Pruebas Funcionales

### 4.2 Pruebas de Validación de Datos

### 4.3 Pruebas de Seguridad

### 4.4 Pruebas de Rendimiento

## 5.HERRAMIENTAS UTILIZADAS

## 6.PROCEDIMIENTO GENERAL DE PRUEBA

## 7.VALIDACIONES CLAVE

## 8.BUENAS PRÁCTICAS

## 9.ENFOQUE FINAL

## 10.Diagrama de Flujo

# 1. INTRODUCCIÓN

## 1.1 Propósito

Este documento describe los **fundamentos del API Testing**, su importancia dentro del proceso de **calidad de software** y la forma correcta de validar servicios mediante pruebas técnicas. Sirve como guía para comprender cómo se prueban las integraciones entre sistemas sin necesidad de utilizar la interfaz gráfica.

## 1.2 Objetivo

Establecer lineamientos claros para ejecutar pruebas sobre APIs, asegurando que los servicios funcionen correctamente, respondan con datos válidos y mantengan la integridad de la información intercambiada entre sistemas.

## 2. CONCEPTOS GENERALES DE API

## 2.1 ¿Qué es una API?

Una API (Interfaz de Programación de Aplicaciones) permite que diferentes sistemas se comuniquen entre sí mediante solicitudes y respuestas estructuradas.

## 2.2 Métodos HTTP más utilizados

Método	Función
GET	Obtener información
POST	Crear registros
PUT	Actualizar datos
DELETE	Eliminar registros

## 2.3 Estructura de una petición

Una solicitud API generalmente contiene:

1. Endpoint: URL del servicio.
2. Método HTTP
3. Headers: Información adicional (tokens, tipo de contenido, etc.)
4. Body: Datos enviados (cuando aplica)

## 3. IMPORTANCIA DEL API TESTING

El API Testing permite:

1. Detectar errores antes de que lleguen a la interfaz de usuario
2. Validar reglas de negocio internas
3. Probar integraciones entre sistemas
4. Reducir fallos en producción
5. Asegurar estabilidad y rendimiento del servicio

Es una de las capas más críticas en la validación de software moderno.

## 4. TIPOS DE PRUEBAS EN APIS

### 4.1 Pruebas Funcionales

- Validan que la API cumpla su función principal.

Ejemplo:

- Crear un registro y verificar que se guarde correctamente.

#### **4.2 Pruebas de Validación de Datos**

- Verifican que el sistema maneje correctamente datos válidos e inválidos.

Ejemplo:

- Enviar campos vacíos
- Enviar datos con formato incorrecto

#### **4.3 Pruebas de Seguridad**

- Aseguran que solo usuarios autorizados puedan acceder.

Ejemplo:

- Probar tokens vencidos
- Probar sin autenticación

#### **4.4 Pruebas de Rendimiento**

- Evalúan cómo responde la API bajo carga o múltiples solicitudes.

### **5. HERRAMIENTAS UTILIZADAS**

Para realizar API Testing se utilizan herramientas como:

1. Postman – Pruebas manuales y automatizadas
2. Swagger – Documentación y prueba de endpoints
3. JMeter – Pruebas de rendimiento
4. Automatización con scripts (según proyecto)

### **6. PROCEDIMIENTO GENERAL DE PRUEBA**

1. Revisar la documentación del endpoint
2. Identificar método, parámetros y respuesta esperada
3. Preparar la solicitud en la herramienta de prueba
4. Enviar la petición

Validar:

1. Código de estado (200, 400, 500, etc.)
2. Estructura de la respuesta
3. Datos devueltos
4. Registrar evidencias de la prueba

## **7. VALIDACIONES CLAVE**

Durante una prueba API siempre se debe verificar:

1. Código de respuesta correcto
2. Mensajes de error claros
3. Integridad de datos
4. Cumplimiento de reglas de negocio
5. Seguridad de acceso

## **8. BUENAS PRÁCTICAS**

1. Probar escenarios positivos y negativos
2. No asumir que la API siempre devuelve datos correctos
3. Validar también casos límite
4. Documentar cada prueba ejecutada
5. Mantener organizadas las colecciones de pruebas

## **9. ENFOQUE FINAL**

El API Testing no solo busca verificar que un servicio responda, sino garantizar que:

1. Los datos sean correctos
2. Las reglas del sistema se respeten
3. La seguridad se mantenga
4. Las integraciones funcionen sin fallos
5. Un API estable significa un sistema confiable.

10. Diagrama de Flujo:

