

# Guia de Postman para API Testing

Aprende a probar APIs desde cero con Postman

Guia completa para QA Testers

# API

Mundo QA - [mundoqa.com](https://mundoqa.com)

Nivel: Basico a Intermedio

Autor: Jorge Andres Escorcía | QA Engineer

[contacto@mundoqa.com](mailto:contacto@mundoqa.com)

# Contenido

#	Seccion	Pag.
1	Que es Postman y para que sirve	3
2	Instalacion y configuracion inicial	3
3	La interfaz de Postman explicada	4
4	Tu primera peticion HTTP	4
5	Metodos HTTP en API Testing	5
6	Parametros, Headers y Body	6
7	Autenticacion en APIs	7
8	Variables de entorno y colecciones	8
9	Tests automatizados en Postman	9
10	Pre-request Scripts	10
11	Flujo de trabajo QA con Postman	11
12	Glosario de terminos	12

## 1. Que es Postman y para que sirve

Postman es una plataforma de colaboracion para el desarrollo y prueba de APIs. Es la herramienta mas utilizada por QA Engineers y desarrolladores para enviar peticiones HTTP, analizar respuestas, automatizar pruebas y documentar APIs.

### Por que Postman es esencial para un QA?

- Permite probar APIs sin necesidad de escribir codigo complejo
- Organiza las peticiones en colecciones reutilizables
- Automatiza validaciones con scripts de prueba
- Simula diferentes escenarios (exito, error, limites)
- Se integra con pipelines CI/CD (Jenkins, GitHub Actions)
- Genera documentacion automatica de la API

**Tip:** Postman tiene una version gratuita mas que suficiente para comenzar en QA. Descargalo en [postman.com](https://postman.com)

## 2. Instalacion y configuracion inicial

## Pasos de instalacion

- Ve a postman.com y haz clic en 'Download the App'
- Selecciona tu sistema operativo (Windows, Mac o Linux)
- Ejecuta el instalador y sigue los pasos
- Crea una cuenta gratuita (opcional pero recomendado para sincronizar)
- Al abrir, selecciona 'Skip and go to the app' si no quieres registrarte aun

## Configuracion recomendada para QA

Configuracion	Valor recomendado	Donde encontrarlo
SSL verification	OFF (en ambientes QA)	Settings > General
Request timeout	30000 ms	Settings > General
Automatically follow redirects	ON	Settings > General
Send no-cache header	ON	Settings > General
Language detection	JSON	Settings > General

**Atencion:** Nunca desactives SSL verification en ambientes de produccion. Solo usalo en QA o desarrollo.

### 3. La interfaz de Postman explicada

Postman tiene varias areas clave que debes conocer:

Area	Descripcion	Uso principal
Sidebar (izquierda)	Colecciones, Entornos, APIs guardadas	Organizar y acceder a tus peticiones
Request Builder	URL, metodo, headers, body, params	Construir y configurar cada peticion
Response Panel	Status, tiempo, tamaño, body, headers	Analizar la respuesta del servidor
Tests Tab	Editor de scripts de prueba en JavaScript	Automatizar validaciones
Console	Log de todas las peticiones y errores	Depurar problemas en las llamadas
Environment Selector	Menu desplegable arriba a la derecha	Cambiar entre ambientes (QA/Staging/Prod)

**Tip:** Usa Ctrl+Alt+C para abrir la consola de Postman. Es tu mejor aliado para depurar peticiones.

### 4. Tu primera peticion HTTP

Vamos a hacer tu primera peticion usando la API publica de prueba JSONPlaceholder, ideal para practicar sin necesidad de un backend propio.

#### Paso a paso

- Haz clic en el boton '+' para abrir una nueva pestana de peticion
- Selecciona el metodo GET en el menu desplegable
- Escribe la URL: <https://jsonplaceholder.typicode.com/posts/1>
- Haz clic en el boton azul 'Send'
- Observa la respuesta en el panel inferior

#### Que debes observar en la respuesta

Elemento	Descripcion	Valor esperado
Status Code	Codigo de estado HTTP	200 OK
Time	Tiempo de respuesta	Menos de 500ms
Size	Tamaño de la respuesta	Visible en bytes
Body	Contenido de la respuesta	JSON con id, title, body, userId
Headers	Cabeceras de respuesta	Content-Type: application/json

**Nota:** JSONPlaceholder (jsonplaceholder.typicode.com) es una API publica y gratuita. Perfecta para practicar.

## 5. Metodos HTTP en API Testing

Los metodos HTTP definen la accion que quieres realizar sobre un recurso. Como QA debes conocer los 5 principales y cuando usar cada uno.

Metodo	Accion	Ejemplo de uso	Codigo exitoso
GET	Obtener un recurso	Listar usuarios, obtener un pedido	200 OK
POST	Crear un recurso	Registrar usuario, crear ticket	201 Created
PUT	Actualizar completo	Reemplazar todos los datos de un usuario	200 OK
PATCH	Actualizar parcial	Cambiar solo el email de un usuario	200 OK
DELETE	Eliminar recurso	Borrar un comentario, cancelar sesion	200 / 204

### Codigos de estado HTTP mas importantes

Codigo	Nombre	Cuando aparece
200	OK	Peticion exitosa (GET, PUT, PATCH, DELETE)
201	Created	Recurso creado exitosamente (POST)
204	No Content	Exito sin cuerpo de respuesta (DELETE)
400	Bad Request	Datos enviados incorrectos o mal formados
401	Unauthorized	Falta autenticacion o token invalido
403	Forbidden	Sin permisos para este recurso
404	Not Found	El recurso no existe
409	Conflict	Conflicto (ej: email duplicado)
422	Unprocessable Entity	Validacion fallida del servidor
500	Internal Server Error	Error en el servidor backend
503	Service Unavailable	Servidor caido o sobrecargado

**Tip:** Como QA, debes probar tanto los casos exitosos (happy path) como los casos de error. No te limites solo al 200 OK.

## 6. Parametros, Headers y Body

## 6.1 Query Parameters

Los query parameters se agregan al final de la URL despues del signo '?'. Se usan para filtrar, paginar o modificar el resultado de un GET.

```
GET https://api.ejemplo.com/usuarios?pagina=1&limite;=10&activo;=true
```

En Postman puedes agregarlos en la pestana **Params** sin escribirlos manualmente en la URL.

## 6.2 Headers

Los headers envian informacion adicional junto a la peticion. Los mas comunes son:

Header	Valor tipico	Para que sirve
Content-Type	application/json	Indica el formato del cuerpo enviado
Accept	application/json	Indica que formato espera la respuesta
Authorization	Bearer {token}	Envia credenciales de acceso
X-API-Key	{tu-api-key}	Clave de acceso a la API
Cache-Control	no-cache	Evita usar cache en la respuesta

## 6.3 Body (cuerpo de la peticion)

El body se usa en POST, PUT y PATCH para enviar datos al servidor. El formato mas comun es JSON. En Postman selecciona Body > raw > JSON:

```
{ "nombre": "Juan Perez", "email": "juan@ejemplo.com", "rol": "admin" }
```

Formato Body	Cuando usarlo
raw > JSON	APIs REST modernas (el mas comun en QA)
form-data	Subida de archivos e imagenes
x-www-form-urlencoded	Formularios HTML tradicionales
binary	Archivos en binario (PDFs, imagenes)
GraphQL	APIs con lenguaje de consulta GraphQL

## 7. Autenticacion en APIs

La mayoría de APIs requieren autenticacion. Postman soporta todos los tipos principales desde la pestana Authorization de cada peticion.

Tipo	Como funciona	Donde configurar en Postman
No Auth	API publica sin restriccion	Auth > No Auth
API Key	Clave en header o query param	Auth > API Key
Bearer Token	Token JWT en el header Authorization	Auth > Bearer Token
Basic Auth	Usuario y contraseña en Base64	Auth > Basic Auth
OAuth 2.0	Token generado por flujo OAuth	Auth > OAuth 2.0
Digest Auth	Similar a Basic con hash MD5	Auth > Digest Auth

### Ejemplo: Autenticacion con Bearer Token

Asi se ve en el header de la peticion una vez configurado:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

### Que probar como QA en la autenticacion?

- Peticion con token valido - debe retornar 200 OK
- Peticion sin token - debe retornar 401 Unauthorized
- Peticion con token expirado - debe retornar 401 con mensaje claro
- Peticion con token de otro usuario - debe retornar 403 Forbidden
- Peticion con token malformado - debe retornar 400 o 401
- Peticion con token de rol sin permisos - debe retornar 403 Forbidden

**Tip:** Guarda tu token como variable de entorno en Postman para no tener que pegarlo en cada peticion.

## 8. Variables de entorno y colecciones

### 8.1 Variables de entorno

Las variables de entorno permiten cambiar valores facilmente segun el ambiente (QA, Staging, Produccion) sin modificar cada peticion.

- Crea un entorno: clic en el icono de ojo arriba a la derecha > Add
- Define variables: base\_url, token, usuario\_id, etc.
- Usalas en la URL y en el body con dobles llaves: `{{base_url}}/api/usuarios`

```
GET {{base_url}}/api/usuarios/{{usuario_id}}
```

Variable	Ambiente QA	Ambiente Prod
base_url	https://qa.miapp.com	https://miapp.com
token	eyJhbGci... (token QA)	eyJhbGci... (token Prod)
usuario_id	123	456
timeout	30000	10000

## 8.2 Colecciones

Una coleccion es un grupo de peticiones relacionadas organizadas en carpetas. Es la forma correcta de organizar el trabajo de API Testing en QA.

- Clic en 'New' > Collection para crear una
- Agrega carpetas por modulo: /Auth, /Usuarios, /Pedidos, /Productos
- Guarda cada peticion en la carpeta correspondiente
- Comparte la coleccion con tu equipo exportandola como JSON
- Ejecuta toda la coleccion con Collection Runner para pruebas de regresion

**Nota:** Nombra tus peticiones con convencion clara: [METODO] Descripcion. Ejemplo: GET Obtener lista de usuarios

## 9. Tests automatizados en Postman

La pestana Tests de Postman permite escribir assertions en JavaScript que se ejecutan automaticamente despues de cada peticion. Es donde el QA agrega valor real al API Testing.

### Estructura basica de un test

```
pm.test("Nombre descriptivo del test", function() { // Tu assertion va aqui
  pm.response.to.have.status(200); });
```

### Tests esenciales que todo QA debe escribir

Que validar	Codigo del test
Status code	<code>pm.test("Status 200", () =&gt; pm.response.to.have.status(200));</code>
Tiempo respuesta	<code>pm.test("Menor 500ms", () =&gt; pm.expect(pm.response.responseTime).to.be.below(500));</code>
Tipo de contenido	<code>pm.test("Es JSON", () =&gt; pm.response.to.have.header("Content-Type", /json/));</code>
Campo en body	<code>pm.test("Tiene id", () =&gt; pm.expect(pm.response.json()).to.have.property("id"));</code>
Valor exacto	<code>pm.test("Email correcto", () =&gt; pm.expect(pm.response.json().email).to.eql("user@test.com"));</code>
Array no vacio	<code>pm.test("Lista no vacia", () =&gt; pm.expect(pm.response.json()).to.be.an("array").that.is.not.empty);</code>

### Test completo de ejemplo: Login exitoso

```
pm.test("Status 200 OK", function() { pm.response.to.have.status(200); });
pm.test("Respuesta contiene token", function() { var jsonData = pm.response.json();
  pm.expect(jsonData).to.have.property("token");
  pm.expect(jsonData.token).to.be.a("string").and.not.empty; }); pm.test("Tiempo de
  respuesta aceptable", function() {
  pm.expect(pm.response.responseTime).to.be.below(1000); }); // Guardar token para
  siguiente peticion pm.environment.set("token", pm.response.json().token);
```

**Tip:** Usa `pm.environment.set()` para pasar datos entre peticiones. Por ejemplo: guarda el token del login y usalo en las demas peticiones.

## 10. Pre-request Scripts

Los Pre-request Scripts se ejecutan ANTES de enviar la peticion. Son utiles para preparar datos dinamicos, generar tokens o calcular valores.

### Casos de uso comunes

- Generar una fecha o timestamp dinamico
- Crear un ID unico para cada peticion de prueba

- Calcular una firma HMAC para APIs que la requieran
- Hacer login automáticamente antes de peticiones protegidas
- Generar datos aleatorios para pruebas

## Ejemplo: Generar datos dinamicos

```
// Generar email unico para cada prueba var timestamp = Date.now();
pm.environment.set("email_unico", "qa_" + timestamp + "@test.com"); // Fecha de hoy en
formato ISO var hoy = new Date().toISOString().split("T")[0];
pm.environment.set("fecha_hoy", hoy); // Numero aleatorio entre 1 y 100 var aleatorio =
Math.floor(Math.random() * 100) + 1; pm.environment.set("id_aleatorio", aleatorio);
```

**Nota:** Postman tiene variables dinamicas integradas como `{{randomEmail}}`, `{{timestamp}}` y `{{guid}}` que puedes usar directamente en la URL o el body sin escribir scripts.

## 11. Flujo de trabajo QA con Postman

Este es el flujo de trabajo recomendado para integrar Postman en el ciclo de pruebas de un equipo QA profesional.

Fase	Actividad	Resultado
1. Analisis	Revisar la documentacion de la API (Swagger/OpenAPI) e endpoints a probar	Lista de endpoints a probar
2. Diseno	Crear coleccion con carpetas por modulo	Estructura organizada
3. Variables	Configurar entornos (QA, Staging)	Variables listas
4. Peticiones	Crear y guardar cada peticion con datos de prueba	Peticiones configuradas
5. Tests	Escribir assertions en la pestana Tests	Validaciones automaticas
6. Ejecucion	Correr con Collection Runner	Reporte de resultados
7. Regresion	Ejecutar coleccion en cada deploy	Deteccion rapida de regresiones
8. CI/CD	Integrar con Newman en pipeline	Testing automatico en CI

### Ejecutar coleccion con Newman (linea de comandos)

Newman es la version de linea de comandos de Postman. Permite ejecutar colecciones en pipelines de CI/CD sin abrir la aplicacion.

```
# Instalar Newman npm install -g newman # Ejecutar coleccion con entorno newman run mi_coleccion.json -e ambiente_qa.json # Generar reporte HTML newman run mi_coleccion.json -e ambiente_qa.json --reporters html --reporter-html-export reporte.html
```

**Tip:** Exporta tu coleccion desde Postman con clic derecho > Export y compartela en el repositorio de tu proyecto para que todo el equipo la use.

## 12. Glosario de terminos

Termino	Definicion
API	Application Programming Interface. Interfaz que permite la comunicacion entre sistemas
REST	Architectural style para APIs que usa HTTP como protocolo de comunicacion
Endpoint	URL especifica de la API que representa un recurso o accion
Request	Peticion enviada al servidor (URL + metodo + headers + body)
Response	Respuesta del servidor a una peticion (status + headers + body)
JSON	JavaScript Object Notation. Formato de datos mas usado en APIs REST
Status Code	Numero de 3 digitos que indica el resultado de la peticion HTTP
Token JWT	JSON Web Token. Credencial encriptada usada para autentificacion en APIs
Header	Metadato enviado con la peticion o respuesta (informacion adicional)
Body	Contenido o datos enviados en el cuerpo de la peticion
Collection	Grupo de peticiones organizadas en Postman
Environment	Conjunto de variables configuradas para un ambiente especifico
Assertion	Validacion que verifica que la respuesta cumple un criterio esperado
Newman	CLI de Postman para ejecutar colecciones desde la terminal
Mock	Servidor simulado que devuelve respuestas predefinidas sin backend real
Swagger	Herramienta para documentar y probar APIs (ahora llamada OpenAPI)

### Continua aprendiendo en Mundo QA

Esta guia cubre los fundamentos de Postman para API Testing. En Mundo QA encontraras tutoriales en video, ejercicios practicos y mas guias como esta para seguir creciendo como QA Tester.

*mundoqa.com | contacto@mundoqa.com*