

## Casos de prueba funcionamiento de endpoints

### Caso 1: Consultar productos de la categoría electronics

**Objetivo:** Validar que el sistema devuelva únicamente los productos de la categoría solicitada.

**Precondiciones:** La API está disponible.

**Pasos:**

1. Enviar una solicitud para listar la categoría “electronics”
2. Revisar que la respuesta incluya productos.
3. Verificar que todos los productos tengan la categoría correcta.

**Resultado esperado:**

La API devuelve una lista de productos de la categoría “electronics”, con sus campos principales.

### Caso 2: Consultar un producto específico

**Objetivo:** Validar que el sistema devuelva la información de un producto por su ID.

**Precondiciones:** La API está disponible.

**Pasos:**

1. Consultar un producto por su ID.
2. Verificar que la respuesta incluya los campos básicos (id, título, precio, descripción).

**Resultado esperado:**

La API devuelve el producto correcto según el ID solicitado.

### Caso 3: Crear un producto

**Objetivo:** Validar que el sistema permita registrar un nuevo producto.

**Precondiciones:** La API está disponible.

**Pasos:**

1. Enviar una solicitud de creación con título, precio, descripción, categoría e imagen.
2. Verificar que se devuelva un identificador único para el producto.

**Resultado esperado:**

La API devuelve el producto creado junto con un id.

Nota: en esta API mock, el producto no queda guardado para consultas posteriores.

#### **Caso 4: Actualizar la imagen de un producto creado**

**Objetivo:** Validar que el sistema permita modificar la información de un producto existente.

**Precondiciones:** Haber creado previamente un producto.

**Pasos:**

1. Actualizar la imagen del producto creado.
2. Verificar que la respuesta incluya el mismo id y la nueva imagen.

**Resultado esperado:**

La API devuelve el producto con la imagen actualizada.

Nota: en esta API mock, el cambio no queda persistido para futuras consultas.

## **Informe de Pruebas de Carga y Estrés**

### **Prueba de carga (150 usuarios, 2 min)**

**Configuración:** 150 usuarios constantes, GET y POST simultáneos.

**Resultados:**

- Promedio: 274 ms
- p95: 435 ms
- Throughput: 192 req/s
- Errores: 0%

**Conclusión:** La API maneja 150 usuarios concurrentes de forma estable y rápida.

### **Prueba de estrés (100 → 1000 usuarios en rampas de 150)**

**Configuración:** incremento cada minuto hasta 1000 usuarios.

**Resultados:**

- Promedio: 512 ms
- p95: 1.35 s (sobrepasa el umbral de 1.2s)

- Throughput: 536 req/s
- Errores: 0%

**Conclusión:** La API soporta hasta 1000 usuarios sin fallos, pero el tiempo de respuesta se degrada al superar los 850 VUs.

**Comparativa de rendimiento**

Prueba	Usuarios	Req/s	Promedio	p95	Errores
Carga	150	192	274 ms	435 ms	0%
Estrés	1000	536	512 ms	1.35 s	0%

**Conclusiones Generales**

Los endpoints de la API funcionan correctamente (GET, POST, PUT).

Los casos negativos se manejan con respuestas vacías en vez de errores (limitación del mock).

La API soporta carga concurrente sin errores, pero el rendimiento comienza a degradarse a partir de 850 usuarios.

```
Microsoft Windows [Versión 10.0.19045.6216]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32\cmd.exe
C:\Users\Andres\OneDrive\Desktop>run load-test-150vus-2m.js

  Grafana
  MR6

execution: local
script: load-test-150vus-2m.js
output: -

scenarios: (100.00%) 1 scenario, 150 max VUs, 2m30s max duration (incl. graceful stop):
  * my_test: 150 looping VUs for 2m0s (gracefulStop: 30s)

THRESHOLDS
http_req_duration
  [p(95)<800] p(95)=513.64ms
http_req_failed
  [rate<0.02] rate=0.00%

TOTAL RESULTS
checks_total.....: 47064 387.524242/s
checks_succeeded...: 100.00% 47064 out of 47064
checks_failed.....: 0.00% 0 out of 47064

GET /products status is 200
GET /products array
POST /products status 200/201
POST /products id

HTTP
http_req_duration.....: avg=266.28ms min=194.28ms med=219.1ms max=867.72ms p(90)=432.77ms p(95)=513.64ms
{ expected_response:true }...: avg=266.28ms min=194.28ms med=219.1ms max=867.72ms p(90)=432.77ms p(95)=513.64ms
http_req_failed.....: 0.00% 0 out of 23532
http_reqs.....: 23532 193.762121/s

EXECUTION
iteration_duration.....: avg=1.53s min=1.39s med=1.45s max=3.22s p(90)=1.78s p(95)=1.82s
iterations.....: 11765 96.88106/s
VUs.....: 60 min=60 max=150
```

```
C:\Windows\System32\cmd.exe
scenarios: (100.00%) 1 scenario, 150 max VUs, 2m30s max duration (incl. graceful stop):
  * my_test: 150 looping VUs for 2m0s (gracefulStop: 30s)

THRESHOLDS
http_req_duration
  * 'p(95)<800' p(95)=513.64ms
http_req_failed
  * 'rate<0.02' rate=0.00%

TOTAL RESULTS
checks_total.....: 47064 387.524242/s
checks_succeeded...: 100.00% 47064 out of 47064
checks_failed.....: 0.00% 0 out of 47064
  * GET /products status is 200
  * GET /products array
  * POST /products status 200/201
  * POST /products id

HTTP
http_req_duration.....: avg=266.28ms min=194.24ms med=219.1ms max=867.72ms p(90)=432.77ms p(95)=513.64ms
  { expected_response:true }...: avg=266.28ms min=194.24ms med=219.1ms max=867.72ms p(90)=432.77ms p(95)=513.64ms
http_req_failed.....: 0.00% 0 out of 23532
http_reqs.....: 23532 193.762121/s

EXECUTION
iteration_duration.....: avg=1.53s min=1.39s med=1.45s max=3.22s p(90)=1.76s p(95)=1.82s
iterations.....: 11766 96.88106/s
vus.....: 69 min=69 max=150
vus_max.....: 150 min=150 max=150

NETWORK
data_received.....: 136 MB 1.1 MB/s
data_sent.....: 3.7 MB 31 KB/s

running (2m01.4s), 000/150 VUs, 11766 complete and 0 interrupted iterations
my_test [=====] 150 VUs 2m0s

C:\Andres\K6>
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versi3n 10.0.19045.6216]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Andres\K6>run stress-test-ramp.js

      Grafana
    /  |  \
   /   |   \
  /____|___\

execution: local
  script: stress-test-ramp.js
  output: -

scenarios: (100.00%) 1 scenario, 1000 max VUs, 7m30s max duration (incl. graceful stop):
  * ramping_test: Up to 1000 looping VUs for 7m0s over 7 stages (gracefulRampDown: 30s, gracefulStop: 30s)

[0289] Request Failed error="Get \"https://fakestoreapi.com/products\": dial tcp 104.21.64.1:443: connect: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond."

THRESHOLDS
http_req_duration
  * 'p(95)<1200' p(95)=776.43ms
http_req_failed
  * 'rate<0.05' rate=0.00%

TOTAL RESULTS
checks_total.....: 257944 612.249233/s
checks_succeeded...: 99.90% 257943 out of 257944
checks_failed.....: 0.00% 1 out of 257944
  * GET status 200
  * GET - 0 128971 / 0 1
  * POST status 200/201

HTTP
http_req_duration.....: avg=384.85ms min=6ms med=114.1ms max=9.18s p(90)=571.93ms p(95)=776.43ms
  { expected_response:true }...: avg=384.85ms min=192.51ms med=114.1ms max=9.18s p(90)=571.93ms p(95)=776.43ms
http_req_failed.....: 0.00% 1 out of 257944
http_reqs.....: 257944 612.249233/s

EXECUTION
iteration_duration.....: avg=1.77s min=1.39s med=1.65s max=24.31s p(90)=2.21s p(95)=2.62s
iterations.....: 128972 986.126517/s
```

```
C:\Windows\System32\cmd.exe
error="get \"https://fakestoreapi.com/products\": dial tcp 104.21.64.1:443: connect: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond."

THRESHOLDS
http_req_duration
  [p(95)<1200] p(95)=776.43ms
http_req_failed
  [rate<0.05] rate=0.00%

TOTAL RESULTS
checks_total.....: 257944 612.249233/s
checks_succeeded....: 99.99% 257941 out of 257944
checks_failed.....: 0.00% 1 out of 257944
  [GET status 200]
  [POST status 200/201]

HTTP
http_req_duration.....: avg=384.85ms min=0s med=114.1ms max=0.18s p(90)=571.93ms p(95)=776.43ms
  { expected_response:true }....: avg=384.86ms min=192.53ms med=114.1ms max=0.18s p(90)=571.93ms p(95)=776.43ms
http_req_failed.....: 0.00% 1 out of 257944
http_reqs.....: 257944 612.249233/s

EXECUTION
iteration_duration.....: avg=1.77s min=1.39s med=1.65s max=24.31s p(90)=2.21s p(95)=2.62s
iterations.....: 128972 386.124617/s
vus.....: 1 min=1 max=999
vus_max.....: 1000 min=1000 max=1000

NETWORK
data_received.....: 1.5 GB 3.5 MB/s
data_sent.....: 41 MB 96 KB/s

Running (7m01.3s), 0000/1000 VUs, 128972 complete and 0 interrupted iterations
Sampling test [=====] 0000/1000 VUs 7m0s
C:\Andres\K6>
```