Prueba técnica William Nieto

Casos de prueba

1. Listar productos electrónicos
    a. Endpoint: /products/category/electronics
    b. Método: GET
    c. Resultado esperado: Status 200, validación por JSON con productos categoría electronics, propiedades del id y títulos presentes.

2. Consulta de producto específico
    a. Endpoint: /products/1
    b. Método: GET
    c. Resultado esperado: Status 200, validación por JSON con producto, propiedades, id, title, price, category, description, image.

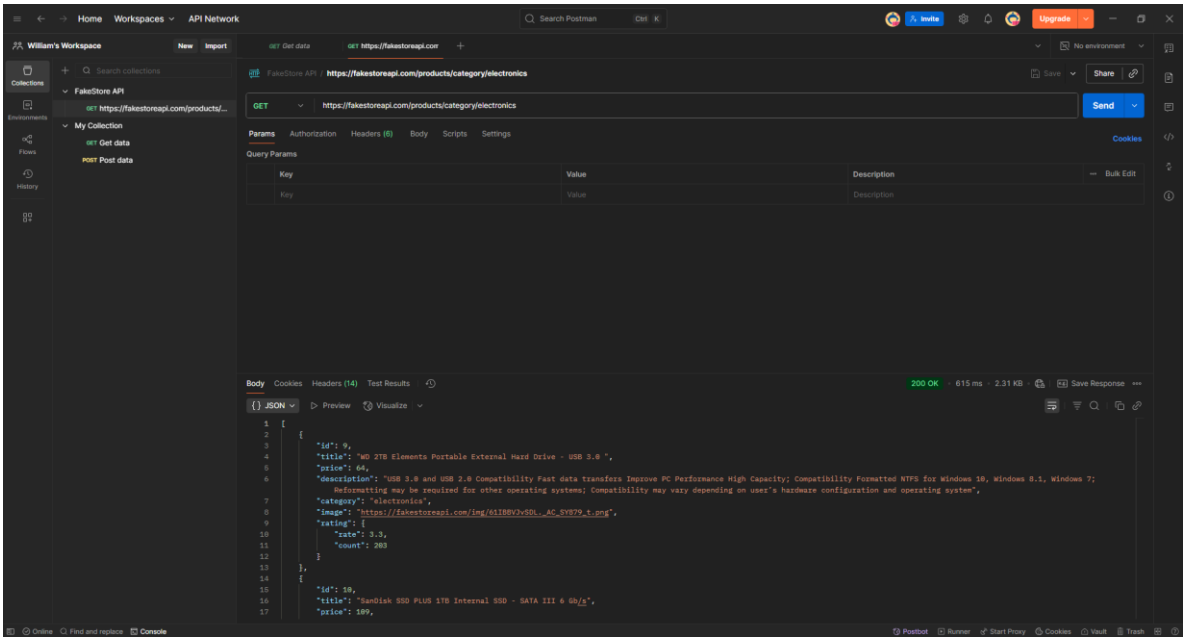
3. Crear producto nuevo
    a. Endpoint: /products
    b. Método: POST
    c. Resultado esperado: Status 201, producto creado con id asignado automáticamente, datos coinciden con el request.

4. Actualizar imagen del producto
    a. Endpoint: /products/21
    b. Método: PUT
    c. Resultado esperado: Status 200, producto actualizado con nueva imagen reflejada, propiedades conservadas.

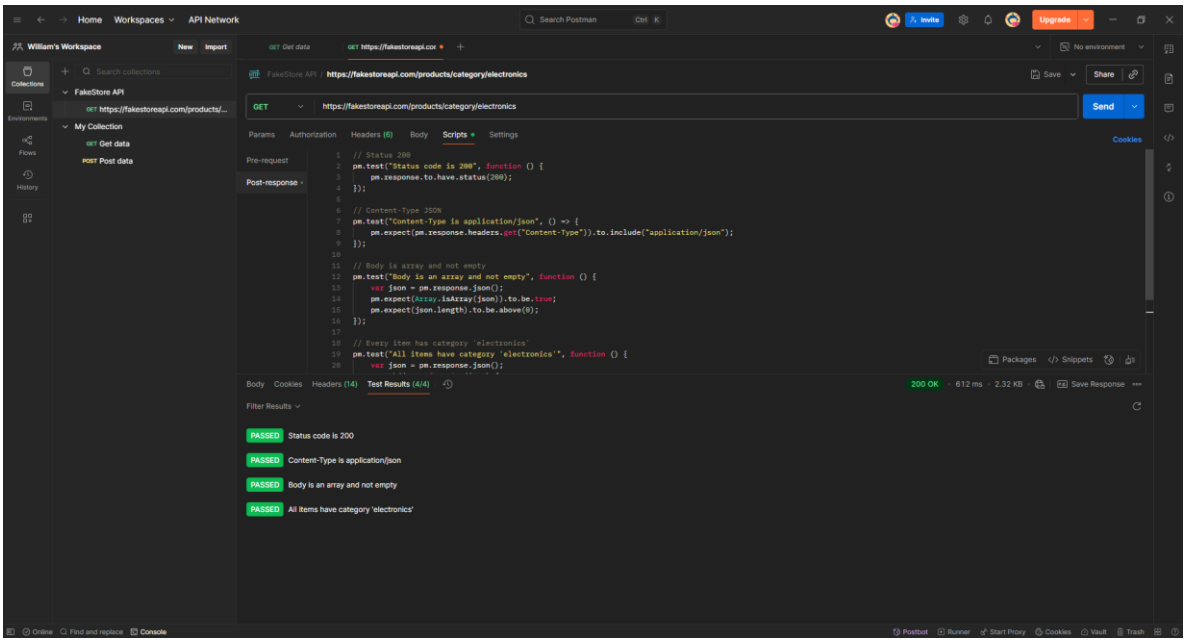
5. Simulación de 150 usuarios
    a. Objetivo: Validar el comportamiento bajo carga de múltiples usuarios.
    b. Configuración: Se realizan dos pruebas en dos fases, la primera fase consta en 100 usuarios simultáneos y la segunda en 50 usuarios simultáneos, haciendo una comparación entre ambas fases, fixed, requests secuenciales.
    c. Métricas: Total requests, tiempo promedio, P90, P95, P99, Error rate.

6. Escalado progresivo
    a. Objetivo: Observar tendencias de rendimiento ante aumento de carga.
    b. Configuración: Se realizan 4 pruebas en 4 fases, con Ramp Up, diferentes intervalos y duraciones, máximo 100 usuarios, por una limitante de Postman.
    c. Métricas: Total requests, tiempo promedio, P90, P95, P99, Error rate, comportamiento bajo incremento y mantenimiento.

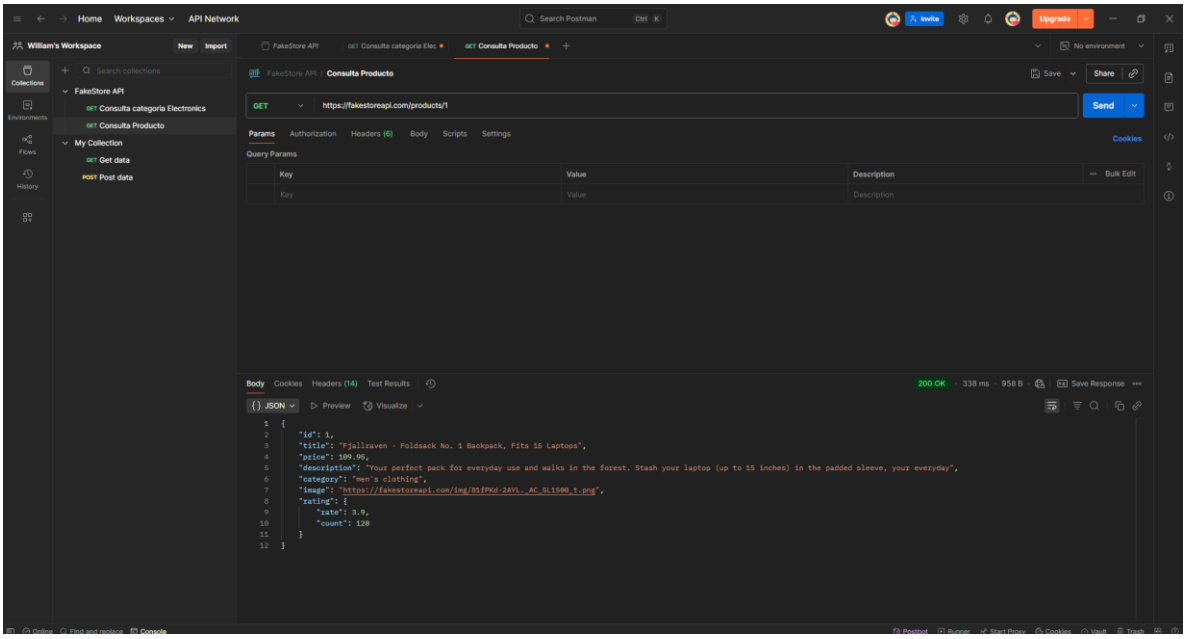
CP1. Listar productos electrónicos



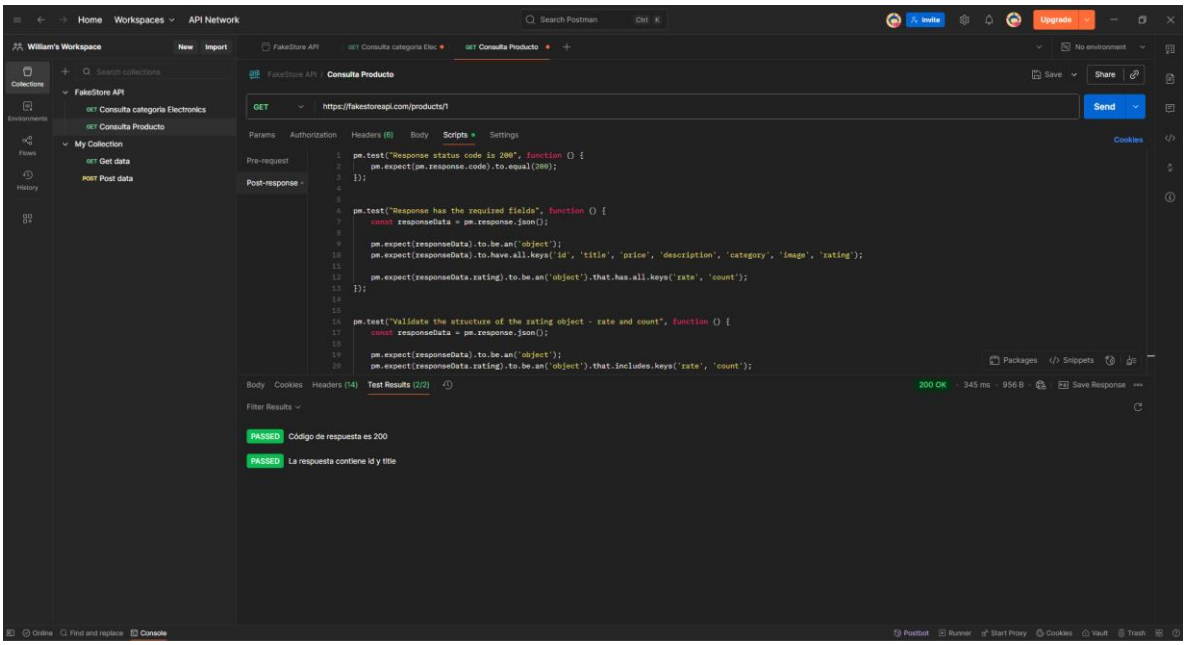
Script test



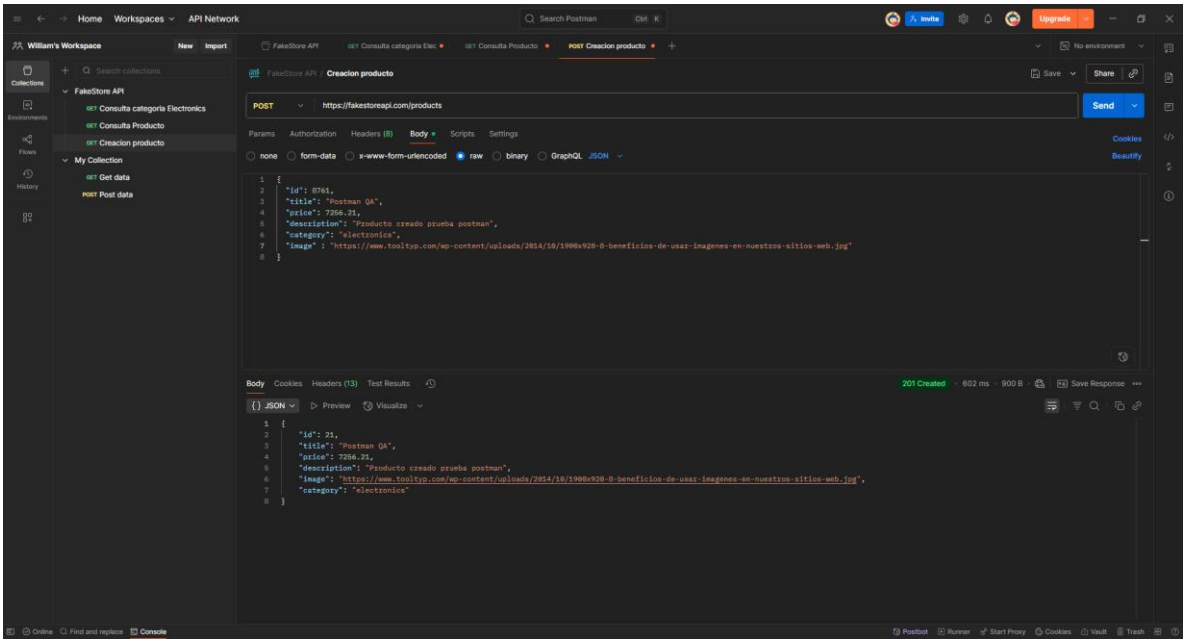
CP2. Consultar un producto específico



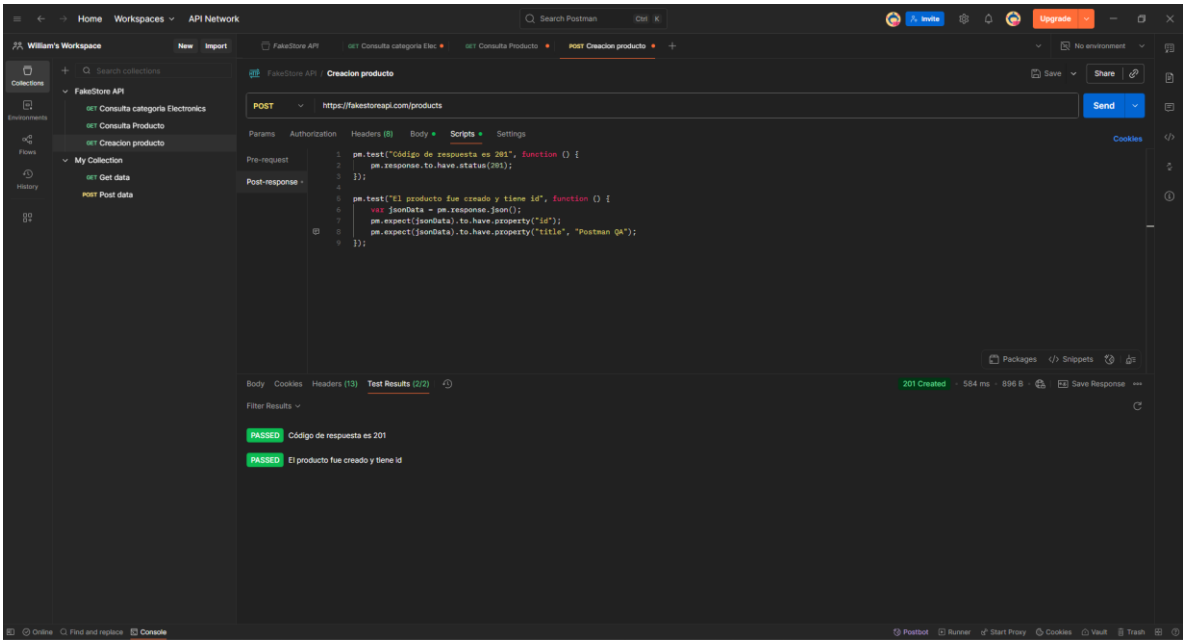
Script test



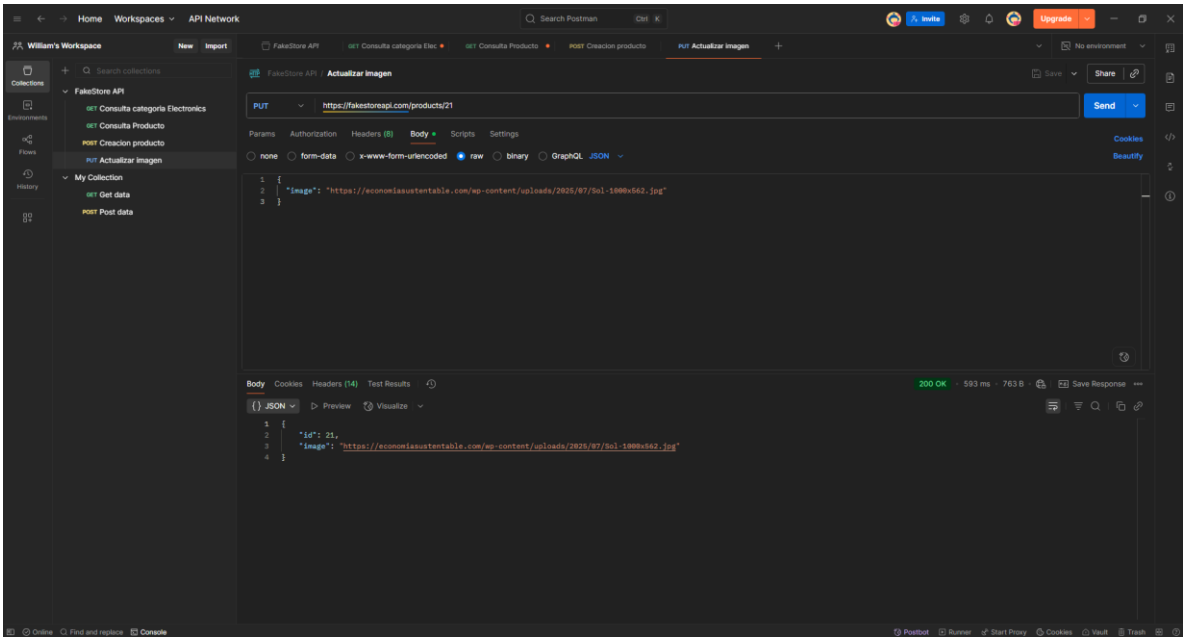
CP3. Crear nuevo producto



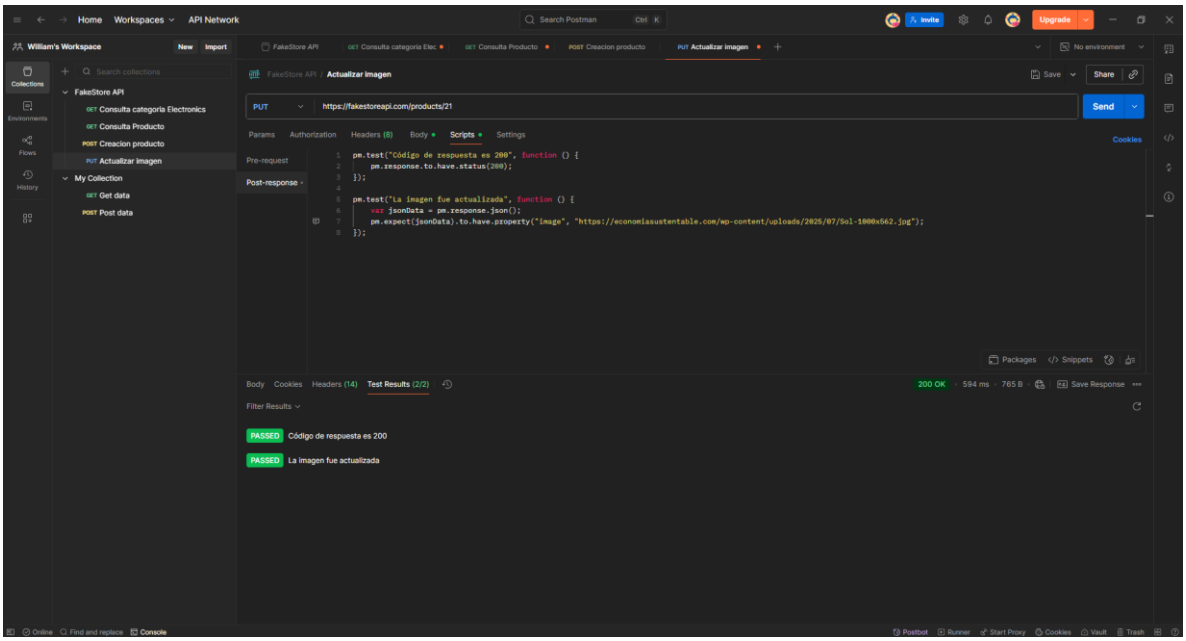
Script test



CP4. Actualizar imagen del producto



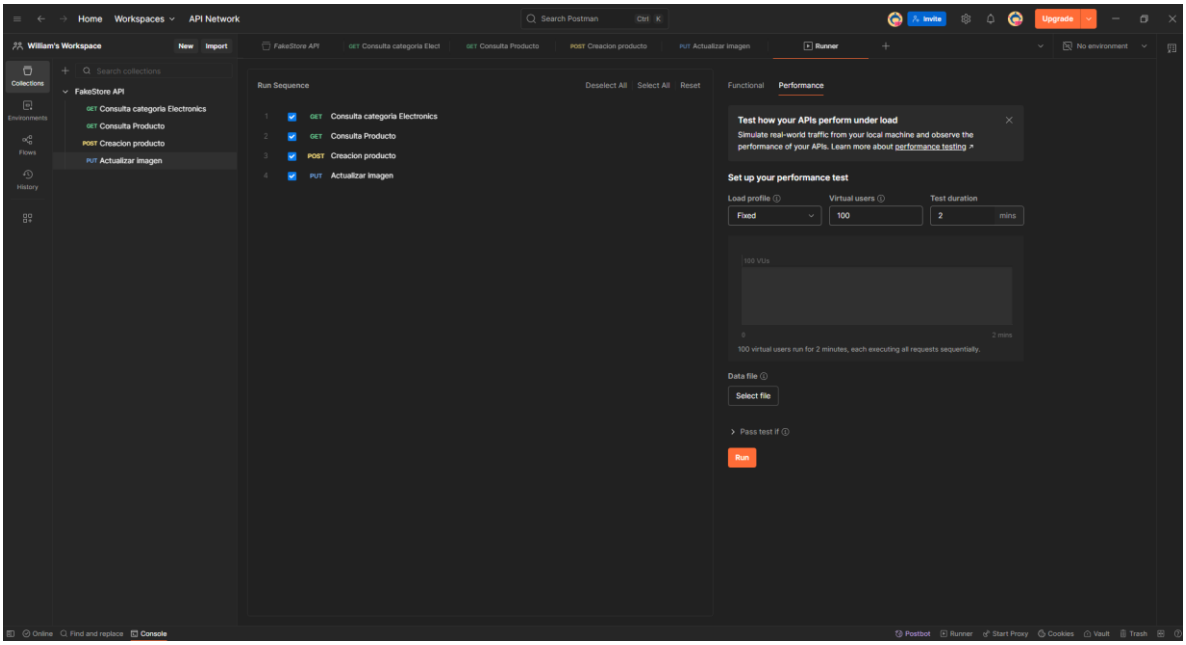
Script test

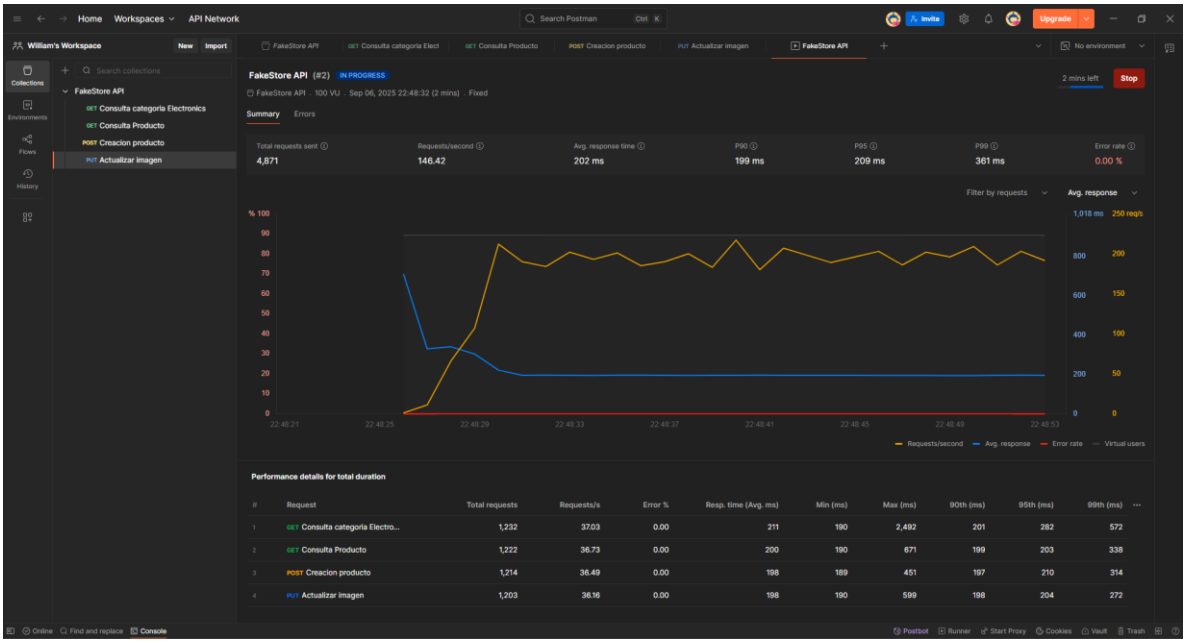


Endpoint	Status	Tiempo Promedio	Tamaño de respuesta	Observaciones
GET /products/category/electronics	200 OK	615 ms	2.31 KB	Listado correcto de productos electrónicos
GET /products/1	200 OK	338 ms	958 B	Información completa de un producto
POST /products	201 OK	584 ms	996 B	Producto creado, id asignado automáticamente
PUT /products/21	200 OK	594 ms	765 B	Imagen actualizada correctamente

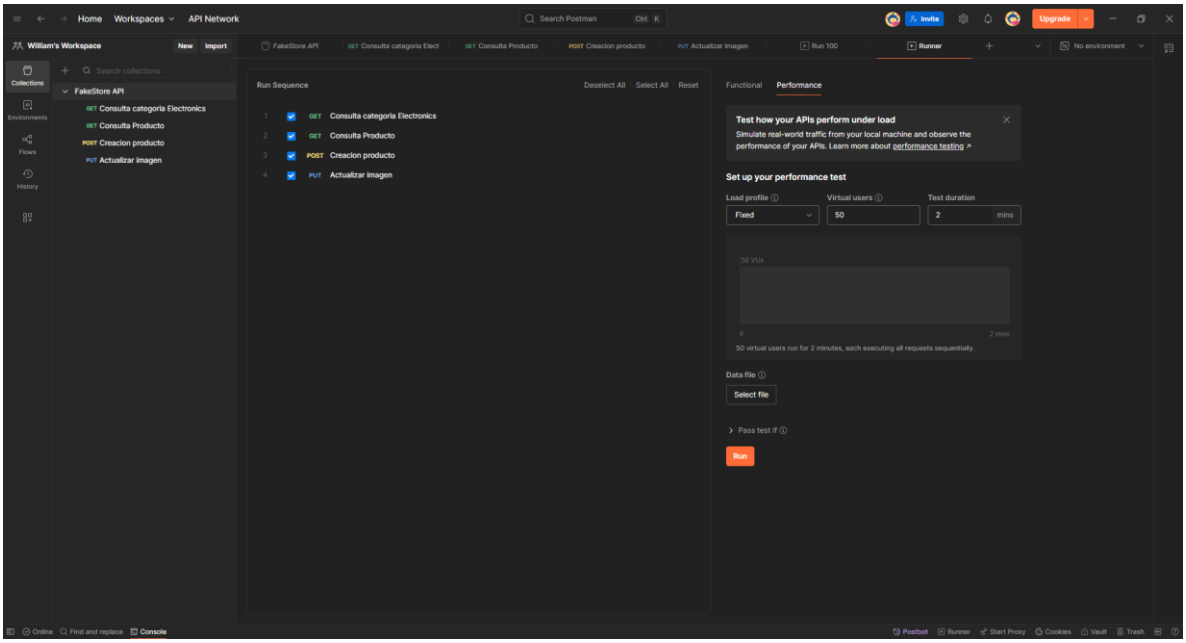
CP5. Simulación 150 usuarios

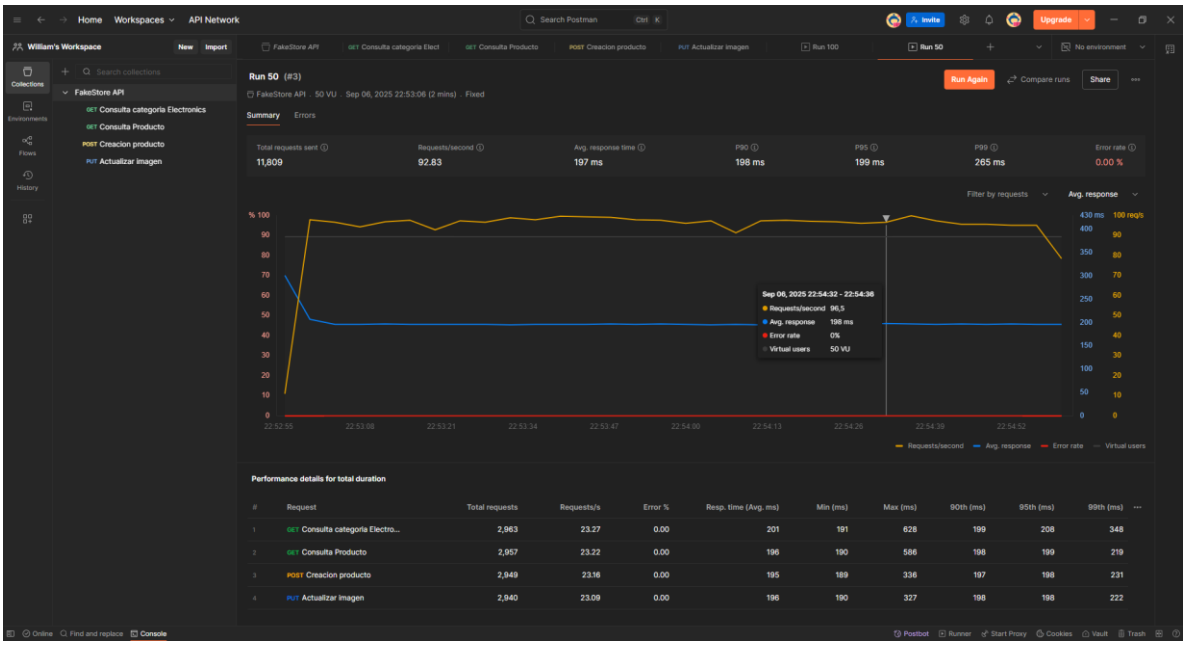
Fase 1: 100 usuarios



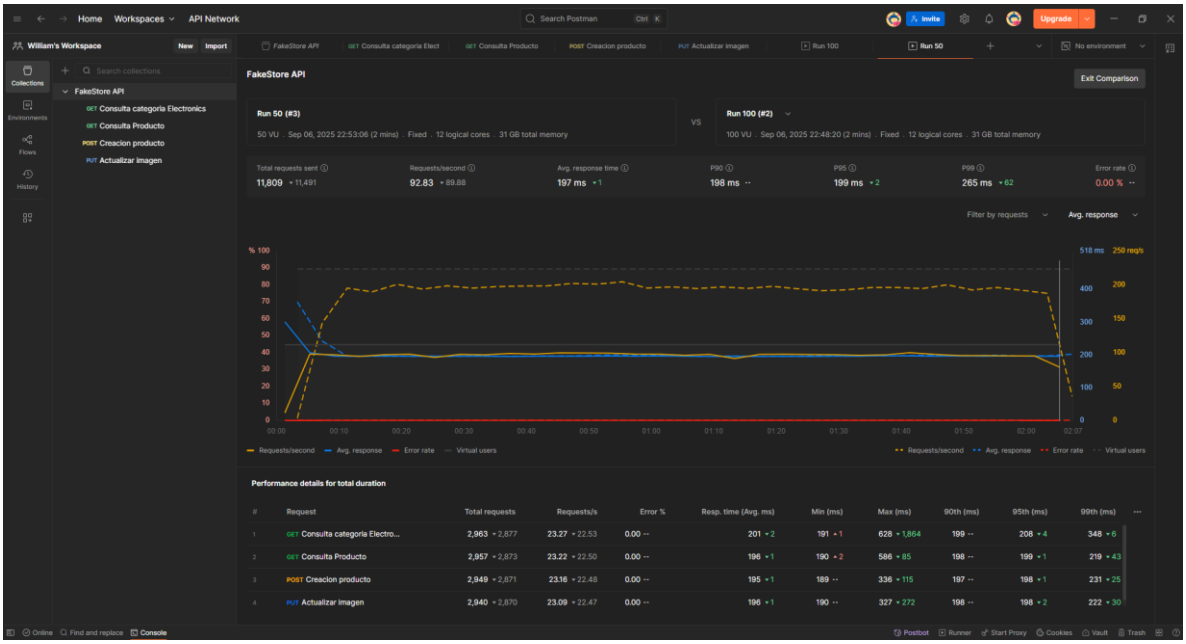


Fase 2: 50 Usuarios





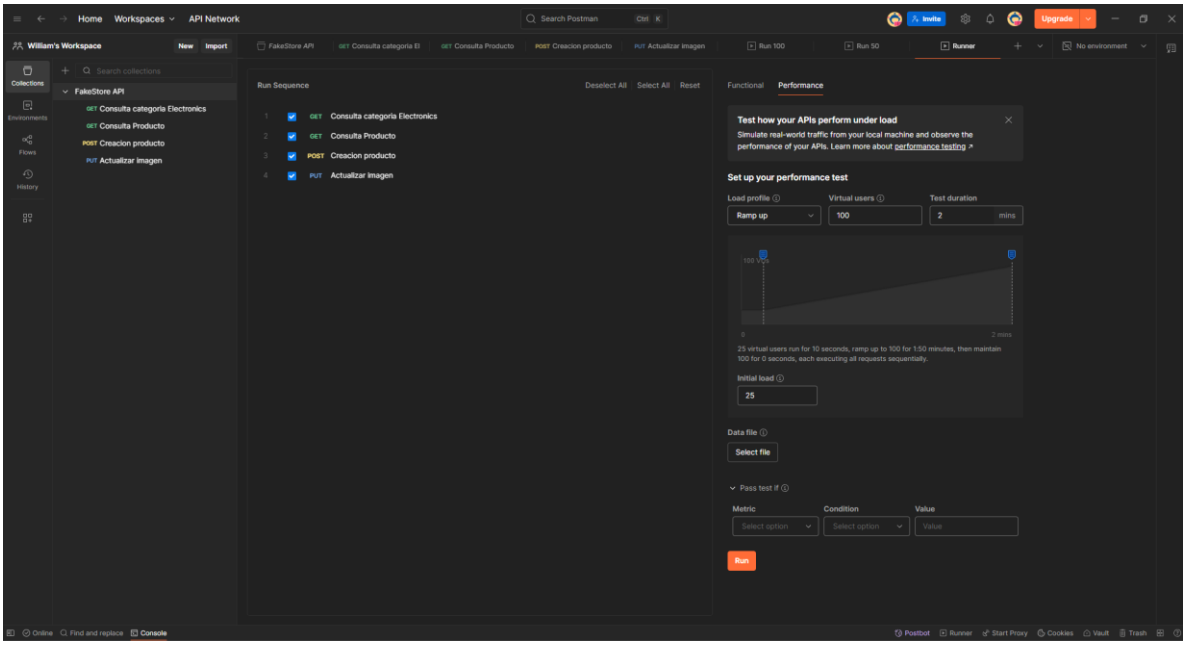
Comparación

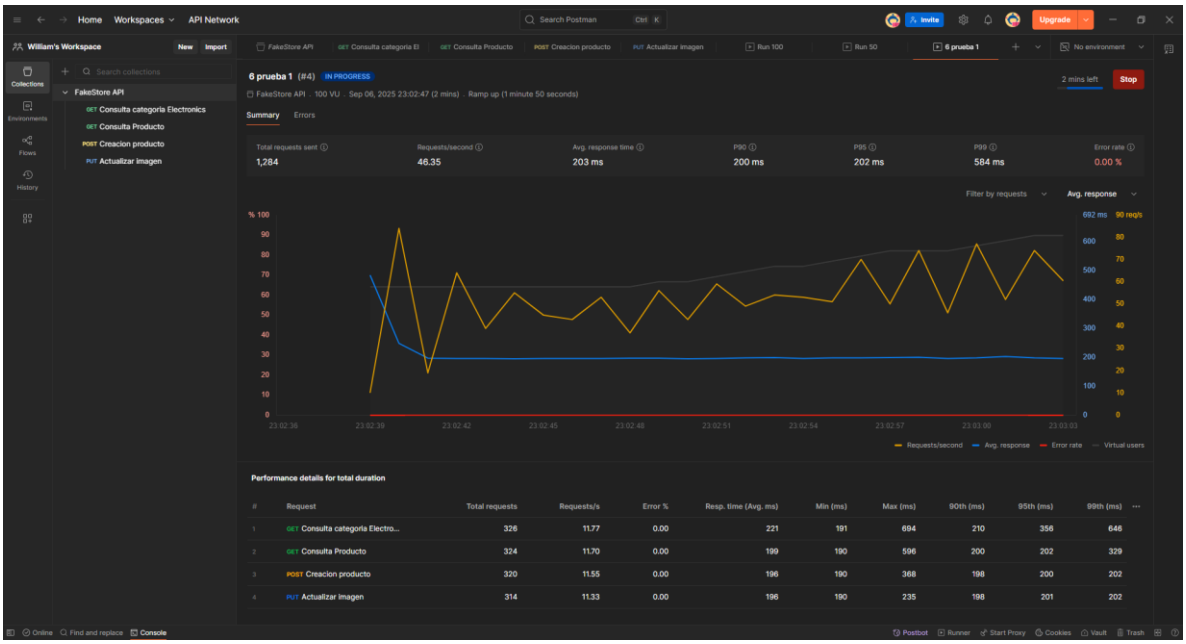


Métricas y comparaciones obtenidas

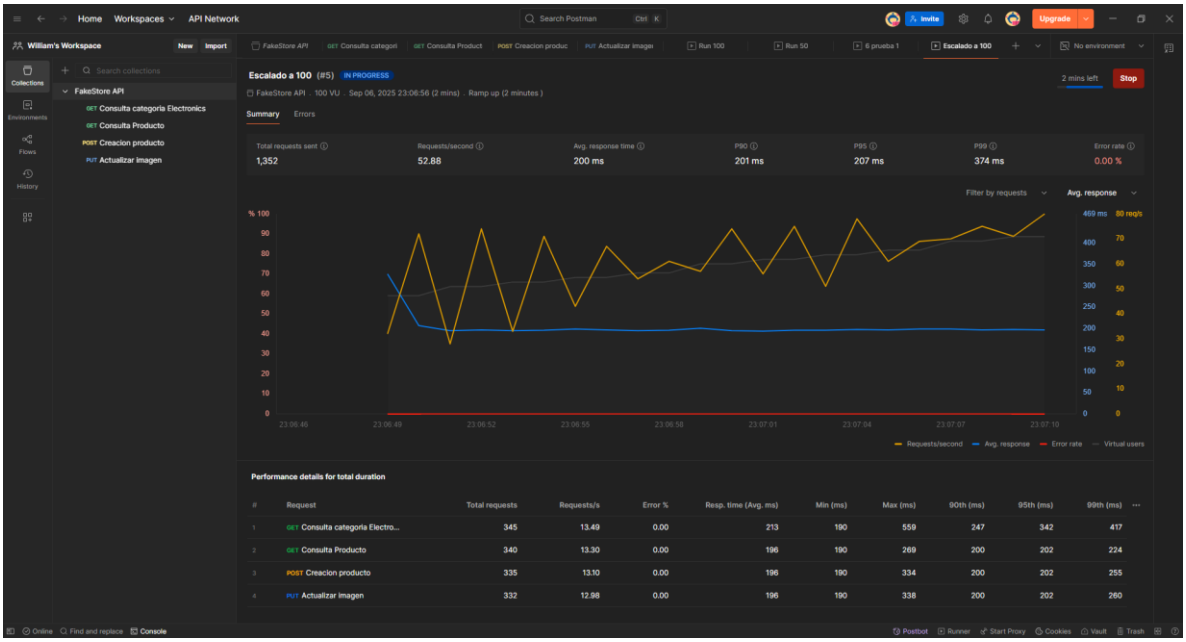
CP6. Escalado progresivo

Fase 1:

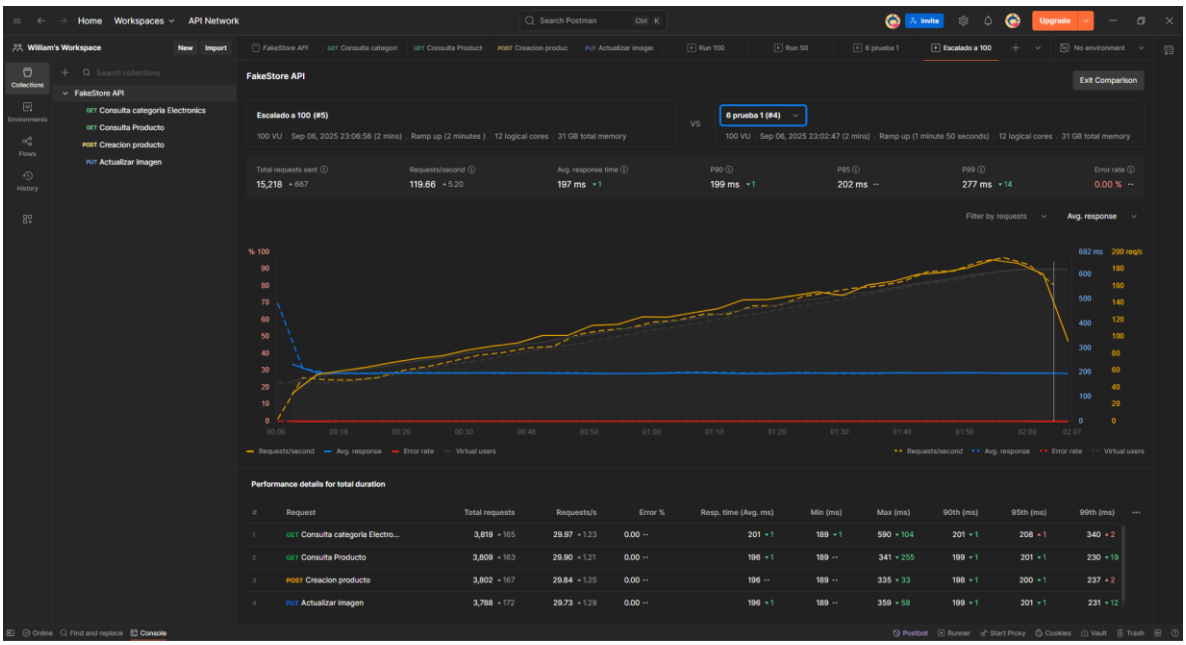




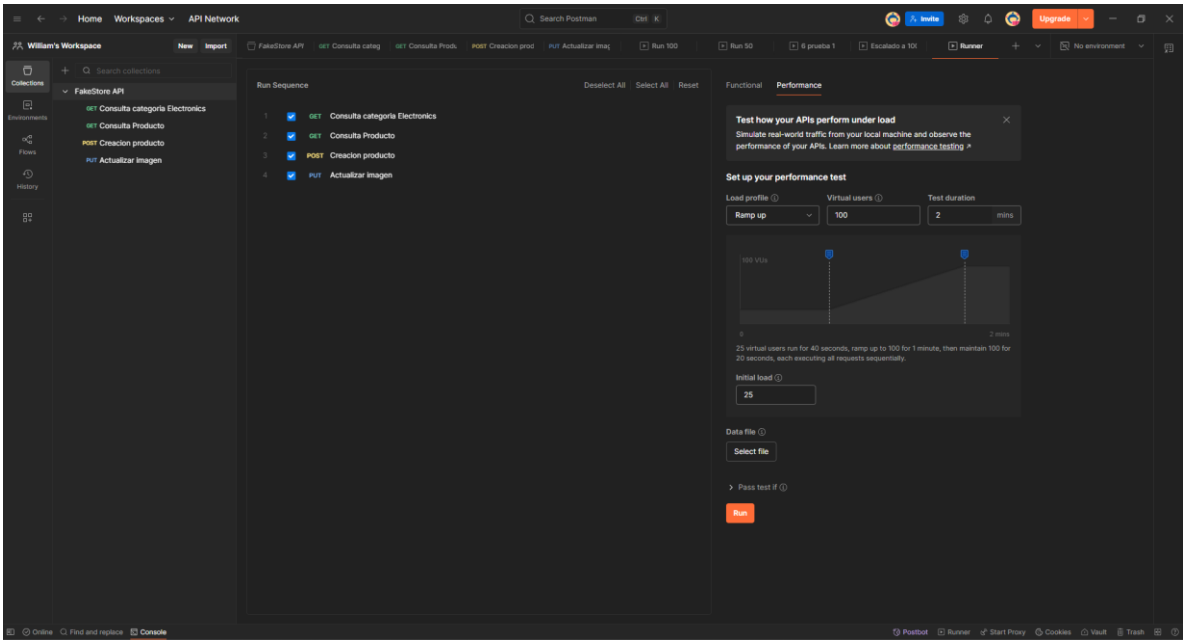
Fase 2:

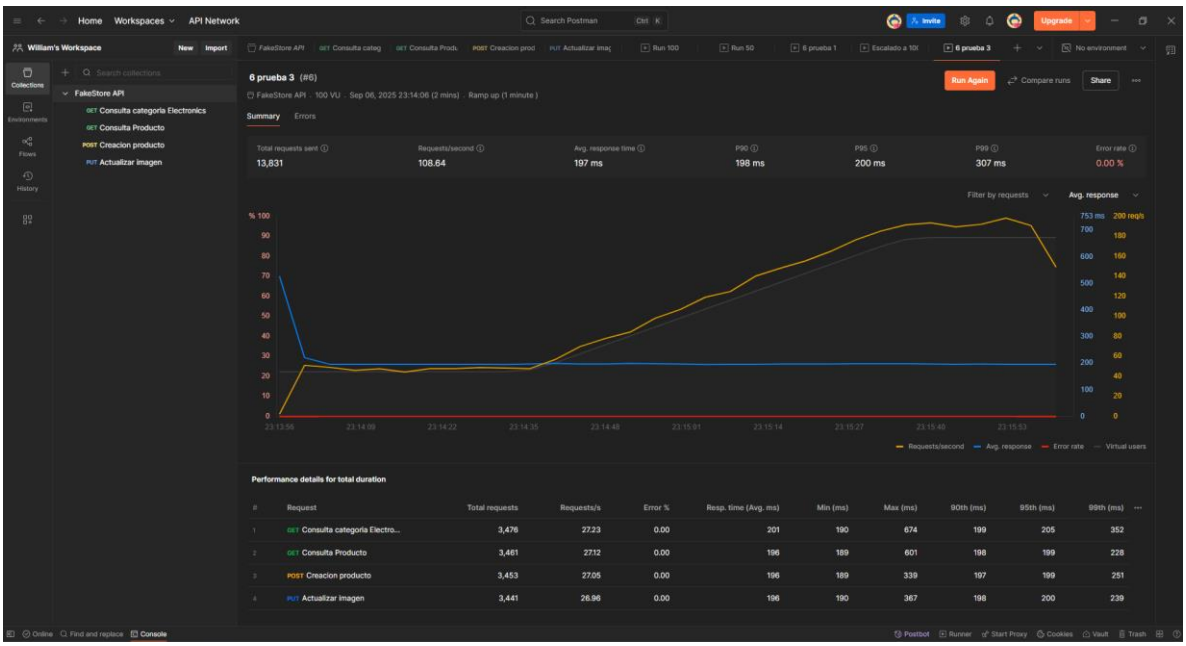


Comparación fase 1 vs fase 2:

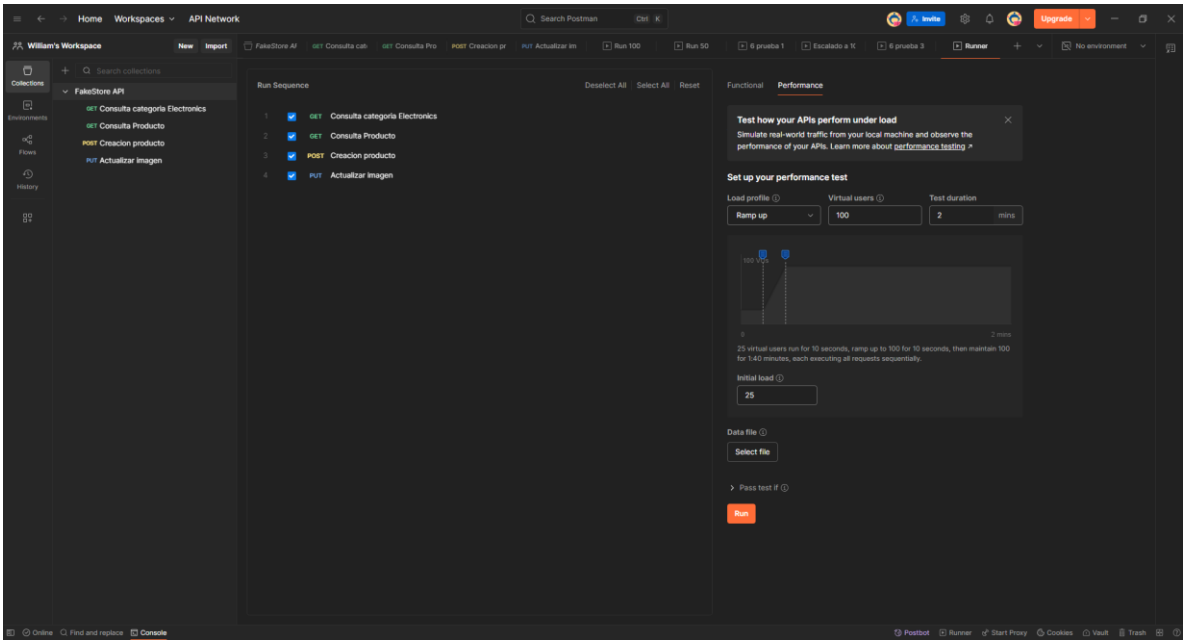


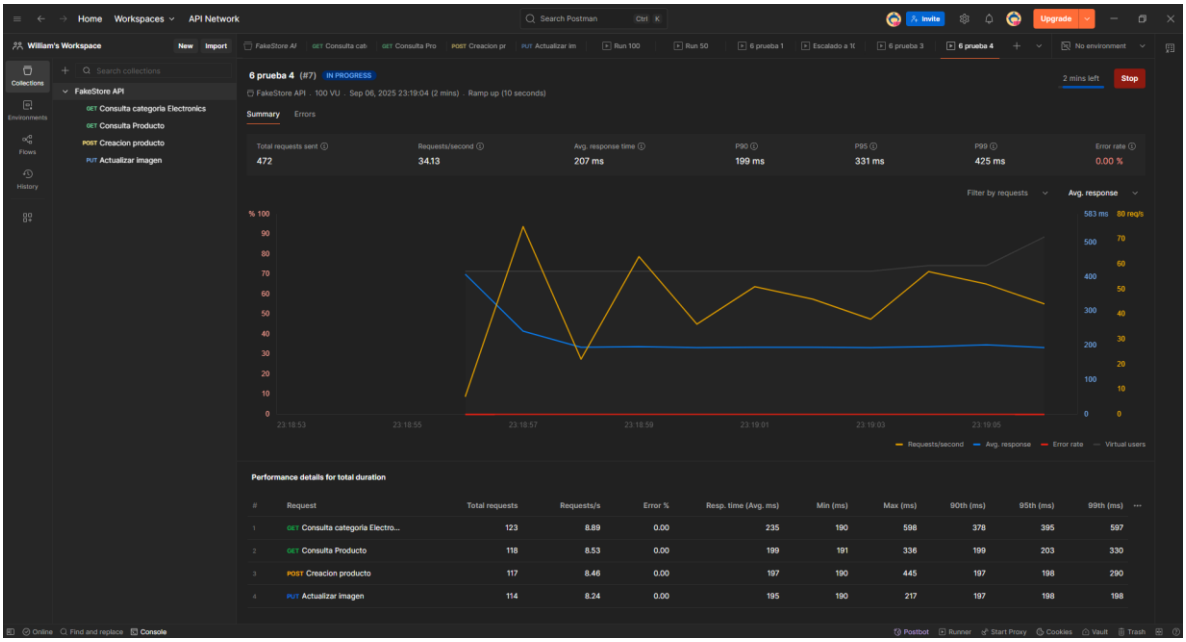
Fase 3:


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
William's Workspace
New
Import
Collections
Environments
Flows
History
FakeStore API
GET Consulta categoría Electronics
GET Consulta Producto
POST Creacion producto
PUT Actualizar imagen
Run Sequence
Deselect All
Select All
Reset
Functional
Performance
Test how your APIs perform under load
Simulate real-world traffic from your local machine and observe the performance of your APIs. Learn more about performance testing
Set up your performance test
Load profile
Virtual users
Test duration
Ramp up
100
2
mins
100 VUs
0
2 mins
25 virtual users run for 40 seconds, ramp up to 100 for 1 minute, then maintain 100 for 20 seconds, each executing all requests sequentially.
Initial load
25
Data file
Select file
Pass test if
Run
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

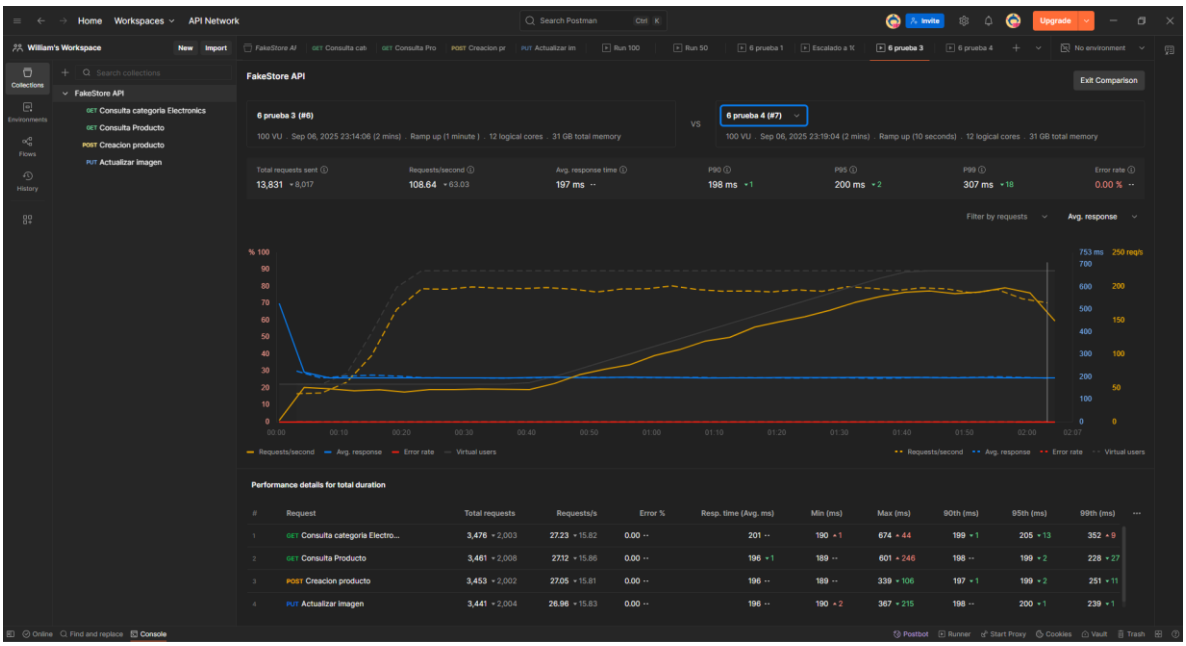

Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
William's Workspace
New
Import
FakeStore API
GET Consulta categoría Electronics
GET Consulta Producto
POST Creacion producto
PUT Actualizar imagen
6 prueba 3 (#6)
FakeStore API · 100 VU · Sep 08, 2025 23:14:06 (2 mins) · Ramp up (1 minute)
Run Again
Compare runs
Share
Summary
Errors
Total requests sent
13,831
Requests/second
108.64
Avg. response time
197 ms
P90
198 ms
P95
200 ms
P99
307 ms
Error rate
0.00 %
Filter by requests
Avg. response
753 ms
200 req/s
Requests/second
Avg. response
Error rate
Virtual users
Performance details for total duration
Request
Total requests
Requests/s
Error %
Resp. time (Avg. ms)
Min (ms)
Max (ms)
90th (ms)
95th (ms)
99th (ms)
GET Consulta categoría Electro...
3,476
27.23
0.00
201
190
674
199
205
352
GET Consulta Producto
3,461
27.12
0.00
196
189
601
198
199
228
POST Creacion producto
3,453
27.05
0.00
196
189
339
197
199
251
PUT Actualizar imagen
3,441
26.96
0.00
196
190
367
198
200
239
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Fase 4:


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
William's Workspace
New
Import
Runner
No environment
Collections
Environments
Flows
History
FakeStore API
GET Consulta categoría Electronics
GET Consulta Producto
POST Creacion producto
PUT Actualizar imagen
Run Sequence
Deselect All
Select All
Reset
Functional
Performance
Test how your APIs perform under load
Simulate real-world traffic from your local machine and observe the performance of your APIs. Learn more about performance testing
Set up your performance test
Load profile
Virtual users
Test duration
Ramp up
100
2
mins
25 virtual users run for 10 seconds, ramp up to 100 for 10 seconds, then maintain 100 for 1:40 minutes, each executing all requests sequentially.
Initial load
25
Data file
Select file
Pass test if
Run
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
William's Workspace
New
Import
6 prueba 4
No environment
Collections
Environments
Flows
History
FakeStore API
GET Consulta categoría Electronics
GET Consulta Producto
POST Creacion producto
PUT Actualizar imagen
6 prueba 4 (#7)
IN PROGRESS
FakeStore API · 100 VU · Sep 06, 2025 23:19:04 (2 mins) · Ramp up (10 seconds)
2 mins left
Stop
Summary
Errors
Total requests sent
472
Requests/second
34.13
Avg. response time
207 ms
P90
199 ms
P95
331 ms
P99
425 ms
Error rate
0.00 %
Filter by requests
Avg. response
Requests/second
Avg. response
Error rate
Virtual users
Performance details for total duration
Request Total requests Requests/s Error % Resp. time (Avg. ms) Min (ms) Max (ms) 90th (ms) 95th (ms) 99th (ms)
1 GET Consulta categoría Electro... 123 8.89 0.00 235 190 598 378 395 597
2 GET Consulta Producto 118 8.53 0.00 199 191 336 199 203 330
3 POST Creacion producto 117 8.46 0.00 197 190 445 197 198 290
4 PUT Actualizar imagen 114 8.24 0.00 195 190 217 197 198 198
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Comparación Fase 3 vs Fase 4



Métricas

Prueba	Total request	Request/second	Avg. Response time	P99	Error rate
Prueba 1	14, 551	114.47	198 ms	291 ms	0.00%
Prueba 2	15, 218	119.66	197 ms	277 ms	0.00%
Prueba 3	13,831	108.64	197 ms	307 ms	0.00%
Prueba 4	21,848	171.68	197 ms	325 ms	0.00%

Conclusiones

- Las API soporta incrementos progresivos de carga sin errores
- Los tiempos promedio se mantienen estables (~197–198 ms)
- Los percentiles P95/P99 muestran ligeros incrementos bajo máxima carga, dentro de los rangos aceptables
- La simulación secuencial permite observar tendencias de rendimiento y comportamiento ante múltiples solicitudes, aunque no sea concurrencia real
- Existe una limitante frente a Postman para el valor total de usuarios recurrentes y obtener métricas