# Portafolio 6 y 7

## Introducción

En este informe se detalla el proceso de validación funcional y de rendimiento para la nueva API REST de la plataforma de salud digital MediPlus. El objetivo es asegurar el correcto funcionamiento, la seguridad y la escalabilidad de los servicios.

Para simular la API de MediPlus, se usarán los endpoints públicos de https://reqres.in/

Se usará GitHub como repositorio tanto del código como de parte de la documentación.

**Repositorio: https://github.com/FabianTorres/portafolio-6-7**

## Lección 1

### Documentación de Endpoints

|  |  |  |
| --- | --- | --- |
| **Método HTTP** | **URI** | **Descripción** |
| GET | /api/users?page=2 | Obtiene una lista paginada de usuarios. |
| GET | /api/users/{id} | Obtiene los datos de un único usuario por su ID. |
| POST | /api/users | Crea un nuevo usuario con los datos del cuerpo. |
| PUT | /api/users/{id} | Actualiza los datos de un usuario existente. |
| DELETE | /api/users/{id} | Elimina un usuario existente por su ID. |
| POST | /api/register | Registra un nuevo usuario en el sistema para obtener un token. |
| POST | /api/login | Autentica a un usuario para obtener un token de sesión. |

### Estructura del Proyecto

Las pruebas se crearán en la carpeta estándar para código de prueba en Maven:

api-tests/

├── src/

│ ├── main/java/

│ └── test/java/

│ └── cl/cursos/

│ ├── FunctionalTests.java

│ └── SecurityTests.java

└── pom.xml

Las dependencias son:

* REST Assured: Librería principal para la automatización de pruebas sobre APIs REST en Java.
* JUnit 5: Framework para la definición y ejecución de los casos de prueba.
* Hamcrest: Librería de matchers que permite escribir aserciones (validaciones) más declarativas y legibles.

## Lección 2

En esta fase, se implementan pruebas automatizadas para validar que cada endpoint se comporte según lo esperado. Se usará REST Assured.

Descripción de los Casos de Prueba Funcionales de FunctionalTests.java

* testListarUsuariosRetornoOK: Verifica que se pueda obtener una lista paginada de usuarios.
* testObtenerUsuarioUnicoRetornoOK: Comprueba la obtención de un usuario por su ID. Se valida el código de estado 200 (OK)
* testCrearUsuarioRetornoCreado: Valida la creación de un nuevo usuario. Se envía un POST con los datos y se espera un código de estado 201
* testRegistroExitosoRetornaToken: Prueba la funcionalidad de registro de un nuevo usuario. Se espera un código 200 (OK) y que la respuesta incluya un ID y un token.
* testActualizarUsuarioRetornoOK: Verifica la actualización completa de un usuario existente mediante PUT. Se espera un código 200 (OK) y que el campo modificado se refleje en la respuesta.
* testEliminarUsuarioRetornoSinContenido: Valida la eliminación de un usuario. Se espera un código de estado 204 (No Content), que indica que la operación fue exitosa pero no hay contenido que devolver.
* testObtenerUsuarioNoEncontrado (Prueba Negativa): Simula una petición para obtener un usuario con un ID que no existe. Se valida que la API responda correctamente con un código de error 404.
* testRegistroFallidoSinPassword (Prueba Negativa): Simula un intento de registro sin el campo obligatorio password. Se valida que la API rechace la solicitud con un código 400 (Bad Request) y devuelva un mensaje de error específico.

## Lección 3

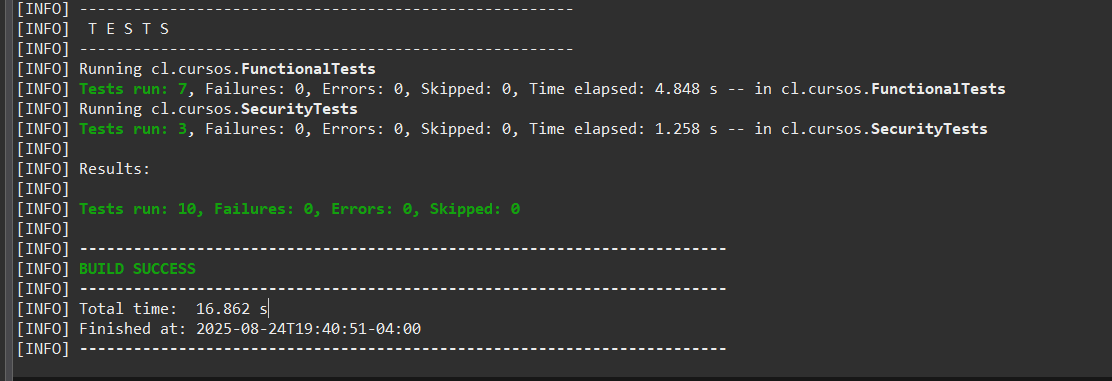
La seguridad de una API sirve para proteger los datos y prevenir accesos no autorizados. En esta sección, se validan los mecanismos de autenticación simulados.

La API de reqres.in usa un mecanismo de seguridad basado en API Keys. Cada solicitud a sus endpoints debe incluir un header específico (x-api-key) con un valor.

Descripción de los Casos de Prueba de Seguridad SecurityTests.java

* testLoginExitosoRetornaToken: Simula un login con credenciales válidas. Se verifica que la API responda con un código 200 (OK)
* testLoginFallidoRetornaError: Simula un intento de login con datos incompletos (sin contraseña). Se valida que el sistema rechace la petición con un código de error 400 (Bad Request)
* testAccesoSinApiKeyRetornaError: Valida el mecanismo de protección principal de la API. Se usa una petición a un endpoint sin incluir el header x-api-key requerido. Se espera que la API deniegue el acceso con un código 401 (Unauthorized)

### Evidencias de Ejecución (Logs)



## Lección 4

Se harán las pruebas de carga simuladas de acuerdo a las configuraciones:

* GET masivo
* POST masivo
* GET+POST combinado

### **Estructura de carpetas de resultados de Jmeter**

Dentro de cada una de estas estructuras de carpetas se encuentran los reportes en html de cada configuración de las pruebas de carga.  
  
Resultados/

├── GET\_POST\_Masivo/

│ ├── 10\_Usuarios/

│ ├── 50\_Usuarios/

│ └── 100\_Usuarios/

└── Combinado/

├── 10\_Usuarios/

├── 50\_Usuarios/

└── 100\_Usuarios/

### Log de Ejecución

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_10\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\10\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\10\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_10\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:30:31 PM CLT (1756081831836)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 1 in 00:00:01 = 0.8/s Avg: 554 Min: 554 Max: 554 Err: 0 (0.00%) Active: 4 Started: 4 Finished: 0

summary + 19 in 00:00:10 = 2.0/s Avg: 340 Min: 113 Max: 702 Err: 0 (0.00%) Active: 0 Started: 20 Finished: 20

summary = 20 in 00:00:11 = 1.9/s Avg: 351 Min: 113 Max: 702 Err: 0 (0.00%)

Tidying up ... @ August 24, 2025 8:30:42 PM CLT (1756081842823)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_50\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\50\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\50\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_50\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:30:58 PM CLT (1756081858525)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 1 in 00:00:01 = 0.7/s Avg: 775 Min: 775 Max: 775 Err: 0 (0.00%) Active: 4 Started: 4 Finished: 0

summary + 62 in 00:00:31 = 2.0/s Avg: 269 Min: 117 Max: 655 Err: 0 (0.00%) Active: 4 Started: 66 Finished: 62

summary = 63 in 00:00:32 = 1.9/s Avg: 277 Min: 117 Max: 775 Err: 0 (0.00%)

summary + 37 in 00:00:18 = 2.1/s Avg: 241 Min: 116 Max: 409 Err: 0 (0.00%) Active: 0 Started: 100 Finished: 100

summary = 100 in 00:00:50 = 2.0/s Avg: 264 Min: 116 Max: 775 Err: 0 (0.00%)

Tidying up ... @ August 24, 2025 8:31:49 PM CLT (1756081909081)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_100\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\100\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\GET\_POST\_Masivo\100\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_GET\_POST\_Masivo\_100\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:31:57 PM CLT (1756081917158)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 5 in 00:00:03 = 1.7/s Avg: 361 Min: 117 Max: 751 Err: 0 (0.00%) Active: 2 Started: 6 Finished: 4

summary + 59 in 00:00:30 = 2.0/s Avg: 251 Min: 112 Max: 594 Err: 0 (0.00%) Active: 3 Started: 66 Finished: 63

summary = 64 in 00:00:33 = 1.9/s Avg: 260 Min: 112 Max: 751 Err: 0 (0.00%)

summary + 55 in 00:01:06 = 0.8/s Avg: 242 Min: 112 Max: 376 Err: 2 (3.64%) Active: 0 Started: 200 Finished: 200

summary = 119 in 00:01:39 = 1.2/s Avg: 252 Min: 112 Max: 751 Err: 2 (1.68%)

Tidying up ... @ August 24, 2025 8:33:36 PM CLT (1756082016416)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_10\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\10\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\10\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_10\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:33:49 PM CLT (1756082029552)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 18 in 00:00:10 = 1.7/s Avg: 316 Min: 125 Max: 627 Err: 0 (0.00%) Active: 2 Started: 10 Finished: 8

summary + 2 in 00:00:02 = 1.2/s Avg: 368 Min: 273 Max: 464 Err: 0 (0.00%) Active: 0 Started: 10 Finished: 10

summary = 20 in 00:00:12 = 1.6/s Avg: 321 Min: 125 Max: 627 Err: 0 (0.00%)

Tidying up ... @ August 24, 2025 8:34:01 PM CLT (1756082041871)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_50\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\50\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\50\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_50\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:34:12 PM CLT (1756082052074)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 33 in 00:00:18 = 1.9/s Avg: 243 Min: 125 Max: 538 Err: 0 (0.00%) Active: 3 Started: 18 Finished: 15

summary + 61 in 00:00:31 = 2.0/s Avg: 201 Min: 119 Max: 317 Err: 0 (0.00%) Active: 3 Started: 49 Finished: 46

summary = 94 in 00:00:49 = 1.9/s Avg: 216 Min: 119 Max: 538 Err: 0 (0.00%)

summary + 6 in 00:00:04 = 1.7/s Avg: 238 Min: 137 Max: 311 Err: 0 (0.00%) Active: 0 Started: 50 Finished: 50

summary = 100 in 00:00:52 = 1.9/s Avg: 217 Min: 119 Max: 538 Err: 0 (0.00%)

Tidying up ... @ August 24, 2025 8:35:04 PM CLT (1756082104403)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>jmeter -n -t "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_100\_usuarios.jmx" -l "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\100\_Usuarios\resultados.jtl" -e -o "C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Resultados\Combinado\100\_Usuarios\reporte\_html"

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:\Tareas\Portafolio 6 y 7\Planes Jmeter\Escenario\_Combinado\_100\_usuarios.jmx

Starting standalone test @ August 24, 2025 8:35:21 PM CLT (1756082121766)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445

summary + 15 in 00:00:08 = 1.8/s Avg: 228 Min: 127 Max: 525 Err: 0 (0.00%) Active: 3 Started: 9 Finished: 6

summary + 61 in 00:00:30 = 2.0/s Avg: 210 Min: 117 Max: 470 Err: 0 (0.00%) Active: 2 Started: 39 Finished: 37

summary = 76 in 00:00:39 = 2.0/s Avg: 214 Min: 117 Max: 525 Err: 0 (0.00%)

summary + 42 in 00:01:00 = 0.7/s Avg: 203 Min: 124 Max: 292 Err: 0 (0.00%) Active: 0 Started: 100 Finished: 100

summary = 118 in 00:01:39 = 1.2/s Avg: 210 Min: 117 Max: 525 Err: 0 (0.00%)

Tidying up ... @ August 24, 2025 8:37:00 PM CLT (1756082220987)

... end of run

C:\Programas\jmeter\apache-jmeter-5.6.3\bin>

## Lección 5 Una vez ejecutadas las 6 pruebas de carga, el paso final es interpretar los resultados para comprender el comportamiento de la API, identificar cuellos de botella y proponer mejoras. La principal herramienta de análisis es el Dashboard Report HTML generado por JMeter, de donde se extrajeron los siguientes datos:

### Análisis Comparativo de Resultados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Escenario** | **Usuarios** | **Throughput (req/seg)** | **Tiempo Resp. Prom. (ms)** | **Percentil 95 (ms)** | **Tasa de Error (%)** |
| GET y POST Masivo | 10 | 1.98 | 351.05 | 699.75 | 0.00% |
|  | 50 | 2.01 | 264.18 | 566.7 | 0.00% |
|  | 100 | 2.01 | 252.11 | 395 | 1.68% |
| Carga Combinada | 10 | 1.74 | 321.5 | 623.15 | 0.00% |
|  | 50 | 1.95 | 217.71 | 463.8 | 0.00% |
|  | 100 | 2 | 210.14 | 292.25 | 0.00% |

### Interpretación de Métricas Clave

**Throughput (Rendimiento):** El rendimiento del sistema se estanca muy rápido, alcanzando un máximo de solo ~2 peticiones por segundo. El hecho de que el throughput no aumente al pasar de 50 a 100 usuarios es un indicador claro de que la API ha alcanzado un cuello de botella severo que le impide procesar más solicitudes, sin importar cuántos usuarios se añadan. Considerando que la web es publica, tiene sentido este comportamiento.

**Tiempos de Respuesta (Promedio y Percentiles)**: Los datos muestran un comportamiento anómalo: los tiempos de respuesta van a la baja a medida que aumenta la carga. Una posible explicación es un efecto de "calentamiento" o caché en el servidor. Las primeras pruebas (con 10 usuarios) pueden encontrar un sistema "frío", y las pruebas subsiguientes (con 50 y 100) se benefician de datos ya cacheados, resultando en respuestas más rápidas.

**Tasa de Error**: Este es el hallazgo más crítico. En el escenario "GET y POST Masivo" con 100 usuarios, apareció una tasa de error del 1.68%. Cualquier porcentaje de error bajo carga indica que el sistema no lo soporta y no puede manejar la demanda. Se sugiere un problema de concurrencia ya que los errores solo aparecen en el escenario con hilos paralelos.

### Hallazgos y Recomendaciones de Mejora

* La capacidad máxima de la API está limitada a aproximadamente 2 peticiones por segundo, un rendimiento muy bajo para una aplicación moderna.
* El sistema se vuelve inestable y genera errores con una carga de 100 usuarios concurrentes realizando operaciones de lectura y escritura en paralelo.

Recomendaciones:

La prioridad número uno debe ser identificar la causa del bajo throughput. Se recomienda ejecutar las pruebas de carga nuevamente mientras se monitorean los recursos del servidor en tiempo real (CPU, memoria, I/O de disco y conexiones de base de datos). Es muy probable que el problema este en un pool de conexiones a la base de datos muy pequeño, consultas SQL ineficientes o un mal manejo de recursos en el código de la aplicación.