

## REPORTE DE PLAN DE PRUEBAS

AUTOR:

BRAYAN ALEXANDER PUENTES MARTINEZ

INSTRUCTOR:

ANDRES RUBIANO CUCARIAN

SENA

ANALISIS Y DESARROLLO DE SOFTWARE(ADSO)

CENTRO DE SERVICIOS FINANCIEROS

BOGOTÁ D.C. 7 DE MAYO DE 2024

## Tabla de contenido

INTRODUCCIÓN .....	3
HERRAMIENTAS PARA UTILIZAR EN LOS CASOS DE PRUEBA.....	3
POSTMAN .....	3
JMETER .....	3
END POINT .....	4
RUTA INICIAR SESIÓN .....	4
RUTA DE REGISTRO .....	4
RUTA DE CAMBIO DE CONTRASEÑA .....	4
RUTA DE MEDIDAS CORPORALES .....	4
URL DEL VIDEO.....	5
CASOS DE PRUEBA.....	5
CASO DE PRUEBA INICIAR SESION .....	5
CASO DE PRUEBA REGISTRO .....	6
CASO DE PRUEBA CAMBIO DE CONTRASEÑA.....	7
CASO DE PRUEBA MEDIDAS CORPORALES .....	8
REPORTE DE PLAN DE PRUEBAS .....	9
RUTA INICIAR SESIÓN .....	9
CAPTURAS DE PANTALLA .....	9
CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS.....	9
BASE DE DATOS .....	12
RUTA DE REGISTRO .....	13
CAPTURAS DE PANTALLA .....	13
CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS.....	13
BASE DE DATOS .....	17
RUTA DE CAMBIO DE CONTRASEÑA .....	18
CAPTURAS DE PANTALLA .....	18
CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS.....	18
BASE DE DATOS .....	24
RUTA DE MEDIDAS CORPORALES .....	25
CAPTURAS DE PANTALLA .....	25
CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS.....	25
BASE DE DATOS .....	33
CONCLUSIONES .....	34
BIBLIOGRAFIA.....	34

## **INTRODUCCIÓN**

En este trabajo de investigación, exploraremos el proceso de prueba de software aplicado a la plataforma web "Hypertrophy Fitness". Nuestro enfoque se centra en comprender la importancia de las pruebas funcionales, no funcionales, estructurales y aquellas relacionadas con cambios en el sistema. Utilizaremos herramientas como Postman y JMeter para evaluar el rendimiento y la calidad de la aplicación. A lo largo de este estudio, buscamos proporcionar una visión clara y accesible de cómo estas pruebas contribuyen a garantizar la calidad y fiabilidad de la plataforma.

## **HERRAMIENTAS PARA UTILIZAR EN LOS CASOS DE PRUEBA**

### **POSTMAN**

Postman es una herramienta popular utilizada para realizar pruebas de API (Interfaz de Programación de Aplicaciones) y desarrollo de software. Permite a los desarrolladores enviar solicitudes HTTP a un servidor web y recibir respuestas para probar la funcionalidad de las API. Además de las pruebas, Postman también proporciona funciones para crear, compartir y documentar API, lo que lo convierte en una herramienta integral para el desarrollo y el análisis de API. (postman, s.f.)

### **JMETER**

Apache JMeter es una herramienta de código abierto diseñada para realizar pruebas de carga y rendimiento en aplicaciones web. Permite simular múltiples usuarios virtuales que interactúan con una aplicación web al enviar solicitudes HTTP y medir el tiempo de respuesta, el rendimiento y la estabilidad del sistema bajo diferentes niveles de carga. JMeter es altamente configurable y puede utilizarse para evaluar el rendimiento de aplicaciones web, servidores FTP, bases de datos y otros servicios de red. (Jmeter, s.f.)

## **END POINT**

### **RUTA INICIAR SESIÓN**

**URL:** <http://localhost:3000/IniciarSesion>

**Ruta de la API:** <http://localhost:4000/api/usuarios>

### **RUTA DE REGISTRO**

**URL:** <http://localhost:3000/Registro>

**Ruta de la API:** <http://localhost:4000/api/usuarios>

### **RUTA DE CAMBIO DE CONTRASEÑA**

**URL:** <http://localhost:3000/cambioContrasena>

**Ruta de la API:** <http://localhost:4000/api/auth>

### **RUTA DE MEDIDAS CORPORALES**

**URL:** <http://localhost:3000/MedidasCorporales>

**Ruta de la API:** <http://localhost:4000/api/medidascorporales>

## URL DEL VIDEO

### LINK

[https://www.youtube.com/watch?v=bZqQN665pdA&ab\\_channel=BRAYANALEXA  
NDERPUENTESMARTINEZ](https://www.youtube.com/watch?v=bZqQN665pdA&ab_channel=BRAYANALEXA<br/>NDERPUENTESMARTINEZ)

## CASOS DE PRUEBA

### CASO DE PRUEBA INICIAR SESION

HIPERTROPHY FITNESS					
Autor del caso de prueba	Brayan Alexander Puentes Martinez				
ID	01				
Fecha de revisión	7/05/2024				
Descripción	Postman	<ul style="list-style-type: none"><li>• GET.</li><li>• POST.</li><li>• PUT.</li><li>• DELETE</li></ul>	Ambiente de prueba	<ul style="list-style-type: none"><li>• Versión web.</li><li>• Windows.</li><li>• Navegador Google chrome</li></ul>	
	Jmeter	<ul style="list-style-type: none"><li>• GET</li></ul>			
Herramienta utilizada	<ul style="list-style-type: none"><li>• Postman.</li><li>• Jmeter.</li></ul>				
Datos de entrada	<ul style="list-style-type: none"><li>• Cedula.</li><li>• Contraseña.</li></ul>				
Salida esperada	El usuario puede iniciar sesión exitosamente.				
Salida obtenida	El usuario inicia sesión correctamente en el sistema. El sistema lo redirige hacia la página de consejos.				
Resultado	Aprobado				
Seguimiento	N/A				
Severidad	ALTO		MADIO	BAJO	
	N/A		N/A	N/A	

## CASO DE PRUEBA REGISTRO

HIPERTROPHY FITNESS					
Autor del caso de prueba	Brayan Alexander Puentes Martinez				
ID	02				
Fecha de revisión	7/05/2024				
Descripción	Postman	<ul style="list-style-type: none"> <li>• GET.</li> <li>• POST.</li> <li>• PUT.</li> <li>• DELETE</li> </ul>	Ambiente de prueba	<ul style="list-style-type: none"> <li>• Versión web.</li> <li>• Windows.</li> <li>• Navegador Google chrome</li> </ul>	
	Jmeter	<ul style="list-style-type: none"> <li>• GET</li> </ul>			
Herramienta utilizada	<ul style="list-style-type: none"> <li>• Postman.</li> <li>• Jmeter.</li> </ul>				
Datos de entrada	<ul style="list-style-type: none"> <li>• Nombre completo.</li> <li>• Cedula.</li> <li>• Celular.</li> <li>• Contraseña.</li> <li>• Confirmar contraseña.</li> </ul>				
Salida esperada	El usuario poder registrarse exitosamente.				
Salida obtenida	El usuario se registra correctamente en el sistema. El sistema lo redirige hacia la página de inicio de sesión.				
Resultado	Aprobado				
Seguimiento	N/A				
Severidad	ALTO		MADIO	BAJO	
	N/A		N/A	N/A	

## CASO DE PRUEBA CAMBIO DE CONTRASEÑA

HIPERTROPHY FITNESS					
Autor del caso de prueba	Brayan Alexander Puentes Martinez				
ID	03				
Fecha de revisión	7/05/2024				
Descripción	Postman	<ul style="list-style-type: none"> <li>• GET.</li> <li>• POST.</li> <li>• PUT.</li> <li>• DELETE</li> </ul>	Ambiente de prueba	<ul style="list-style-type: none"> <li>• Versión web.</li> <li>• Windows.</li> <li>• Navegador Google chrome</li> </ul>	
	Jmeter	<ul style="list-style-type: none"> <li>• GET</li> </ul>			
Herramienta utilizada	<ul style="list-style-type: none"> <li>• Postman.</li> <li>• Jmeter.</li> </ul>				
Datos de entrada	<ul style="list-style-type: none"> <li>• Cedula.</li> <li>• Nueva contraseña.</li> <li>• Confirme la contraseña.</li> </ul>				
Salida esperada	El usuario debe poder cambiar su contraseña exitosamente.				
Salida obtenida	El usuario cambia su contraseña correctamente en el sistema. El sistema lo redirige hacia la página de inicio de sesión.				
Resultado	Aprobado				
Seguimiento	N/A				
Severidad	ALTO	MADIO	BAJO		
	N/A	N/A	N/A		

## CASO DE PRUEBA MEDIDAS CORPORALES

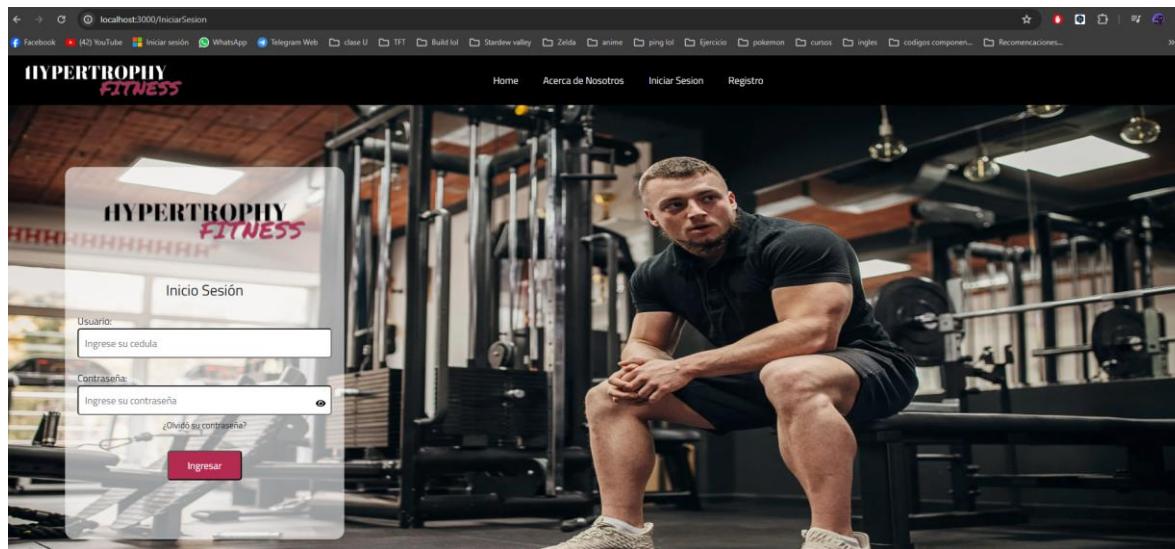
HIPERTROPHY FITNESS					
Autor del caso de prueba	Brayan Alexander Puentes Martinez				
ID	01				
Fecha de revisión	7/05/2024				
Descripción	Postman	<ul style="list-style-type: none"> <li>• GET.</li> <li>• POST.</li> <li>• PUT.</li> <li>• DELETE</li> </ul>	Ambiente de prueba	<ul style="list-style-type: none"> <li>• Versión web.</li> <li>• Windows.</li> <li>• Navegador Google chrome</li> </ul>	
	Jmeter	<ul style="list-style-type: none"> <li>• GET</li> </ul>			
Herramienta utilizada	<ul style="list-style-type: none"> <li>• Postman.</li> <li>• Jmeter.</li> </ul>				
Datos de entrada	<ul style="list-style-type: none"> <li>• Hombros.</li> <li>• Pecho.</li> <li>• Cintura.</li> <li>• Glúteos.</li> <li>• Bíceps derecho.</li> <li>• Bíceps izquierdo.</li> <li>• Antebrazo derecho.</li> <li>• Antebrazo izquierdo.</li> <li>• Pierna derecha.</li> <li>• Pierna izquierda.</li> </ul>				
Salida esperada	El usuario puede registrar sus medidas corporales exitosamente.				
Salida obtenida	El usuario registra correctamente sus medidas corporales en el sistema.				
Resultado	Aprobado				
Seguimiento	N/A				
Severidad	ALTO		MADIO	BAJO	
	N/A		N/A	N/A	

# REPORTE DE PLAN DE PRUEBAS

## RUTA INICIAR SESIÓN

URL: <http://localhost:3000/IniciarSesion>

### CAPTURAS DE PANTALLA



### CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS

#### POSTMAN

Ruta de la API: <http://localhost:4000/api/usuarios>

#### GET

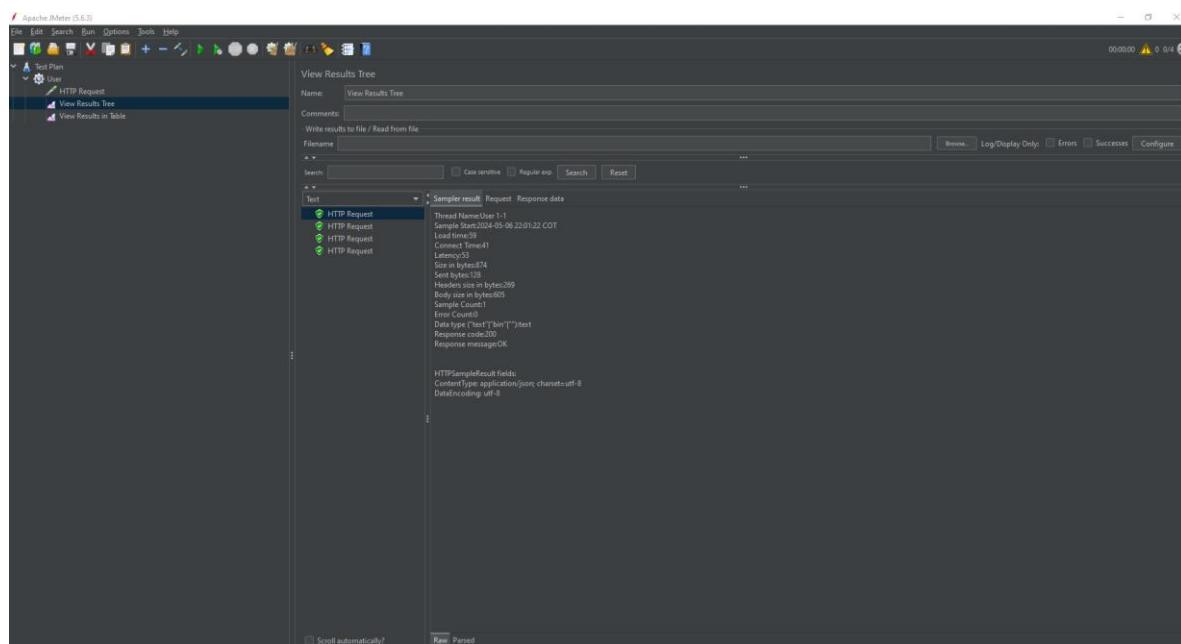
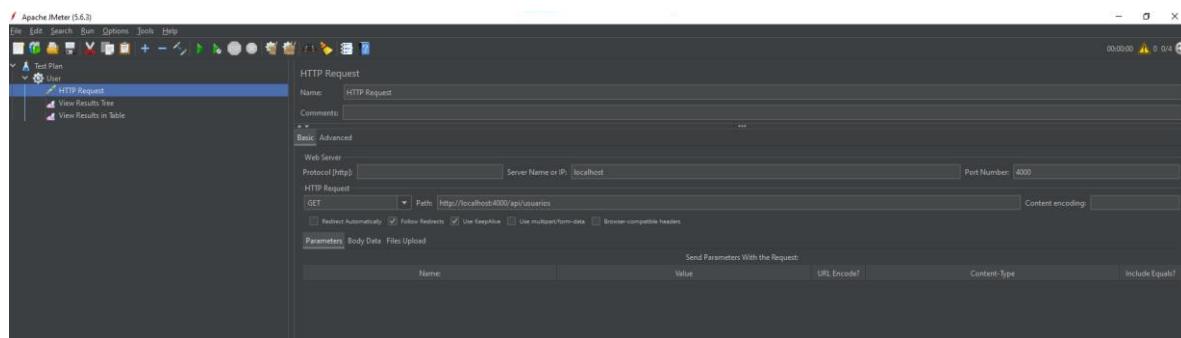
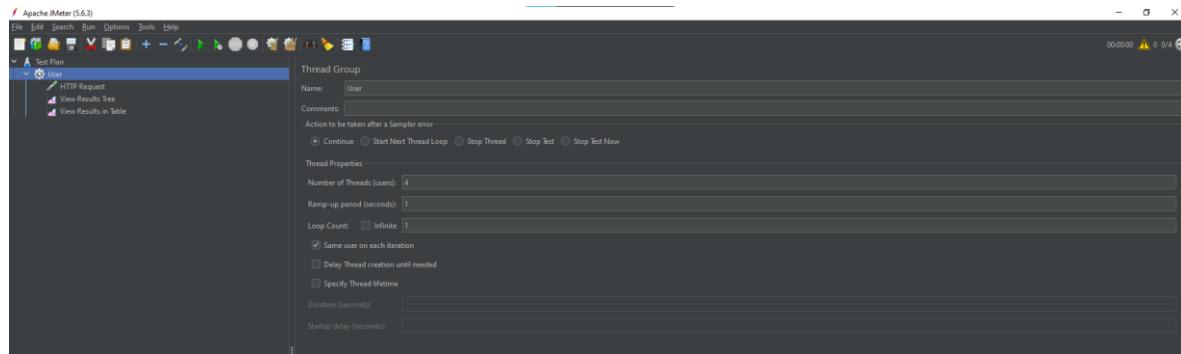
A screenshot of the Postman application interface. On the left, there's a sidebar with a collection named 'Gym pruebas' containing several requests. The main area shows a 'New Request' dialog for a 'GET' request to 'http://localhost:4000/api/usuarios'. The 'Body' tab is selected, showing a JSON response. The response body is a list of user objects, each with properties like 'exitoso', 'status', and 'body'. The 'body' field contains multiple user records, each with 'Cedula', 'Nombre\_Completo', and 'Celular' fields. The JSON is formatted with line numbers and indentation for readability.

En este caso solo se van a realizar pruebas con el método GET ya que el usuario previamente se registró a la página web y el método GET es para consultar si el usuario existe en la base de datos. Las pruebas del POST, PUT y DELETE se van a realizar cuando el usuario se registre.

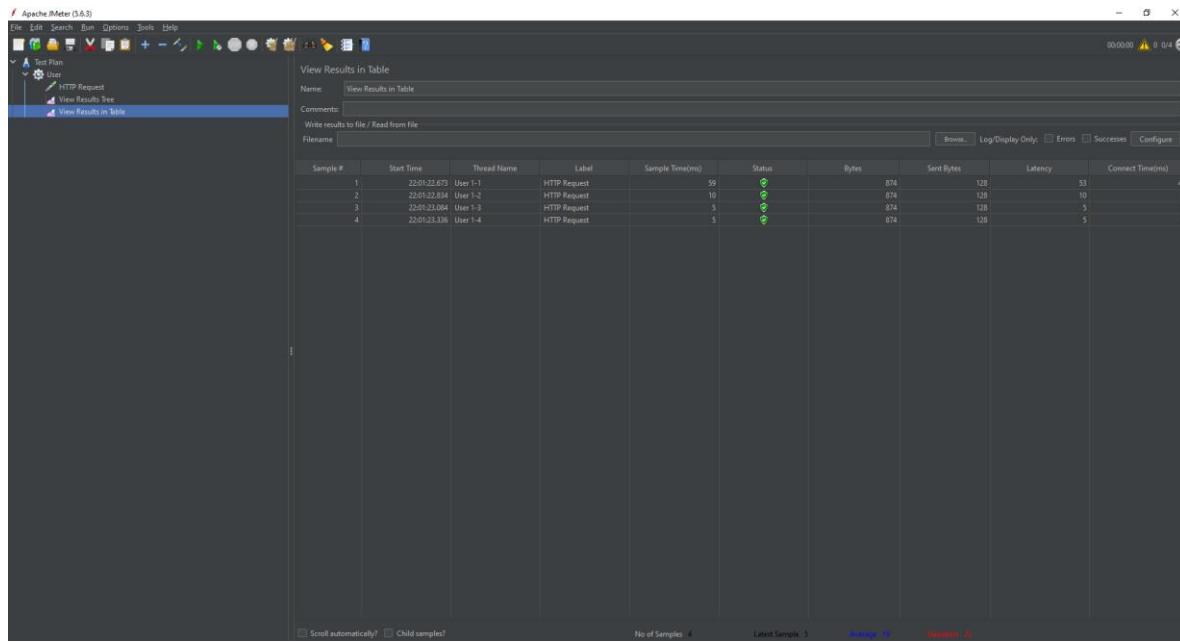
## JMETER

### Prueba con 4 usuarios

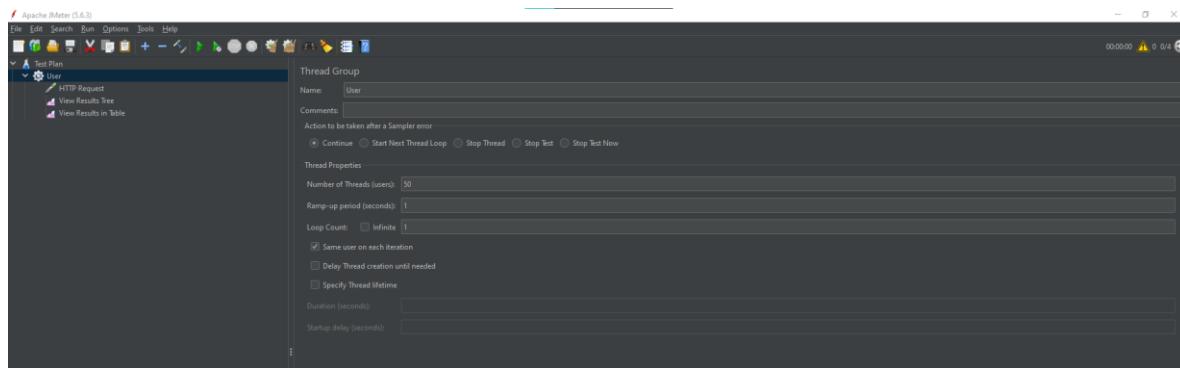
Ruta de la API: <http://localhost:4000/api/usuarios>



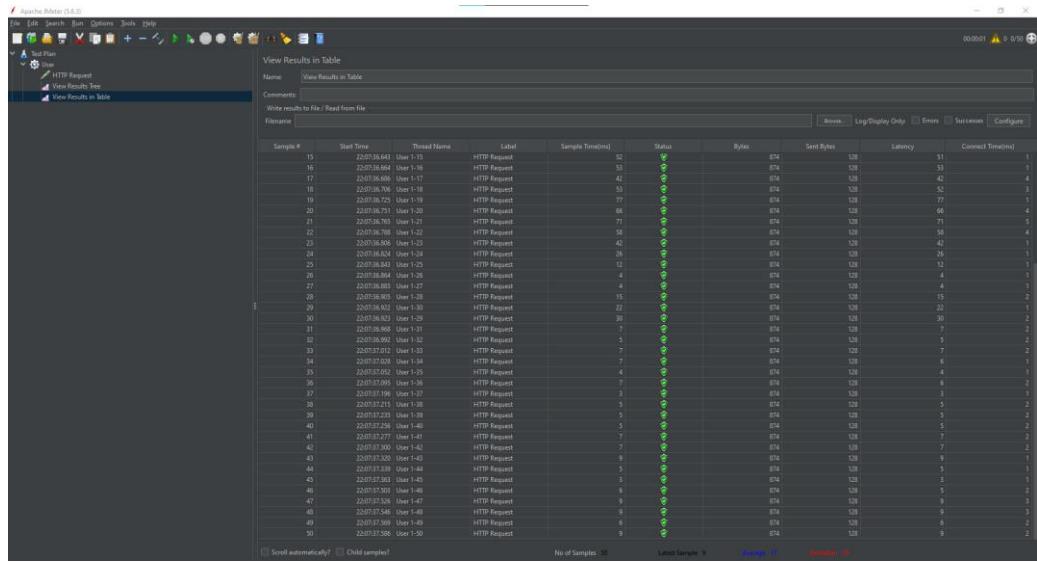
## Tiempos de respuesta



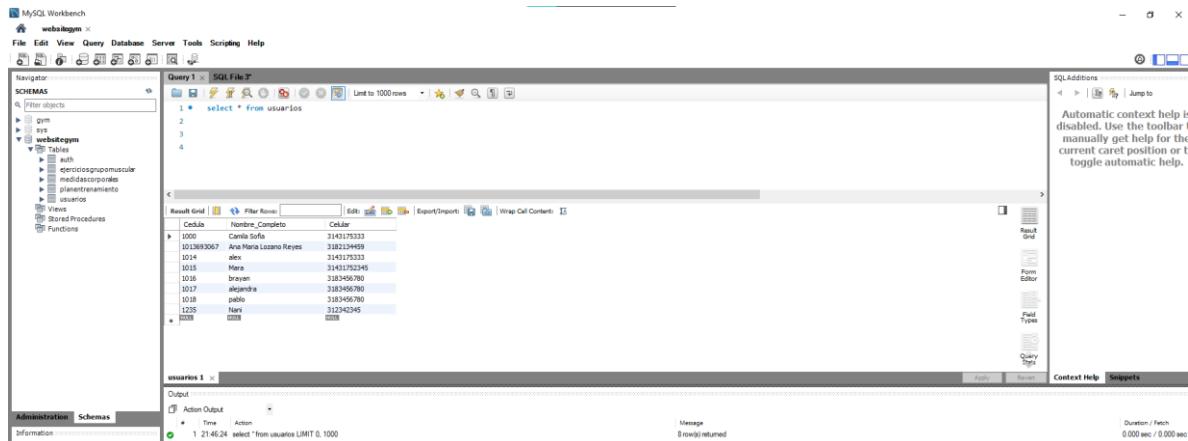
## Prueba con 50 usuarios



## Tiempos de respuesta



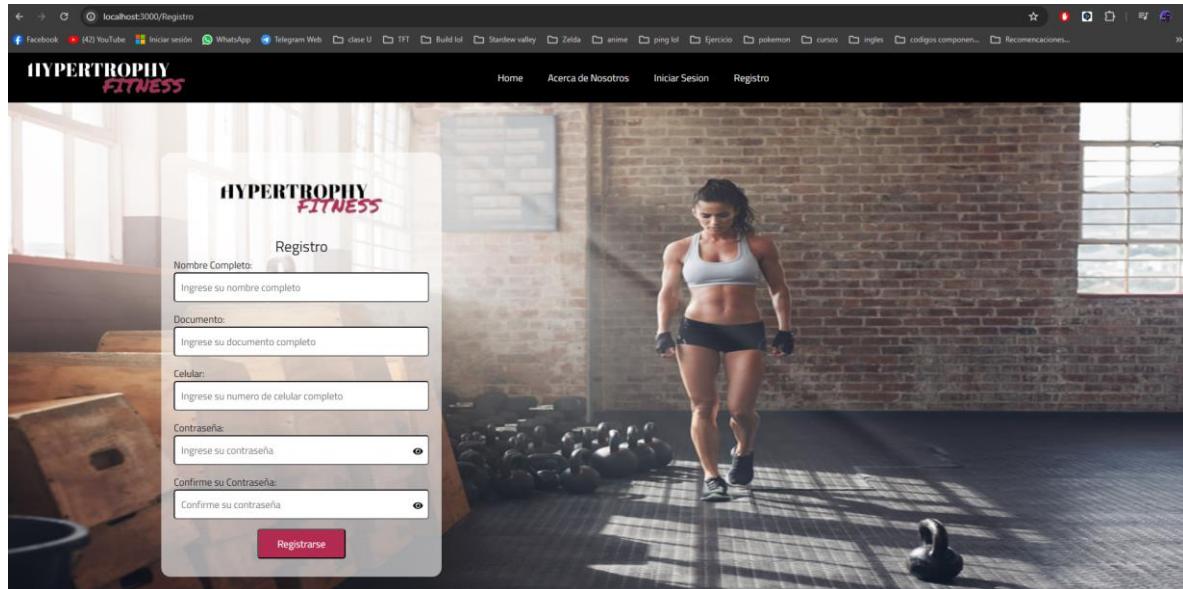
## BASE DE DATOS



## RUTA DE REGISTRO

URL: <http://localhost:3000/Registro>

### CAPTURAS DE PANTALLA



### CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS

#### POSTMAN

Ruta de la API: <http://localhost:4000/api/usuarios>

#### POST

A screenshot of the Postman application interface. On the left, there's a sidebar with collections like "Gym pruebas" and "Users". In the center, a "New Request" dialog is open for a "POST" method to the URL "http://localhost:4000/api/usuarios". The "Body" tab is selected, showing a JSON payload with multiple user objects. Each object has a "Cedula" (ID) and "Nombre\_Completo" (Full Name) field, along with a "Celular" (Cellular) field. The "Pretty" tab shows the JSON code, and the "Test Results" tab shows a successful response with status 200 OK and time 7 ms. At the bottom, there are tabs for "Pretty", "Raw", "Preview", "Visualize", and "JSON".

POST <http://localhost:4000/api/usuarios>

```

1 {
2   "cedula": "41050031",
3   "Nombre_Completo": "Sandra Rocio Cardenas Lopez",
4   "celular": "3182346699",
5   "Contraseña": "Sandy123"
6 }

```

Status: 201 Created Time: 20 ms Size: 333 B Save as example

GET <http://localhost:4000/api/usuarios>

```

1 [
2   {
3     "cedula": "41050031",
4     "Nombre_Completo": "Sandra Rocio Cardenas Lopez",
5     "celular": "3182346699",
6     "Contraseña": "Sandy123"
7   }
8 ]
9
10 [
11   {
12     "cedula": "101A",
13     "Nombre_Completo": "maria",
14     "celular": "31431752345"
15   },
16   {
17     "cedula": "101A",
18     "Nombre_Completo": "bryan",
19     "celular": "3183466788"
20   },
21   {
22     "cedula": "1017",
23     "Nombre_Completo": "alejandra",
24     "celular": "3183466789"
25   },
26   {
27     "cedula": "1018",
28     "Nombre_Completo": "pablo",
29     "celular": "3183466788"
30   },
31   {
32     "cedula": "1238",
33     "Nombre_Completo": "Nani",
34     "celular": "3123424348"
35   },
36   {
37     "cedula": "41560342",
38     "Nombre_Completo": "Sandra Rocio Cardenas Lopez",
39     "celular": "3182346699"
40   }
41 ]
42
43 [
44   {
45     "cedula": "41560342",
46     "Nombre_Completo": "Sandra Rocio Cardenas Lopez",
47     "celular": "3182346699"
48   }
49 ]
50
51 ]

```

Status: 200 OK Time: 7 ms Size: 967 B Save as example

## PUT

PUT <http://localhost:4000/api/usuarios>

```

1 {
2   "cedula": "41050031",
3   "Nombre_Completo": "Sandra Rocio Cardenas Lopez",
4   "celular": "3182346699"
5 }

```

Status: 200 OK Time: 8 ms Size: 967 B Save as example

Postman API Network - Gym pruebas / Usuarios - New Request

PUT http://localhost:4000/api/usuarios

Body (raw, JSON)

```

1 {
2   "Cedula": "41058342",
3   "Nombre_Completo": "Sandra Rocío Cárdenas López",
4   "Celular": "39861236579"
5 }

```

Test Results (Pretty, Raw, Preview, Visualize, JSON)

```

1 {
2   "error": false,
3   "status": 201,
4   "body": "Item actualizado con éxito."
5 }

```

Status: 201 Created Time: 15 ms Size: 336 B Save as example

Postman API Network - Gym pruebas / Usuarios - New Request

GET http://localhost:4000/api/usuarios

Body (raw, JSON)

```

1 [
2   {
3     "Cedula": "31050345",
4     "Nombre_Completo": "Sandra Rocío Cárdenas López",
5     "Celular": "39861236579"
6   },
7   {
8     "Cedula": "1816",
9     "Nombre_Completo": "brayan",
10    "Celular": "3183464708"
11  },
12  {
13    "Cedula": "1817",
14    "Nombre_Completo": "alejandra",
15    "Celular": "3183464708"
16  },
17  {
18    "Cedula": "1818",
19    "Nombre_Completo": "pablo",
20    "Celular": "3183464708"
21  },
22  {
23    "Cedula": "1235",
24    "Nombre_Completo": "Nani",
25    "Celular": "3123423405"
26  },
27  [
28    {
29      "Cedula": "41058342",
30      "Nombre_Completo": "Sandra Rocío Cárdenas López",
31      "Celular": "39861236579"
32    }
33  ]
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

Status: 200 OK Time: 8 ms Size: 987 B Save as example

## DELETE

Postman API Network - Gym pruebas / Usuarios - New Request

DELETE http://localhost:4000/api/usuarios

Body (raw, JSON)

```

1 [
2   {
3     "Cedula": "41058342"
4   }
5 ]

```

Test Results (Pretty, Raw, Preview, Visualize, JSON)

```

1 [
2   {
3     "Cedula": "3885",
4     "Nombre_Completo": "Mass",
5     "Celular": "314317823465"
6   },
7   {
8     "Cedula": "1816",
9     "Nombre_Completo": "brayan",
10    "Celular": "3183464708"
11  },
12  {
13    "Cedula": "1817",
14    "Nombre_Completo": "alejandra",
15    "Celular": "3183464708"
16  },
17  {
18    "Cedula": "1818",
19    "Nombre_Completo": "pablo",
20    "Celular": "3183464708"
21  },
22  {
23    "Cedula": "1235",
24    "Nombre_Completo": "Nani",
25    "Celular": "3123423405"
26  },
27  [
28    {
29      "Cedula": "41058342",
30      "Nombre_Completo": "Sandra Rocío Cárdenas López",
31      "Celular": "39861236579"
32    }
33  ]
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

Status: 200 OK Time: 7 ms Size: 967 B Save as example

**DELETE Request:**

```

1: {
2:   "Cedula": "416568342"
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:
25:
26:
27:
28:
29:
30:
31:
32:
33:
34:
35:
36:
37:
38:
39:
40:
41:
42:
43:
44:
45:
46:

```

**GET Request Response:**

```

1: [
2:   {
3:     "Nombre_Completo": "alex",
4:     "Celular": "3143176333"
5:   },
6:   {
7:     "Cedula": "49161",
8:     "Nombre_Completo": "Mara",
9:     "Celular": "31431762345"
10: },
11:   {
12:     "Cedula": "49164",
13:     "Nombre_Completo": "brayan",
14:     "Celular": "31834646708"
15: },
16:   {
17:     "Cedula": "49167",
18:     "Nombre_Completo": "alejandra",
19:     "Celular": "31834646708"
20: },
21:   {
22:     "Cedula": "49168",
23:     "Nombre_Completo": "sabio",
24:     "Celular": "31834646708"
25: },
26:   {
27:     "Cedula": "49256",
28:     "Nombre_Completo": "Nani",
29:     "Celular": "312342348"
30: }
31:
32:
33:
34:
35:
36:
37:
38:
39:
40:
41:
42:
43:
44:
45:
46:

```

## JMETER

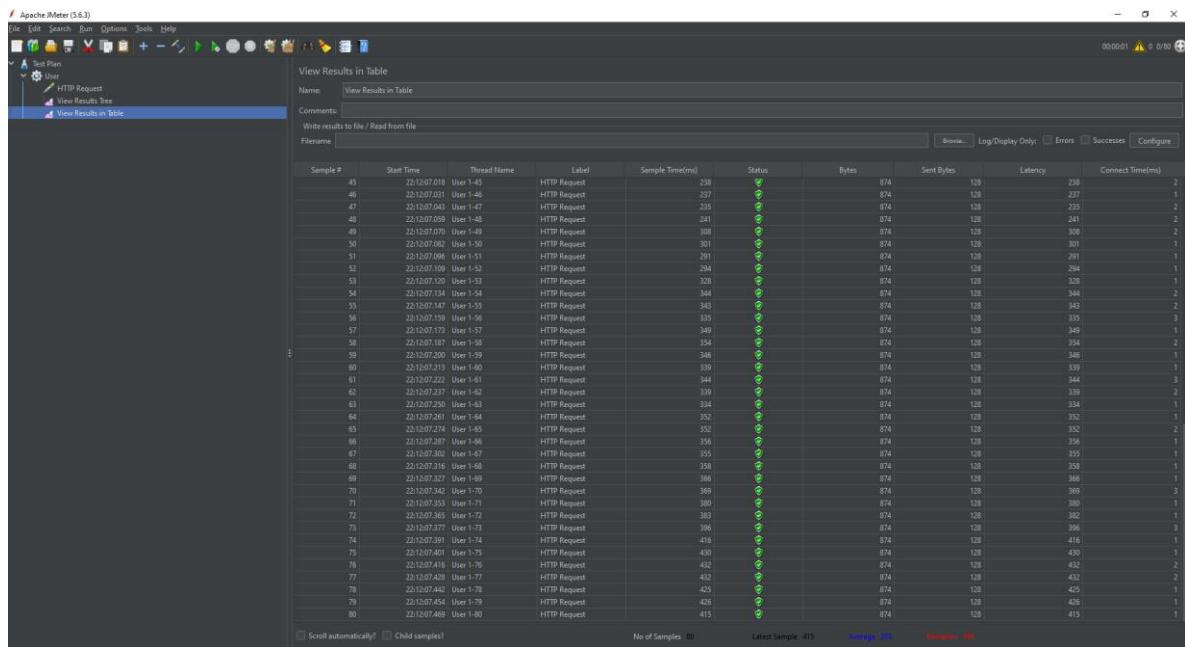
Prueba con 80 usuarios

Ruta de la API: <http://localhost:4000/api/usuarios>

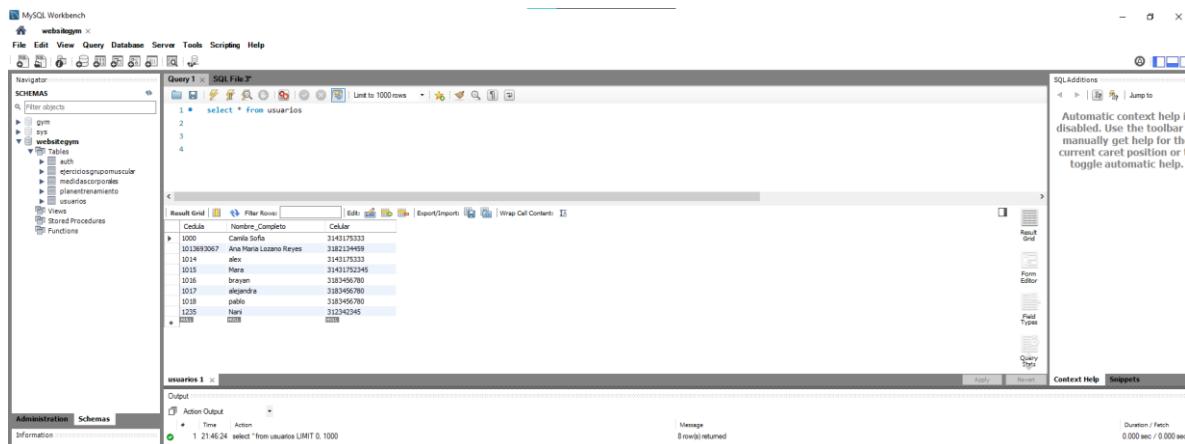
**Thread Group Settings:**

- Name: User
- Comments:
- Action to be taken after a Sampler error:
  - Continue
  - Start Next Thread Loop
  - Stop Thread
  - Stop Test
  - Stop Test Now
- Number of Threads (users): 80
- Ramp-up period (seconds): 1
- Loop Count: 1
- Same user on each iteration:
- Delay Thread creation until needed:
- Specify Thread lifetime:
- Duration (seconds):
- Startup delay (seconds):

## Tiempos de respuesta



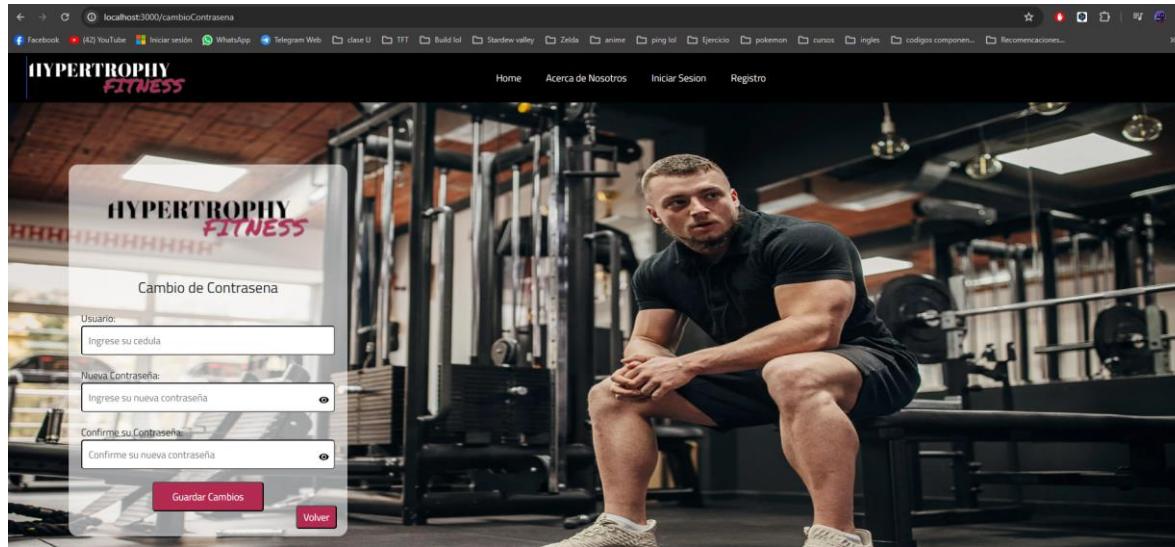
## BASE DE DATOS



## RUTA DE CAMBIO DE CONTRASEÑA

URL: <http://localhost:3000/cambioContrasena>

### CAPTURAS DE PANTALLA



### CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS

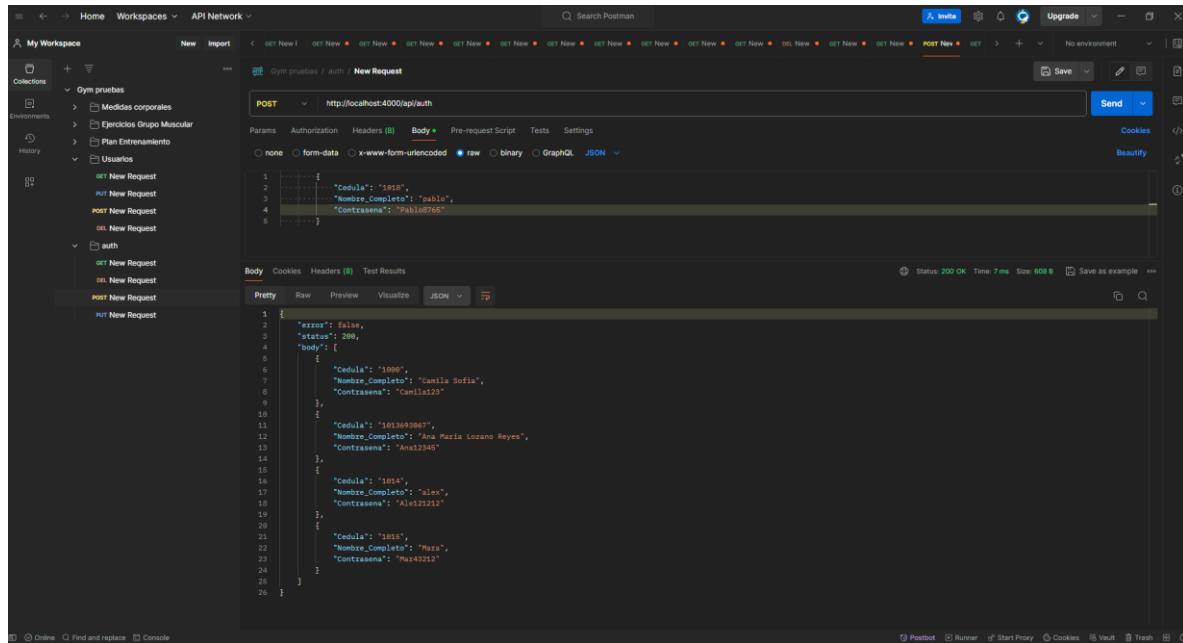
#### POSTMAN

Ruta de la API: <http://localhost:4000/api/auth>

#### GET

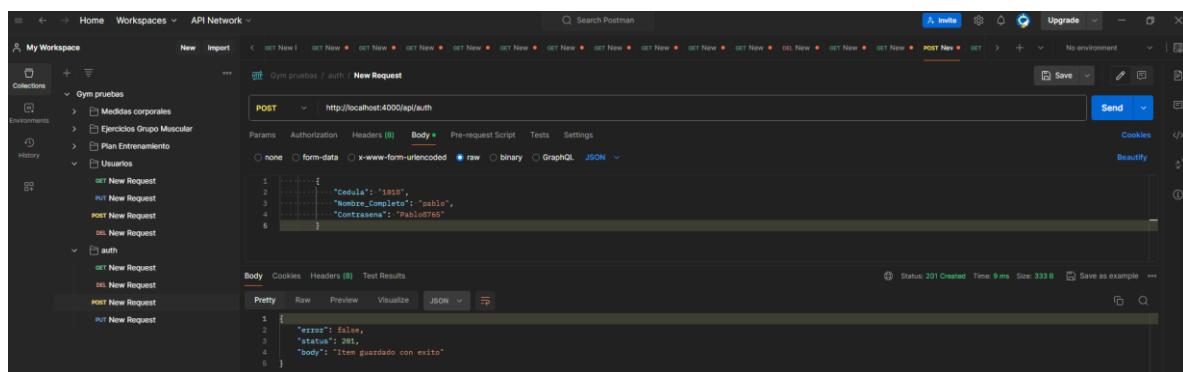
La captura de pantalla de Postman muestra una solicitud GET dirigida a la URL <http://localhost:4000/api/auth>. La respuesta JSON devolvió una lista de usuarios con sus respectivas cédulas y contraseñas. Aquí se presentan los datos devueltos:

# POST



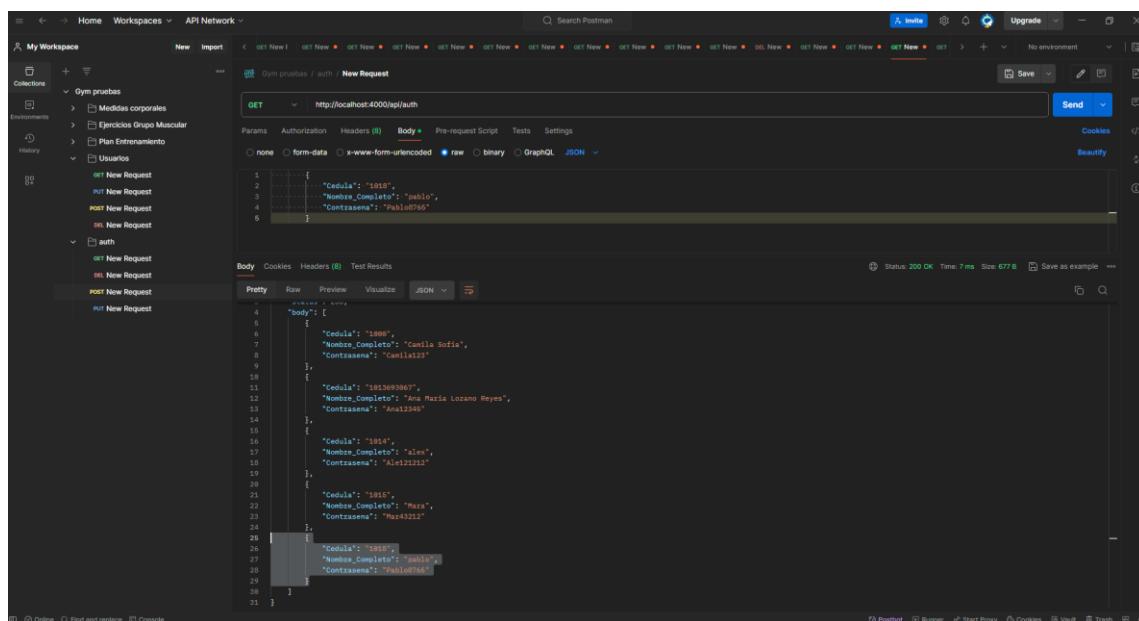
Postman screenshot showing a POST request to `http://localhost:4000/api/auth`. The request body is a JSON array of user records:

```
[{"Cedula": "1000", "Nombre_Completo": "Camila Sofia", "Contrasena": "Camila123"}, {"Cedula": "1013938677", "Nombre_Completo": "Ana Maria Lozano Reyes", "Contrasena": "Ana12345"}, {"Cedula": "1014", "Nombre_Completo": "alex", "Contrasena": "Aled12312"}, {"Cedula": "1015", "Nombre_Completo": "Mara", "Contrasena": "Mara43212"}]
```



Postman screenshot showing a POST request to `http://localhost:4000/api/auth`. The request body is a JSON object representing a single user record:

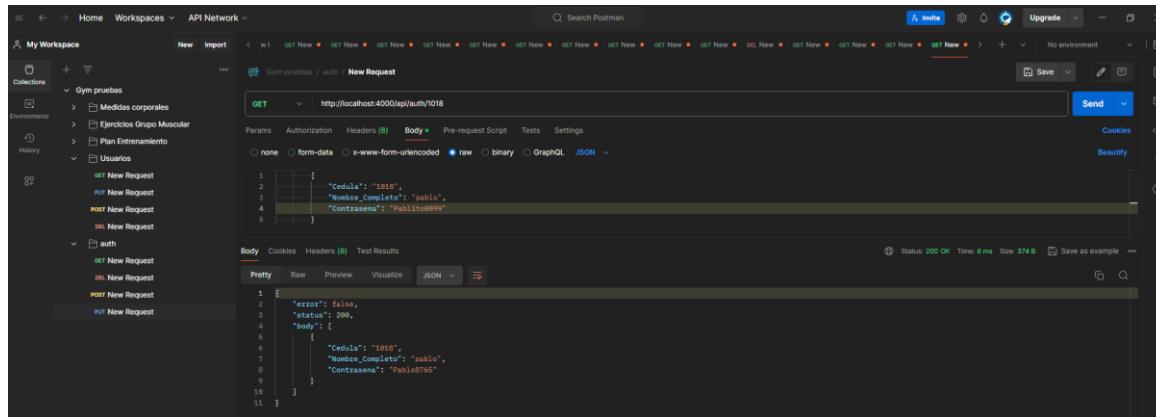
```
{"Cedula": "1000", "Nombre_Completo": "pablo", "Contrasena": "Pabolo765"}
```



Postman screenshot showing a GET request to `http://localhost:4000/api/auth`. The response body is a JSON array of user records:

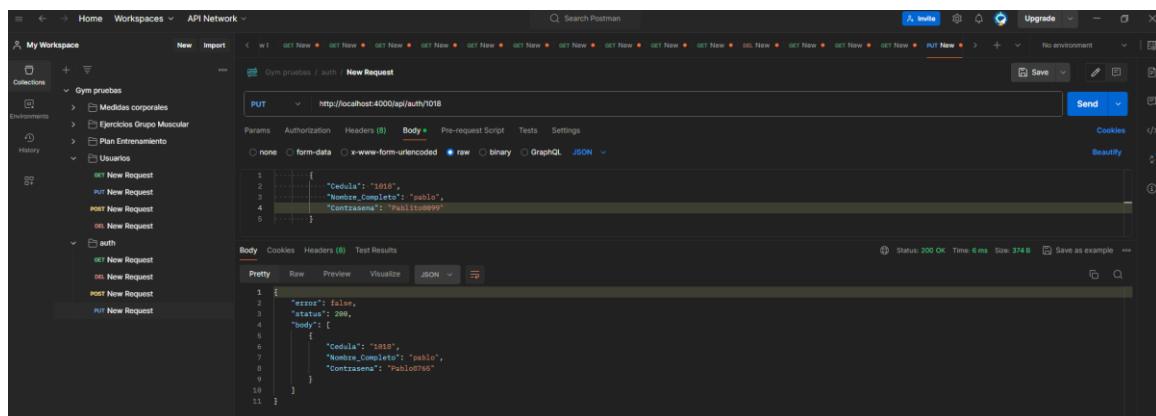
```
[{"Cedula": "1000", "Nombre_Completo": "Camila Sofia", "Contrasena": "Camila123"}, {"Cedula": "1013938677", "Nombre_Completo": "Ana Maria Lozano Reyes", "Contrasena": "Ana12345"}, {"Cedula": "1014", "Nombre_Completo": "alex", "Contrasena": "Aled12312"}, {"Cedula": "1015", "Nombre_Completo": "Mara", "Contrasena": "Mara43212"}, {"Cedula": "1016", "Nombre_Completo": "pablo", "Contrasena": "Pabolo765"}]
```

## PUT



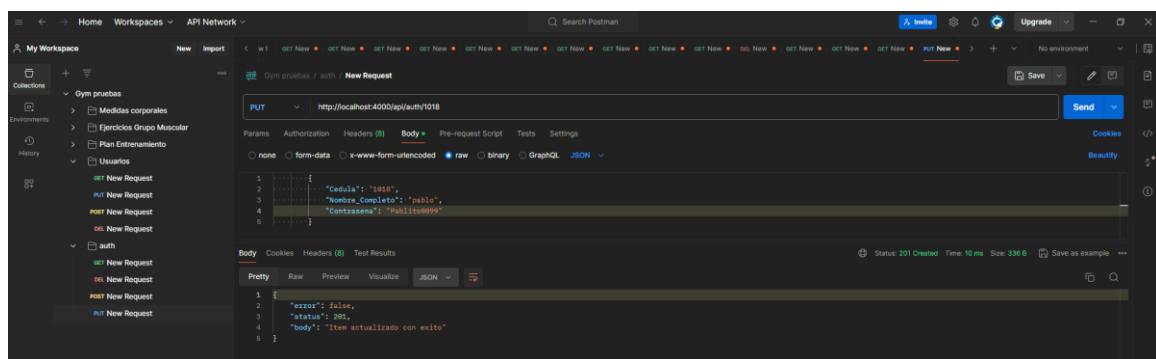
Postman screenshot showing a successful PUT request to `http://localhost:4000/api/auth/1018`. The response status is 200 OK, and the response body is:

```
[{"error": false, "status": 200, "body": [{"Cedula": "1010", "Nombre_Completo": "pablo", "Contrasena": "Pablito0999"}]}
```



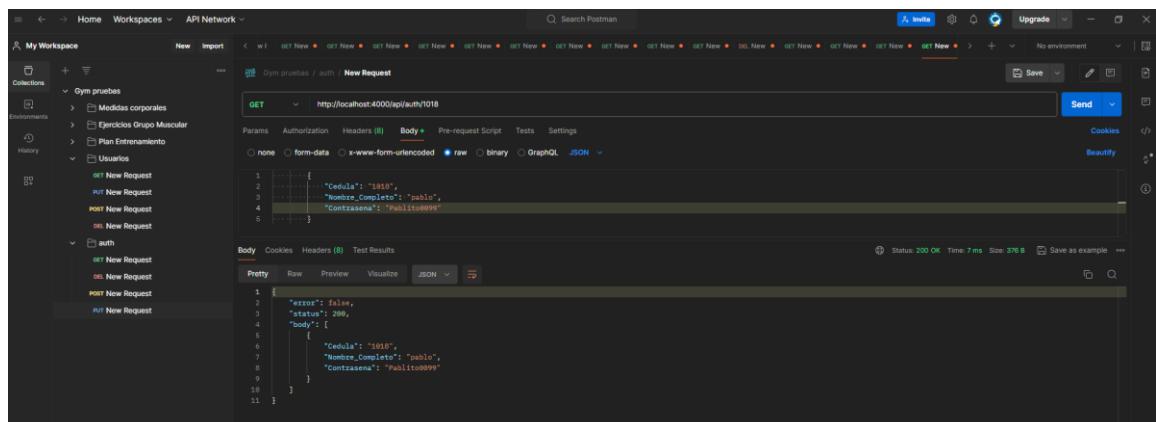
Postman screenshot showing a successful PUT request to `http://localhost:4000/api/auth/1018`. The response status is 200 OK, and the response body is:

```
[{"error": false, "status": 200, "body": [{"Cedula": "1010", "Nombre_Completo": "pablo", "Contrasena": "Pablito0999"}]}
```



Postman screenshot showing a successful PUT request to `http://localhost:4000/api/auth/1018`. The response status is 201 Created, and the response body is:

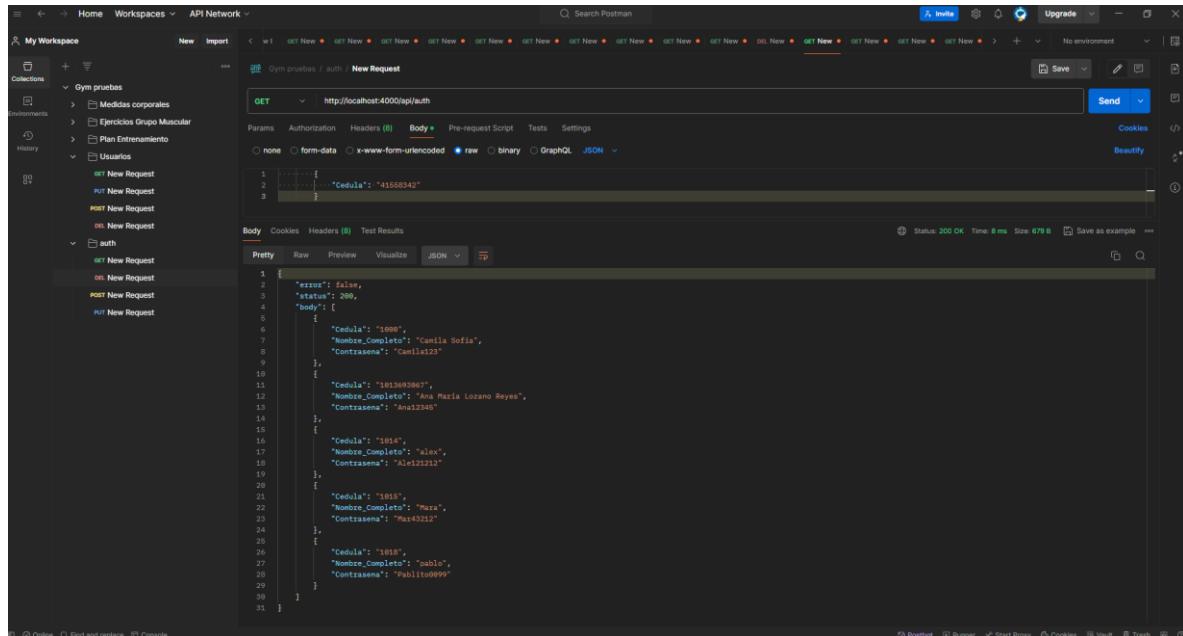
```
[{"error": false, "status": 201, "body": {"mensaje": "Item actualizado con exito"}}
```



Postman screenshot showing a successful GET request to `http://localhost:4000/api/auth/1018`. The response status is 200 OK, and the response body is:

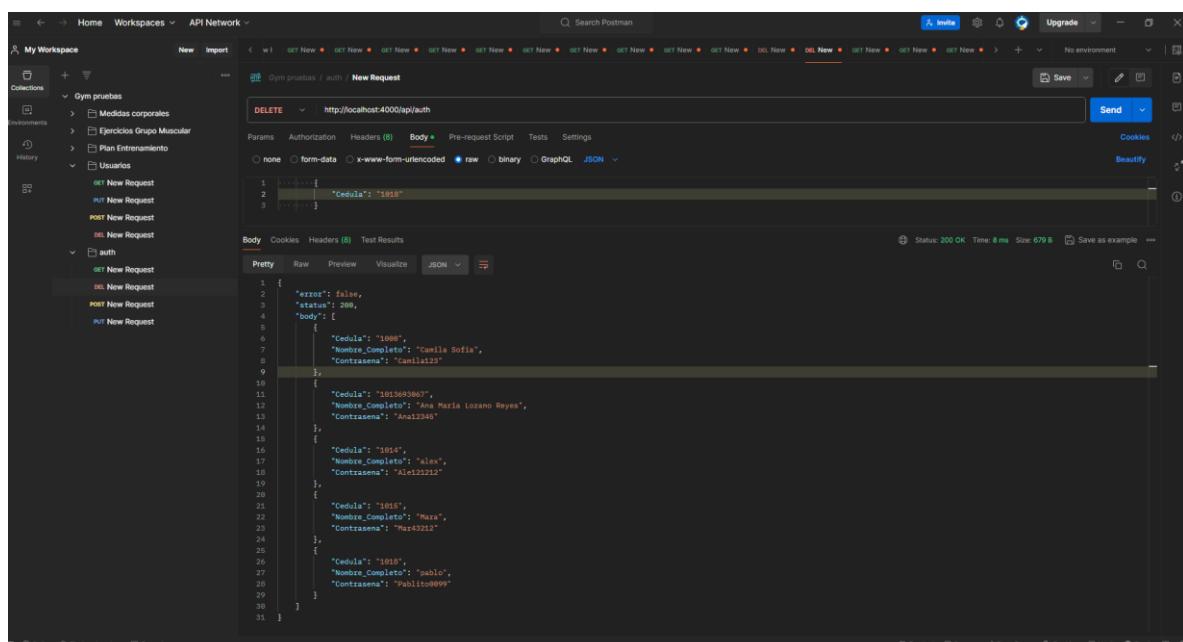
```
[{"error": false, "status": 200, "body": [{"Cedula": "1010", "Nombre_Completo": "pablo", "Contrasena": "Pablito0999"}]}
```

# DELETE



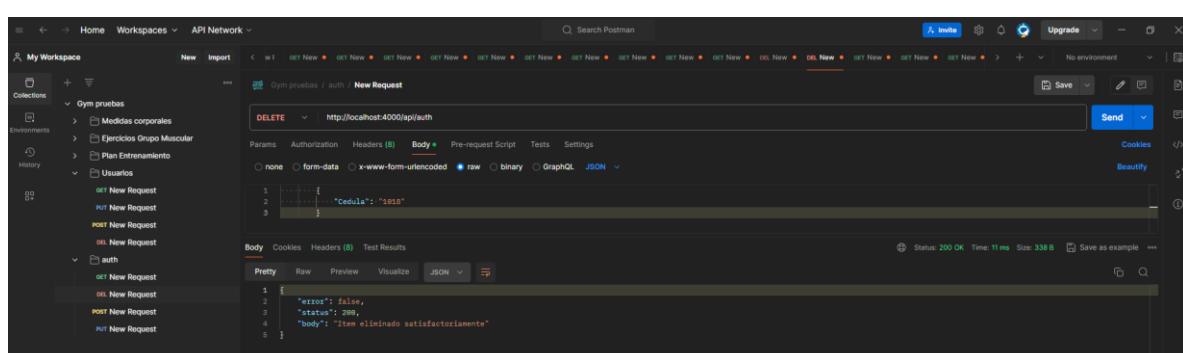
Postman screenshot showing a successful GET request to `http://localhost:4000/api/auth`. The response is a JSON array of users:

```
[{"Cedula": "1098", "Nombre_Completo": "Camila Sofia", "Contraseña": "Camila23"}, {"Cedula": "10136903867", "Nombre_Completo": "Ana Maria Lozano Reyes", "Contraseña": "Anal2345"}, {"Cedula": "1014", "Nombre_Completo": "alex", "Contraseña": "Alie1231212"}, {"Cedula": "1015", "Nombre_Completo": "Maria", "Contraseña": "Mar4321"}, {"Cedula": "1018", "Nombre_Completo": "pablo", "Contraseña": "Pabito0999"}]
```



Postman screenshot showing a successful DELETE request to `http://localhost:4000/api/auth` with Cedula: 1018. The response is a JSON array of users:

```
[{"Cedula": "1098", "Nombre_Completo": "Camila Sofia", "Contraseña": "Camila23"}, {"Cedula": "10136903867", "Nombre_Completo": "Ana Maria Lozano Reyes", "Contraseña": "Anal2345"}, {"Cedula": "1014", "Nombre_Completo": "alex", "Contraseña": "Alie1231212"}, {"Cedula": "1015", "Nombre_Completo": "Maria", "Contraseña": "Mar4321"}, {"Cedula": "1018", "Nombre_Completo": "pablo", "Contraseña": "Pabito0999"}]
```



Postman screenshot showing a successful DELETE request to `http://localhost:4000/api/auth` with Cedula: 1018. The response is a JSON object:

```
{"error": false, "status": 200, "body": "Item eliminado satisfactoriamente"}
```

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections like 'Gym pruebas', environments, and history. In the center, a 'New Request' dialog is open for a 'GET' method to 'http://localhost:4000/api/auth'. The 'Body' tab is selected, showing a JSON payload:

```

1  {
2     "errors": false,
3     "status": 200,
4     "body": [
5         {
6             "Cedula": "1000",
7             "Nombre_Completo": "Camila Sofia",
8             "Contrasenna": "Camila123"
9         },
10        {
11            "Cedula": "0103090907",
12            "Nombre_Completo": "Ana Maria Lozano Reyes",
13            "Contrasenna": "Ana123456"
14        },
15        {
16            "Cedula": "1004",
17            "Nombre_Completo": "alex",
18            "Contrasenna": "Alex123456"
19        },
20        {
21            "Cedula": "1010",
22            "Nombre_Completo": "Marta",
23            "Contrasenna": "Marta3212"
24        }
25    ]
26 }

```

At the bottom right, the status bar shows 'Status: 200 OK Time: 7ms Size: 608 B'. Below the main window, a toolbar includes 'Online', 'Find and replace', 'Console', and other Postman-specific icons.

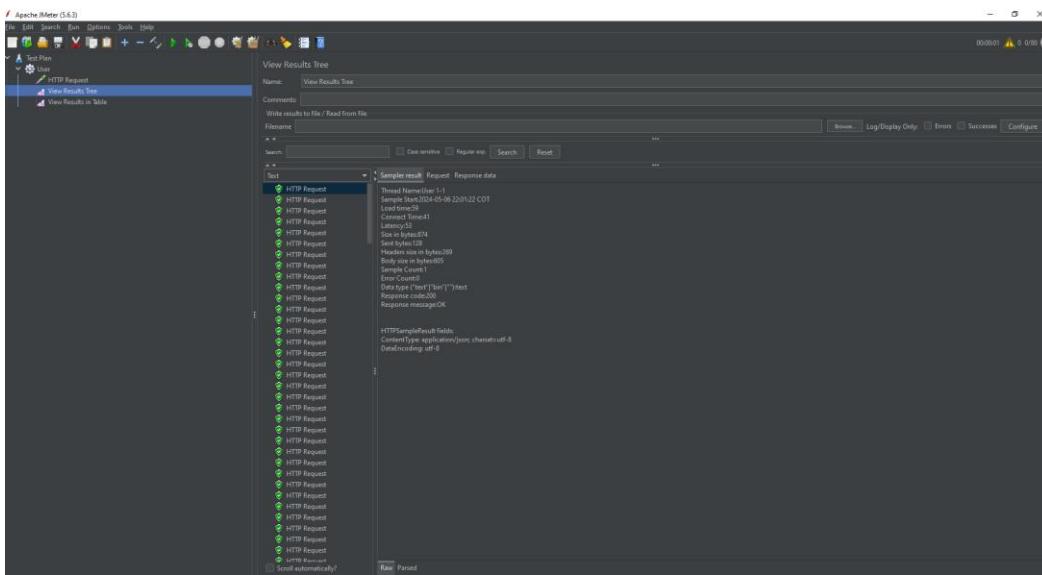
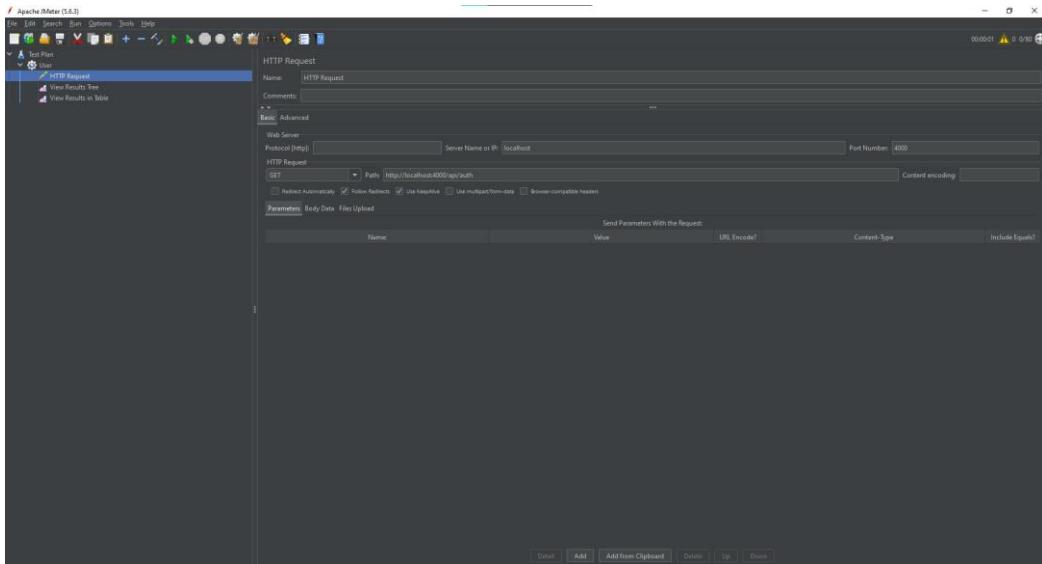
## JMETER

Prueba con 80 usuarios

Ruta de la API: <http://localhost:4000/api/auth>

The screenshot shows the Apache JMeter 5.6.2 interface. On the left, the 'Test Plan' tree view shows a 'User' thread group containing an 'HTTP Request' sampler. The 'User' thread group properties are displayed on the right:

- Name:** User
- Comments:**
- Action to be taken after a Sampler error:** Continue
- Thread Properties:**
  - Number of Threads (users):** 80
  - Ramp-up period (seconds):** 1
  - Loop Count:** infinite
  - Same user on each iteration
  - Delay Thread creation until needed
  - Specify Thread lifetime
- Duration (seconds):**
- Startup delay (seconds):**



## Tiempos de respuesta

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
45	23/11/2018 1:45	User 1-45	HTTP Request	381	OK	608	608	124	385
46	23/11/2018 1:46	User 1-46	HTTP Request	361	OK	608	608	124	363
47	23/11/2018 1:47	User 1-47	HTTP Request	361	OK	608	608	124	363
48	23/11/2018 1:48	User 1-48	HTTP Request	384	OK	608	608	124	384
49	23/11/2018 1:49	User 1-49	HTTP Request	397	OK	608	608	124	397
50	23/11/2018 1:50	User 1-50	HTTP Request	380	OK	608	608	124	380
51	23/11/2018 1:51	User 1-51	HTTP Request	402	OK	608	608	124	402
52	23/11/2018 1:52	User 1-52	HTTP Request	389	OK	608	608	124	389
53	23/11/2018 1:53	User 1-53	HTTP Request	381	OK	608	608	124	381
54	23/11/2018 1:54	User 1-54	HTTP Request	273	OK	608	608	124	273
55	23/11/2018 1:55	User 1-55	HTTP Request	261	OK	608	608	124	261
56	23/11/2018 1:56	User 1-56	HTTP Request	260	OK	608	608	124	260
57	23/11/2018 1:57	User 1-57	HTTP Request	272	OK	608	608	124	272
58	23/11/2018 1:58	User 1-58	HTTP Request	251	OK	608	608	124	250
59	23/11/2018 1:59	User 1-59	HTTP Request	251	OK	608	608	124	251
60	23/11/2018 1:59	User 1-60	HTTP Request	250	OK	608	608	124	250
61	23/11/2018 1:59	User 1-61	HTTP Request	251	OK	608	608	124	251
62	23/11/2018 1:59	User 1-62	HTTP Request	270	OK	608	608	124	270
63	23/11/2018 1:59	User 1-63	HTTP Request	268	OK	608	608	124	268
64	23/11/2018 1:59	User 1-64	HTTP Request	261	OK	608	608	124	261
65	23/11/2018 1:59	User 1-65	HTTP Request	261	OK	608	608	124	261
66	23/11/2018 1:59	User 1-66	HTTP Request	251	OK	608	608	124	251
67	23/11/2018 1:59	User 1-67	HTTP Request	212	OK	608	608	124	212
68	23/11/2018 1:59	User 1-68	HTTP Request	201	OK	608	608	124	201
69	23/11/2018 1:59	User 1-69	HTTP Request	193	OK	608	608	124	193
70	23/11/2018 1:59	User 1-70	HTTP Request	190	OK	608	608	124	190
71	23/11/2018 1:59	User 1-71	HTTP Request	186	OK	608	608	124	186
72	23/11/2018 1:59	User 1-72	HTTP Request	200	OK	608	608	124	200
73	23/11/2018 1:59	User 1-73	HTTP Request	218	OK	608	608	124	218
74	23/11/2018 1:59	User 1-74	HTTP Request	220	OK	608	608	124	220
75	23/11/2018 1:59	User 1-75	HTTP Request	230	OK	608	608	124	230
76	23/11/2018 1:59	User 1-76	HTTP Request	230	OK	608	608	124	230
77	23/11/2018 1:59	User 1-77	HTTP Request	246	OK	608	608	124	225
78	23/11/2018 1:59	User 1-78	HTTP Request	231	OK	608	608	124	230
79	23/11/2018 1:59	User 1-79	HTTP Request	231	OK	608	608	124	230
80	23/11/2018 1:59	User 1-80	HTTP Request	231	OK	608	608	124	230

Scoll automatically? Child samples? No of Samples: 200 Latest Sample: 200 Average: 200 Standard Deviation: 200

## BASE DE DATOS

The screenshot shows the MySQL Workbench interface with the 'websitegym' database selected. The 'auth' table is currently being viewed. The results grid displays the following data:

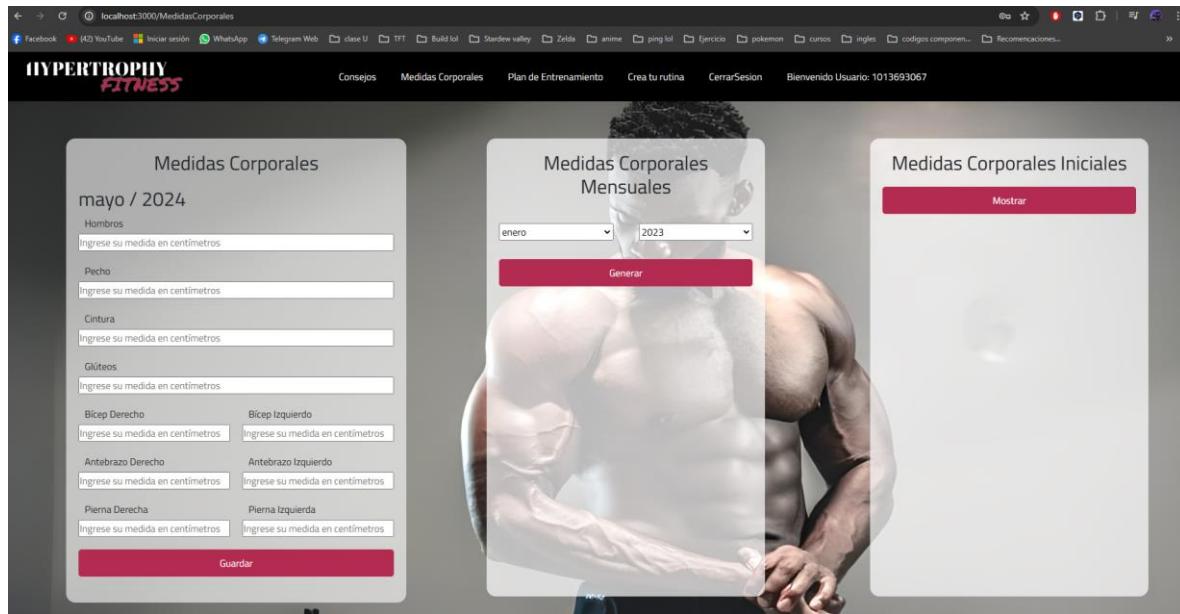
Cédula	Nombre_Completo	Contraseña
1000	Carola Sofia	Carola123
1013930067	Maria Lourdes Reyes	Ale123456
1014	alex	Ale12312
1015	Mara	Mara43212
1028	BBB	BBB

The SQL editor window contains the query: `select * from auth`. The status bar at the bottom right indicates: Duration / Fetch 0.00 sec / 0.000 sec, 4 row(s) returned.

## RUTA DE MEDIDAS CORPORALES

URL: http://localhost:3000/MedidasCorporales

### CAPTURAS DE PANTALLA



### CAPTURA DE HERRAMIENTA PARA REALIZAR PRUEBAS

#### POSTMAN

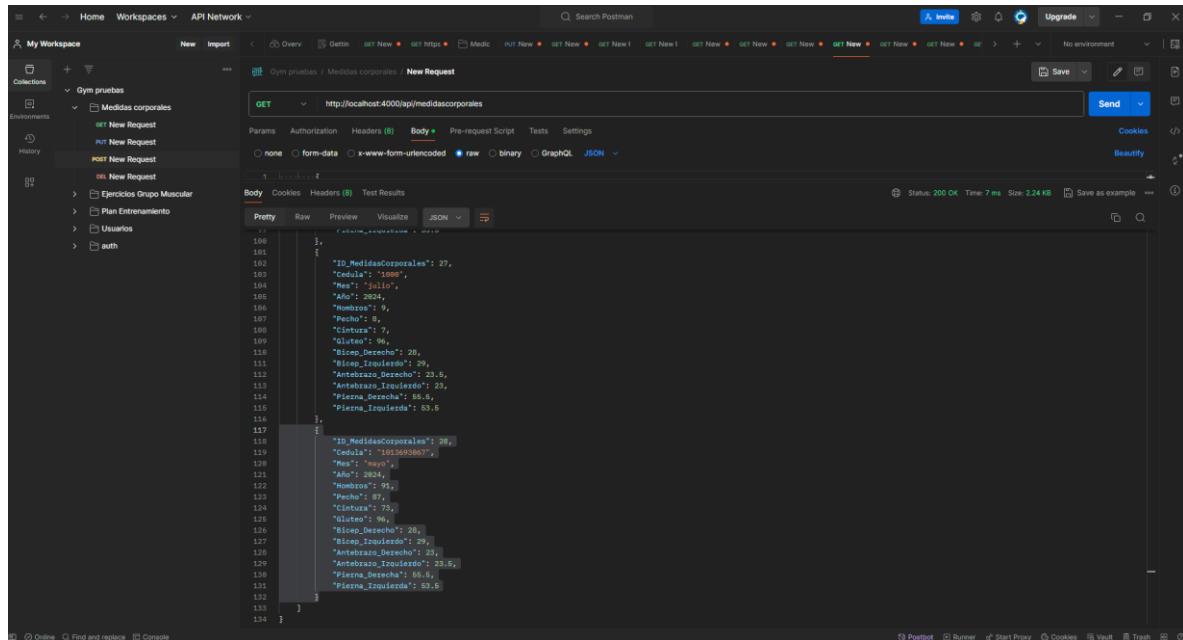
Ruta de la API: http://localhost:4000/api/medidascorporales

#### GET

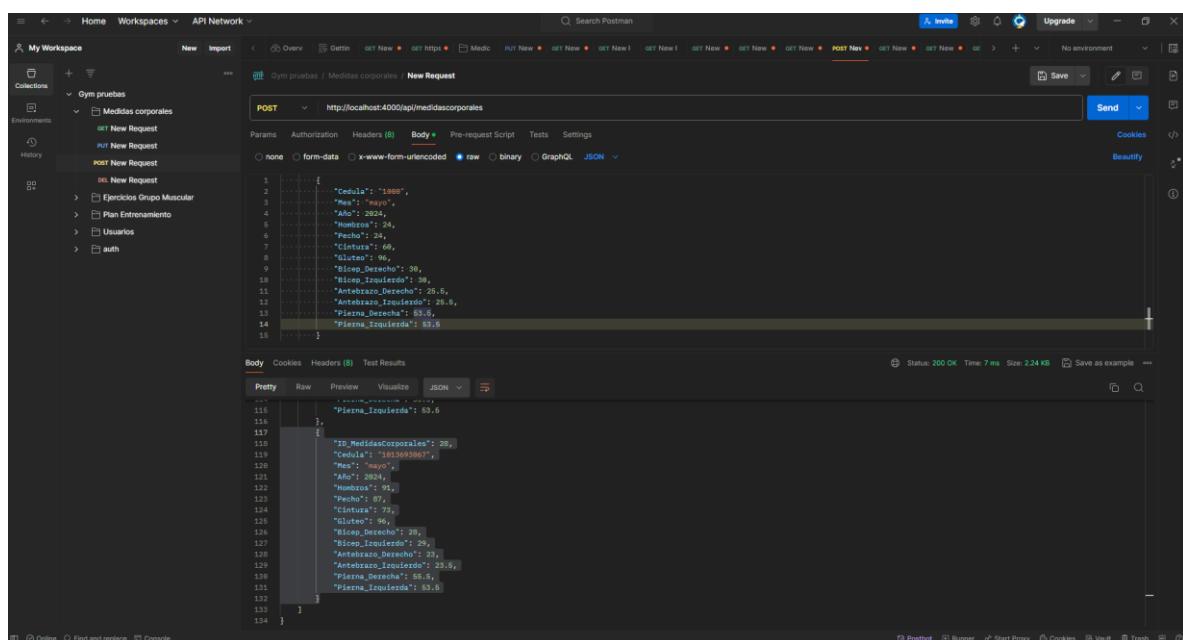
The screenshot shows a Postman interface with a collection named 'Gym pruebas'. A 'Medidas corporales' request is selected, showing a GET method to 'http://localhost:4000/api/medidascorporales'. The response body is displayed as JSON, showing two entries of measurement data:

```
180 [
181   {
182     "ID_MedidasCorporales": 27,
183     "Cedula": "1000",
184     "Mes": "Julio",
185     "Año": 2024,
186     "Hombro": 93,
187     "Pecho": 84,
188     "Cintura": 7,
189     "Gluteo": 96,
190     "Bicep_Derecho": 20,
191     "Antebrazo_Derecho": 29,
192     "Antebrazo_Izquierdo": 23.5,
193     "Bicep_Izquierdo": 23,
194     "Perna_Derecha": 68.5,
195     "Perna_Izquierda": 63.6
196   },
197   {
198     "ID_MedidasCorporales": 28,
199     "Cedula": "10136938067",
200     "Mes": "Agosto",
201     "Año": 2024,
202     "Hombro": 91,
203     "Pecho": 87,
204     "Cintura": 73,
205     "Gluteo": 94,
206     "Bicep_Derecho": 20,
207     "Antebrazo_Izquierdo": 29,
208     "Bicep_Izquierdo": 23,
209     "Perna_Derecha": 68.5,
210     "Perna_Izquierda": 63.6
211   }
212 ]
```

# POST



```
1, [
2,   {
3,     "ID_MedidasCorporales": 27,
4,     "Cedula": "10888",
5,     "Nombre": "Julio",
6,     "Apellido": "2624",
7,     "Hombres": 9,
8,     "Tallas": 7,
9,     "Cintura": 7,
10,    "Pecho": 96,
11,    "Biceps_Derecho": 29,
12,    "Biceps_Izquierdo": 29,
13,    "Antebrazo_Derecho": 23.6,
14,    "Antebrazo_Izquierdo": 23,
15,    "Pierna_Derecha": 88.6,
16,    "Pierna_Izquierda": 63.6
17,  },
18,  {
19,    "ID_MedidasCorporales": 28,
20,    "Cedula": "101010930867",
21,    "Nombre": "Mayo",
22,    "Apellido": "91",
23,    "Hombres": 9,
24,    "Tallas": 7,
25,    "Pecho": 87,
26,    "Cintura": 73,
27,    "Biceps_Derecho": 29,
28,    "Biceps_Izquierdo": 29,
29,    "Antebrazo_Derecho": 23.6,
30,    "Antebrazo_Izquierdo": 23.6,
31,    "Pierna_Derecha": 88.6,
32,    "Pierna_Izquierda": 63.6
33,  }
34 ]
```



```
1, [
2,   {
3,     "Cedula": "10888",
4,     "Nombre": "Julio",
5,     "Apellido": "2624",
6,     "Hombres": 24,
7,     "Tallas": 6,
8,     "Cintura": 68,
9,     "Pecho": 96,
10,    "Biceps_Derecho": 39,
11,    "Biceps_Izquierdo": 39,
12,    "Antebrazo_Derecho": 25.5,
13,    "Antebrazo_Izquierdo": 25.5,
14,    "Pierna_Derecha": 83.8,
15,    "Pierna_Izquierda": 83.8
16,  },
17,  {
18,    "ID_MedidasCorporales": 29,
19,    "Cedula": "101010930867",
20,    "Nombre": "Mayo",
21,    "Apellido": "91",
22,    "Hombres": 91,
23,    "Tallas": 67,
24,    "Cintura": 73,
25,    "Pecho": 96,
26,    "Biceps_Derecho": 29,
27,    "Biceps_Izquierdo": 29,
28,    "Antebrazo_Derecho": 23,
29,    "Antebrazo_Izquierdo": 23.5,
30,    "Pierna_Derecha": 88.6,
31,    "Pierna_Izquierda": 63.6
32,  }
33 ]
```

```

1   {
2     "Cedula": "1888",
3     "Mes": "mayo",
4     "Año": 2024,
5     "Hombros": 24,
6     "Pecho": 24,
7     "Cintura": 69,
8     "Estatura": 166,
9     "Bicep_Derecho": 38,
10    "Bicep_Izquierdo": 38,
11    "Antebrazo_Derecho": 28.5,
12    "Antebrazo_Izquierdo": 28.5,
13    "Perna_Derecha": 63.5,
14    "Perna_Izquierda": 63.5
15  }

```

Body Cookies Headers Test Results

```

1   {
2     "error": false,
3     "status": 201,
4     "body": "Item guardado con exito"
5   }

```

```

1   [
2     {
3       "Cedula": "1888",
4       "Mes": "mayo",
5       "Año": 2024,
6       "Hombros": 24,
7       "Pecho": 24,
8       "Cintura": 69,
9       "Estatura": 166,
10      "Bicep_Derecho": 38,
11      "Bicep_Izquierdo": 38,
12      "Antebrazo_Derecho": 28.5,
13      "Antebrazo_Izquierdo": 28.5,
14      "Perna_Derecha": 63.5,
15      "Perna_Izquierda": 63.5
16    }
17  ]

```

Body Cookies Headers Test Results

```

1   [
2     {
3       "Cedula": "1888",
4       "Mes": "mayo",
5       "Año": 2024,
6       "Hombros": 24,
7       "Pecho": 24,
8       "Cintura": 69,
9       "Estatura": 166,
10      "Bicep_Derecho": 38,
11      "Bicep_Izquierdo": 38,
12      "Antebrazo_Derecho": 28.5,
13      "Antebrazo_Izquierdo": 28.5,
14      "Perna_Derecha": 63.5,
15      "Perna_Izquierda": 63.5
16    }
17  ]

```

Online Find and replace Console

## PUT

```

1   {
2     "ID_MedidasCorporales": 8,
3     "Cedula": "1888",
4     "Mes": "abril",
5     "Año": 2024,
6     "Hombros": 92,
7     "Pecho": 68,
8     "Cintura": 69,
9     "Estatura": 166,
10    "Bicep_Derecho": 29,
11    "Bicep_Izquierdo": 29,
12    "Antebrazo_Derecho": 28.5,
13    "Antebrazo_Izquierdo": 28.5,
14    "Perna_Derecha": 64.5,
15    "Perna_Izquierda": 63.5
16  }

```

Body Cookies Headers Test Results

```

1   [
2     {
3       "Cedula": "1888",
4       "Mes": "abril",
5       "Año": 2024,
6       "Hombros": 92,
7       "Pecho": 68,
8       "Cintura": 69,
9       "Estatura": 166,
10      "Bicep_Derecho": 29,
11      "Bicep_Izquierdo": 29,
12      "Antebrazo_Derecho": 28.5,
13      "Antebrazo_Izquierdo": 28.5,
14      "Perna_Derecha": 64.5,
15      "Perna_Izquierda": 63.5
16    }
17  ]

```

Body Cookies Headers Test Results

```

1   [
2     {
3       "Cedula": "1888",
4       "Mes": "abril",
5       "Año": 2024,
6       "Hombros": 92,
7       "Pecho": 68,
8       "Cintura": 69,
9       "Estatura": 166,
10      "Bicep_Derecho": 38,
11      "Bicep_Izquierdo": 38,
12      "Antebrazo_Derecho": 28.5,
13      "Antebrazo_Izquierdo": 28.5,
14      "Perna_Derecha": 63.5,
15      "Perna_Izquierda": 63.5
16    }
17  ]

```

Online Find and replace Console

The screenshot shows the Postman interface with a PUT request to `http://localhost:4000/api/medidascorporales`. The request body is a JSON object containing various measurements for a user with ID 1008. The response status is 200 OK, indicating success.

```
PUT http://localhost:4000/api/medidascorporales
{
    "ID_MedidaCorporal": 29,
    "Cedula": "1008",
    "Mes": "diciembre",
    "Año": 2024,
    "Hombro": 24,
    "Pecho": 24,
    "Cintura": 66,
    "Gluteo": 96,
    "Bicep_Derecho": 39,
    "Bicep_Izquierdo": 39,
    "Antebrazo_Derecho": 26.6,
    "Antebrazo_Izquierdo": 26.6,
    "Pantora_Derecha": 63.6,
    "Pantora_Izquierda": 63.6
}
```

The screenshot shows the Postman interface with a successful response (201 Created) after the PUT request. The response body indicates the item was updated successfully.

```
{
    "error": false,
    "status": 201,
    "body": "Item actualizado con exito"
}
```

# DELETE

The screenshot shows two separate API requests in the Postman application:

**GET Request:**

- URL: `http://localhost:4000/api/medidascorporales`
- Method: GET
- Body: JSON (Pretty)
- Response Status: 200 OK
- Response Body (Pretty):

```
1  [
2     {
3         "ID_MedidasCorporales": 25,
4         "Cedula": "1000",
5         "Mes": "marzo",
6         "Año": 2023,
7         "Hombros": 10,
8         "Pecho": 11,
9         "Cintura": 12,
10        "Quadriceps": 13,
11        "Bicep_Derecho": 14,
12        "Bicep_Izquierdo": 15,
13        "Antebrazo_Derecho": 16,
14        "Antebrazo_Izquierdo": 17,
15        "Plasma_Derecha": 18,
16        "Plasma_Izquierda": 19
17    }
18 ]
19 
```

**DELETE Request:**

- URL: `http://localhost:4000/api/medidascorporales`
- Method: DELETE
- Body: JSON (Pretty)
- Response Status: 200 OK
- Response Body (Pretty):

```
1  [
2     {
3         "ID_MedidasCorporales": 29,
4         "Cedula": "1000",
5         "Mes": "diciembre",
6         "Año": 2023,
7         "Hombros": 24,
8         "Pecho": 24,
9         "Cintura": 66,
10        "Quadriceps": 96,
11        "Bicep_Derecho": 30,
12        "Bicep_Izquierdo": 30,
13        "Antebrazo_Derecho": 26,6,
14        "Antebrazo_Izquierdo": 26,6,
15        "Plasma_Derecha": 63,6,
16        "Plasma_Izquierda": 63,6
17    }
18 ]
19 
```

Postman API Network

My Workspace

Gym pruebas / Medidas corporales New Request

DELETE http://localhost:4000/api/medidascorporales

Body (raw JSON)

```
[{"ID_MedidasCorporales": 29, "Cedula": "1000", "Mes": "diciembre", "Año": 2024, "Hombres": 24, "Pecho": 24, "Estomago": 40, "Gluteo": 96, "Bicep_Derecho": 38, "Bicep_Izquierdo": 38, "Antebrazo_Derecho": 38.8, "Antebrazo_Izquierdo": 38.8, "Pierna_Derecha": 63.4, "Pierna_Izquierda": 63.6}
```

Status: 200 OK Time: 9 ms Size: 336 B Save as example

Body (Pretty JSON)

```
{ "error": false, "status": 200, "body": "Item eliminado satisfactoriamente" }
```

Postman API Network

My Workspace

Gym pruebas / Medidas corporales New Request

GET http://localhost:4000/api/medidascorporales

Body (raw JSON)

```
[{"ID_MedidasCorporales": 29, "Cedula": "1000", "Mes": "diciembre", "Año": 2024, "Hombres": 24, "Pecho": 24, "Estomago": 40, "Gluteo": 96, "Bicep_Derecho": 38, "Bicep_Izquierdo": 38, "Antebrazo_Derecho": 38.8, "Antebrazo_Izquierdo": 38.8, "Pierna_Derecha": 63.4, "Pierna_Izquierda": 63.6}, {"ID_MedidasCorporales": 28, "Cedula": "101000000000000000", "Mes": "diciembre", "Año": 2024, "Hombres": 15, "Pecho": 97, "Estomago": 45, "Gluteo": 96, "Bicep_Derecho": 28, "Bicep_Izquierdo": 29, "Antebrazo_Derecho": 35, "Antebrazo_Izquierdo": 35, "Pierna_Derecha": 66.5, "Pierna_Izquierda": 63.6}]
```

Status: 200 OK Time: 9 ms Size: 2.27 KB Save as example

# JMETER

Prueba con 80 usuarios

Ruta de la API: http://localhost:4000/api/medidascorporales

The screenshot shows the Apache JMeter interface version 5.6.3. The left sidebar displays the 'Test Plan' structure, which includes a 'User' element containing an 'HTTP Request' sampler. The main panel is divided into two sections: 'Thread Group' and 'HTTP Request'.

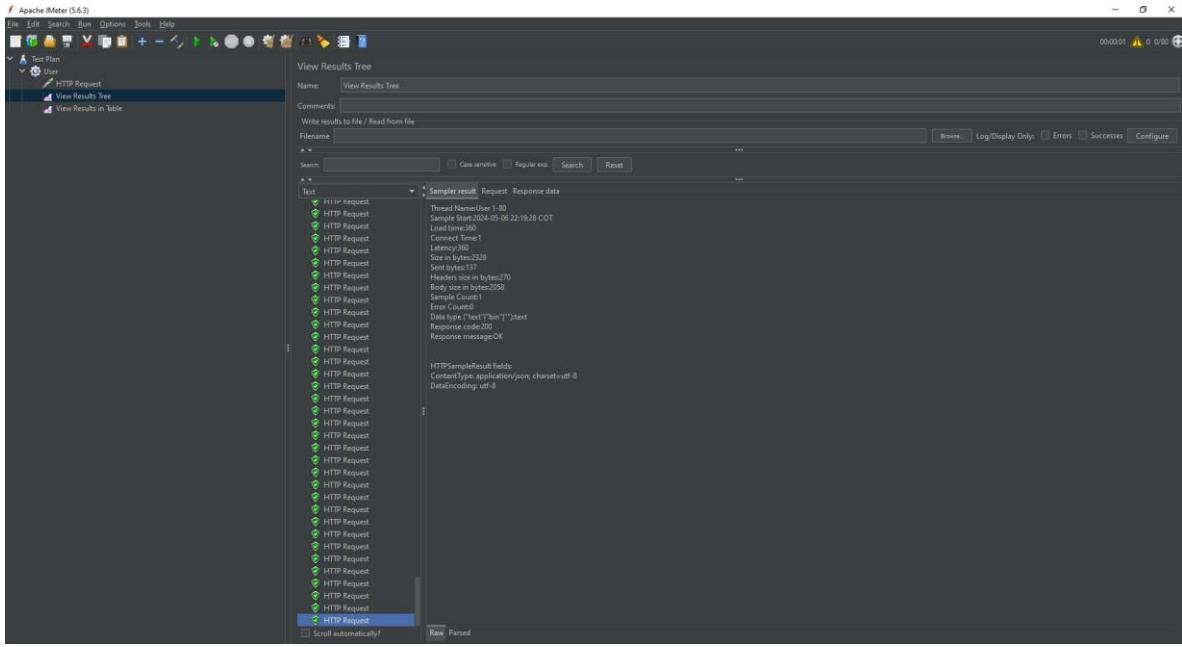
**Thread Group:**

- Name: User
- Comments: Action to be taken after a Sampler error:
  - Continue
  - Start Next Thread Loop
  - Stop Thread
  - Stop Test
  - Stop Test Now
- Number of Threads (users): 80
- Ramp-up period (seconds): 1
- Loop Count:  Infinite 1
- Same user on each iteration
- Delay Thread creation until needed
- Specify Thread lifetime
- Duration (seconds):
- Startup delay (seconds):

**HTTP Request:**

- Name: HTTP Request
- Comments: \*\*\*
- Web Server:
  - Protocol [http]: http
  - Server Name or IP: localhost
  - Port Number: 4000
- HTTP Request:
  - Method: GET
  - Path: http://localhost:4000/api/medidascorporales
  - Checkboxes: Redirect Automatically, Follow Redirections, Use Keepalive, Use multipart/form-data, Browser-compatible headers
- Parameters: Body Data, Files Upload
- Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
- Buttons at the bottom: Detail, Add, Add from Clipboard, Delete, Up, Down



## Tiempos de respuesta

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
45	22:19:28.465	User 1-45	HTTP Request	235	✓	2328	137	235	1
46	22:19:28.480	User 1-46	HTTP Request	255	✓	2328	137	255	1
47	22:19:28.495	User 1-47	HTTP Request	258	✓	2328	137	258	1
48	22:19:28.505	User 1-48	HTTP Request	256	✓	2328	137	256	2
49	22:19:28.518	User 1-49	HTTP Request	281	✓	2328	137	281	1
50	22:19:28.535	User 1-50	HTTP Request	323	✓	2328	137	323	1
51	22:19:28.547	User 1-51	HTTP Request	316	✓	2328	137	316	1
52	22:19:28.552	User 1-52	HTTP Request	309	✓	2328	137	309	2
53	22:19:28.573	User 1-53	HTTP Request	301	✓	2328	137	301	1
54	22:19:28.595	User 1-54	HTTP Request	363	✓	2328	137	363	3
55	22:19:28.601	User 1-55	HTTP Request	354	✓	2328	137	354	1
56	22:19:28.613	User 1-56	HTTP Request	343	✓	2328	137	343	2
57	22:19:28.627	User 1-57	HTTP Request	340	✓	2328	137	340	1
58	22:19:28.637	User 1-58	HTTP Request	342	✓	2328	137	342	2
59	22:19:28.681	User 1-59	HTTP Request	342	✓	2328	137	342	2
60	22:19:28.692	User 1-60	HTTP Request	260	✓	2328	137	340	1
61	22:19:28.693	User 1-61	HTTP Request	343	✓	2328	137	343	2
62	22:19:28.699	User 1-62	HTTP Request	344	✓	2328	137	344	2
63	22:19:28.704	User 1-63	HTTP Request	349	✓	2328	137	349	1
64	22:19:28.718	User 1-64	HTTP Request	352	✓	2328	137	352	1
65	22:19:28.729	User 1-65	HTTP Request	347	✓	2328	137	347	1
66	22:19:28.744	User 1-66	HTTP Request	351	✓	2328	137	351	1
67	22:19:28.759	User 1-67	HTTP Request	344	✓	2328	137	347	1
68	22:19:28.765	User 1-68	HTTP Request	352	✓	2328	137	352	1
69	22:19:28.780	User 1-69	HTTP Request	331	✓	2328	137	331	2
70	22:19:28.795	User 1-70	HTTP Request	342	✓	2328	137	342	1
71	22:19:28.808	User 1-71	HTTP Request	337	✓	2328	137	337	2
72	22:19:28.821	User 1-72	HTTP Request	329	✓	2328	137	329	1
73	22:19:28.834	User 1-73	HTTP Request	335	✓	2328	137	335	1
74	22:19:28.847	User 1-74	HTTP Request	327	✓	2328	137	327	2
75	22:19:28.858	User 1-75	HTTP Request	321	✓	2328	137	321	1
76	22:19:28.869	User 1-76	HTTP Request	325	✓	2328	137	325	1
77	22:19:28.883	User 1-77	HTTP Request	330	✓	2328	137	330	0
78	22:19:28.898	User 1-78	HTTP Request	349	✓	2328	137	349	2
79	22:19:28.911	User 1-79	HTTP Request	366	✓	2328	137	366	1
80	22:19:28.924	User 1-80	HTTP Request	360	✓	2328	137	360	1

No of Samples: 80      Latest Sample: 360      Average: 336      Standard Deviation: 334

Scrolled automatically? Child samples?

## BASE DE DATOS

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- gym
- sys
- websitegym**
  - Tables
  - auth
  - ejerciciospumuscular
  - medidascorporales
  - parametros
  - usuarios
- Views
- Stored Procedures
- Functions

Query1 SQL File 2

```
1 • select * from medidascorporales
2
3
4
```

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content |

ID_MedidasCorporales	Cedula	Mes	Año	Hombres	Pecho	Cintura	Gürtel	Biceps_Derecha	Biceps_Izquierda	Antebrazo_Derecho	Antebrazo_Izquierdo	Pierna_Derecha	Pierna_Izquierda
8	1015	Abri	2023	92.00	85.00	70.00	98.00	20.00	29.00	23.50	21.00	94.50	53.50
21	1013693067	Agosto	2024	93.00	86.00	71.00	99.00	21.00	30.00	24.50	22.00	95.50	54.00
22	1000	diciembre	2024	24.00	24.00	60.00	96.00	30.00	30.00	25.50	25.50	53.50	53.50
23	1014	marzo	2024	5.00	5.00	5.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
24	1015	marzo	2024	30.00	4.00	5.00	6.00	4.00	4.00	4.00	4.00	5.00	5.00
26	1000	diciembre	2024	24.00	24.00	60.00	96.00	30.00	30.00	25.50	25.50	53.50	53.50
27	1000	diciembre	2024	24.00	24.00	60.00	96.00	30.00	30.00	25.50	25.50	53.50	53.50
28	1013693067	mayo	2024	91.00	87.00	73.00	96.00	28.00	29.00	23.50	23.50	55.50	53.50
3028	0208	0208	0208	0208	0208	0208	0208	0208	0208	0208	0208	0208	0208

Result Grid | Form Editor | Field Types | Query Help | Reset | Context Help | Snippets

medidascorporales 3

Action Output

Time	Action
1 21:46:24	select * from usuarios LIMIT 0, 1000
2 21:48:02	select * from auth LIMIT 0, 1000
3 21:48:38	select * from medidascorporales LIMIT 0, 1000

Message | Duration / Fetch

0 rows returned 0.000 sec / 0.000 sec

4 rows returned 0.000 sec / 0.000 sec

8 rows returned 0.000 sec / 0.000 sec

Administration Schemas Information No object selected

## CONCLUSIONES

En conclusión, las pruebas realizadas tanto en Postman como en JMeter arrojaron resultados altamente satisfactorios para la plataforma web "**Hypertrophy Fitness**". En Postman, se ejecutaron pruebas para los métodos GET, PUT, POST y DELETE, confirmando que todas las operaciones se llevaron a cabo sin inconvenientes y que la interacción con la API se realizó de manera adecuada.

Por otro lado, las pruebas en JMeter se realizaron con diferentes cargas de usuarios, incluyendo 4, 50 y 80 usuarios simultáneos, y todas resultaron exitosas. A pesar de las variaciones en la cantidad de usuarios conectados, el servidor de la página web "**Hypertrophy Fitness**" se mantuvo estable y continuó funcionando de manera óptima, sin experimentar caídas ni tiempos de inactividad.

Además, se evaluaron las principales rutas de la página web, incluyendo la página de inicio de sesión, registro, cambio de contraseña y medidas corporales. En todas estas áreas, el tiempo de respuesta del servidor fue satisfactorio, manteniéndose por debajo de los 2 milisegundos en todo momento, lo que indica un rendimiento eficiente y una capacidad de respuesta adecuada incluso bajo cargas de usuarios elevadas.

## BIBLIOGRAFIA

Postman API platform. (s/f). Postman.com. Recuperado el 7 de mayo de 2024, de <https://www.postman.com/>

Apache JMETERTM. (s/f). Apache.org. Recuperado el 7 de mayo de 2024, de <https://jmeter.apache.org/>