

Actividad – Modulo1-Lección3:

GET/info ruta que envía un mensaje con una breve descripción del programa:

```
Actividad-Modulo1-Leccion3 > client.rest > POST /crear_usuario
1 # Probar la ruta GET /info
  Send Request
2 GET http://127.0.0.1:5000/info
3 Content-Type: application/json
4
5 ###
6
7 # Probar la ruta POST /crear_usuario
  Send Request
8 POST http://127.0.0.1:5000/crear_usuario
9 Content-Type: application/json
10
11 {
12   "Id": 1,
13   "nombre": "Adriana Marrero",
14   "correo": "adri#M2s@gmail.com"
15 }
16
17 ###
```

```
http...info
http...fo
http...es
http...fo
http...tos

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.3
3 Date: Wed, 23 Apr 2025 21:41:42 GMT
4 Content-Type: application/json
5 Content-Length: 62
6 Connection: close
7
8 {
9   "mensaje": "Sistema de gestion de productos y usuarios"
10 }
```

POST/crear_usuario ruta que recibe los datos del usuario y valida el correo, nombre de este para guardarlos en una lista:

```
app.py client.rest X Response(9ms) X
Actividad-Modulo1-Leccion3 > client.rest > POST /crear_usuario
6
7 # Probar la ruta POST /crear_usuario
  Send Request
8 POST http://127.0.0.1:5000/crear_usuario
9 Content-Type: application/json
10
11 {
12   "Id": 1,
13   "nombre": "Adriana Marrero",
14   "correo": "adri#M2s@gmail.com"
15 }
16
17 ###
18
19 # Probar la ruta GET /usuarios
  Send Request
20 GET http://127.0.0.1:5000/usuarios
21 Content-Type: application/json
22
23 ###
24
25 # Probar la ruta POST /crear_producto
```

```
http...info
http...ario
http...arios
http...ducto
http...uctos

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.3
3 Date: Wed, 23 Apr 2025 21:45:29 GMT
4 Content-Type: application/json
5 Content-Length: 148
6 Connection: close
7
8 {
9   "mensaje": "Usuario creado exitosamente",
10  "usuario": {
11    "Id": 1,
12    "correo": "adri#M2s@gmail.com",
13    "nombre": "Adriana Marrero"
14  }
15 }
```

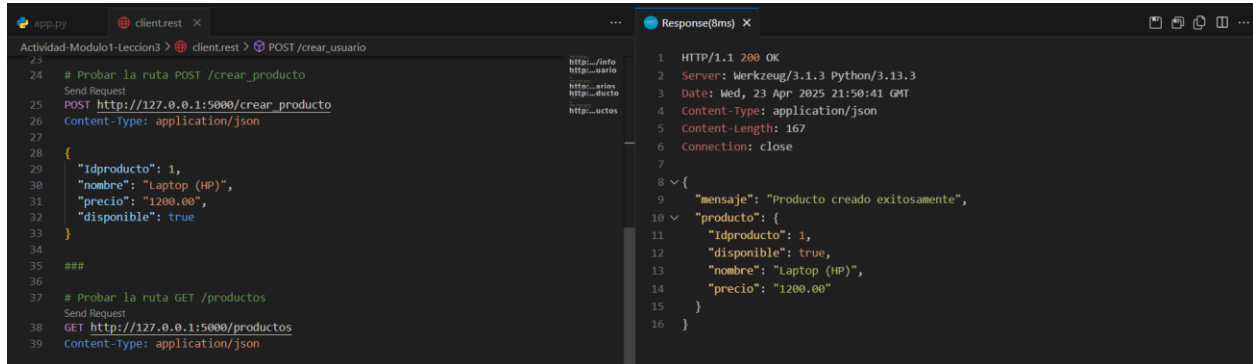
GET/usuarios ruta que devuelve la información del usuario creado que proviene de la lista de usuarios:

```
app.py client.rest X Response(3ms) X
Actividad-Modulo1-Leccion3 > client.rest > POST /crear_usuario
19 # Probar la ruta GET /usuarios
  Send Request
20 GET http://127.0.0.1:5000/usuarios
21 Content-Type: application/json
22
23 ###
24
25 # Probar la ruta POST /crear_producto
  Send Request
26 POST http://127.0.0.1:5000/crear_producto
27 Content-Type: application/json
28
29 {
30   "Idproducto": 1,
31   "nombre": "Laptop (HP)",
32   "precio": "1200.00",
33   "disponible": true
34 }
35
36 ###
```

```
http...info
http...ario
http...arios
http...ducto
http...uctos

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.3
3 Date: Wed, 23 Apr 2025 21:47:39 GMT
4 Content-Type: application/json
5 Content-Length: 123
6 Connection: close
7
8 {
9   "usuarios": [
10     {
11       "Id": 1,
12       "correo": "adri#M2s@gmail.com",
13       "nombre": "Adriana Marrero"
14     }
15   ]
16 }
```

POST/crear_producto ruta que recibe los datos de un producto los valida y los guarda en una lista:

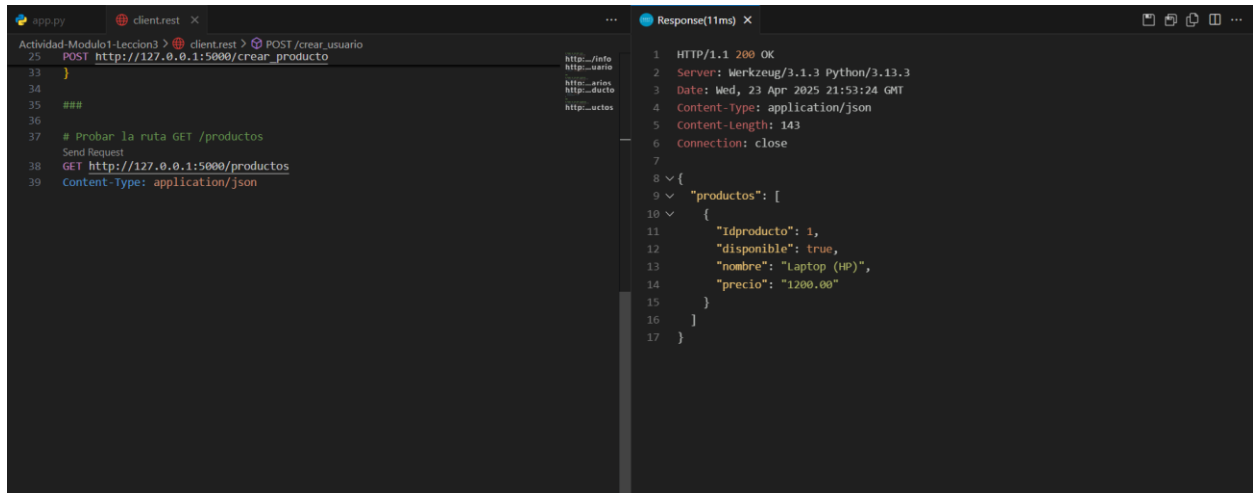


The screenshot shows a REST client interface with two panes. The left pane displays the request details for a POST request to `http://127.0.0.1:5000/crear_producto` with a JSON body: `{ "Idproducto": 1, "nombre": "Laptop (HP)", "precio": "1200.00", "disponible": true }`. The right pane shows the response, which is a 200 OK status with a JSON body: `{ "mensaje": "Producto creado exitosamente", "producto": { "Idproducto": 1, "disponible": true, "nombre": "Laptop (HP)", "precio": "1200.00" } }`.

```
23 # Probar la ruta POST /crear_producto
24
25 Send Request
26 POST http://127.0.0.1:5000/crear_producto
27 Content-Type: application/json
28 {
29   "Idproducto": 1,
30   "nombre": "Laptop (HP)",
31   "precio": "1200.00",
32   "disponible": true
33 }
34
35 ###
36
37 # Probar la ruta GET /productos
38 Send Request
39 GET http://127.0.0.1:5000/productos
40 Content-Type: application/json
```

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.3
3 Date: Wed, 23 Apr 2025 21:50:41 GMT
4 Content-Type: application/json
5 Content-Length: 167
6 Connection: close
7
8 {
9   "mensaje": "Producto creado exitosamente",
10  "producto": {
11    "Idproducto": 1,
12    "disponible": true,
13    "nombre": "Laptop (HP)",
14    "precio": "1200.00"
15  }
16 }
```

GET/productos ruta que devuelve la información del producto creado que proviene de la lista de productos:



The screenshot shows a REST client interface with two panes. The left pane displays the request details for a GET request to `http://127.0.0.1:5000/productos`. The right pane shows the response, which is a 200 OK status with a JSON body: `{ "productos": [{ "Idproducto": 1, "disponible": true, "nombre": "Laptop (HP)", "precio": "1200.00" }] }`.

```
33 POST http://127.0.0.1:5000/crear_producto
34 }
35 ###
36
37 # Probar la ruta GET /productos
38 Send Request
39 GET http://127.0.0.1:5000/productos
40 Content-Type: application/json
```

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.3
3 Date: Wed, 23 Apr 2025 21:53:24 GMT
4 Content-Type: application/json
5 Content-Length: 143
6 Connection: close
7
8 {
9   "productos": [
10    {
11      "Idproducto": 1,
12      "disponible": true,
13      "nombre": "Laptop (HP)",
14      "precio": "1200.00"
15    }
16  ]
17 }
```

Enlace al repositorio de Github:

https://github.com/BenyahirMartinez2004/Comp_2052/tree/main/Actividad-Modulo1-Leccion3