Génesis Ojeda

Mod 1 Lección 3

API Flask para Gestión de Usuarios y Productos

Este proyecto es un servidor Flask que gestiona **Usuarios** y **Productos** mediante una API REST. Proporciona rutas para obtener información sobre el sistema, crear usuarios y productos, y obtener listas de usuarios y productos almacenados en memoria. Se utilizó Flask, Python y Postman.

Propuesta de estructura de datos:

Usuarios:

nombre: El nombre del usuario.

correo: Correo electrónico del usuario.

Productos:

id: Identificador único del producto.

nombre: Nombre del producto.

descripcion: Una descripción breve del producto.

precio: El precio del producto.

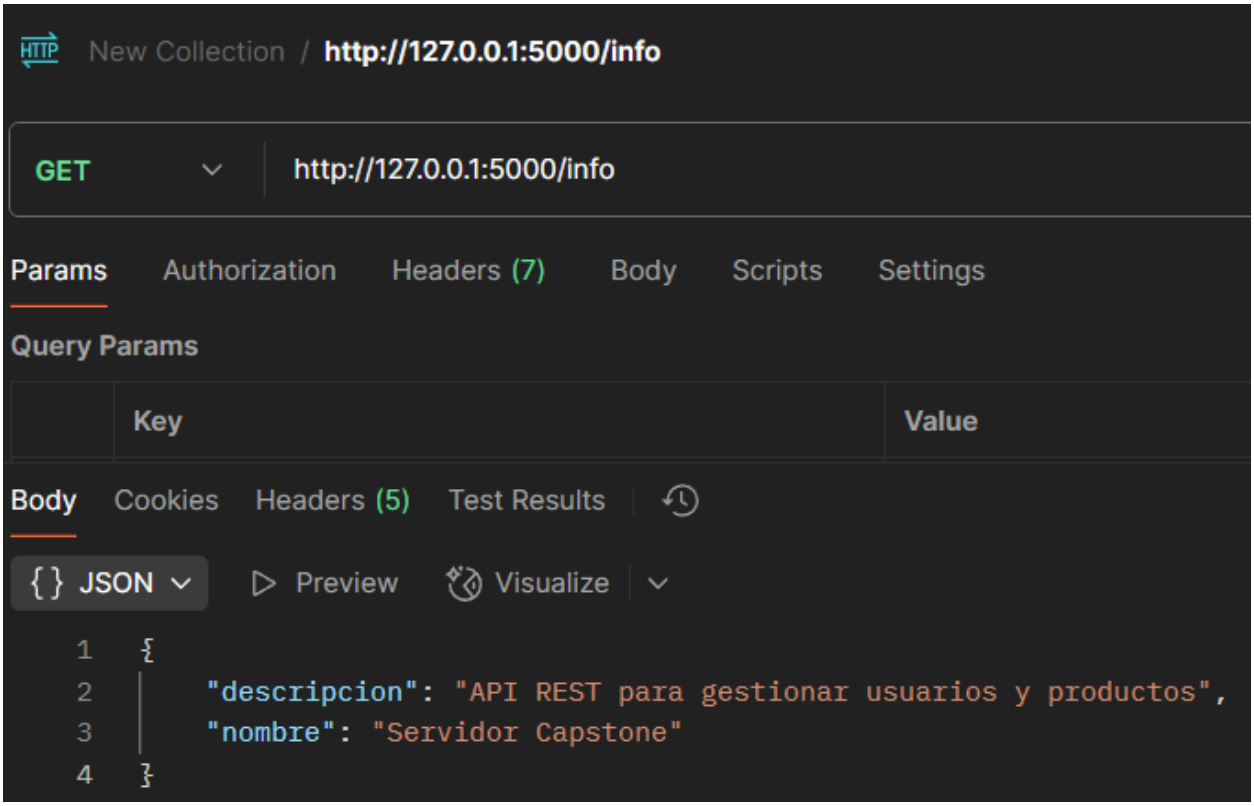
Formato JSON

{ "nombre": "Juan Pérez", "correo": "juan@example.com" }

{ "nombre": "Producto 3", "precio": 200 }

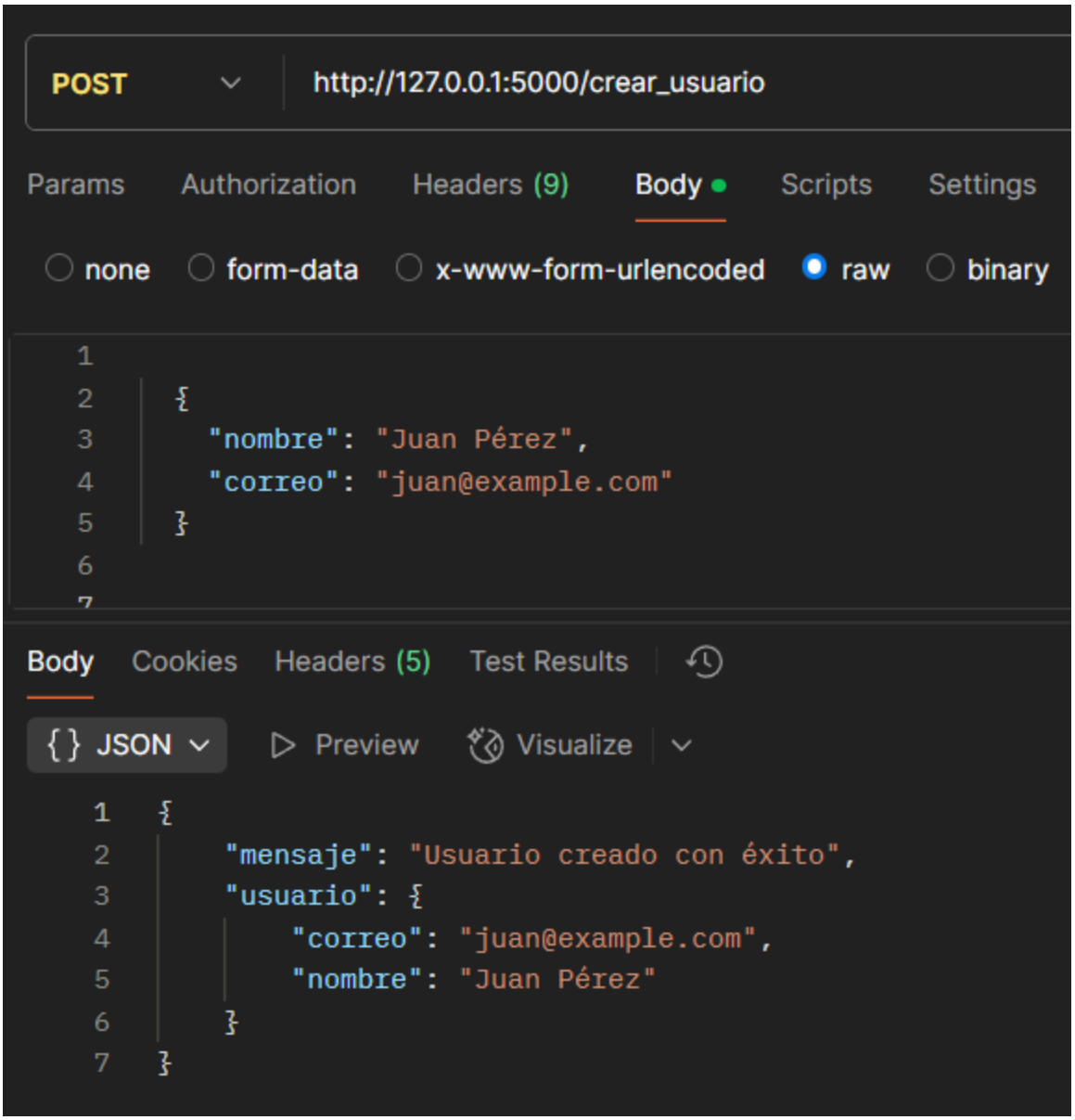
Get info

Trae la descripción y nombre guardada.



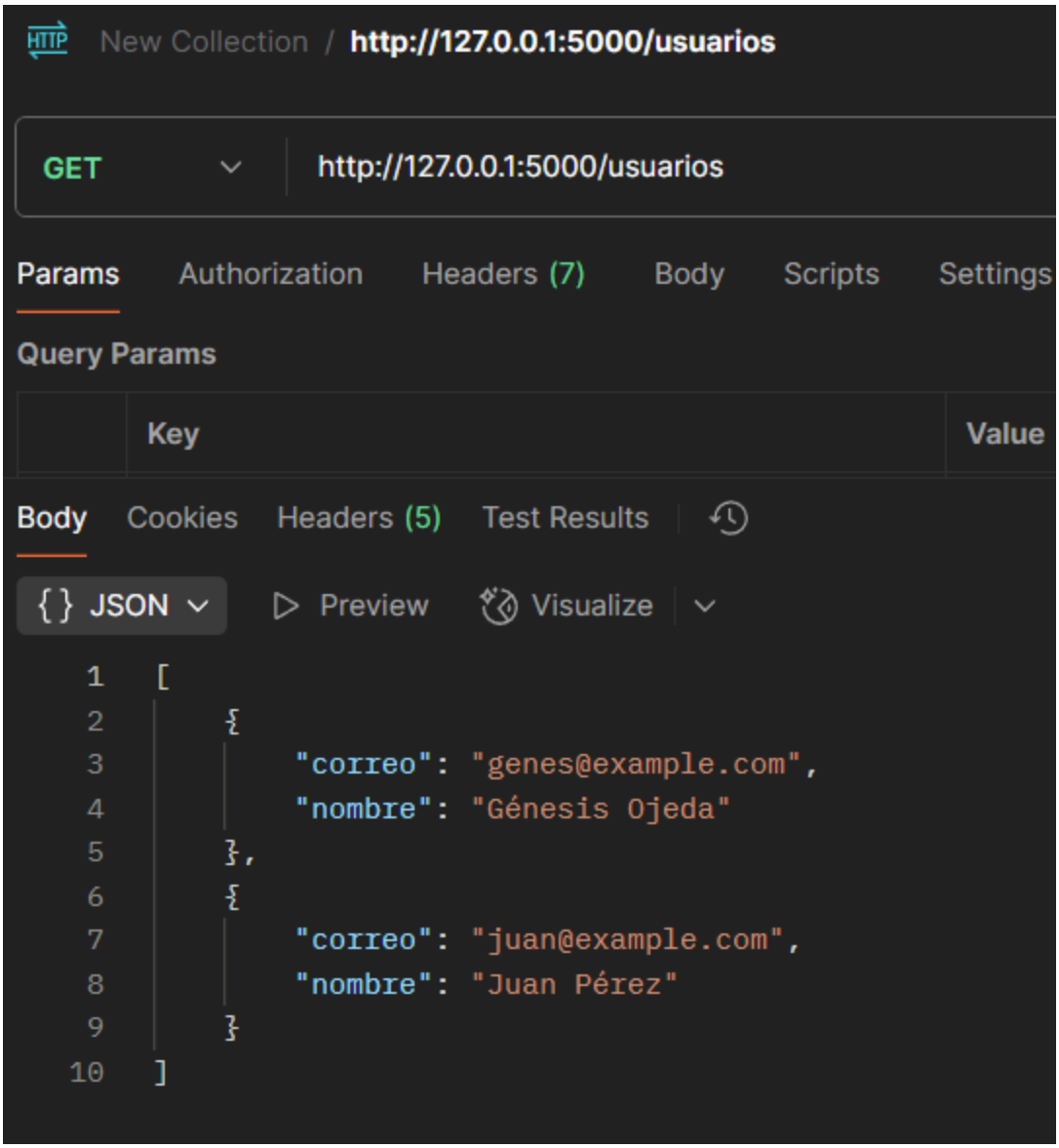
Post Crear usuario

Se le envía dos argumentos, nombre y correo, cuando las recibe, envía un mensaje.



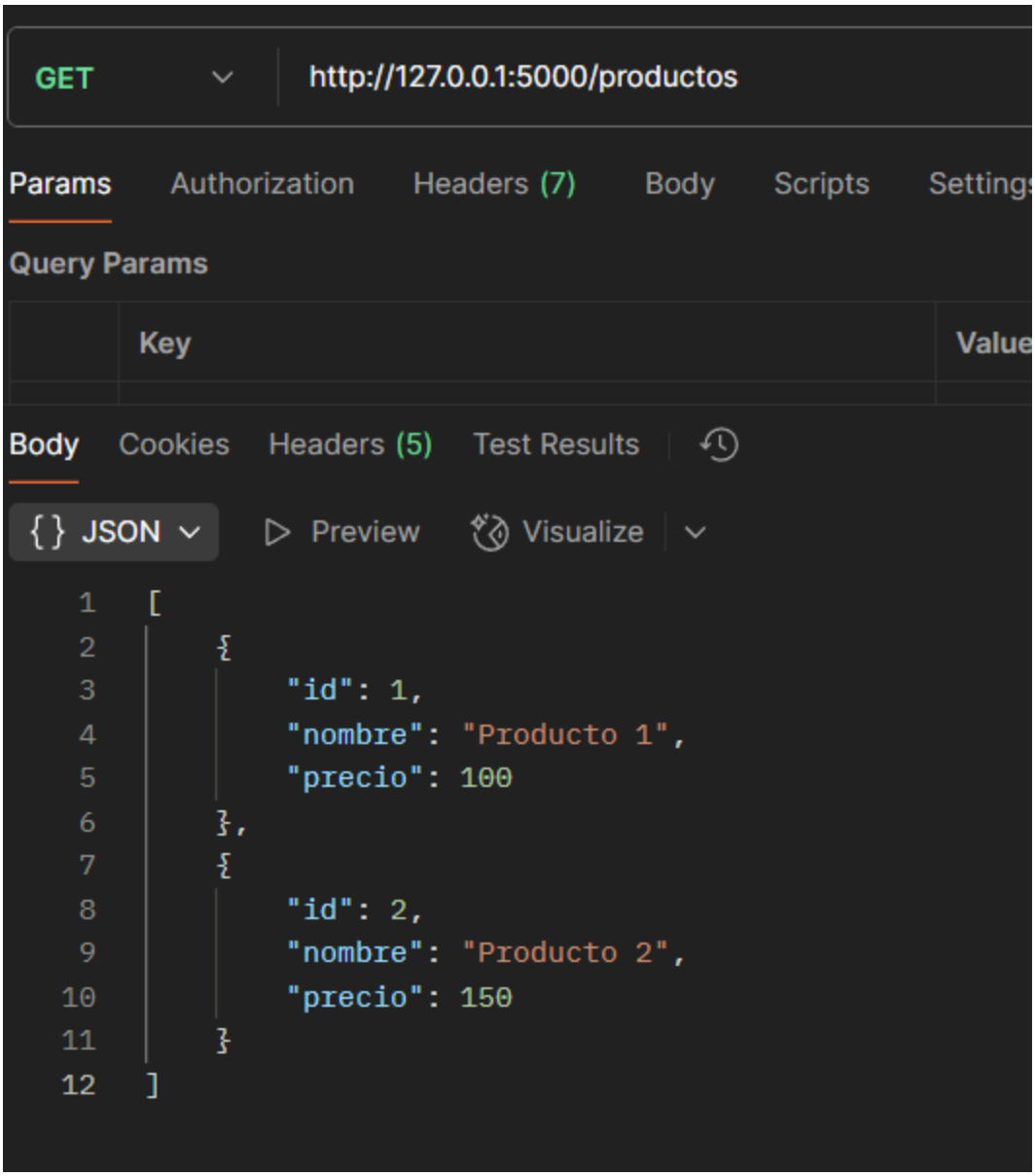
Get usuarios

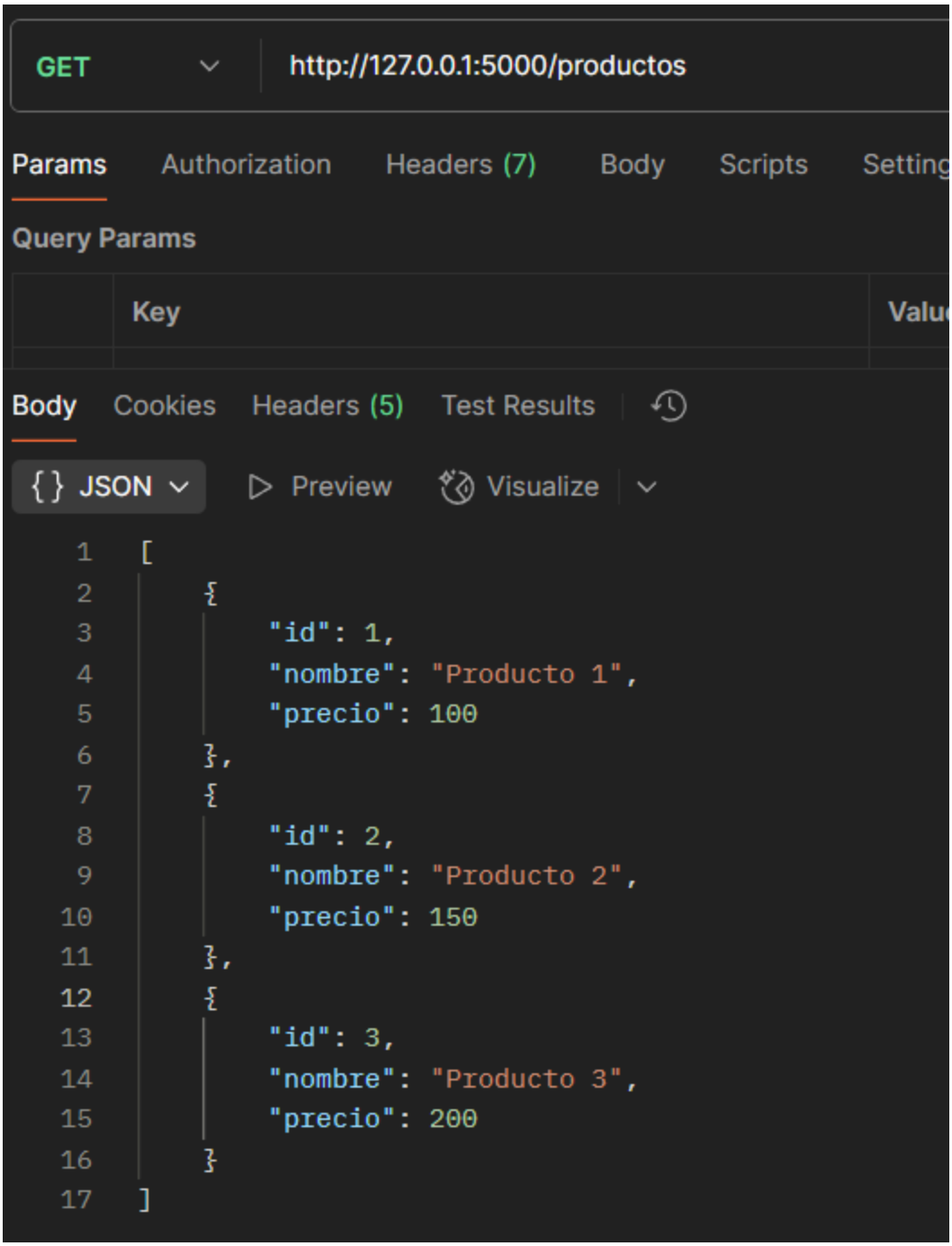
Trae todos los usuarios creados.



Get productos

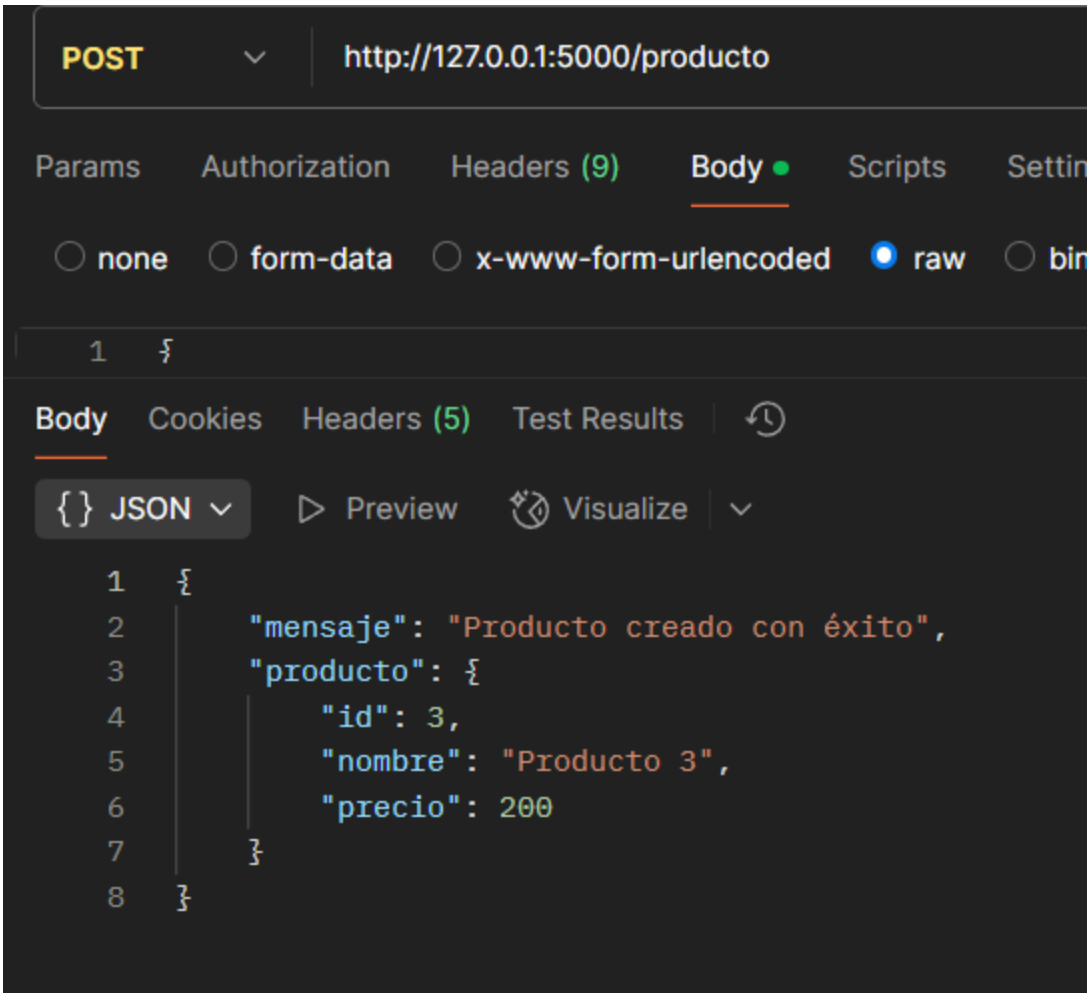
Trae todos los productos creados. Cada uno con ID único, nombre y precio.


GET
http://127.0.0.1:5000/productos
Params
Authorization
Headers (7)
Body
Scripts
Settings
Query Params
Key
Value
Body
Cookies
Headers (5)
Test Results
JSON
Preview
Visualize
[
{
"id": 1,
"nombre": "Producto 1",
"precio": 100
},
{
"id": 2,
"nombre": "Producto 2",
"precio": 150
}
]



Post product

Crea un product nuevo a base de nombre y precio que se le envíe como parámetro.


POST
http://127.0.0.1:5000/producto
Params
Authorization
Headers (9)
Body
Scripts
Settin
none
form-data
x-www-form-urlencoded
raw
bin
1 {
Body
Cookies
Headers (5)
Test Results
JSON
Preview
Visualize
1 {
2 "mensaje": "Producto creado con éxito",
3 "producto": {
4 "id": 3,
5 "nombre": "Producto 3",
6 "precio": 200
7 }
8 }