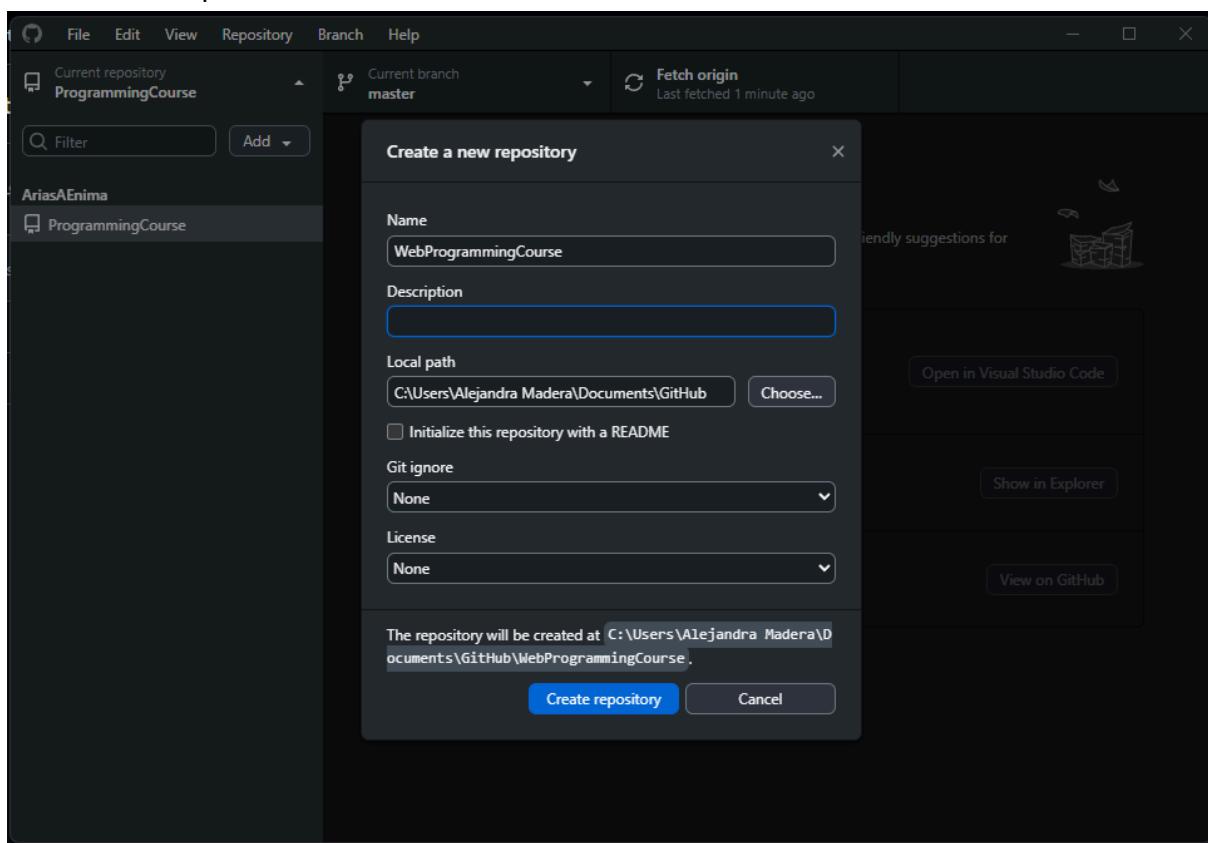
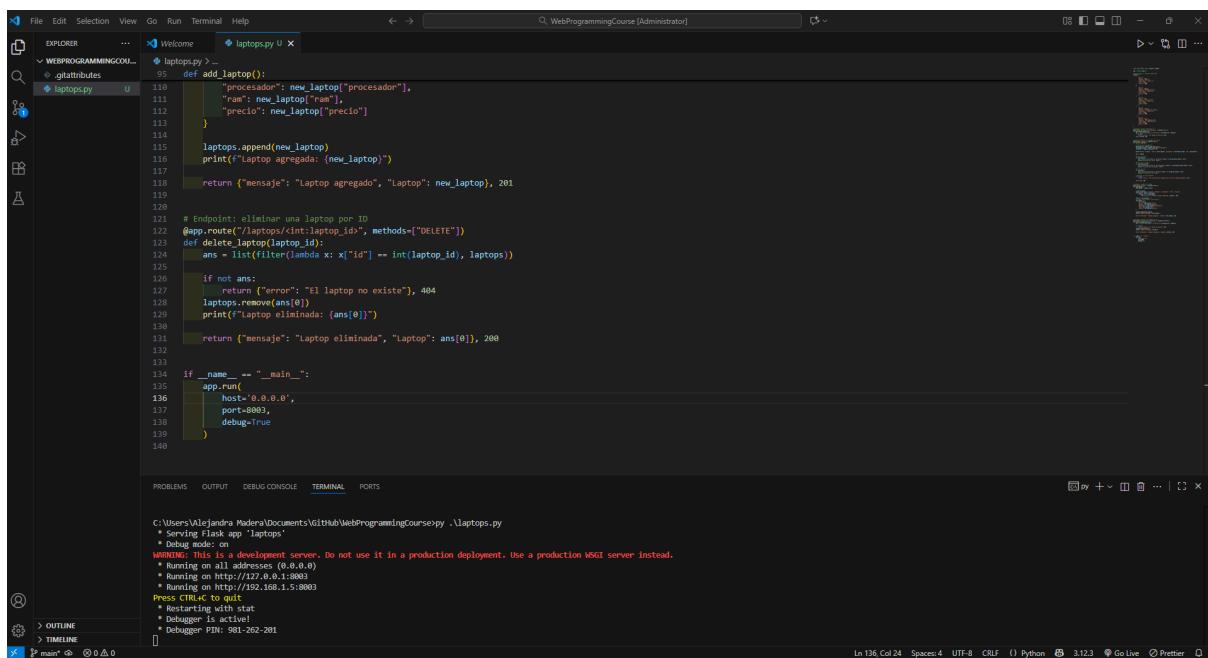


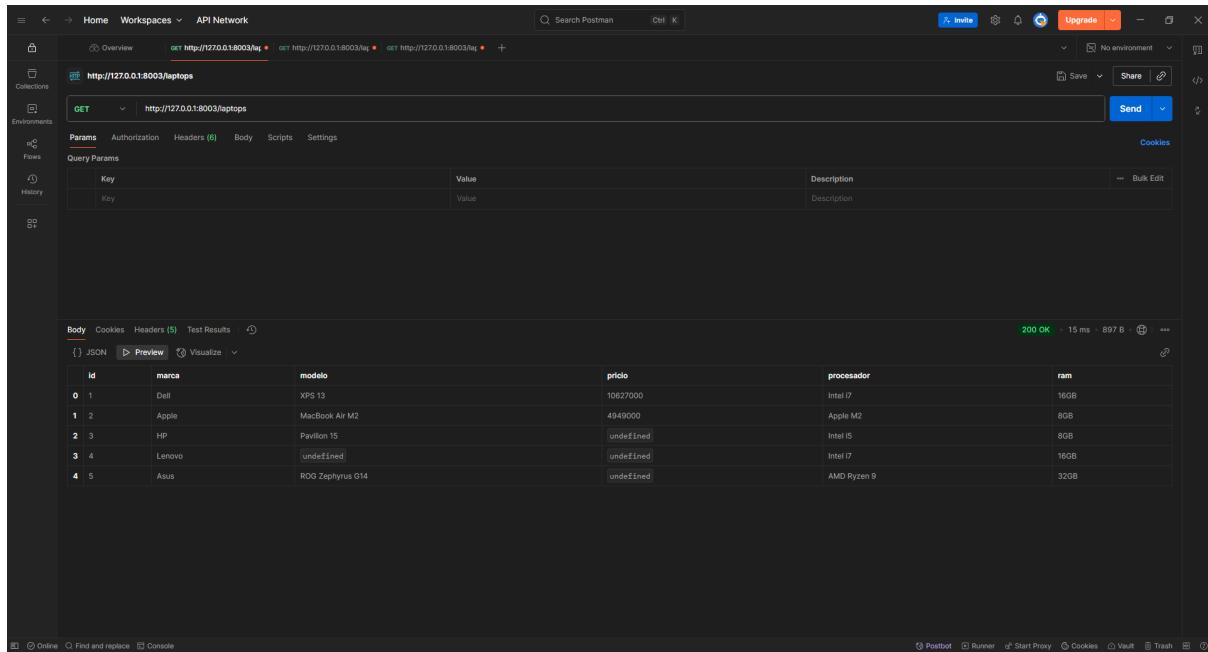
Creación del repositorio:



Correr el servidor



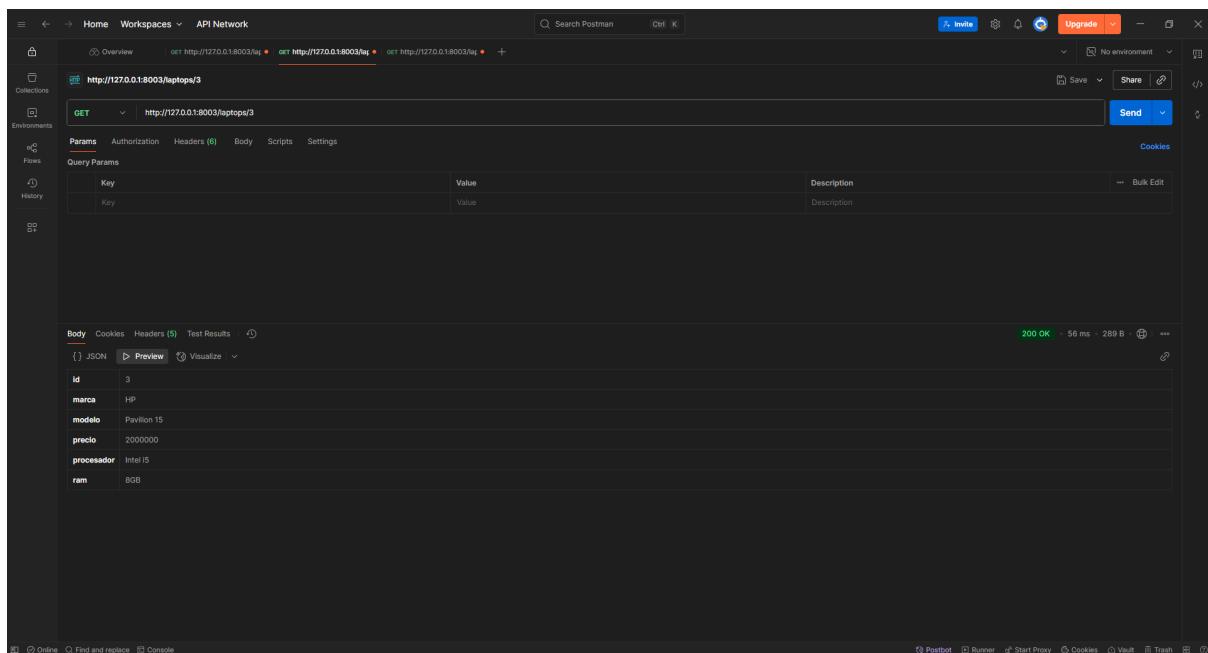
Metodo GET obtener todos los laptops



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8003/laptops`. The response is a 200 OK status with 15 ms latency and 897 B size. The response body is a table with 5 rows of laptop data:

	id	marca	modelo	precio	processador	ram
0	1	Dell	XPS 13	10627000	Intel i7	16GB
1	2	Apple	MacBook Air M2	4949000	Apple M2	8GB
2	3	HP	Pavilion 15	undefined	Intel i5	8GB
3	4	Lenovo	undefined	undefined	Intel i7	16GB
4	5	Asus	ROG Zephyrus G14	undefined	AMD Ryzen 9	32GB

Metodo GET obtener laptop por ID



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8003/laptops/3`. The response is a 200 OK status with 56 ms latency and 289 B size. The response body is a table with 1 row of laptop data:

	id	marca	modelo	precio	processador	ram
1	3	HP	Pavilion 15	2000000	Intel i5	8GB

Método GET para filtrar por marca

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8003/laptops?marca=Asus`. The 'Params' tab is selected, showing a query parameter `marca` with the value `Asus`. The response body is displayed in JSON format, showing a single laptop entry:

```
1 [ 2 { 3 "id": 6, 4 "marca": "Asus", 5 "modelo": "ROG Zephyrus G14", 6 "precio": 5899999, 7 "procesador": "AMD Ryzen 9", 8 "ram": "32GB" 9 } ]
```

Método GET para filtrar por procesador y ram

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8003/laptops?procesador=Intel i7&ram=16GB`. The 'Params' tab is selected, showing query parameters `procesador` set to `Intel i7` and `ram` set to `16GB`. The response body is displayed in JSON format, showing two laptop entries:

```
1 [ 2 { 3 "id": 1, 4 "marca": "Dell", 5 "modelo": "XPS 13", 6 "precio": 18627000, 7 "procesador": "Intel i7", 8 "ram": "16GB" 9 }, 10 { 11 "id": 4, 12 "marca": "Lenovo", 13 "modelo": "IdeaPad Xi Carbon", 14 "precio": 22699999, 15 "procesador": "Intel i7", 16 "ram": "16GB" 17 } ]
```

Método POST para agregar un nuevo laptop

The screenshot shows the Postman interface with a collection named "API Network". A POST request is made to `http://127.0.0.1:8003/laptops`. The request body contains the following JSON:

```

1 {
2   "marca": "Acer",
3   "modelo": "Aspire 7",
4   "procesador": "AMD Ryzen 7",
5   "ram": "16GB",
6   "precio": 3150000
7 }

```

The response status is `201 CREATED` with a response time of 16 ms and a total size of 359 B. The response body is:

```

1 {
2   "laptop": {
3     "id": 6,
4     "marca": "Acer",
5     "modelo": "Aspire 7",
6     "precio": 3150000,
7     "procesador": "AMD Ryzen 7",
8     "ram": "16GB"
9   },
10   "mensaje": "Laptop agregado"
11 }

```

Método DELETE para eliminar una laptop por ID

The screenshot shows the Postman interface with a workspace named "My Workspace". A DELETE request is made to `http://127.0.0.1:8003/laptops/1`. The response status is `200 OK` with a response time of 40 ms and a total size of 351 B. The response body is:

```

1 {
2   "laptop": [
3     {
4       "id": 1,
5       "marca": "Dell",
6       "modelo": "Vostro 5470",
7       "procesador": "Intel i7",
8       "ram": "16GB"
9     }
10   ],
11   "mensaje": "Laptop eliminada"
12 }

```