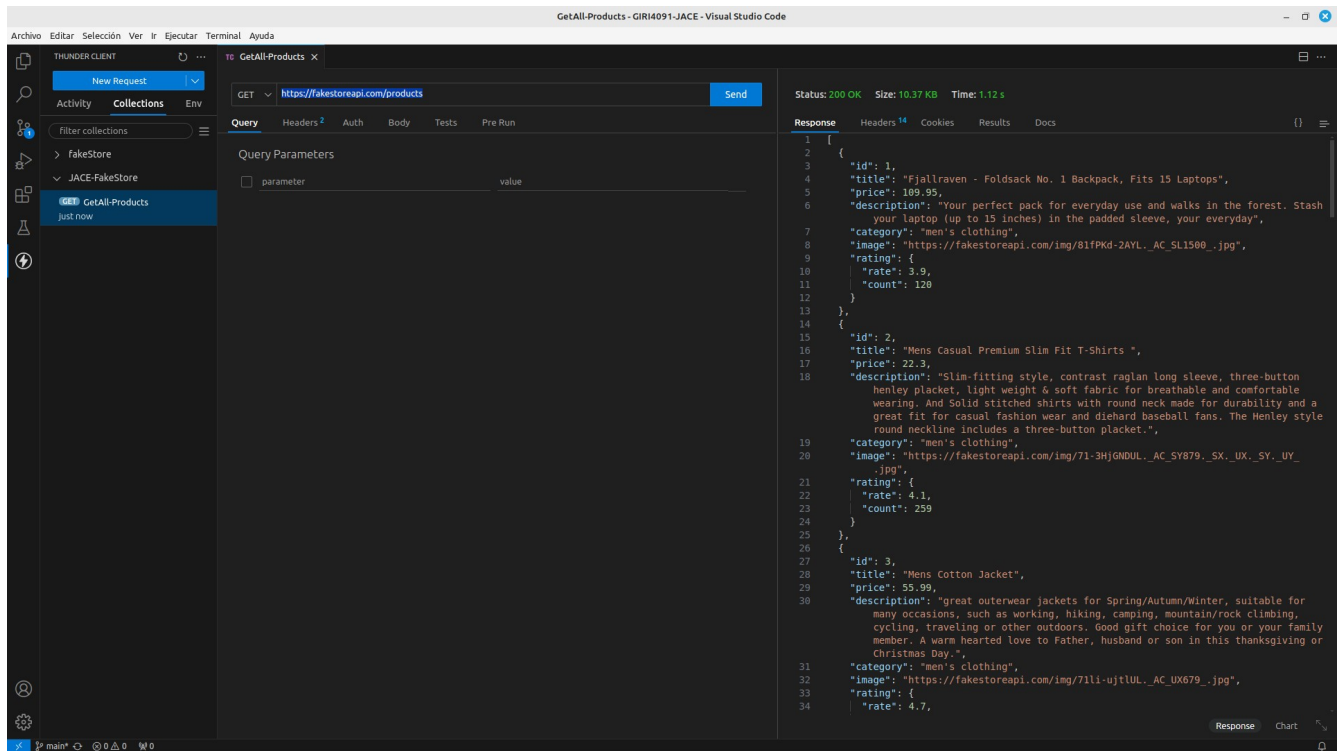


GetAll-Products

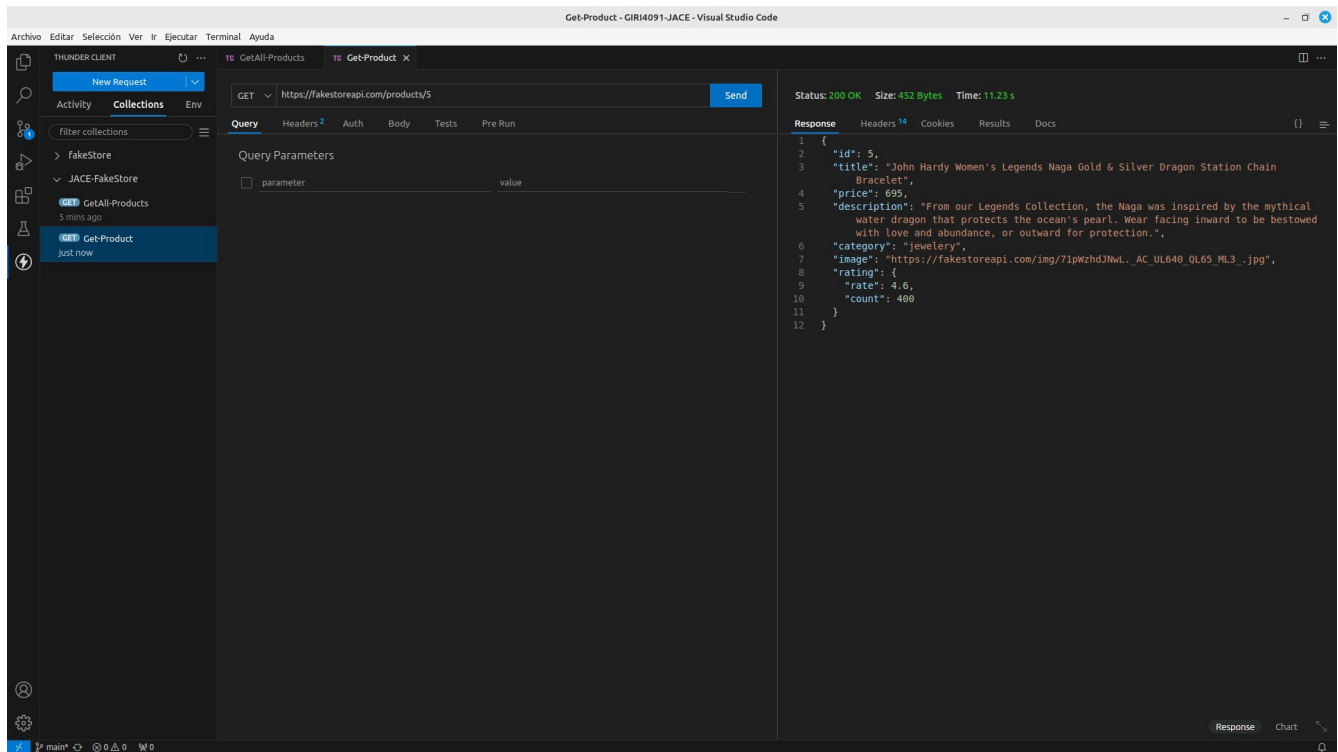
Thunder Client



Python

```
def GetAllProducts():
    url = "https://fakestoreapi.com/products"
    respuesta = requests.get(url).json()
    print("Listado de productos")
    print("-----")
    print(json.dumps(respuesta, indent=4, ensure_ascii=False))
```

GetProduct: Thunder Client



Python:

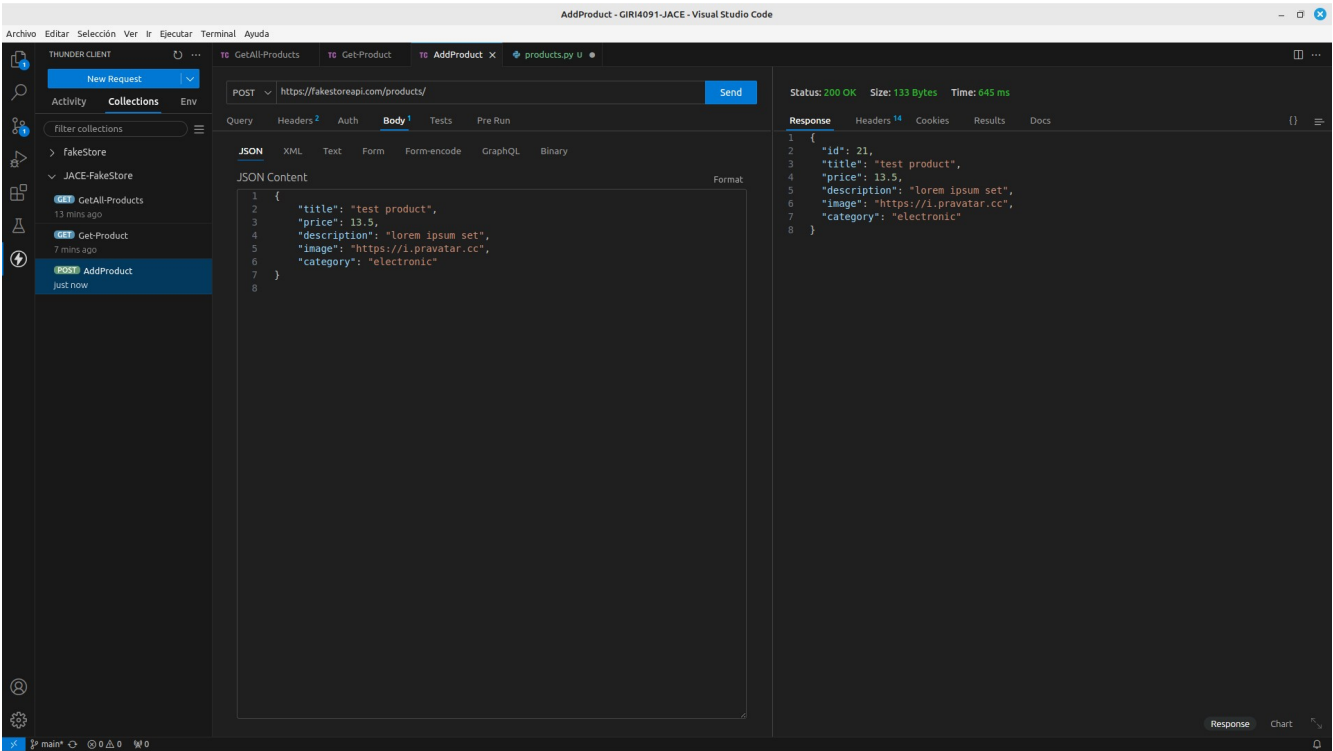
```
def GetProduct():
    try:
        id = input("Ingresa el ID del producto: ")
        url = "https://fakestoreapi.com/products/" + id
        response = requests.get(url)

        if response.status_code == 404:
            print("El ID no existe. Por favor, verifica el ID e intenta nuevamente.")
        else:
            response.raise_for_status()

            try:
                producto = response.json()
                if not producto:
                    print("El ID no existe o no se encontró el producto.")
                else:
                    print("Listado de productos")
                    print("-----")
                    print(json.dumps(producto, indent=4, ensure_ascii=False))
                    print("Búsqueda de producto completada")
            except ValueError:
                print("El ID no existe. ")
    except requests.exceptions.RequestException as e:
        print("Error al realizar la solicitud:", e)
    except Exception as e:
        print("Ha ocurrido un error:", e)
```

AddProduct

Tunder Client



Python:

```
def AddProduct():

    print("\n"+"Agregar producto"+"\\n")
    url = "https://fakestoreapi.com/products"
    titleProduct = (input("Ingresa el título del producto: "+"\\n"))
    priceProduct = (input("Ingresa el precio del producto: "+"\\n"))
    descriptionProduct = (input("Ingresa la descripción del producto: "+"\\n"))
    categoryProduct = (input("Ingresa la categoría del producto: "+"\\n"))

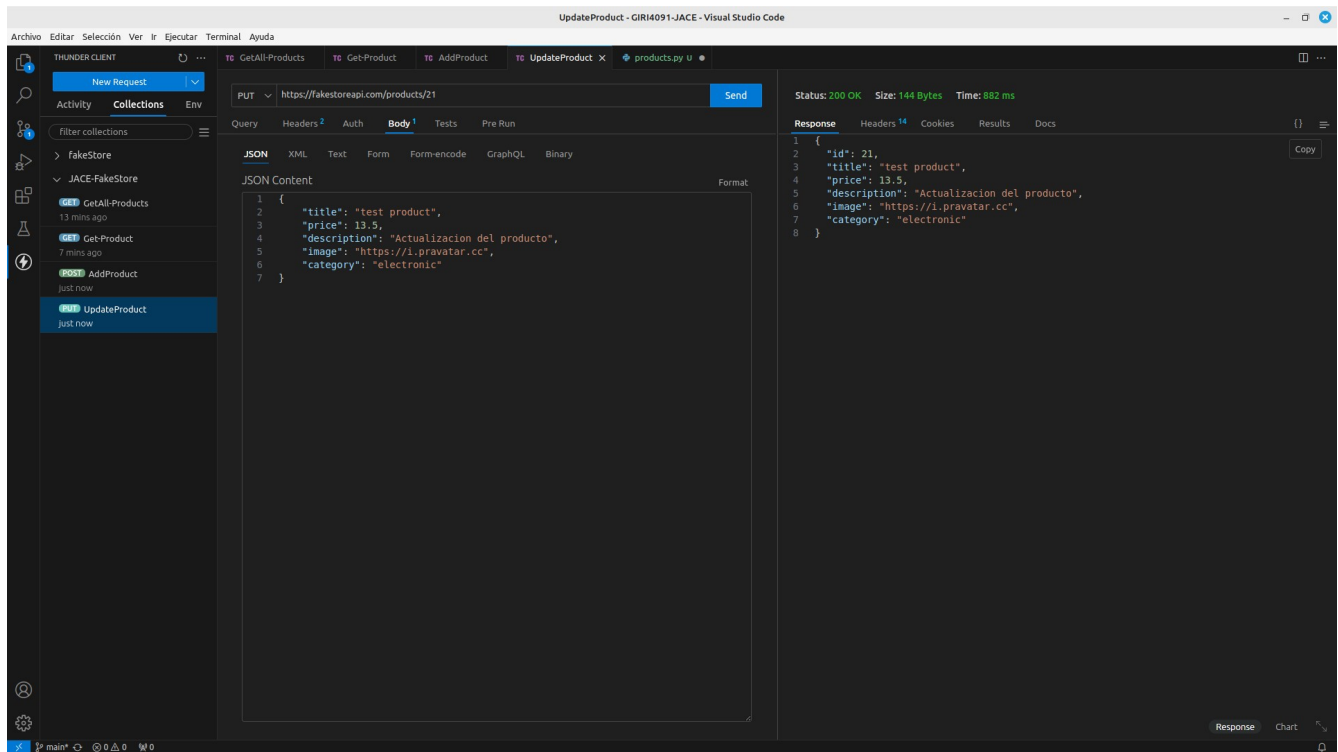
    payload = {
        "title": " "+titleProduct,
        "price": " "+priceProduct,
        "description": " "+descriptionProduct,
        "category": " "+categoryProduct,
        "image": "https://fakestoreapi.com/img/placeholder.jpg",
        "rating": {
            "rate": 4.5,
            "count": 10
        }
    }

    headers = {
        "Content-Type": "application/json"
    }

    response = requests.post(url, json=payload, headers=headers)

    if response.status_code == 200:
        print("\n"+"Producto creado exitosamente ")
        print(response.json())
    else:
        print("\n"+"Error al crear el producto ")
        print(response.status_code)
        print(response.text)
```

UpdateProduct Tunder Client:

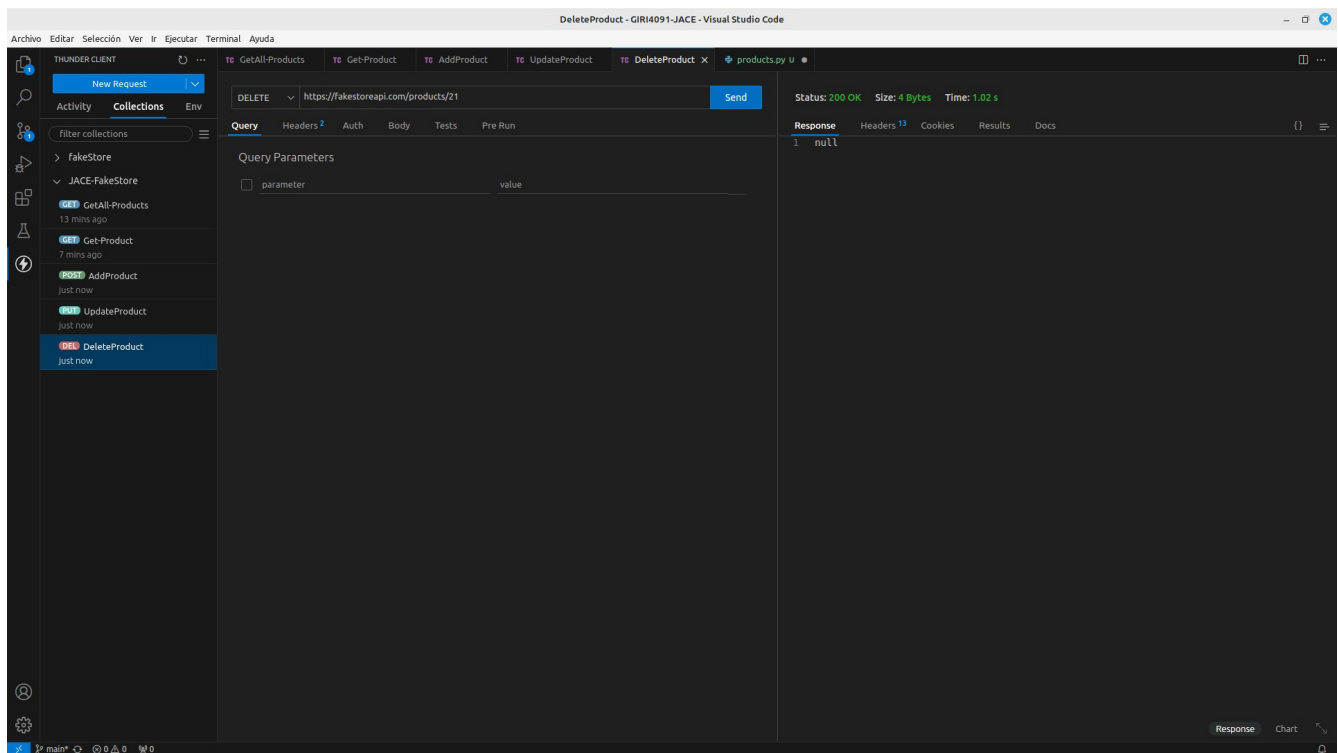


Python:

```
def UpdateProduct():
    try:
        id = input("Ingresa el ID del producto a Modificar ")
        url = "https://fakestoreapi.com/products/" + id
        data = {
            "title": input("Nuevo título: "),
            "price": float(input("Nuevo precio: ")),
            "description": input("Nueva descripción: "),
            "category": input("Nueva categoría: "),
            "image": "https://fakestoreapi.com/img/placeholder.jpg",
        }

        response = requests.put(url, json=data)
        response.raise_for_status()
        updated_product = response.json()
        print("-----")
        print("Producto actualizado con éxito")
        print(json.dumps(updated_product, indent=4, ensure_ascii=False))
    except requests.exceptions.RequestException as e:
        print("Error al realizar la solicitud:", e)
    except ValueError:
        print("Respuesta inválida. No se pudo parsear el JSON.")
    except Exception as e:
        print("Ha ocurrido un error:", e)
```


DeleteProduct Tunder Client:



Python:

```
def DeleteProduct():
    try:
        id = input("Ingresa el ID del producto a eliminar: ")
        url = "https://fakestoreapi.com/products/" + id
        response = requests.delete(url)

        if response.status_code == 404:
            print("Producto no encontrado. Por favor, verifica el ID e intenta nuevamente.")
        else:
            response.raise_for_status()

            try:
                respuesta_json = response.json()
            except ValueError:
                respuesta_json = None

            if respuesta_json is None:
                print("El producto no existe o no se encontró. No se realizó ninguna eliminación.")
            elif "error" in respuesta_json or not respuesta_json:
                print("El producto no se encontró o no se pudo eliminar.")
            else:
                print("Producto eliminado exitosamente.")
                print(json.dumps(respuesta_json, indent=4, ensure_ascii=False))
    except requests.exceptions.RequestException as e:
        print("Error al realizar la solicitud:", e)
    except Exception as e:
        print("Ha ocurrido un error:", e)
```