

Ilustración 2Post

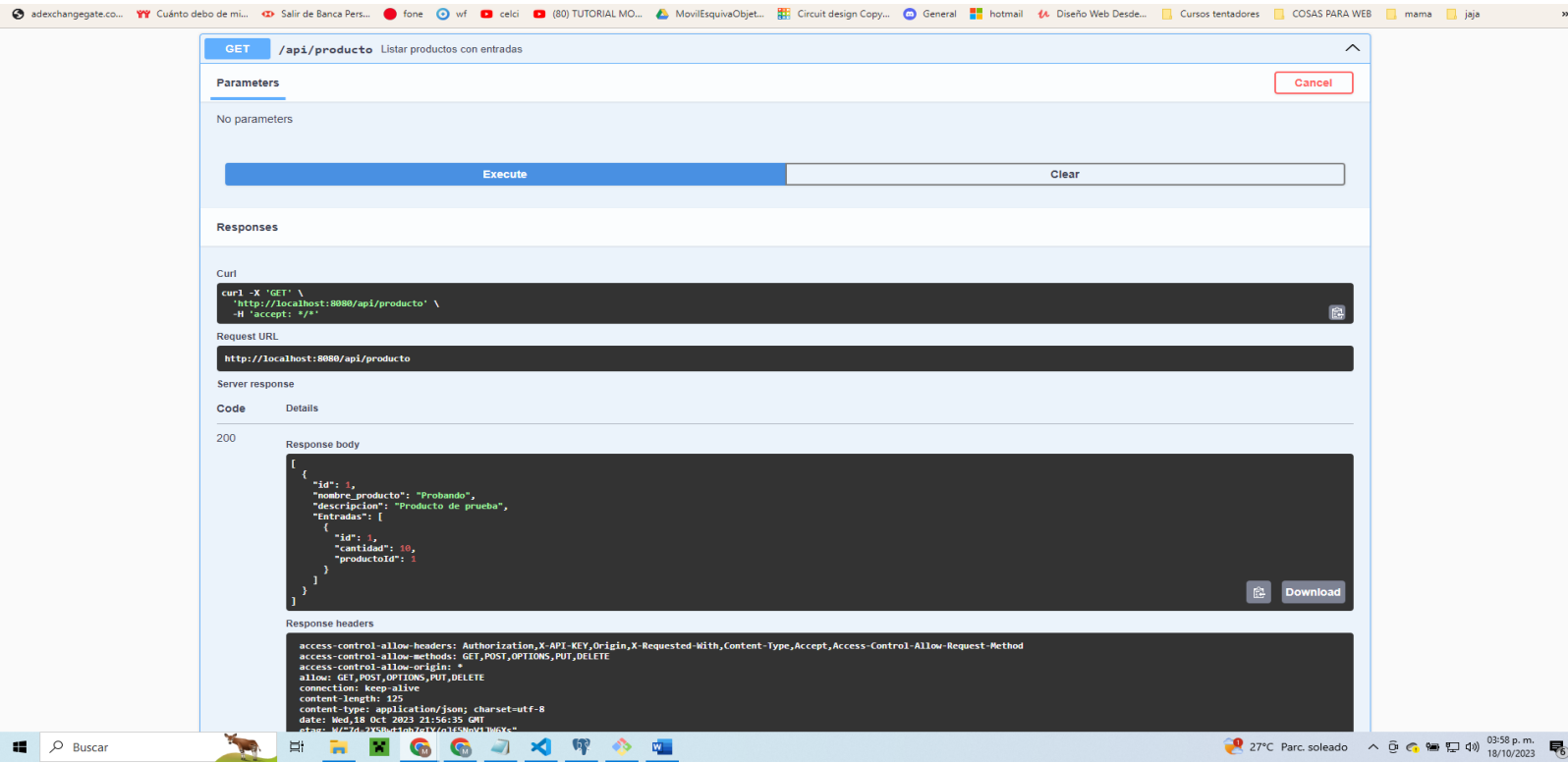


Ilustración 1Gest despues del Post

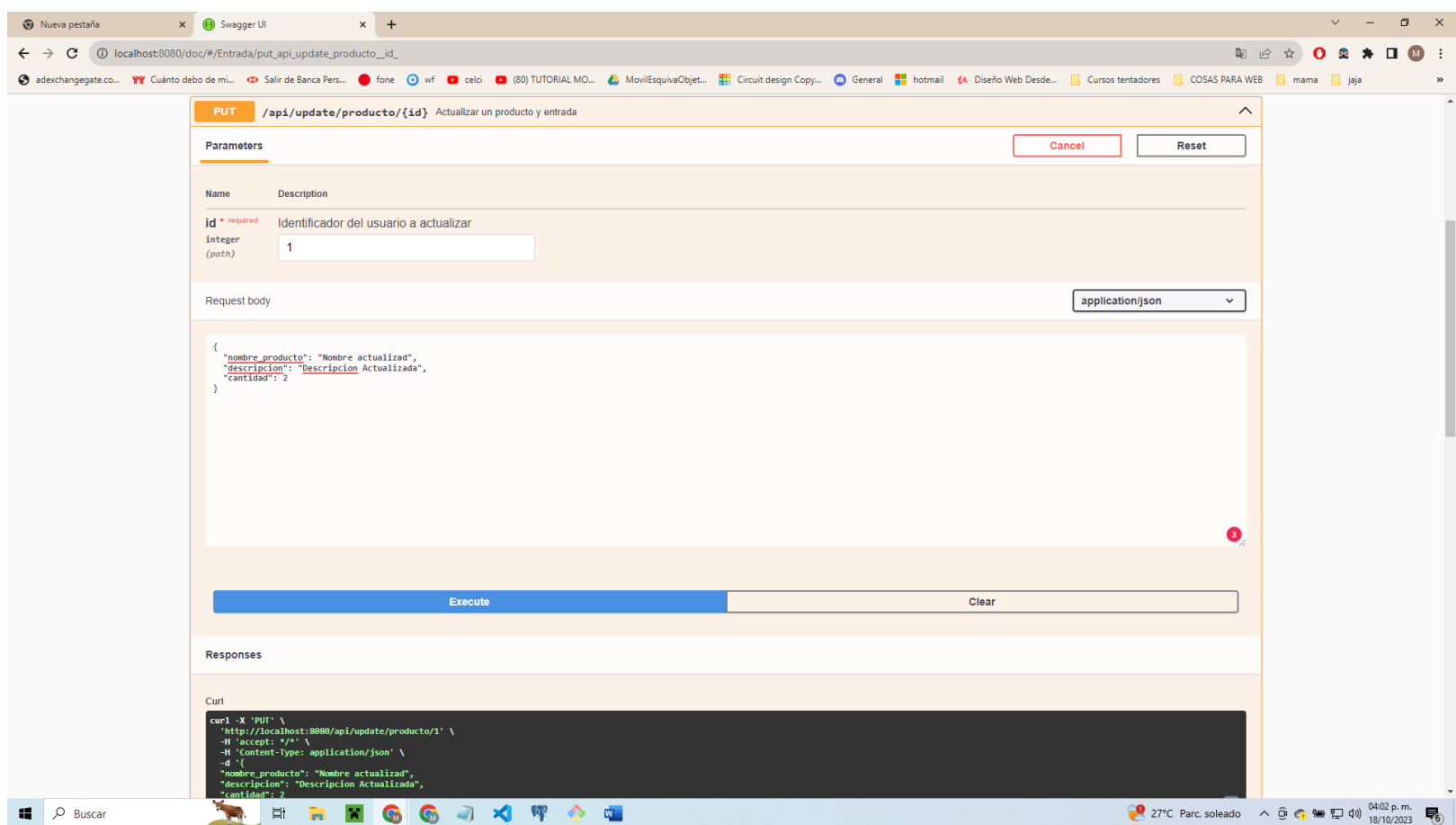


Ilustración 4Put

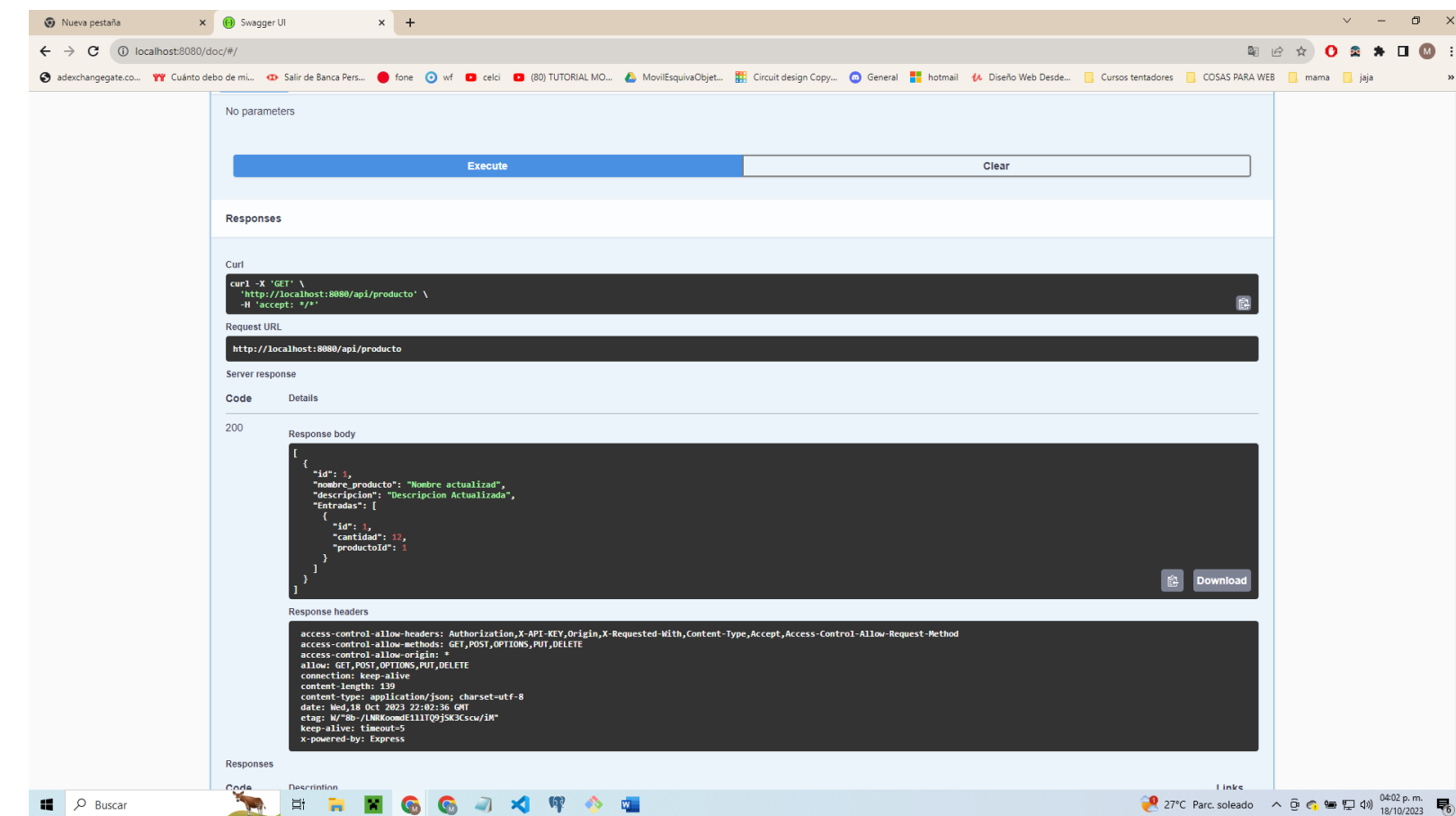


Ilustración 3Get despues dle put

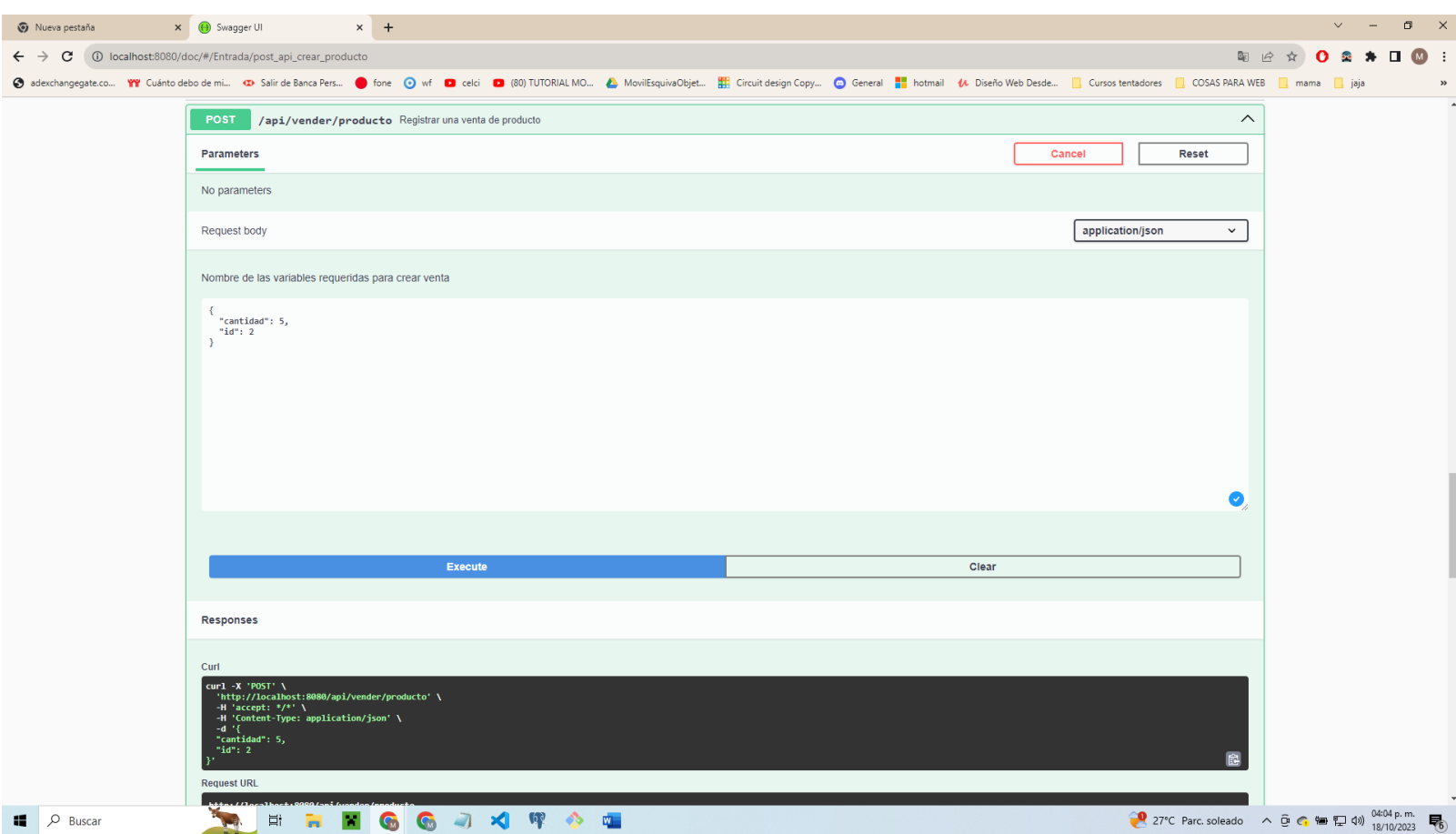


Ilustración 6Post venta

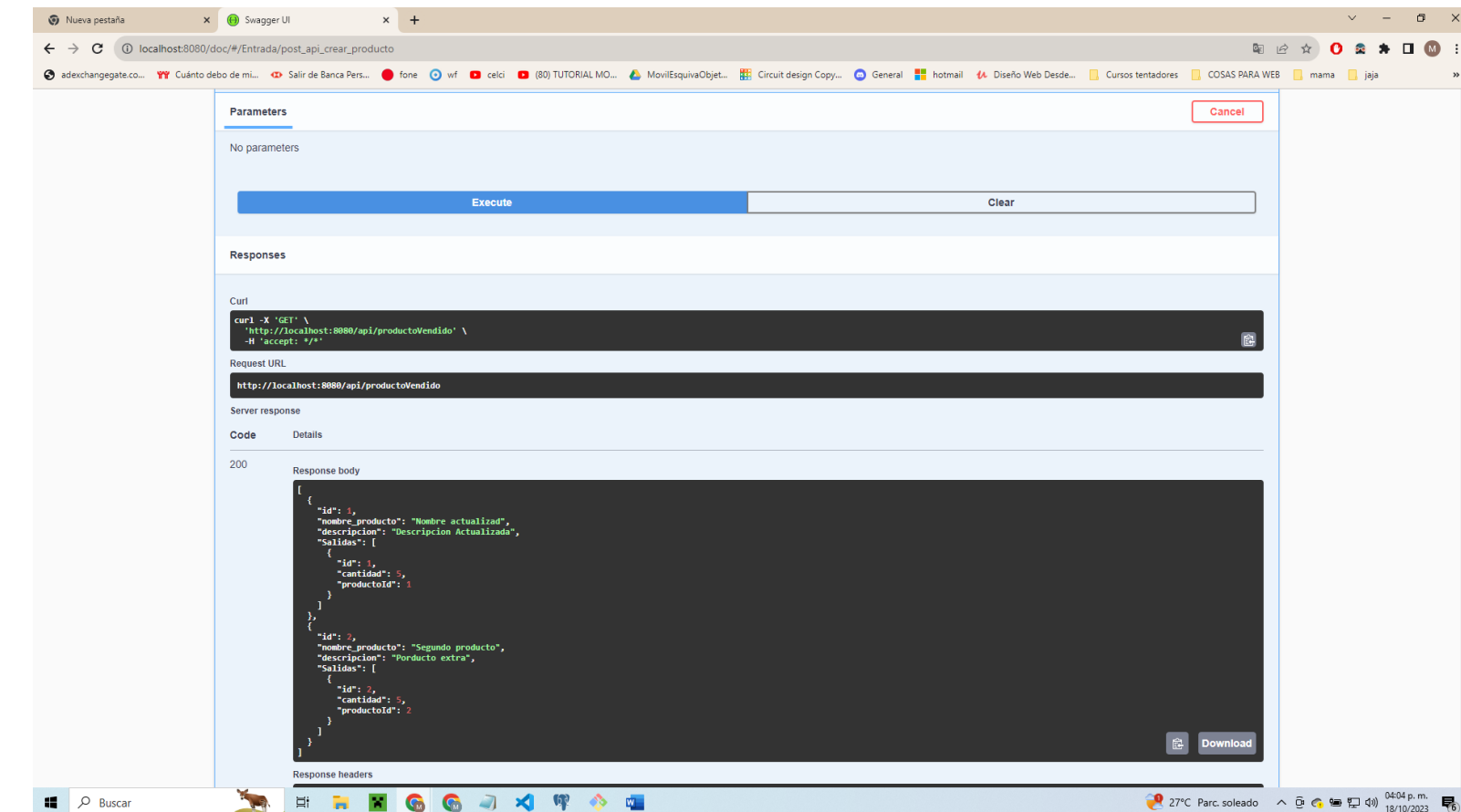


Ilustración 5Get de venta

Swagger UI interface showing a PUT request configuration for the endpoint `/api/update/venta/{id}`. The request body is set to `application/json` with the payload `{ "cantidad": 4 }`. The response section shows the corresponding curl command.

```
curl -X 'PUT' \
  'http://localhost:8080/api/update/venta/2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "cantidad": 4
  }'
```

Ilustración 8Put de venta

Swagger UI interface showing a GET request configuration for the endpoint `/api/productoVendido`. The response section shows the corresponding curl command and the server response body.

```
curl -X 'GET' \
  'http://localhost:8080/api/productoVendido' \
  -H 'accept: */*' \
  -d ''
```

Request URL: `http://localhost:8080/api/productoVendido`

Server response

Code	Details
200	<pre>{   "id": 1,   "nombre_producto": "Nombre actualizado",   "descripcion": "Descripcion Actualizada",   "Salidas": [     {       "id": 1,       "cantidad": 5,       "productoId": 1     }   ],   "id": 2,   "nombre_producto": "Segundo producto",   "descripcion": "Producto extra",   "Salidas": [     {       "id": 2,       "cantidad": 9,       "productoId": 2     }   ] }</pre>

Ilustración 7Get despues de put

-- Crea una vista que muestra el listado del inventario con información del producto

CREATE OR REPLACE VIEW vista\_inventario\_producto AS

SELECT

p.id AS producto\_id,

p.nombre\_producto AS producto\_nombre,

p.descripcion AS producto\_descripcion,

i.cantidad\_actual AS inventario\_cantidad\_actual

FROM

"Productos" p

LEFT JOIN

"Inventarios" i ON p.id = i."productoid";

-- Crea un procedimiento almacenado (SP) para calcular el saldo del inventario de un producto

CREATE OR REPLACE FUNCTION calcular\_saldo\_inventario(producto\_id INT)

RETURNS INTEGER AS \$\$

DECLARE

saldo INTEGER;

BEGIN

SELECT COALESCE((SELECT cantidad\_actual FROM "Inventarios" WHERE "productoid" =  
producto\_id), 0) -

COALESCE((SELECT SUM(cantidad) FROM "Entradas" WHERE "productoid" = producto\_id), 0)

+

COALESCE((SELECT SUM(cantidad) FROM "Salidas" WHERE "productoid" = producto\_id), 0)

INTO saldo;

RETURN saldo;

END;

\$\$ LANGUAGE plpgsql;

```
SELECT calcular_saldo_inventario(1);
```

```
-- Crea una función que se activará cuando se registre una entrada
```

```
CREATE OR REPLACE FUNCTION aumentar_inventario()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
-- Incrementa el inventario
```

```
UPDATE Inventario
```

```
SET cantidad_actual = cantidad_actual + NEW.cantidad
```

```
WHERE productoid = NEW.productoid;
```

```
RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
-- Crea un trigger que se activará después de insertar en la tabla Entrada
```

```
CREATE TRIGGER trigger_aumentar_inventario
```

```
AFTER INSERT
```

```
ON Entrada
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION aumentar_inventario();
```