

PARCIAL II CORTE

Base de Datos Masivas

Juan David Moreno Pico

ID:852417

Docente. Alex Matallana

Ingeniería de Sistemas.

Corporación Universitaria Uniminuto.

Zipaquirá

2025

INICIAMOS EL PROYECTO CON NPM

```
PS C:\Users\juand\OneDrive\Escritorio\PARCIAL2> npm init -y  
Wrote to C:\Users\juand\OneDrive\Escritorio\PARCIAL2\package.json:
```

```
PS C:\Users\juand\OneDrive\Escritorio\PARCIAL2> npm i express pg dotenv cors
```

VAMOS A SUPABASE Y CREAMOS LA BASE DE DATOS Y LAS TABLAS

```
1 create database parcial;
```

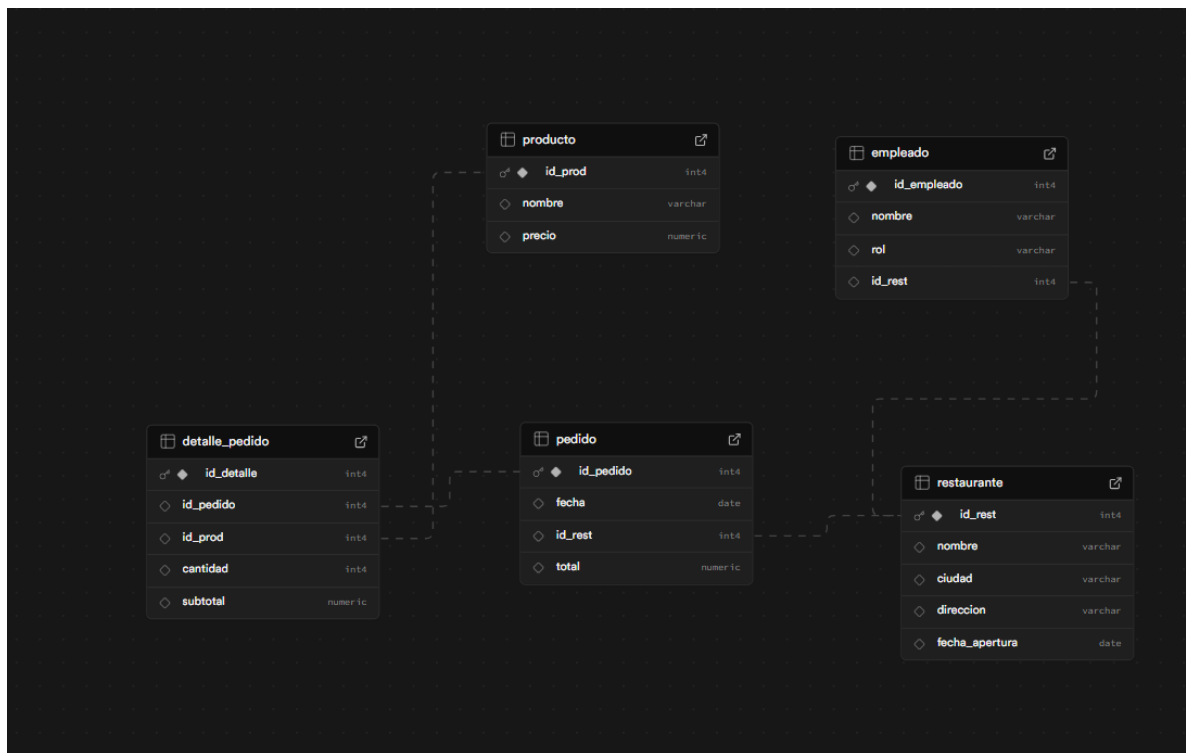
```
CREATE TABLE restaurante (  
  id_rest INT PRIMARY KEY,  
  nombre VARCHAR(100),  
  ciudad VARCHAR(100),  
  direccion VARCHAR(150),  
  fecha_apertura DATE  
);  
  
CREATE TABLE empleado (  
  id_empleado INT PRIMARY KEY,  
  nombre VARCHAR(100),  
  rol VARCHAR(50),  
  id_rest INT,  
  FOREIGN KEY (id_rest) REFERENCES restaurante(id_rest) ON DELETE CASCADE  
);  
  
CREATE TABLE producto (  
  id_prod INT PRIMARY KEY,  
  nombre VARCHAR(100),  
  precio NUMERIC(10,2)  
);
```

```

16 CREATE TABLE producto (
17     id_prod INT PRIMARY KEY,
18     nombre VARCHAR(100),
19     precio NUMERIC(10,2)
20 );
21 CREATE TABLE pedido (
22     id_pedido INT PRIMARY KEY,
23     fecha DATE,
24     id_rest INT,
25     total NUMERIC(10,2),
26     FOREIGN KEY (id_rest) REFERENCES restaurante(id_rest) ON DELETE CASCADE
27 );
28 CREATE TABLE detalle_pedido (
29     id_detalle INT PRIMARY KEY,
30     id_pedido INT,
31     id_prod INT,
32     cantidad INT,
33     subtotal NUMERIC(10,2),
34     FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido) ON DELETE CASCADE,
35     FOREIGN KEY (id_prod) REFERENCES producto(id_prod) ON DELETE CASCADE
36 );

```

VALIDAMOS QUE ESTE CORRECTA LAS TABLAS Y SUS RELACIONES.



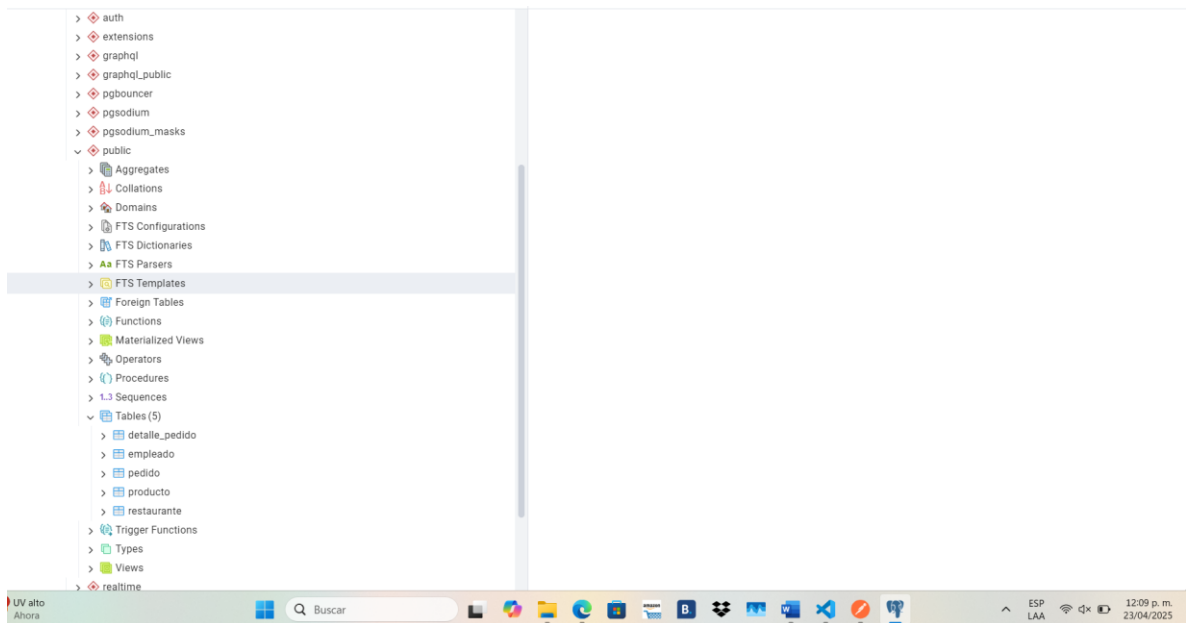
INSERTAMOS REGISTROS

```
1 INSERT INTO producto (id_prod, nombre, precio) VALUES
2 (1, 'Hamburguesa Clásica', 8.99),
3 (2, 'Pizza Margarita', 12.50),
4 (3, 'Ensalada César', 6.75),
5 (4, 'Sopa de Tomate', 4.99),
6 (5, 'Pasta Alfredo', 10.25),
7 (6, 'Tacos de Pollo', 7.50),
8 (7, 'Sándwich de Pavo', 5.99),
9 (8, 'Helado de Vainilla', 3.25),
10 (9, 'Refresco de Cola', 2.00),
11 (10, 'Agua Mineral', 1.50),
12 (11, 'Cerveza Artesanal', 5.75),
13 (12, 'Tarta de Manzana', 4.50),
14 (13, 'Café Americano', 2.25),
15 (14, 'Batido de Fresa', 3.75),
16 (15, 'Alitas Picantes', 9.99),
17 (16, 'Nachos con Queso', 6.25),
18 (17, 'Sushi Roll', 11.50),
19 (18, 'Pollo a la Parrilla', 14.99),
20 (19, 'Lasagna', 10.75),
21 (20, 'Tiramisu', 5.25),
22 (21, 'Burrito de Carne', 8.50),
23 (22, 'Pan de Azúcar', 3.99)
```

Results Chart Export ✓ Source Primary Database Role postgres Run CTRL ↵

Success. No rows returned

CONEXIÓN CON PGADMIN4



CREAMOS LAS APIS DEL CRUD DE CADA TABLA

API Restaurante

```
// Rutas de restaurante

app.get('/restaurantes', async (req, res) => {
  const result = await pool.query('SELECT * FROM restaurante');
  res.json(result.rows);
});

app.post('/restaurantes', async (req, res) => {
  const { nombre, ciudad, direccion, fecha_apertura } = req.body;
  await pool.query('INSERT INTO restaurante(nombre, ciudad, direccion, fecha_apertura) VALUES ($1, $2, $3, $4)', [nombre, ciudad, direccion, fecha_apertura]);
  res.send('Restaurante creado');
});

app.put('/restaurantes/:id', async (req, res) => {
  const { id } = req.params;
  const { nombre, ciudad, direccion, fecha_apertura } = req.body;
  await pool.query('UPDATE restaurante SET nombre=$1, ciudad=$2, direccion=$3, fecha_apertura=$4 WHERE id_rest=$5', [nombre, ciudad, direccion, fecha_apertura, id]);
  res.send('Restaurante actualizado');
});

app.delete('/restaurantes/:id', async (req, res) => {
  await pool.query('DELETE FROM restaurante WHERE id_rest=$1', [req.params.id]);
  res.send('Restaurante eliminado');
});
```

API Empleados

```
// Rutas de empleados

app.get('/empleados', async (req, res) => {
  const result = await pool.query('SELECT * FROM empleado');
  res.json(result.rows);
});

app.post('/empleados', async (req, res) => {
  const { nombre, rol, id_rest } = req.body;
  await pool.query('INSERT INTO empleado(nombre, rol, id_rest) VALUES ($1, $2, $3)', [nombre, rol, id_rest]);
  res.send('Empleado creado');
});

app.put('/empleados/:id', async (req, res) => {
  const { id } = req.params;
  const { nombre, rol, id_rest } = req.body;
  await pool.query('UPDATE empleado SET nombre=$1, rol=$2, id_rest=$3 WHERE id_empleado=$4', [nombre, rol, id_rest, id]);
  res.send('Empleado actualizado');
});

app.delete('/empleados/:id', async (req, res) => {
  await pool.query('DELETE FROM empleado WHERE id_empleado=$1', [req.params.id]);
  res.send('Empleado eliminado');
});
```

API Productos

```
// Rutas de productos

app.get('/productos', async (req, res) => {
  const result = await pool.query('SELECT * FROM producto');
  res.json(result.rows);
});

app.post('/productos', async (req, res) => {
  const { nombre, precio } = req.body;
  await pool.query('INSERT INTO producto(nombre, precio) VALUES ($1, $2)', [nombre, precio]);
  res.send('Producto creado');
});

app.put('/productos/:id', async (req, res) => {
  const { id } = req.params;
  const { nombre, precio } = req.body;
  await pool.query('UPDATE producto SET nombre=$1, precio=$2 WHERE id_prod=$3', [nombre, precio, id]);
  res.send('Producto actualizado');
});

app.delete('/productos/:id', async (req, res) => {
  await pool.query('DELETE FROM producto WHERE id_prod=$1', [req.params.id]);
  res.send('Producto eliminado');
});
```

API Pedidos

```
// Rutas de pedidos

app.get('/pedidos', async (req, res) => {
  const result = await pool.query('SELECT * FROM pedido');
  res.json(result.rows);
});

app.post('/pedidos', async (req, res) => {
  const { fecha, id_rest, total } = req.body;
  await pool.query('INSERT INTO pedido(fecha, id_rest, total) VALUES ($1, $2, $3)', [fecha, id_rest, total]);
  res.send('Pedido creado');
});

app.put('/pedidos/:id', async (req, res) => {
  const { id } = req.params;
  const { fecha, id_rest, total } = req.body;
  await pool.query('UPDATE pedido SET fecha=$1, id_rest=$2, total=$3 WHERE id_pedido=$4', [fecha, id_rest, total, id]);
  res.send('Pedido actualizado');
});

app.delete('/pedidos/:id', async (req, res) => {
  await pool.query('DELETE FROM pedido WHERE id_pedido=$1', [req.params.id]);
  res.send('Pedido eliminado');
});
```

API Detalles

```
// Rutas de detalles de pedido

app.get('/detalles', async (req, res) => {
  const result = await pool.query('SELECT * FROM detalle_pedido');
  res.json(result.rows);
});

app.post('/detalles', async (req, res) => {
  const { id_pedido, id_prod, cantidad, subtotal } = req.body;
  await pool.query('INSERT INTO detalle_pedido(id_pedido, id_prod, cantidad, subtotal) VALUES ($1, $2, $3, $4)', [id_pedido, id_prod, cantidad, subtotal]);
  res.send('Detalle agregado');
});

app.put('/detalles/:id', async (req, res) => {
  const { id } = req.params;
  const { id_pedido, id_prod, cantidad, subtotal } = req.body;
  await pool.query('UPDATE detalle_pedido SET id_pedido=$1, id_prod=$2, cantidad=$3, subtotal=$4 WHERE id_detalle=$5', [id_pedido, id_prod, cantidad, subtotal, id]);
  res.send('Detalle actualizado');
});

app.delete('/detalles/:id', async (req, res) => {
  await pool.query('DELETE FROM detalle_pedido WHERE id_detalle=$1', [req.params.id]);
  res.send('Detalle eliminado');
});
```

PROBAMOS LAS APIS EN POSTMAN

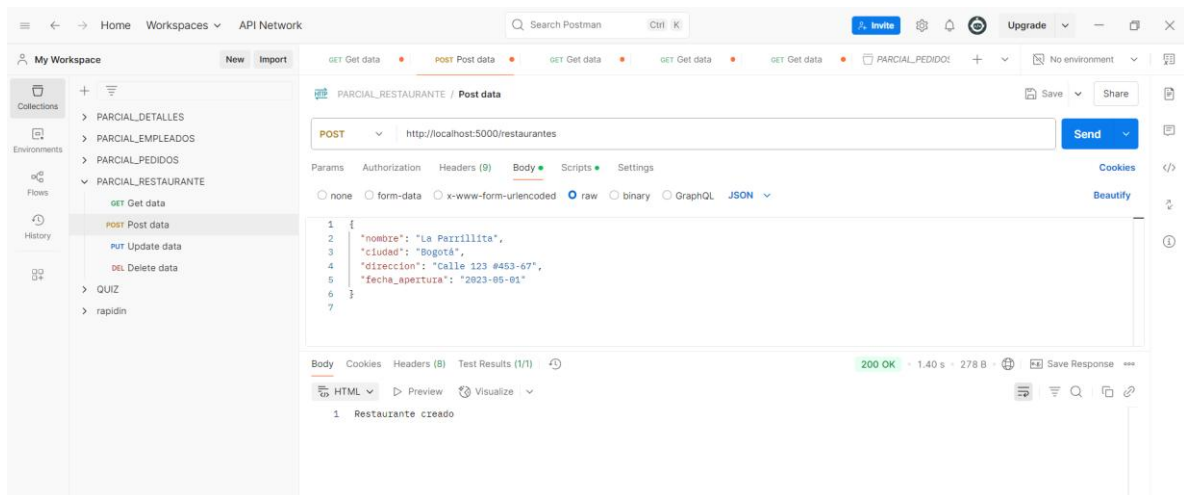
METODO GET EN TABLA RESTAURANTE:

The screenshot shows the Postman interface with a GET request to `http://localhost:5000/restaurantes` successfully executed. The response is a JSON array of restaurant data.

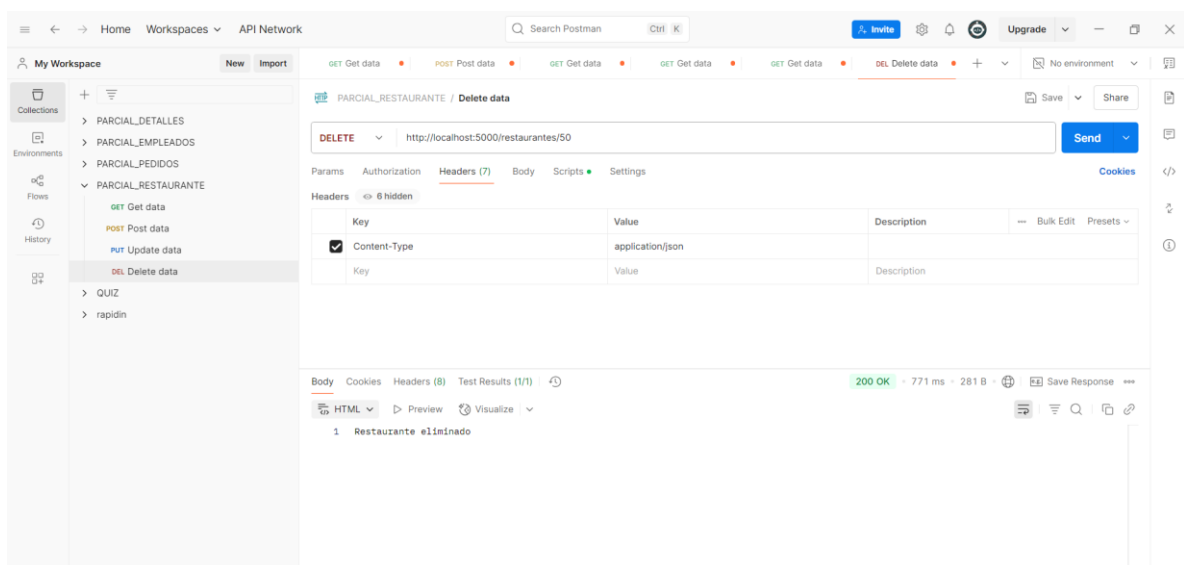
| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

```
1 [
2   {
3     "id_rest": 1,
4     "nombre": "Sabores del Sur",
5     "ciudad": "Buenos Aires",
6     "direccion": "Av. Corrientes 123",
7     "fecha_apertura": "2020-05-15T05:00:00.000Z"
8   },
9   {
10    "id_rest": 2,
11    "nombre": "La Parrilla",
12    "ciudad": "Madrid",
13    "direccion": "Calle Gran Vía 45",
14    "fecha_apertura": "2019-03-10T05:00:00.000Z"
15  },
16  {
17    "id_rest": 3,
18    "nombre": "El Marinero",
19    "ciudad": "Barcelona",
20    "direccion": "Paseo Marítim de la Barceloneta 363"
21  }
22 ]
```

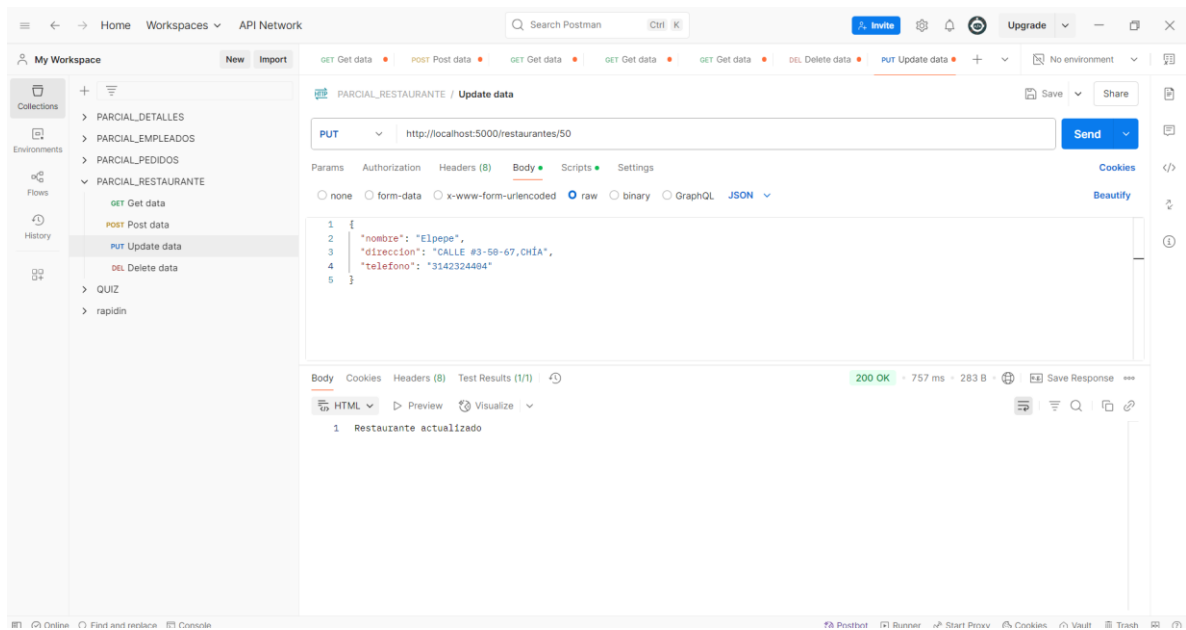
METODO POST EN RESTAURANTE



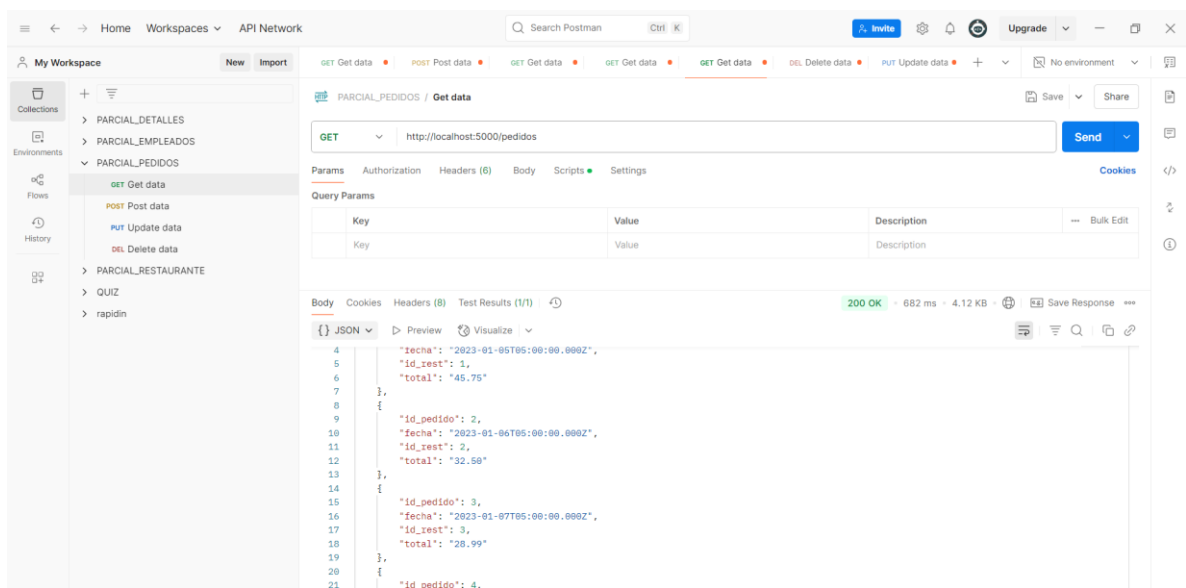
METODO DELETE EN RESTAURANTE



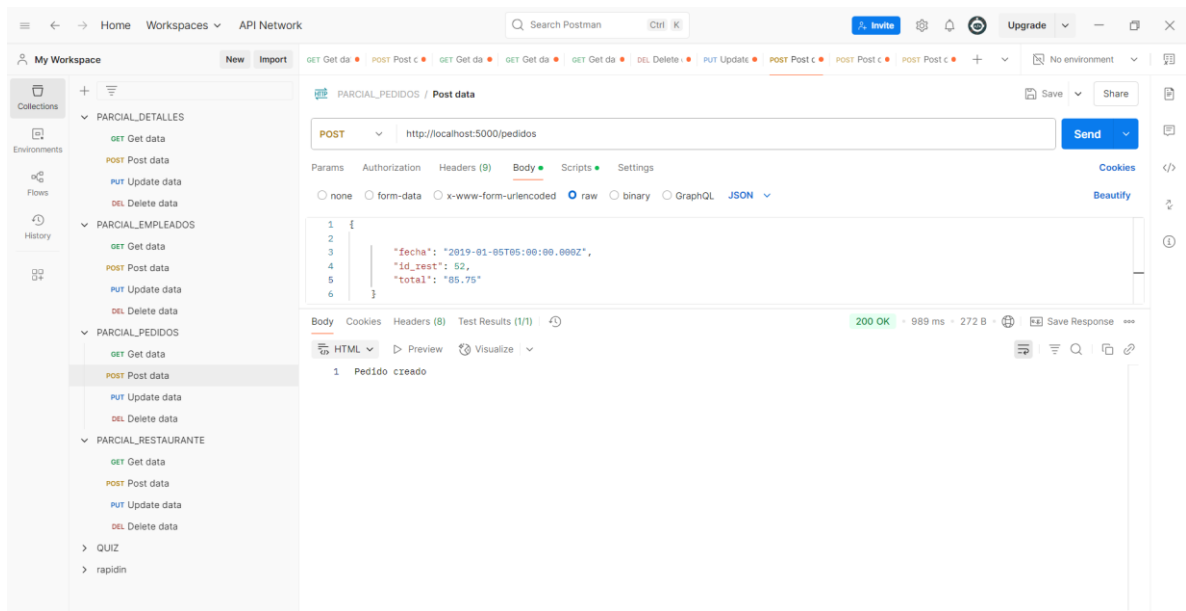
METODO PUT EN RESTAURANTE



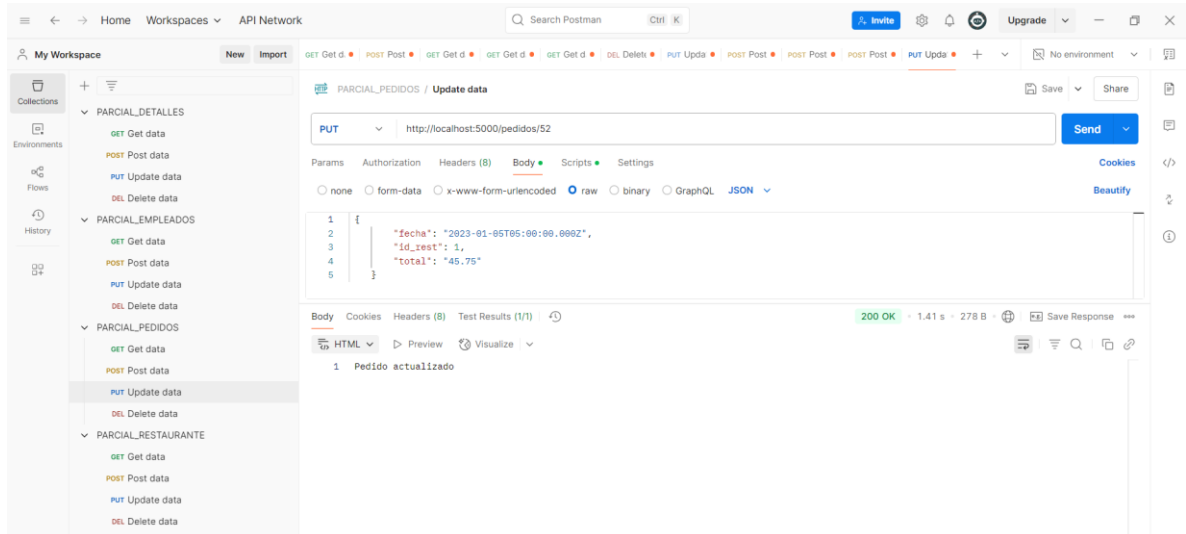
METODO GET EN TABLA PEDIDO



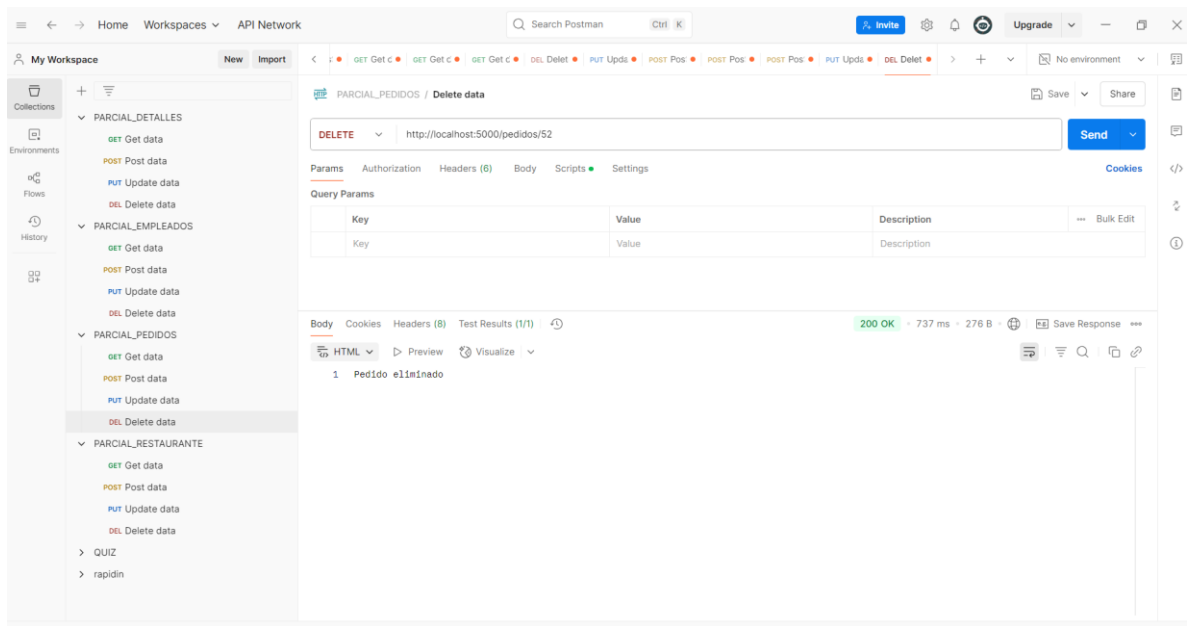
METODO POST EN TABLA PEDIDO



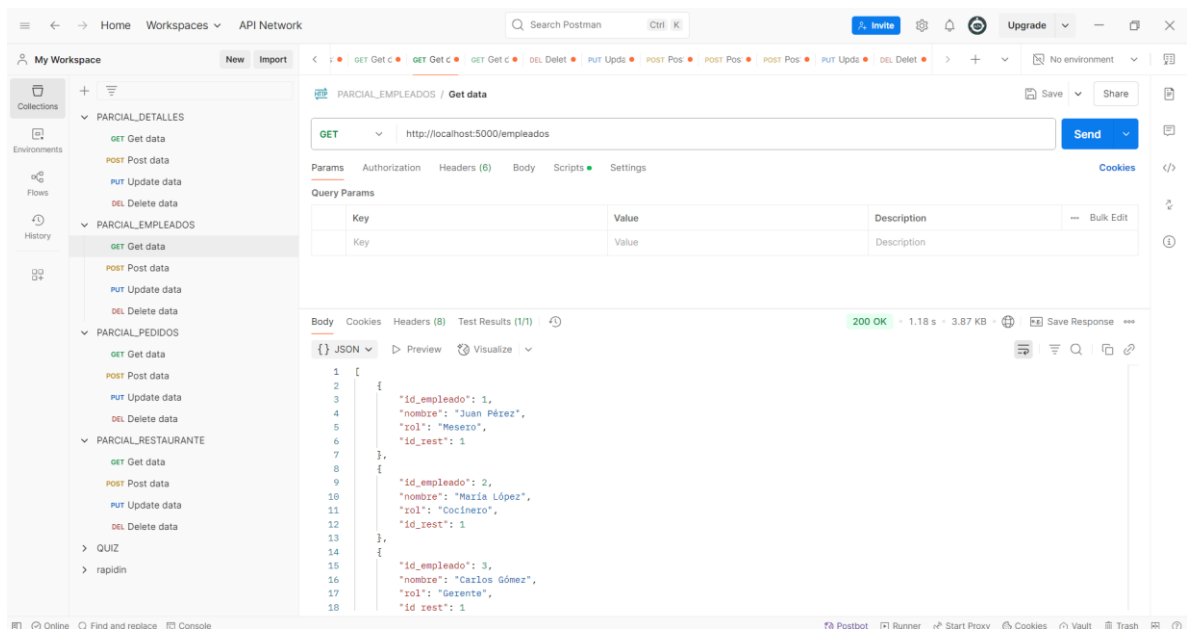
METODO PUT EN TABLA PEDIDO



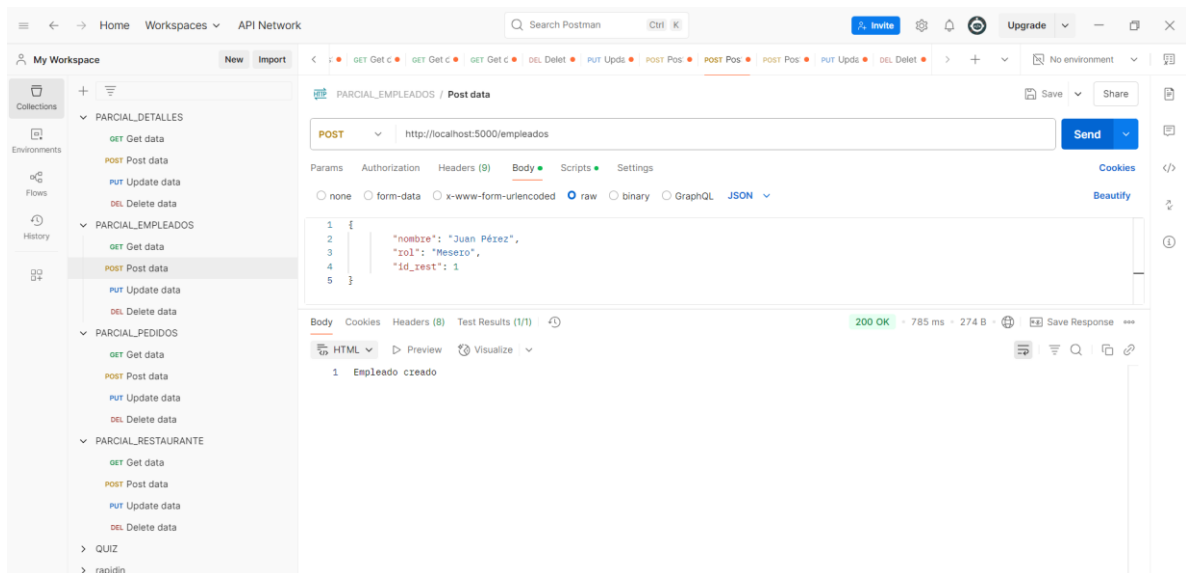
METODO DELETE EN TABLA PEDIDO



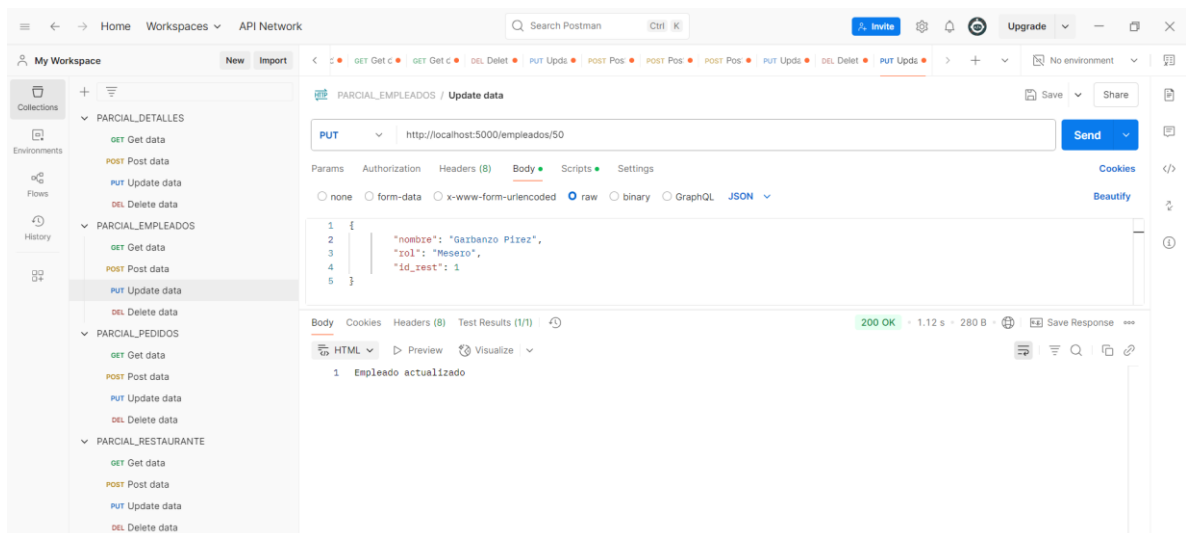
METODO GET EN TABLA EMPLEADO



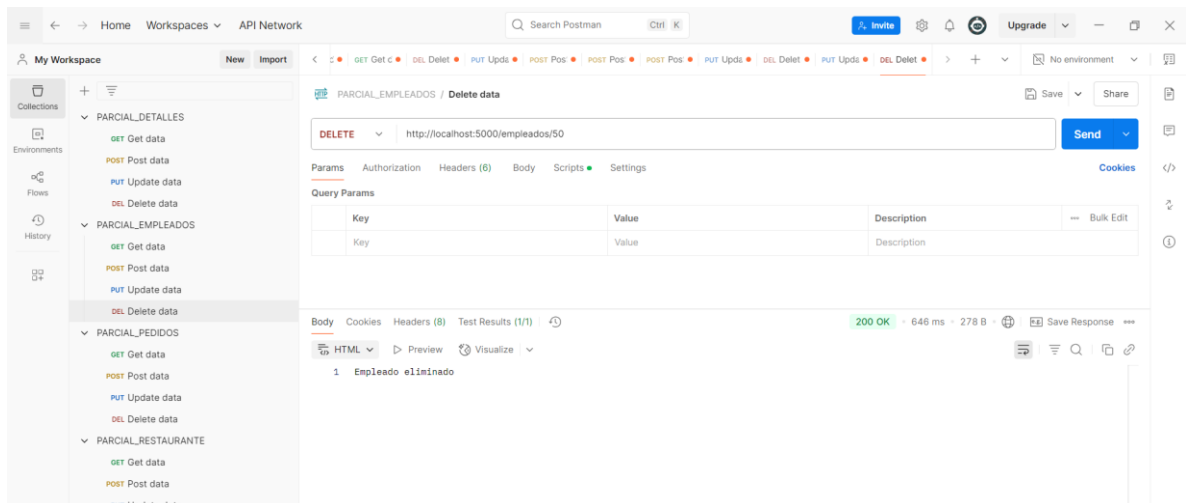
METODO POST EN TABLA EMPLEADO



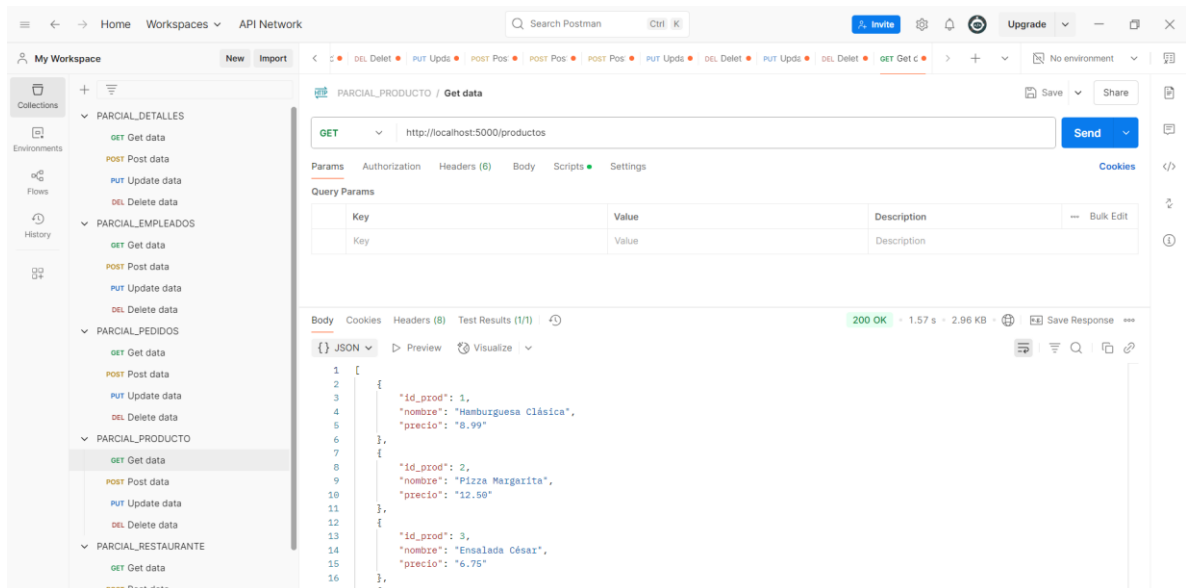
METODO PUT EN TABLA EMPLEADO



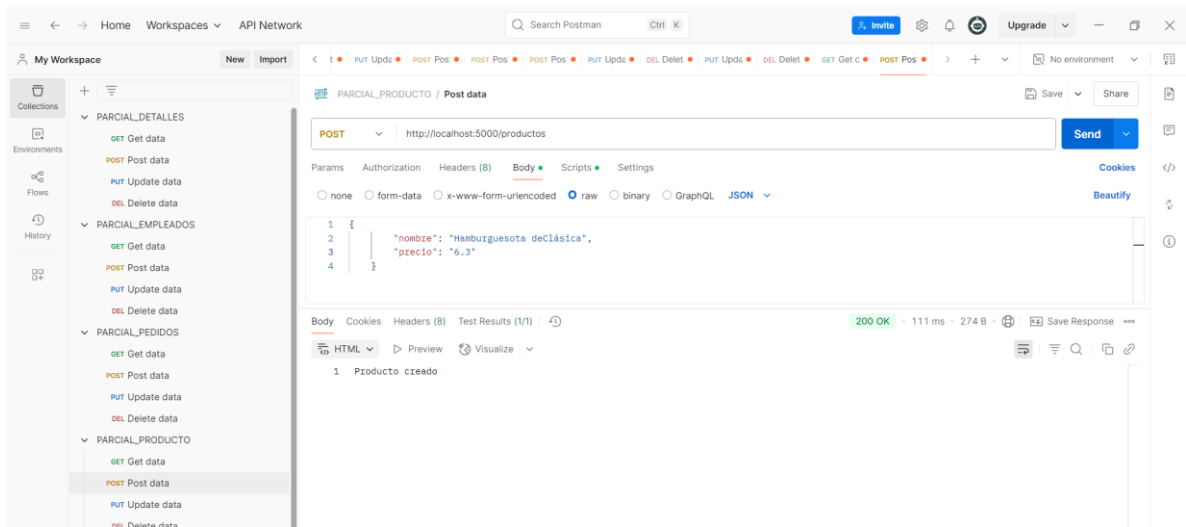
METODO DELETE EN TABLA EMPLEADO



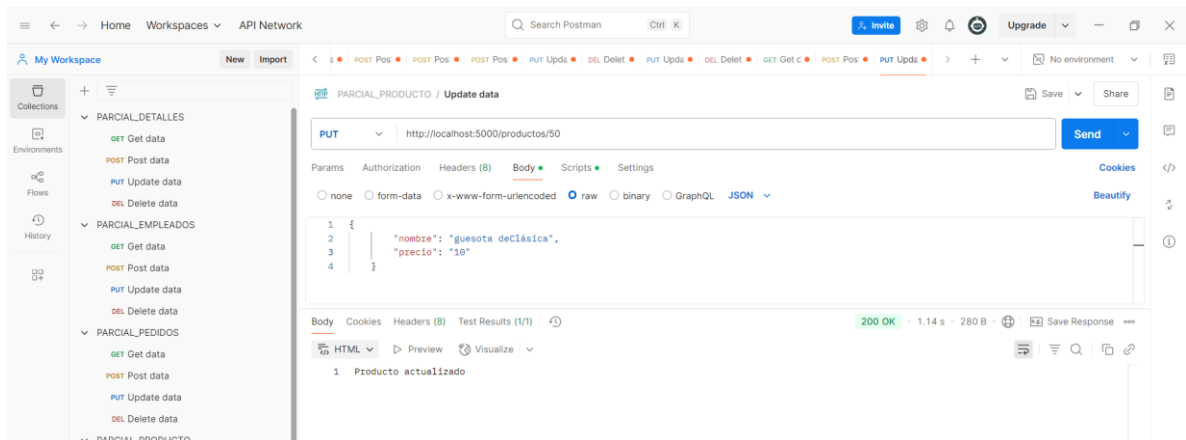
METODO GET EN TABLA PRODUCTO



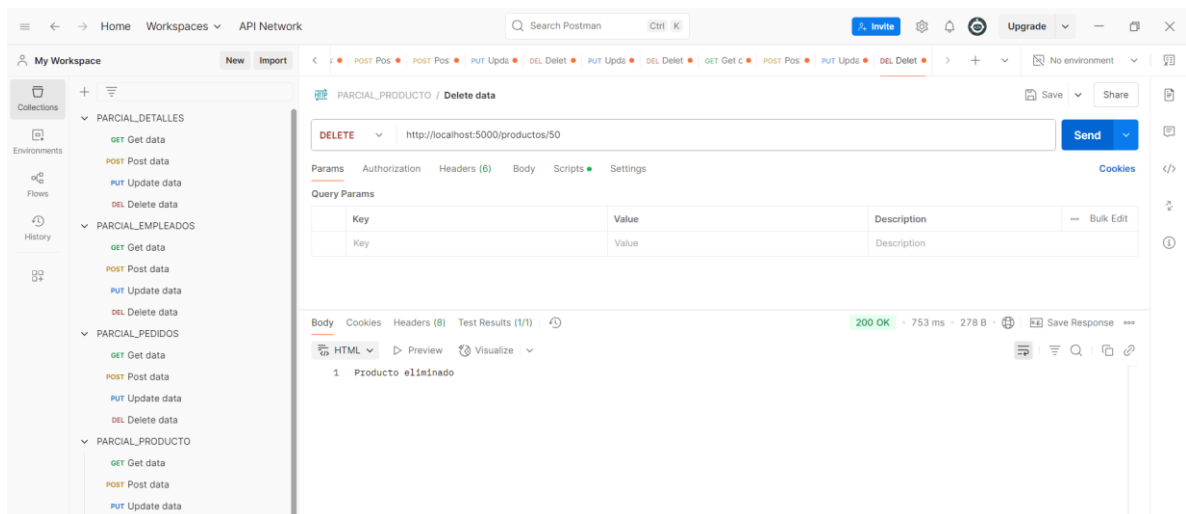
METODO POST EN TABLA PRODUCTO



METODO PUT EN TABLA PRODUCTO



METODO DELETE EN TABLA PRODUCTO



METODO GET EN TABLA DETALLE_PEDIDO

PARCIAL_DETALLES / Get data

GET http://localhost:5000/detalles

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (8) Test Results (1/1)

200 OK - 874 ms - 3.93 KB

JSON Preview Visualize

```
1 {
2   {
3     "id_detalle": 1,
4     "id_pedido": 1,
5     "id_prod": 1,
6     "cantidad": 2,
7     "subtotal": "17.98"
8   },
9   {
10    "id_detalle": 2,
11    "id_pedido": 1,
12    "id_prod": 3,
13    "cantidad": 1,
14    "subtotal": "6.75"
15  },
16  {
17    "id_detalle": 3,
```

METODO POST EN TABLA DETALLE_PEDIDO

PARCIAL_DETALLES / Post data

POST http://localhost:5000/detalles

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id_pedido": 4,
3   "id_prod": 8,
4   "cantidad": 2,
5   "subtotal": "28.98"
6 }
```

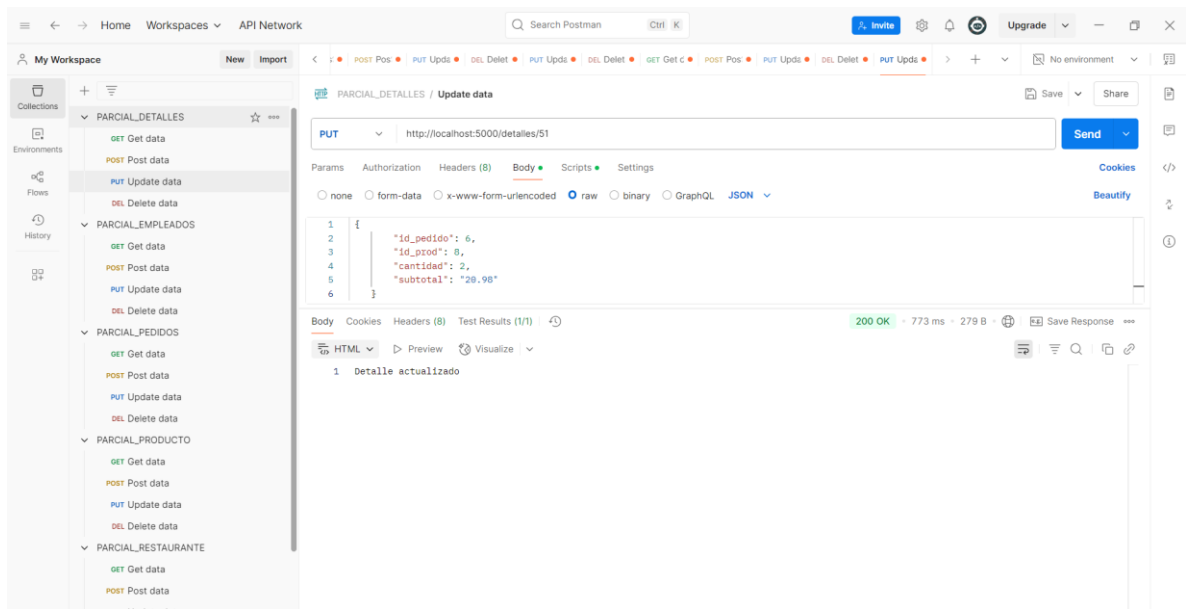
Body Cookies Headers (8) Test Results (1/1)

200 OK - 740 ms - 276 B

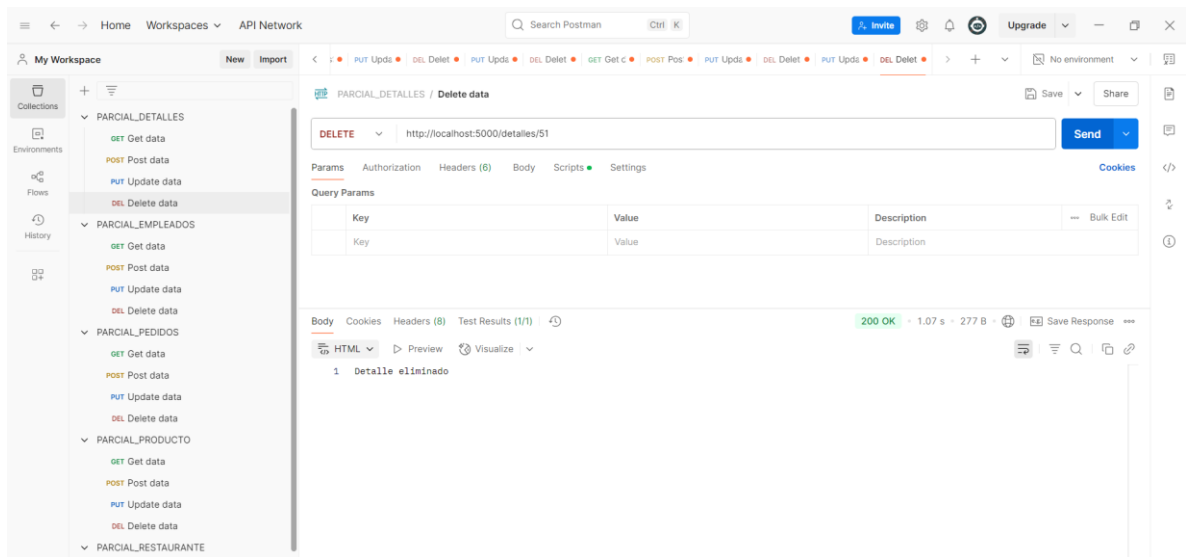
HTML Preview Visualize

1 Detalle agregado

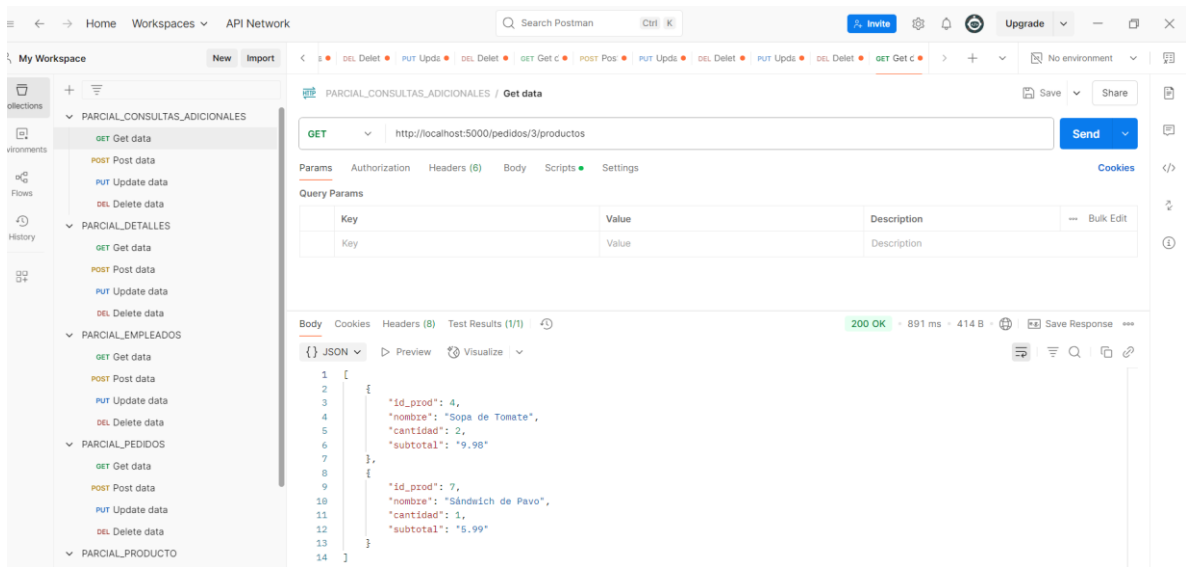
METODO PUT EN TABLA DETALLE_PEDIDO



METODO DELETE EN TABLA DETALLE_PEDIDO



OBTENER TODOS LOS PRODUCTOS DE UN PEDIDO ESPECÍFICO



PARCIALCONSULTASADICIONALES / Get data

GET http://localhost:5000/pedidos/3/productos

Params Authorization Headers (6) Body Scripts Settings Cookies

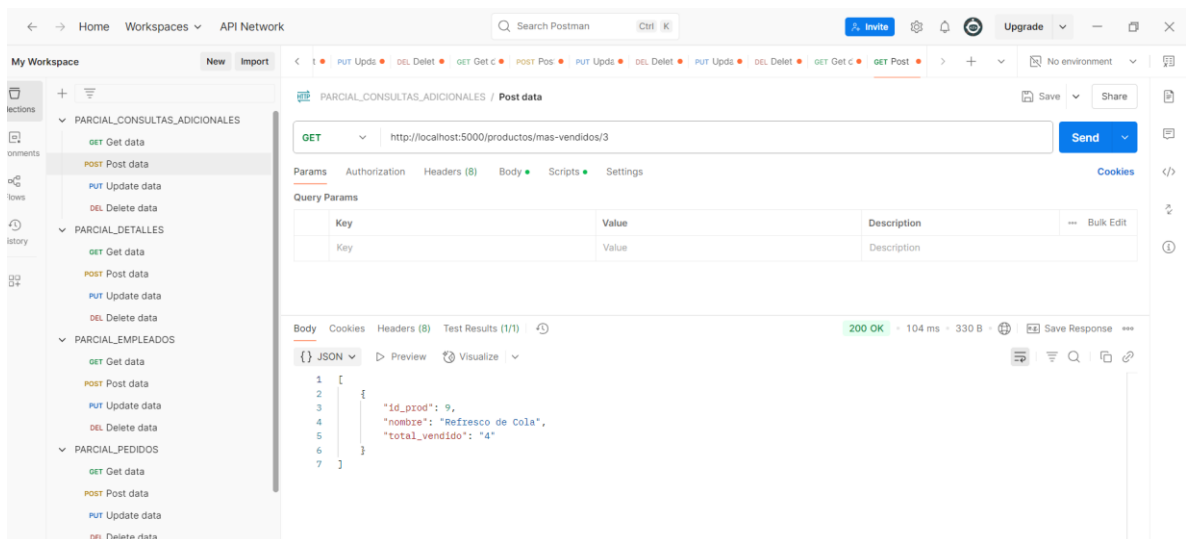
Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (8) Test Results (1/1) 200 OK 891 ms 414 B Save Response

```
{
  "id_prod": 4,
  "nombre": "Sopa de Tomate",
  "cantidad": 2,
  "subtotal": "9.98"
},
{
  "id_prod": 7,
  "nombre": "Sándwich de Pavo",
  "cantidad": 1,
  "subtotal": "5.99"
}
```

OBTENER LOS PRODUCTOS MÁS VENDIDOS (MÁS DE X UNIDADES)



PARCIALCONSULTASADICIONALES / Post data

GET http://localhost:5000/productos/mas-vendidos/3

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (8) Test Results (1/1) 200 OK 104 ms 330 B Save Response

```
{
  "id_prod": 9,
  "nombre": "Refresco de Cola",
  "total_vendido": "4"
}
```

OBTENER EL TOTAL DE VENTAS POR RESTAURANTE

PARCIAL CONSULTAS ADICIONALES / Update data

GET http://localhost:5000/ventas/restaurantes

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (8) Test Results (1/1) 200 OK · 1.18 s · 3.33 KB Save Response

```
{
  "id_rest": 52,
  "nombre": "La Parrillita",
  "total_ventas": "85.75"
},
{
  "id_rest": 13,
  "nombre": "El Fogón",
  "total_ventas": "75.25"
},
{
  "id_rest": 24,
  "nombre": "La Trattoria",
  "total_ventas": "62.75"
}
```

OBTENER LOS PEDIDOS REALIZADOS EN UNA FECHA ESPECÍFICA

PARCIAL CONSULTAS ADICIONALES / Delete data

GET http://localhost:5000/pedidos/fecha/2023-01-06

Params Authorization Headers (6) Body Scripts Settings Cookies

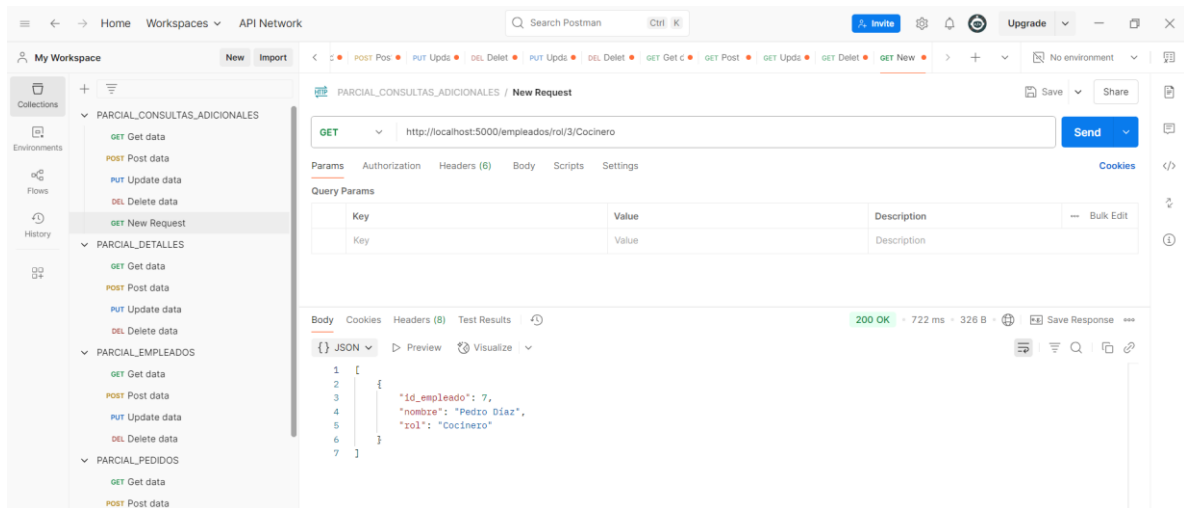
Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (8) Test Results (1/1) 200 OK · 664 ms · 347 B Save Response

```
{
  "id_pedido": 2,
  "fecha": "2023-01-06T05:00:00.000Z",
  "id_rest": 2,
  "total": "32.50"
}
```

OBTENER LOS EMPLEADOS POR ROL EN UN RESTAURANTE



APIS PARA LAS CONSULTAS ADICIONALES

```
//APIS CONSULTAS ADICIONALES
app.get('/pedidos/:id/productos', async (req, res) => {
  const { id } = req.params;
  try {
    const result = await pool.query(`
      SELECT p.id_prod, pr.nombre, p.cantidad, p.subtotal
      FROM detalle_pedido p
      INNER JOIN producto pr ON p.id_prod = pr.id_prod
      WHERE p.id_pedido = $1
    `, [id]);
    res.json(result.rows);
  } catch (error) {
    console.error(error);
    res.status(500).send('Error al obtener productos del pedido');
  }
});

app.get('/productos/mas-vendidos/:cantidad', async (req, res) => {
  const { cantidad } = req.params;
  const result = await pool.query(`
    SELECT p.id_prod, p.nombre, SUM(dp.cantidad) AS total_vendido
    FROM detalle_pedido dp
    JOIN producto p ON dp.id_prod = p.id_prod
    GROUP BY p.id_prod, p.nombre
    HAVING SUM(dp.cantidad) > $1
    ORDER BY total_vendido DESC
  `, [cantidad]);
  res.json(result.rows);
});
```

```
app.get('/ventas/restaurantes', async (req, res) => {
  const result = await pool.query(`
    SELECT r.id_rest, r.nombre, COALESCE(SUM(p.total), 0) AS total_ventas
    FROM restaurante r
    LEFT JOIN pedido p ON r.id_rest = p.id_rest
    GROUP BY r.id_rest, r.nombre
    ORDER BY total_ventas DESC
  `);
  res.json(result.rows);
});

app.get('/pedidos/fecha/:fecha', async (req, res) => {
  const { fecha } = req.params;
  const result = await pool.query(`
    SELECT * FROM pedido
    WHERE DATE(fecha) = $1
  `, [fecha]);
  res.json(result.rows);
});

app.get('/empleados/rol/:id_rest/:rol', async (req, res) => {
  const { id_rest, rol } = req.params;
  const result = await pool.query(`
    SELECT id_empleado, nombre, rol
    FROM empleado
    WHERE id_rest = $1 AND rol = $2
  `, [id_rest, rol]);
  res.json(result.rows);
});
```