

Parcial 2

David Santiago Noguera Perez - 827823

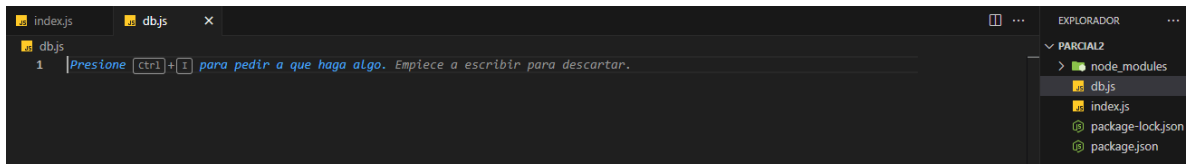
Corporación Universitaria Minuto de Dios

Ing. William Alexander Matallana Porras

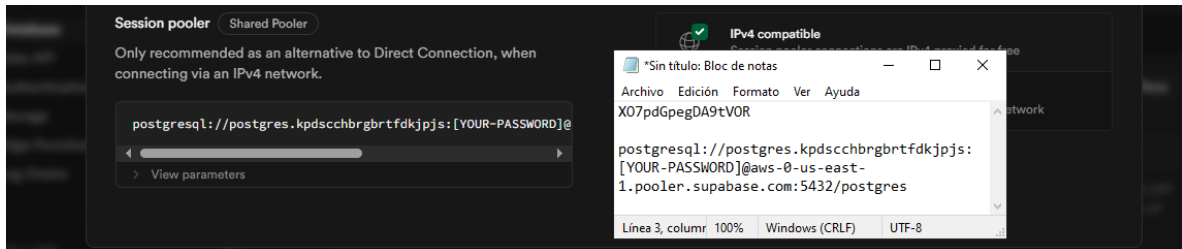
10-60747: Bases de datos

2025

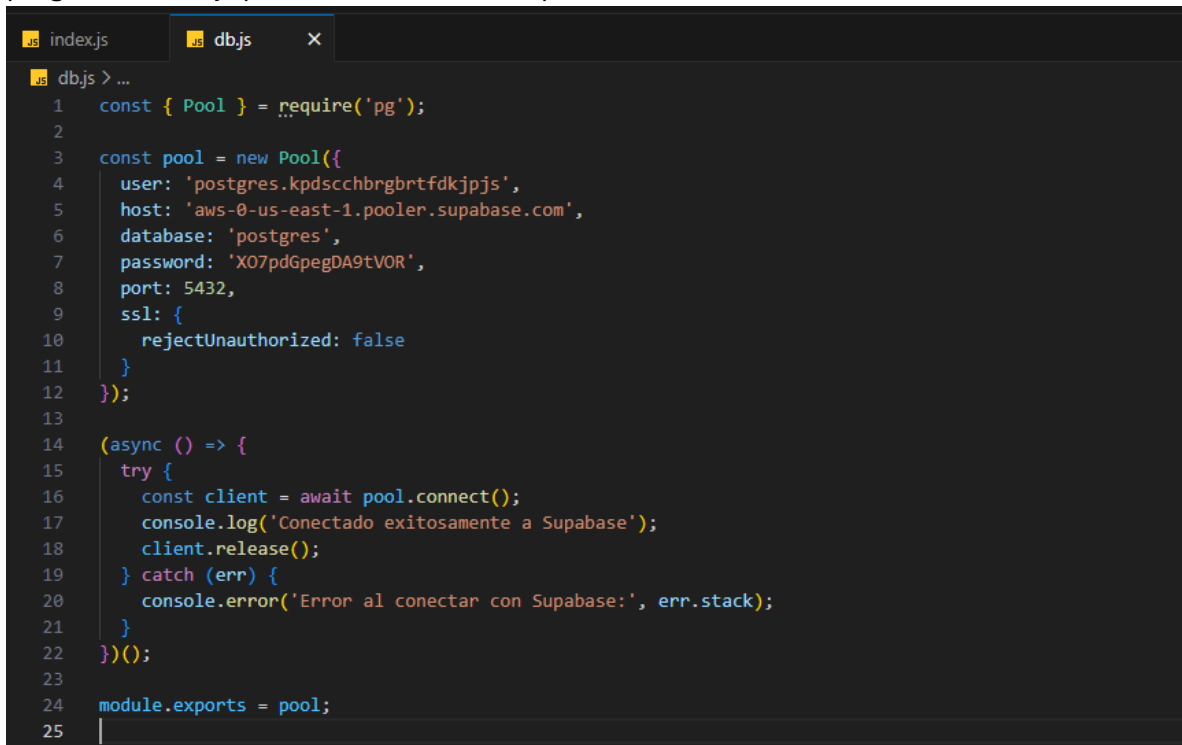
Creación de archivos



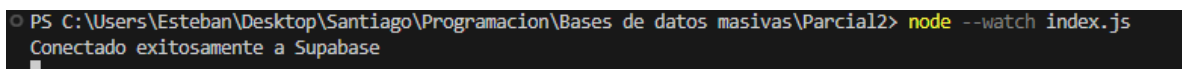
Datos para la conexión



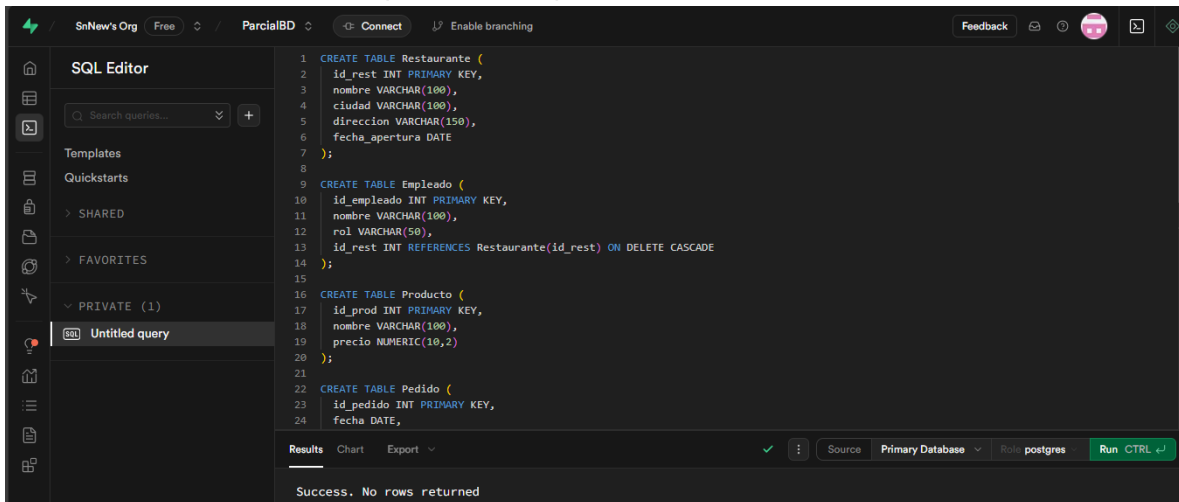
programar el db.js para la conexión con supabase



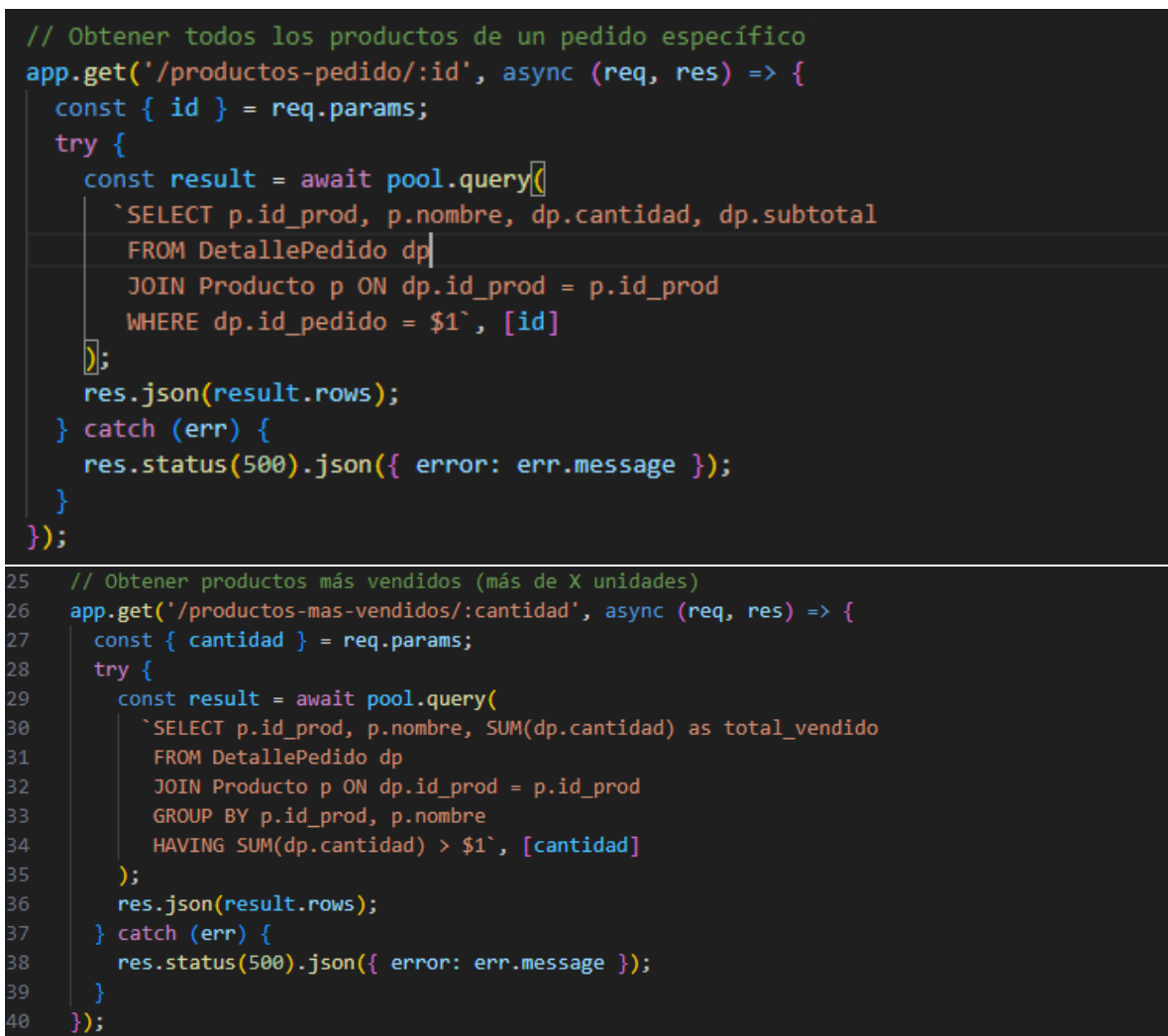
Usamos el **node --watch index.js** para lanzar los servicios y verificar la conexión en la terminal de visual



Creación de las tablas en el sql editor de supabase



Consultas

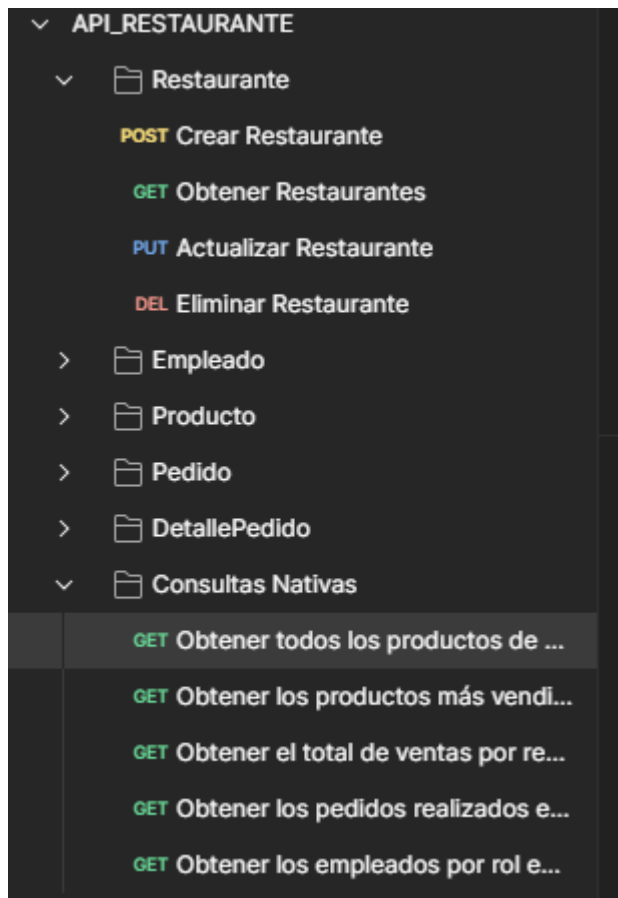


```

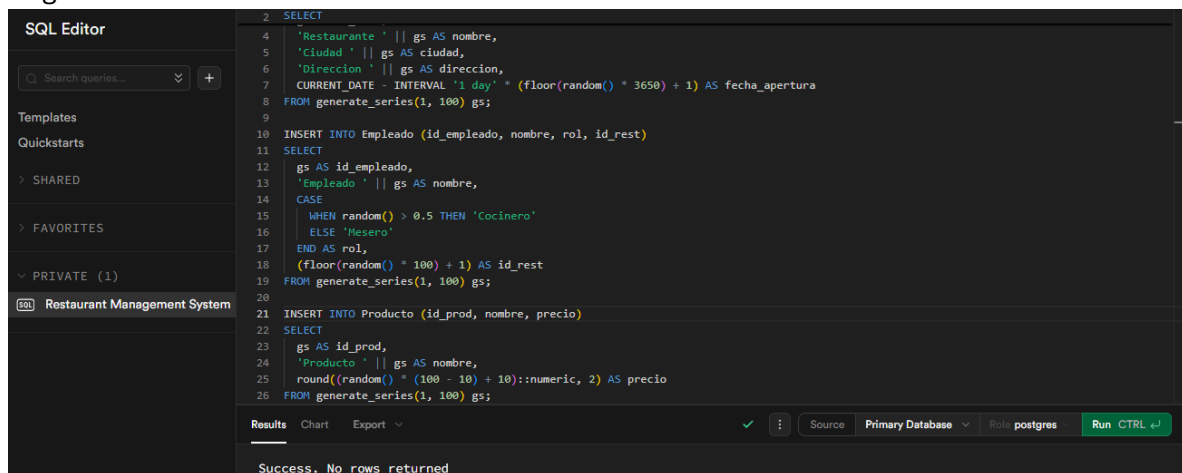
42 // Obtener el total de ventas por restaurante
43 app.get('/ventas-por-restaurante', async (req, res) => {
44   try {
45     const result = await pool.query(
46       `SELECT r.id_rest, r.nombre, SUM(p.total) as total_ventas
47        FROM Pedido p
48        JOIN Restaurante r ON p.id_rest = r.id_rest
49        GROUP BY r.id_rest, r.nombre`
50     );
51     res.json(result.rows);
52   } catch (err) {
53     res.status(500).json({ error: err.message });
54   }
55 });
56
57 // Obtener los pedidos realizados en una fecha específica
58 app.get('/pedidos-por-fecha/:fecha', async (req, res) => {
59   const { fecha } = req.params;
60   try {
61     const result = await pool.query(
62       `SELECT * FROM Pedido WHERE fecha = $1`, [fecha]
63     );
64     res.json(result.rows);
65   } catch (err) {
66     res.status(500).json({ error: err.message });
67   }
68 });
69
70 // Obtener empleados por rol en un restaurante
71 app.get('/empleados-por-rol/:id_rest/:rol', async (req, res) => {
72   const { id_rest, rol } = req.params;
73   try {
74     const result = await pool.query(
75       `SELECT * FROM Empleado WHERE id_rest = $1 AND rol ILIKE $2`, [id_rest, rol]
76     );
77     res.json(result.rows);
78   } catch (err) {
79     res.status(500).json({ error: err.message });
80   }
81 });

```

Coleccion de postman



Registros



Capturas de las consultas

API RESTAURANTE

GET <http://localhost:3000/restaurantes> Send

Params Authorization Headers (7) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results 200 OK · 894 ms · 13.22 KB Save Response

JSON Preview Visualize

```
1 [
2   {
3     "id_rest": 1,
4     "nombre": "Restaurante 1",
5     "ciudad": "Ciudad 1",
6     "direccion": "Direccion 1",
7     "fecha_apertura": "2021-10-15T05:00:00.000Z"
8   },
9   {
10    "id_rest": 2,
11    "nombre": "Restaurante 2",
12    "ciudad": "Ciudad 2",
13    "direccion": "Direccion 2",
14    "fecha_apertura": "2018-11-04T05:00:00.000Z"
15  },
16 ]
```

Obtener todos los productos de un pedido específico

API RESTAURANTE / Consultas Nativas / Obtener todos los productos de un pedido específico

GET <http://localhost:3000/pedidos/1/productos> Send

Params Authorization Headers (7) Body Scripts Settings

Body Cookies Headers (8) Test Results 200 OK · 828 ms · 342 B Save Response

JSON Preview Visualize

```
1 [
2   {
3     "nombre": "Producto 14",
4     "precio": "59.89",
5     "cantidad": 5,
6     "subtotal": "81.98"
7   }
8 ]
```

Obtener los productos más vendidos (más de X unidades)

API RESTAURANTE / Consultas Nativas / Obtener los productos más vendidos (más de X unidades)

GET <http://localhost:3000/productos/mas-vendidos/10> Send

Params Authorization Headers (7) Body Scripts Settings

Query Params

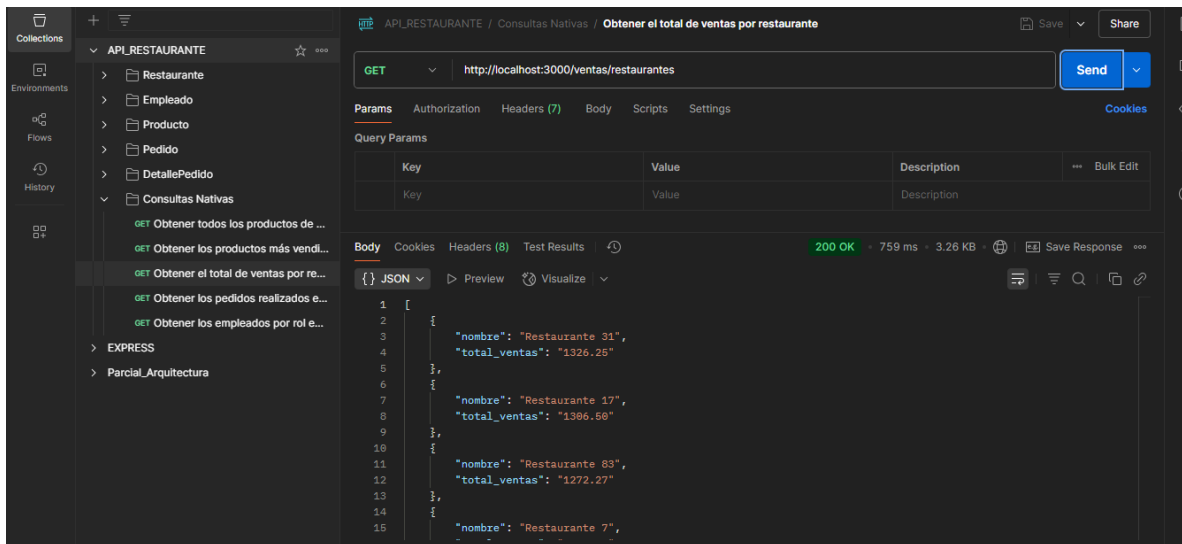
Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results 200 OK · 112 ms · 1.03 KB Save Response

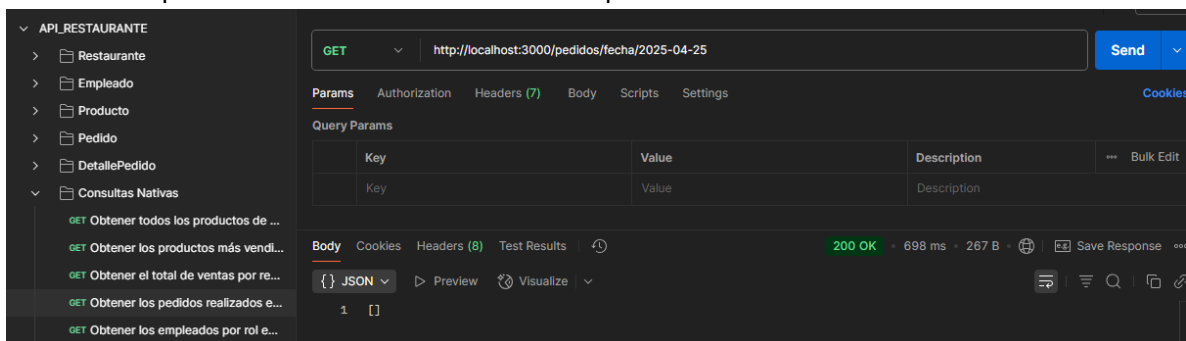
JSON Preview Visualize

```
1 [
2   {
3     "nombre": "Producto 76",
4     "total_vendido": "26"
5   },
6   {
7     "nombre": "Producto 45",
8     "total_vendido": "20"
9   },
10  {
11    "nombre": "Producto 22",
12    "total_vendido": "17"
13  },
14  {
15    "nombre": "Producto 85",
16    "total_vendido": "15"
17  }
18 ]
```

Obtener el total de ventas por restaurante



Obtener los pedidos realizados en una fecha específica



Obtener los empleados por rol en un restaurante

