

**PARCIAL API REST CON NODE.JS – EXPRESS - SUPABASE**

**JORGE ANDRES CASTRO PACHON**

**ID:827833**

**BASES DE DATOS MASIVAS**

**DOCENTE:**

**WILLIAM ALEXANDER MATALLANA**

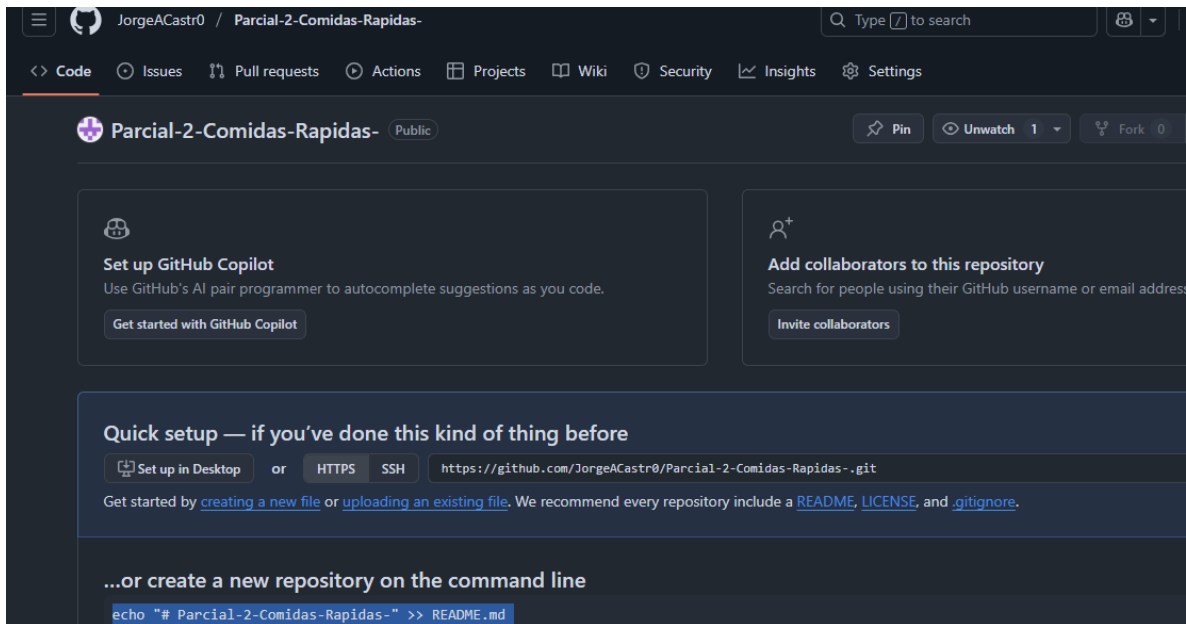
**CORPORACIÓN UNIVERSITARIA MINUTO DE**

**DIOS**

**INGENERIA DE SISTEMAS**

**ZIPQUIRÁ, CUNDINAMARCA 2025**

## 1. Creamos repositorio en github



## 2. Enlazamos la carpeta con GIT

```
SDA-48-239@LAPTOP-IPD2JEJG MINGW64 ~/OneDrive - uniminuto.edu/6to Semestre/Bases de datos Masivas/Parcial 2do Corte
$ echo "# Parcial-2-Comidas-Rapidas-" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JorgeACastro0/Parcial-2-Comidas-Rapidas-.git
git push -u origin main
Initialized empty Git repository in C:/Users/SDA-48-239/OneDrive - uniminuto.edu/6to Semestre/Bases de datos Masivas/Parcial 2do Corte/.git/
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) c349166] first commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 121.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JorgeACastro0/Parcial-2-Comidas-Rapidas-.git
 * [new branch]      main -> main
```

## 3. Creamos archivos e instalamos dependencias

```
PS C:\Users\SDA-48-239\OneDrive - uniminuto.edu\6to Semestre\Bases de datos Masivas\Parcial 2do Corte> npm install express postgres dotenv cors
added 84 packages in 12s

15 packages are looking for funding
run 'npm fund' for details
```

```

PS C:\Users\SDA-48-239\OneDrive - uniminuto.edu\6to Semestre\Bases de datos Masivas\Parcial 2do Corte> npm init -y
Wrote to C:\Users\SDA-48-239\OneDrive - uniminuto.edu\6to Semestre\Bases de datos Masivas\Parcial 2do Corte\package.json:

{
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^16.5.0",
    "express": "^5.1.0",
    "postgre": "^0.1.8"
  },
  "name": "parcial-2do-corte",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "devDependencies": {},
  "scripts": {

```

#### 4. Creamos el proyecto en supabase y enlazamos la base de datos

**Session pooler** Shared Pooler

Only recommended as an alternative to Direct Connection, when connecting via an IPv4 network.

**IPv4 compatible**  
Session pooler connections are IPv4 proxied for free

**Only use on a IPv4 network**  
Use Direct Connection if connecting via an IPv6 network

postgresql://postgres.fcjvzrjjccscmipthavq:[YOUR-PASSWORD]@

View parameters

host: aws-0-sa-east-1.pooler.supabase.com  
port: 5432  
database: postgres  
user: postgres.fcjvzrjjccscmipthavq  
pool\_mode: session

For security reasons, your database password is never shown.

```

Welcome  JS dbjs  JS index.js IM
JS dbjs > default
You, 2 weeks ago | 1 author (You)
1 import postgres from 'postgres';
2
3 const sql = postgres('postgresql://postgres.fcjvzrjjccscmipthavq:cE$t8$5*v&-mg!w@aws-0-sa-east-1.pooler.supabase.com:5432/p
4
5
6 export default sql; You, 2 weeks ago • Premier commit del quiz ...

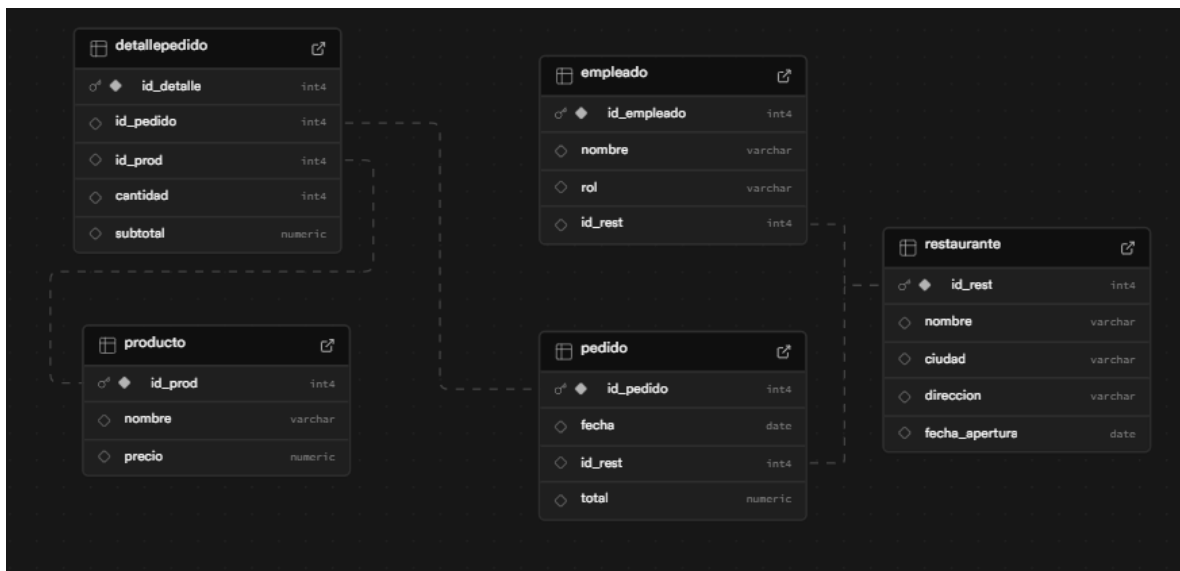
```

5. Ahora crearemos las tablas de la base de datos según la indicación

```
1 CREATE TABLE Restaurante (  
2     id_rest SERIAL PRIMARY KEY,  
3     nombre VARCHAR(100),  
4     ciudad VARCHAR(100),  
5     direccion VARCHAR(150),  
6     fecha_apertura DATE  
7 );  
8  
9 CREATE TABLE Producto (  
10    id_prod SERIAL PRIMARY KEY,  
11    nombre VARCHAR(100),  
12    precio NUMERIC(10,2)  
13 );  
14  
15 CREATE TABLE Empleado (  
16    id_empleado SERIAL PRIMARY KEY,  
17    nombre VARCHAR(100),  
18    rol VARCHAR(50),  
19    id_rest INT REFERENCES Restaurante(id_rest)  
20 );  
21
```

```
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    fecha DATE,  
    id_rest INT REFERENCES Restaurante(id_rest),  
    total NUMERIC(10,2)  
);  
  
CREATE TABLE DetallePedido (  
    id_detalle SERIAL PRIMARY KEY,  
    id_pedido INT REFERENCES Pedido(id_pedido),  
    id_prod INT REFERENCES Producto(id_prod),  
    cantidad INT,  
    subtotal NUMERIC(10,2)  
);
```

Diagrama ER:



6. Ahora insertamos registros en la base de datos

Tabla restaurante:

```
1 INSERT INTO Restaurante (nombre, ciudad, direccion, fecha_apertura) VALUES
2 ('Delicias Rápidas', 'Bogotá', 'Calle 123 #45-67', '2020-01-15'),
3 ('Sabor Criollo', 'Medellín', 'Carrera 10 #20-30', '2021-03-12'),
4 ('Burgertown', 'Cali', 'Avenida 3N #45-12', '2019-06-20'),
5 ('Comida Express', 'Barranquilla', 'Calle 30 #15-25', '2022-02-01'),
6 ('La Parrilla', 'Cartagena', 'Cra 7 #54-88', '2018-11-05'),
7 ('Arepas Ya', 'Manizales', 'Calle 50 #22-13', '2020-08-17'),
8 ('Tostones Grill', 'Pereira', 'Cra 4 #32-20', '2021-09-25'),
9 ('Papas y Algo Más', 'Bucaramanga', 'Av. Santander #19-20', '2019-03-30'),
10 ('Comida Urbana', 'Cúcuta', 'Cra 9 #11-11', '2022-07-10'),
11 ('Rico al Paso', 'Ibagué', 'Calle 21 #14-17', '2023-01-05');
12
```

Tabla Empleado:

```
1 INSERT INTO Empleado (nombre, rol, id_rest) VALUES
2 -- Restaurante 1
3 ('Carlos Pérez', 'Cocinero', 1),
4 ('Ana Gómez', 'Cajero', 1),
5 ('Luis Rojas', 'Mesero', 1),
6 ('Paola Ruiz', 'Administrador', 1),
7 ('David Torres', 'Domiciliario', 1),
8 -- Restaurante 2
9 ('Camila Rodríguez', 'Cocinero', 2),
10 ('Juan Esteban', 'Cajero', 2),
11 ('Marta Jiménez', 'Mesero', 2),
12 ('Diego Morales', 'Administrador', 2),
13 ('Lina Campos', 'Domiciliario', 2),
14 -- Restaurante 3
15 ('José Herrera', 'Cocinero', 3),
16 ('Claudia Arias', 'Cajero', 3),
17 ('Ricardo Lozano', 'Mesero', 3),
18 ('Valentina Rico', 'Administrador', 3),
19 ('Sebastián Quintero', 'Domiciliario', 3),
20 -- Restaurante 4
21 ('Sandra Beltrán', 'Cocinero', 4),
22 ('Julian Ortiz', 'Cajero', 4),
23 ('Adriana Luna', 'Mesero', 4),
24 ('Andrés Suárez', 'Administrador', 4),
25 ('Diana Cortés', 'Domiciliario', 4),
26 -- Restaurante 5
27 ('Nicolás Vargas', 'Cocinero', 5),
28 ('María Cárdenas', 'Cajero', 5),
29 ('Fernando Salas', 'Mesero', 5),
30 ('Isabel Naranjo', 'Administrador', 5),
31 ('Sebastián Pineda', 'Domiciliario', 5);
```

Tabla productos:

```
1 INSERT INTO Producto (nombre, precio) VALUES
2 ('Hamburguesa Clásica', 12000),
3 ('Hamburguesa Doble', 15000),
4 ('Hamburguesa BBQ', 14500),
5 ('Perro Caliente', 9000),
6 ('Salchipapa', 10000),
7 ('Arepa Rellena', 8000),
8 ('Tacos Mixtos', 13000),
9 ('Chorizo Santarrosano', 7000),
10 ('Papas Criollas', 5000),
11 ('Nachos con Queso', 9500),
12 ('Choripan', 10500),
13 ('Empanadas x3', 6000),
14 ('Alitas BBQ x5', 11000),
15 ('Tostones', 7500),
16 ('Arepa con Chorizo', 8500),
17 ('Bebida Gaseosa 350ml', 3000),
18 ('Jugos Naturales', 4000),
19 ('Agua en Botella', 2500),
20 ('Cerveza Nacional', 5000),
21 ('Cerveza Importada', 8000),
22 ('Postre de Tres Leches', 6500),
23 ('Brownie con Helado', 7000),
24 ('Helado en Copa', 5000),
25 ('Pizza Personal', 12000),
26 ('Pizza Mediana', 18000),
27 ('Pizza Grande', 25000),
28 ('Combo Hamburguesa', 18000),
29 ('Combo Perro Caliente', 15000),
```

Tabla pedidos:

```
1 INSERT INTO Pedido (fecha, id_rest, total) VALUES
2 ('2024-04-01', 1, 45000),
3 ('2024-04-01', 2, 32000),
4 ('2024-04-02', 3, 25000),
5 ('2024-04-02', 4, 54000),
6 ('2024-04-03', 5, 39000),
7 ('2024-04-03', 6, 46000),
8 ('2024-04-04', 7, 31000),
9 ('2024-04-04', 8, 52000),
10 ('2024-04-05', 9, 48000),
11 ('2024-04-05', 10, 36000),
12 ('2024-04-06', 1, 37000),
13 ('2024-04-06', 2, 41000),
14 ('2024-04-07', 3, 29000),
15 ('2024-04-07', 4, 47000),
16 ('2024-04-08', 5, 50000),
17 ('2024-04-08', 6, 28000),
18 ('2024-04-09', 7, 33000),
19 ('2024-04-09', 8, 31000),
20 ('2024-04-10', 9, 49000),
21 ('2024-04-10', 10, 35000),
22 ('2024-04-11', 1, 31000),
23 ('2024-04-11', 2, 27000),
24 ('2024-04-12', 3, 51000),
25 ('2024-04-12', 4, 34000),
26 ('2024-04-13', 5, 46000),
27 ('2024-04-13', 6, 42000),
28 ('2024-04-14', 7, 39000),
29 ('2024-04-14', 8, 37000),
30 ('2024-04-15', 9, 53000),
31 ('2024-04-15', 10, 47000)
```

Tabla detallepedido:

```
1 INSERT INTO DetallePedido (id_pedido, id_prod, cantidad, subtotal) VALUES
2 (1, 1, 2, 24000),
3 (2, 5, 1, 10000),
4 (3, 12, 3, 18000),
5 (4, 3, 2, 29000),
6 (5, 20, 4, 20000),
7 (6, 7, 2, 26000),
8 (7, 8, 3, 21000),
9 (8, 15, 2, 17000),
10 (9, 23, 1, 5000),
11 (10, 27, 2, 36000),
12 (11, 2, 2, 30000),
13 (12, 30, 3, 27000),
14 (13, 11, 2, 21000),
15 (14, 13, 3, 33000),
16 (15, 18, 2, 5000),
17 (16, 25, 1, 25000),
18 (17, 6, 2, 16000),
19 (18, 4, 2, 18000),
20 (19, 10, 2, 19000),
21 (20, 9, 2, 10000),
22 (21, 21, 2, 13000),
23 (22, 16, 2, 6000),
24 (23, 22, 1, 5000),
25 (24, 24, 2, 24000),
26 (25, 26, 1, 25000),
27 (26, 17, 2, 8000)
```

7. Ahora configuraremos el archivo index.js para que escuche por el puerto 3000 y traiga las demás dependencias

```

JS index.js > ...
1  import express from 'express';
2  import sql from './db.js';
3
4  const app = express();
5
6  app.use(express.json());
7
8  app.use(express.urlencoded({extended:true}));
9
10 app.get('/api/prueba' , (req, res) => {
11
12     res.send('LA API FUNCIONA');
13 });
14
15 //Crear puerto de conexion del servidor
16 const PORT = 3000;
17
18 //La conexion la va a escuchar por el puerto 3000 y si
19 app.listen(PORT, ()=>{
20     console.log('El servidor esta corriendo');
21
22 });

```

```

(Use node -trace-warnings if
El servidor esta corriendo

```

8. Ahora creamos las APIS para hacer el CRUD

Aquí creamos insertamos los restaurantes

```

app.post('/api/nuevorestaurante', async (req, res) => {
    try {
        const { nombre, ciudad, direccion, fecha_apertura } = req.body;
        const result = await sql`
            INSERT INTO Restaurante (nombre, ciudad, direccion, fecha_apertura)
            VALUES (${nombre}, ${ciudad}, ${direccion}, ${fecha_apertura})
            RETURNING *;
        `;
        res.status(201).json({ success: true, data: result });
    } catch (error) {
        res.status(500).json({ success: false, message: 'Error al crear restaurante', details: error.message });
    }
});

```