QUIZ

Brayan Y. Calderón Tirado

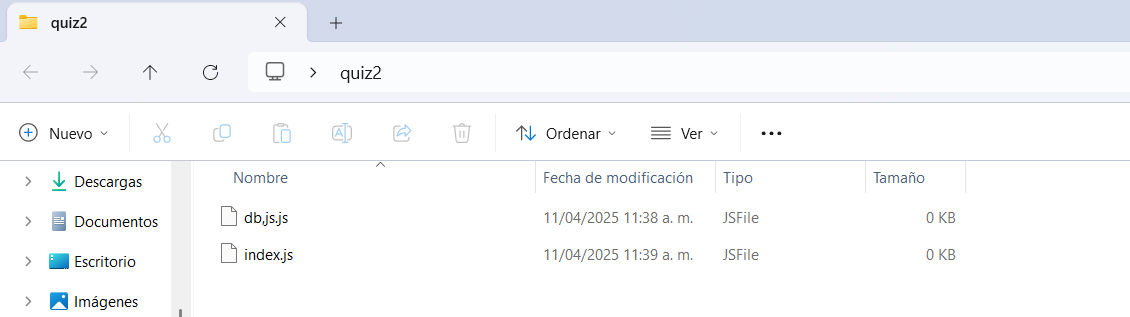
Ingeniería de sistemas, Corporación universitaria minuto Dios

NRC-10-60747, Bases de Datos Masivas

WILLIAM A. MATALLANA PORRAS

11/04/2025

1. Creamos una carpeta donde vamos a guardar nuestro proyecto he incluiremos los archivos index.js y db.js



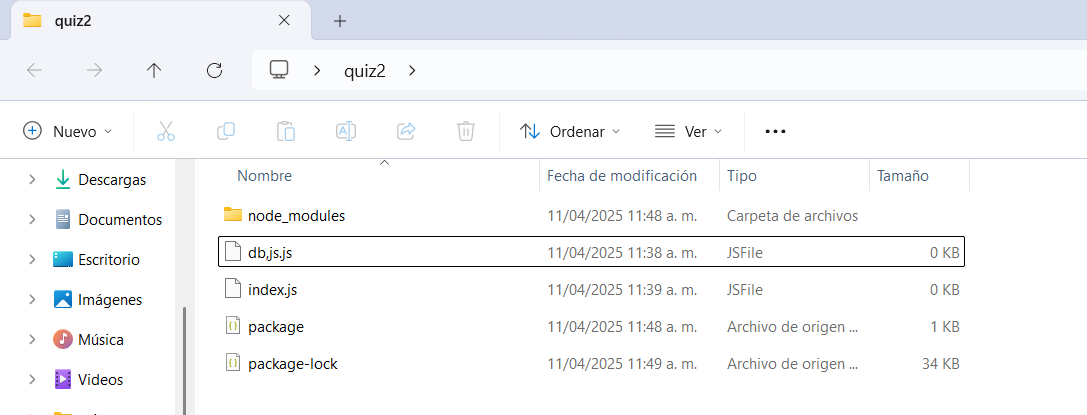
1. Abrimos nuestra carpeta en visual estudio, abrimos una terminal y empezamos a descargar las dependencias de Node.js

* npm init -y
* npm install express cors pg
* npm install pg

El comando npm init -yse utiliza para iniciar un proyecto en Node.js de forma rápida. Al ejecutarlo, se crea automáticamente un archivo llamado package.json, el cual contiene información básica del proyecto como el nombre, la versión y las dependencias. La opción -ypermite aceptar todas las configuraciones por defecto sin necesidad de ir respondiendo preguntas.

Por otro lado, el comando npm install express cors pgsirve para instalar tres dependencias esenciales en el desarrollo de aplicaciones web. La primera es expressun framework ligero que facilita la creación de servidores y rutas en Node.js. La segunda es cors, un middleware que permite habilitar o restringir el acceso a recursos desde diferentes dominios, lo cual es muy útil cuando se trabaja con frontend y backend por separado. La tercera es pg, un cliente que permite establecer conexiones con bases de datos PostgreSQL y ejecutar consultas desde Node.js.

Finalmente, el comando npm install pgpermite instalar únicamente la dependencia pg. Este comando es útil si ya tienes configurado tu proyecto con otras librerías y ahora necesitas agregar soporte para conectarte a una base de datos PostgreSQL.

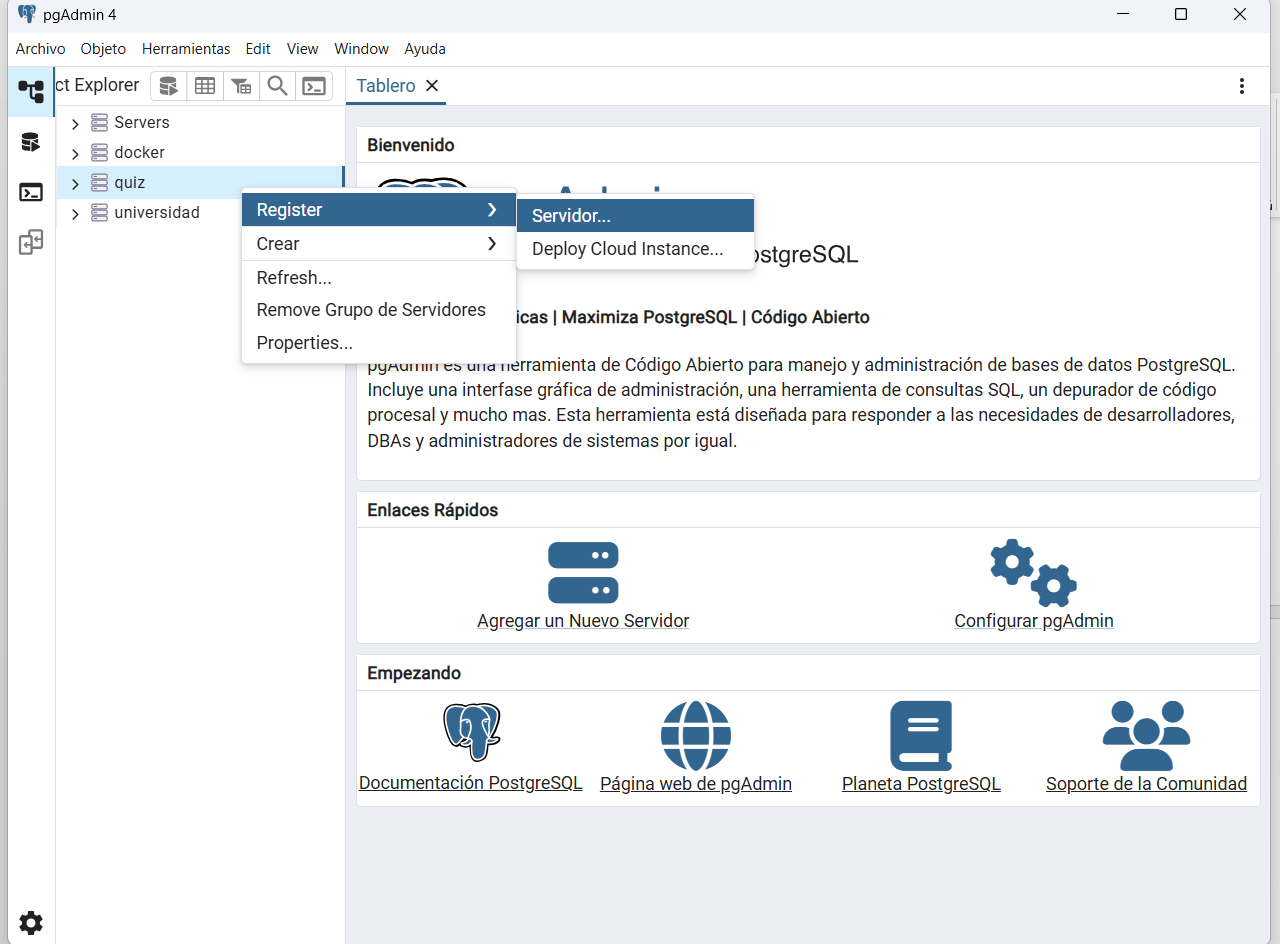


1. procedemos a realizar la conexión de pgadmin con supabase, nos dirigimos al cliente y con clic derecho sobre servers seleccionamos crear grupo de servidores y asignamos un nombre.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Realizamos el mismo procedimiento al nuevo servidor que creamos que se llama quiz y clicamos en la opción crear servidor y le asignamos un nombre.



Le asignamos un nombre y luego procedemos a realizar la conexión con supabase

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

En esta nueva ventana pondremos las credenciales que nos brinda supabase

Interfaz de usuario gráfica, Aplicación

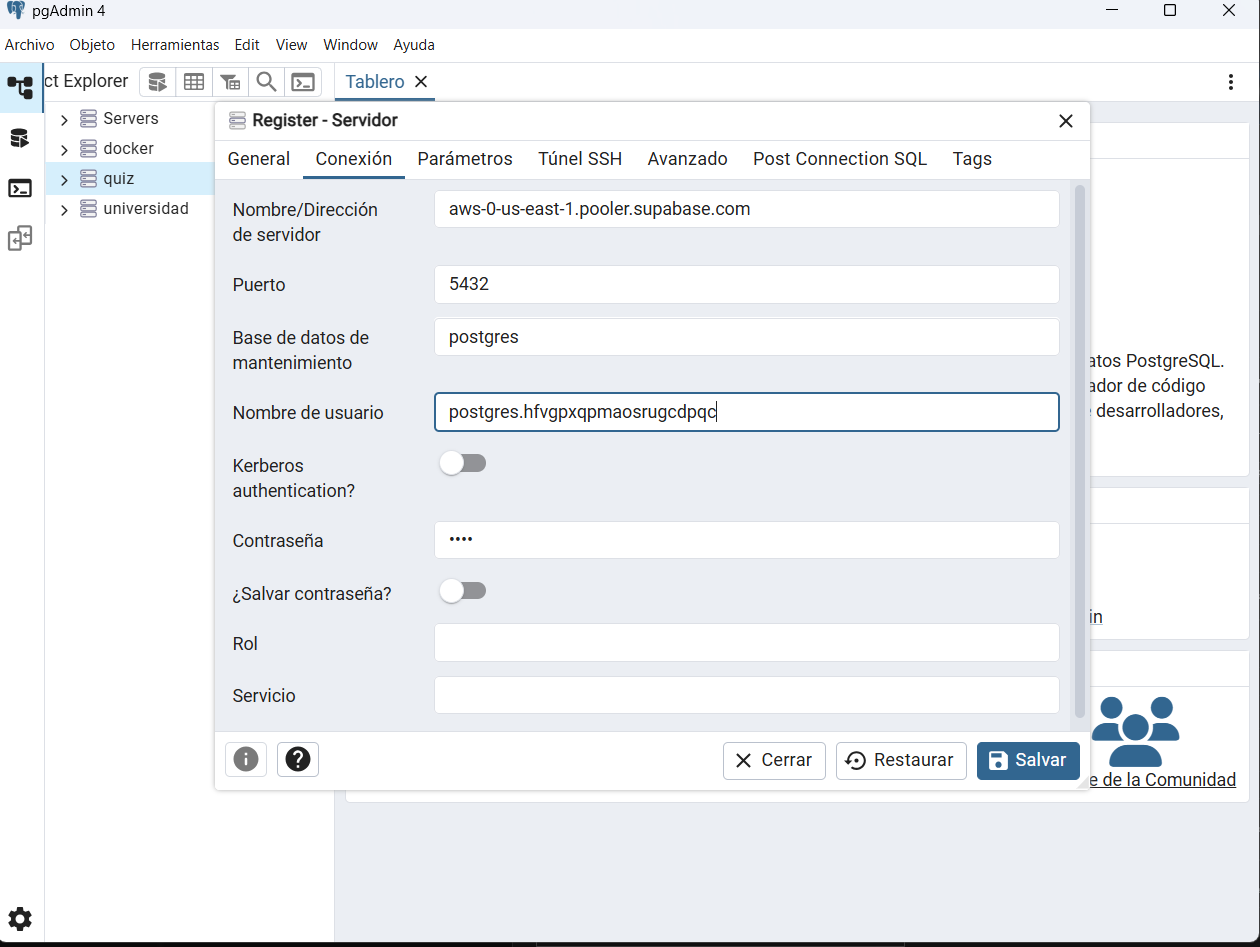
El contenido generado por IA puede ser incorrecto.

Nos dirigimos a supabase, creamos la conexión luego de ingresar en la parte superior nos encontramos con la opción conectar y veremos las credenciales necesarias para realizar dicha conexión.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

En el ultimo recuadro encontramos las credenciales que debemos poner en pgadmin que es el cliente.

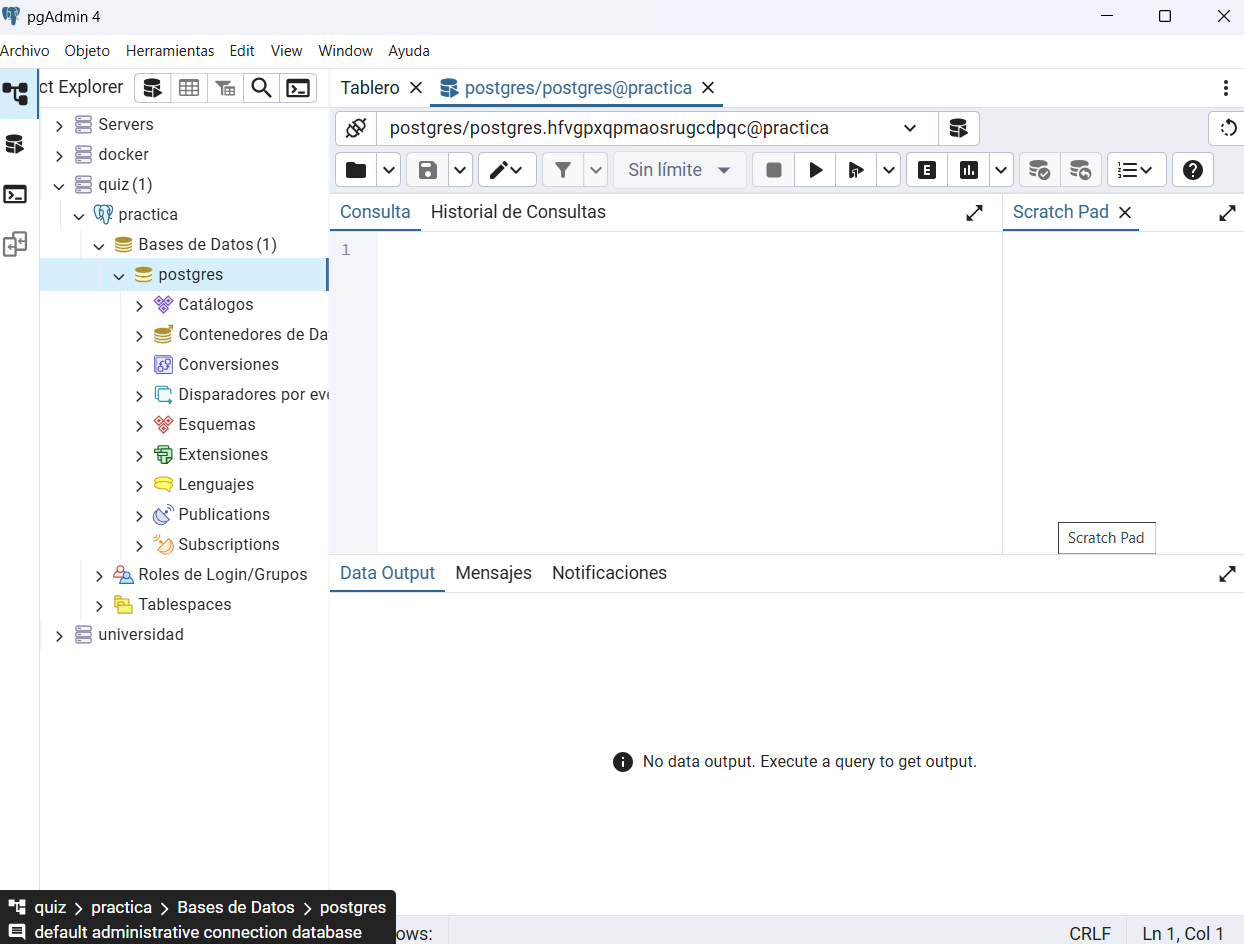


Ya realizada la conexión nos debería aparecer de la siguiente manera

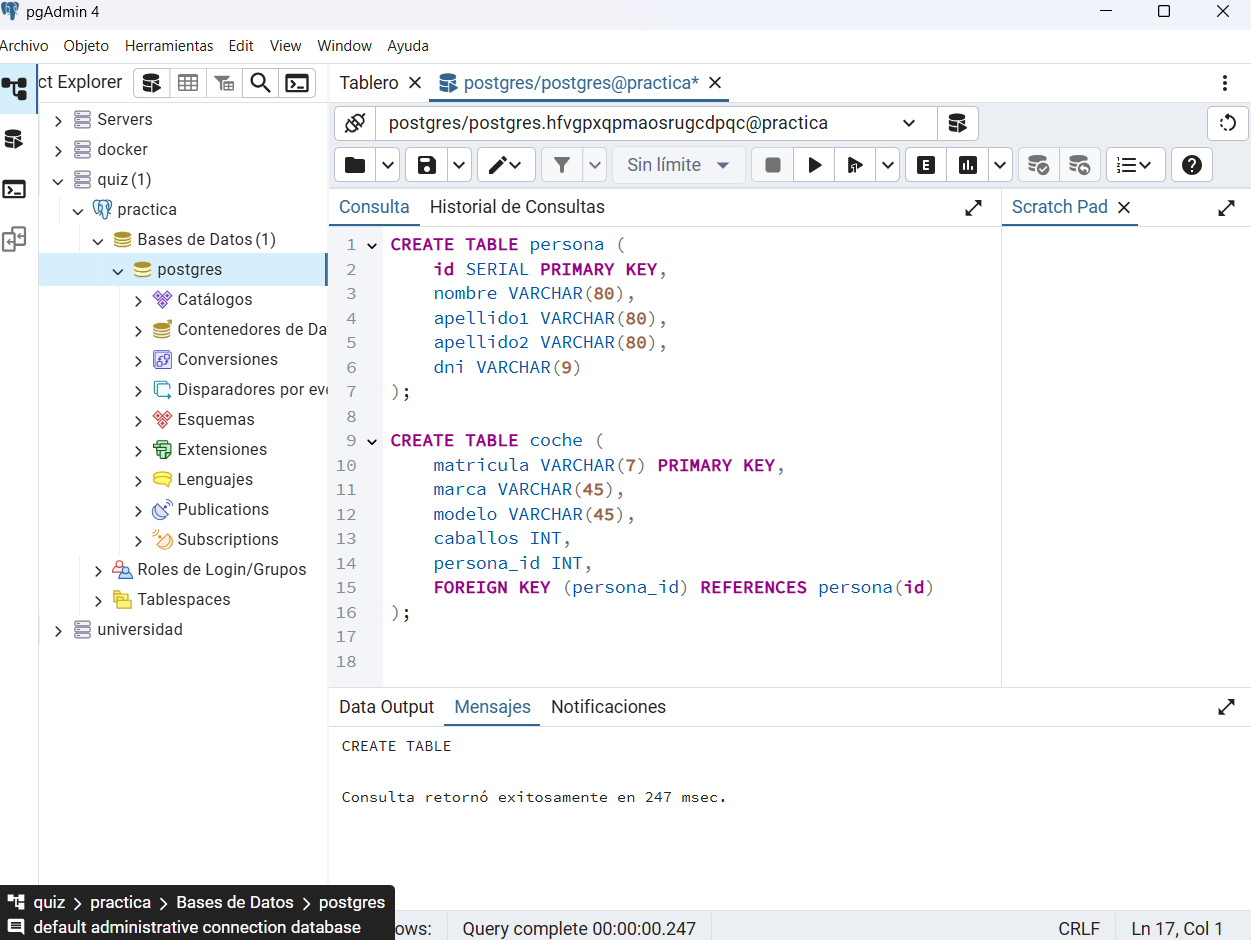
Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

Desplegamos el menú practica y clicamos en bases de datos, clicamos en postgres y presionamos clic derecho y bajamos a la opción que dice herramientas de consulta para realizar la creación de las respectivas tablas y registros.



Realizaremos la creación de las tablas correspondientes



Vamos a realizar los 100 registros de cada una de las tablas

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

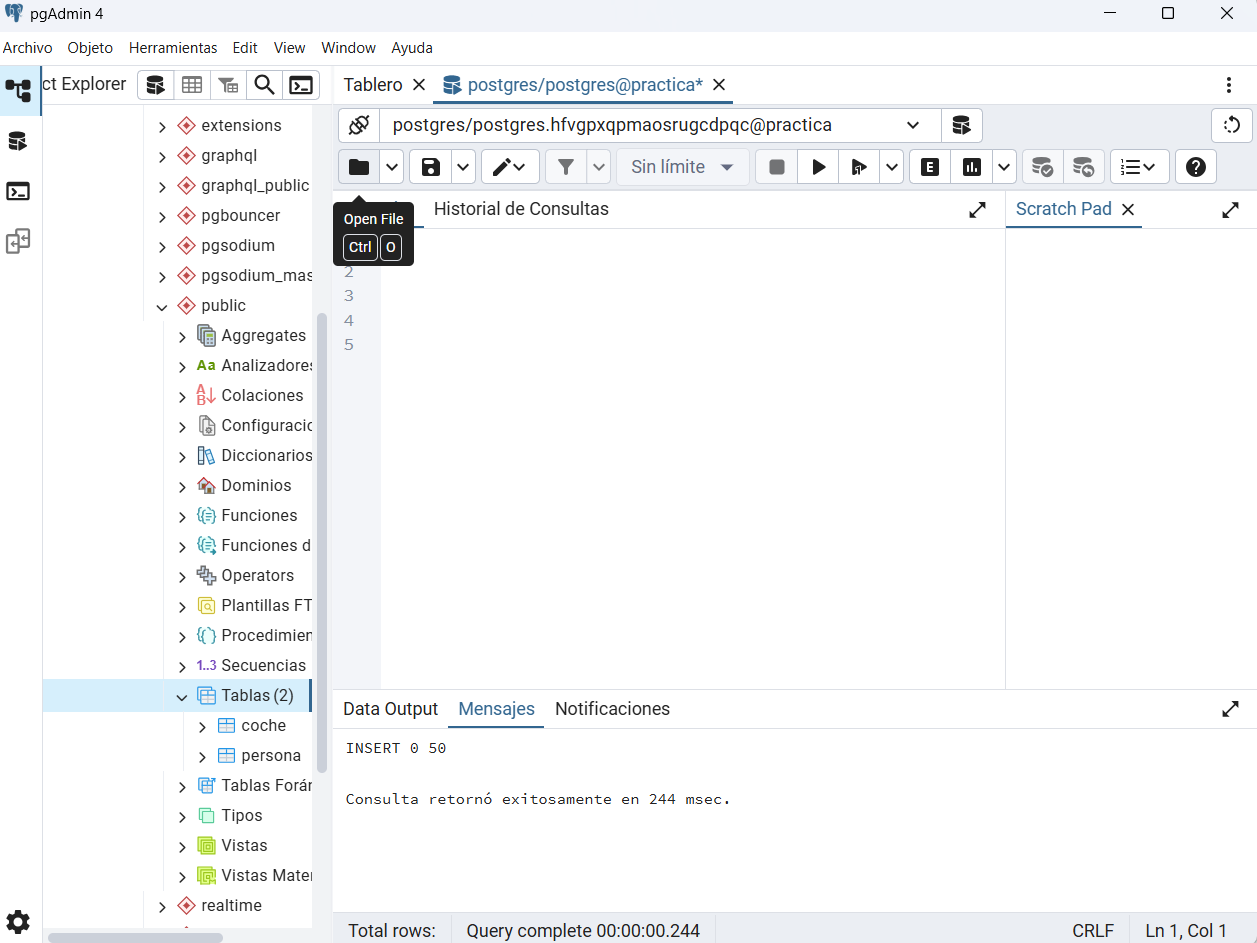
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Con las tablas y los registros ya creados, nos dirigimos a la opcion esquemas, public y luego a tablas y podemos verificar que se crearon las tablas, en la misma opcion en donde se ingresaron los datos podemos realizar un SELECT \* FROME y el nombre de las tablas para verificar si se crearon los datos de manera correcta.



Tabla

El contenido generado por IA puede ser incorrecto.

1. creación de las apis

* El archivo db.jsse encarga de establecer la conexión entre nuestra aplicación Node.js y la base de datos PostgreSQL alojada en Supabase. Utiliza el paquete pg(PostgreSQL) y crea un "pool de conexiones" con los datos necesarios como el usuario, host, contraseña, puerto y nombre de la base de datos. Además, incluye una verificación inicial para confirmar que la conexión fue exitosa o mostrar un error si algo falla.
* Por otro lado, el archivo index.jses el núcleo de la API creada con Express. Defina rutas para realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre tres tablas: estudiantes, cursos e inscripciones. También incluye una ruta para ver los detalles combinados de las inscripciones usando un JOIN. Usa middleware como corsy express.json() para manejar solicitudes HTTP de forma correcta, y finalmente, arranca el servidor en el puerto 3000 mostrando un mensaje cuando todo está listo.

1. Con el comando node --watch index.jspodemos ejecutar nuestra aplicación y hacer que se reinicie automáticamente cada vez que detecte cambios en el archivo index.js (o archivos relacionados). Esto es útil durante el desarrollo, ya que no tenemos que detener y volver a iniciar el servidor manualmente cada vez que modificamos el código.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.