**Tecnológico Nacional de México**

**Instituto Tecnológico De Tláhuac**

**Desarrollo Backend**

**Docente: Ing. Padilla Juárez Jaqueline**

**Practica – U3:**

**Documentación**

**Alumnos:**

**Díaz Vera Lisbeth Cristina**

**Ortega Rico Kevin**

**Badillo Martínez Carlos**

**Ortiz Elizalde Cesar Emanuel**

**Grupo: 8S1**

**Agosto – Diciembre**

Índice.

[**Introducción.** 4](#_Toc210070392)

[**¿Qué es Supbase?** 5](#_Toc210070393)

[**Características** 5](#_Toc210070394)

[**Ventajas** 5](#_Toc210070395)

[**Desventajas** 6](#_Toc210070396)

[**Instalación.** 6](#_Toc210070397)

[**¿En qué está basado?** 12](#_Toc210070398)

[**Modelado de datos.** 13](#_Toc210070399)

[**Entidades principales** 13](#_Toc210070400)

[**Relaciones entre entidades** 15](#_Toc210070401)

[**Modelo Físico** 15](#_Toc210070402)

[**Conclusión.** 19](#_Toc210070403)

**Figuras.**

[Ilustración 1 "Página de supbase" 6](#_Toc210070688)

[Ilustración 2 "Proyecto creado" 7](#_Toc210070689)

[Ilustración 3 "Creación de una bdd" 7](#_Toc210070690)

[Ilustración 4 "Información general de la BDD" 8](#_Toc210070691)

[Ilustración 5 "Desglose de supabase" 8](#_Toc210070692)

[Ilustración 6 "Visión de Table Editor" 9](#_Toc210070693)

[Ilustración 7 "Información de Tabla admins" 9](#_Toc210070694)

[Ilustración 8 "Tipos de enlaces" 10](#_Toc210070695)

[Ilustración 9 "Conexión de la Bdd" 10](#_Toc210070696)

[Ilustración 10 "Consulta a usuarios" 11](#_Toc210070697)

[Ilustración 11 "Consulta a products" 11](#_Toc210070698)

[Ilustración 12 "Visualización a la versión del gestor" 12](#_Toc210070699)

[Ilustración 13 "Diagrama Entidad–Relación" 14](#_Toc210070700)

[Ilustración 14 "comando SELECT \* FROM products" 15](#_Toc210070701)

[Ilustración 15 "Registros almacenados 15](#_Toc210070702)

[Ilustración 16 "consulta SELECT \* FROM usuarios" 16](#_Toc210070703)

[Ilustración 17 "Registros almacenados de usuarios" 16](#_Toc210070704)

[Ilustración 18 "Registros almacenados de products" 17](#_Toc210070705)

[Ilustración 19 "consulta SELECT \* FROM Products" 17](#_Toc210070706)

[Ilustración 20 "Inserción de un registro en la tabla usuarios” 17](#_Toc210070707)

[Ilustración 21 "Archivo de configuración " 18](#_Toc210070708)

[Ilustración 22 "Ejecutando el proyecto con Vite" 18](#_Toc210070709)

[Ilustración 23 "Repositorio en GitHub" 19](#_Toc210070710)

# **Introducción.**

El presente documento describe el diseño y la implementación de la base de datos de un sistema de gestión de tienda en línea, cuya finalidad es administrar de manera eficiente la información de usuarios, productos, carritos de compra, pagos y facturación. Se presenta un enfoque integral que incluye tanto el modelo conceptual, mediante un Diagrama Entidad–Relación (DER) que identifica las entidades principales y sus relaciones, como el modelo físico, reflejado en consultas SQL que permiten acceder directamente a los datos almacenados en PostgreSQL a través de Supabase.

En el documento se detallan las entidades principales del sistema Usuarios, Productos, Carrito de Compras, Pagos y Facturas, sus atributos y cómo se interrelacionan para garantizar la correcta gestión de información y operaciones de compra. Además, se explican las relaciones entre entidades, incluyendo casos de uno a muchos y muchos a muchos, lo que permite un control efectivo de inventario, la administración de transacciones, el seguimiento de pagos y la generación de comprobantes de compra.

Finalmente, se muestran ejemplos de consultas SQL sobre las tablas de Usuarios y Productos, evidenciando la implementación práctica del modelo físico y la forma en que los datos son almacenados y manipulados dentro del motor de base de datos, asegurando la integridad y consistencia de la información en el sistema.

## **¿Qué es Supabase?**

Es una plataforma que permite a los desarrolladores crear backends sin necesidad de configurarlos desde cero. Funciona sobre PostgreSQL y ofrece funcionalidades modernas como APIs automáticas, autenticación de usuarios, almacenamiento de archivos y suscripciones en tiempo real.

## **Características**

1. **Base de datos**: PostgreSQL gestionado con acceso completo a SQL.
2. **API instantánea**: Genera automáticamente una API RESTful y GraphQL al crear tablas.
3. **Autenticación**: Manejo de usuarios, registros, inicio de sesión y proveedores externos (Google, GitHub, etc.).
4. **Almacenamiento**: Permite guardar y gestionar archivos (imágenes, documentos, videos, etc.).
5. **Tiempo real**: Suscripciones para escuchar cambios en la base de datos en vivo.
6. **Dashboard amigable**: Panel para administrar la base de datos y los servicios sin necesidad de mucho código.
7. **Código abierto**: Transparente, extensible y sin depender de un proveedor propietario.

## **Ventajas**

* Es de código abierto y está basado en PostgreSQL.
* Genera APIs automáticas al crear tablas.
* Permite escalar desde proyectos pequeños hasta grandes.
* Ofrece autenticación integrada con soporte para proveedores sociales.
* Soporta tiempo real para escuchar cambios en la base de datos.
* Panel de control amigable y fácil de usar.
* Tiene una curva de aprendizaje más sencilla que montar un backend desde cero.
* Cuenta con comunidad activa y buena documentación.

## **Desventajas**

* Es una plataforma relativamente nueva frente a Firebase u otros servicios.
* Algunas funciones avanzadas aún están en desarrollo.
* El rendimiento en proyectos muy grandes puede requerir optimización adicional.
* Si usas su hosting, hay dependencia de su infraestructura.
* Tiene menos plugins y ecosistema en comparación con Firebase.

# **Instalación.**

Como primer paso nos tenemos que loguear a la pagina de Supabase: <https://supabase.com>

Iniciar sesión con GitHub.

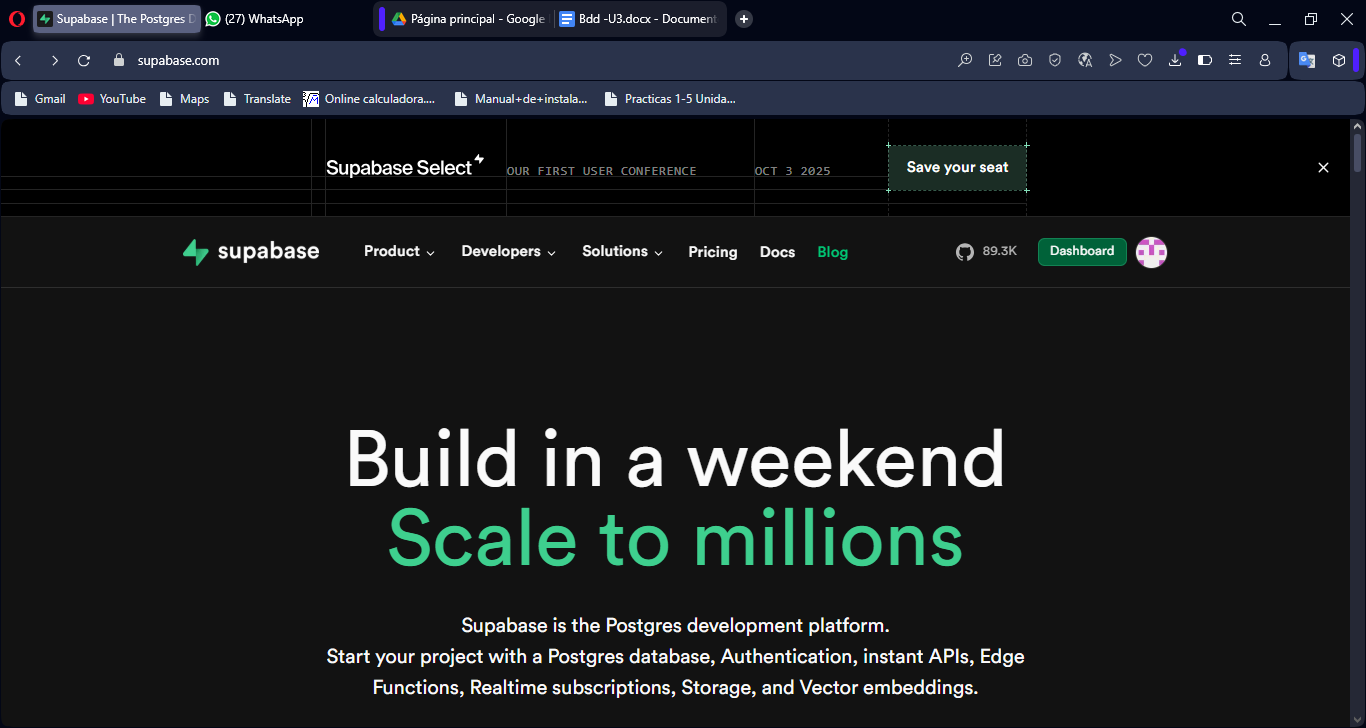


Ilustración 1 "Página de supbase"

Ya iniciada damos clic en Dashboard y nos abrira el menu de Supabase

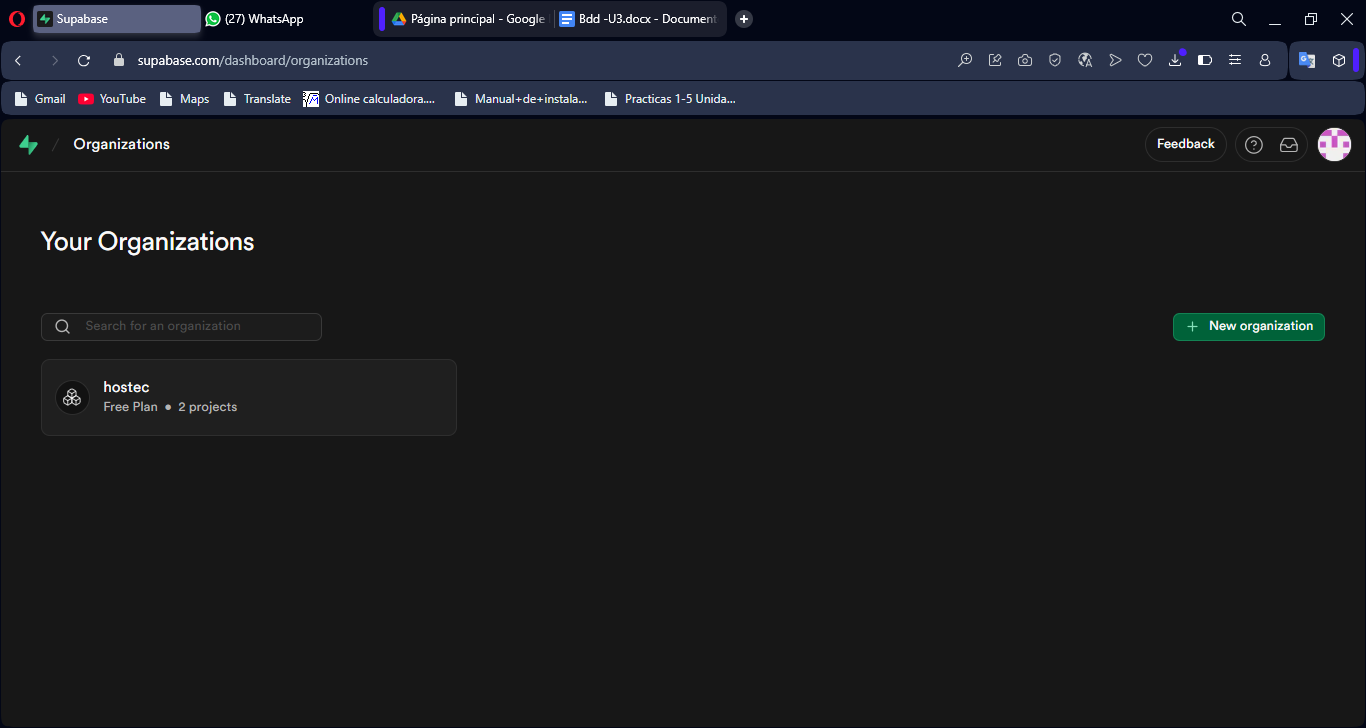


Ilustración 2 "Proyecto creado"

En este caso ya está el proyecto creado. La versión gratis solo permite crear dos proyectos.

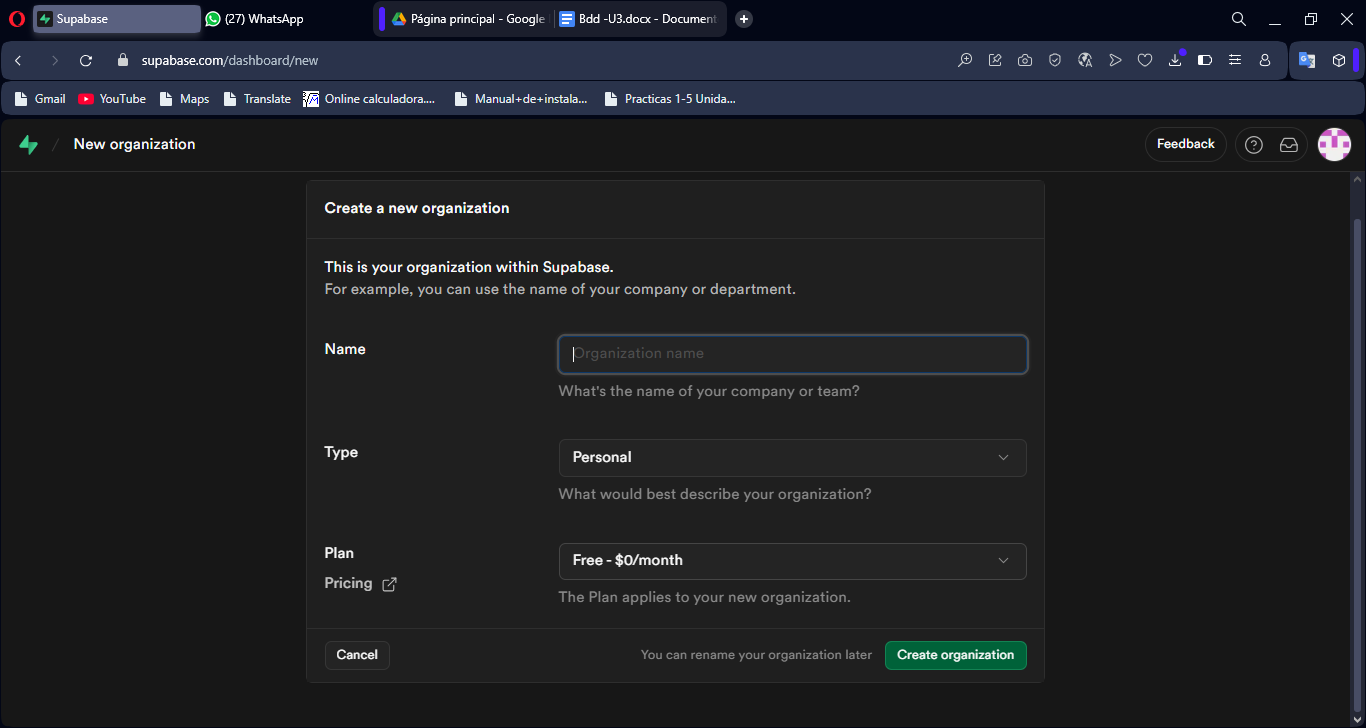
Así se ve la creación de la base de datos:  


Ilustración 3 "Creación de una bdd"

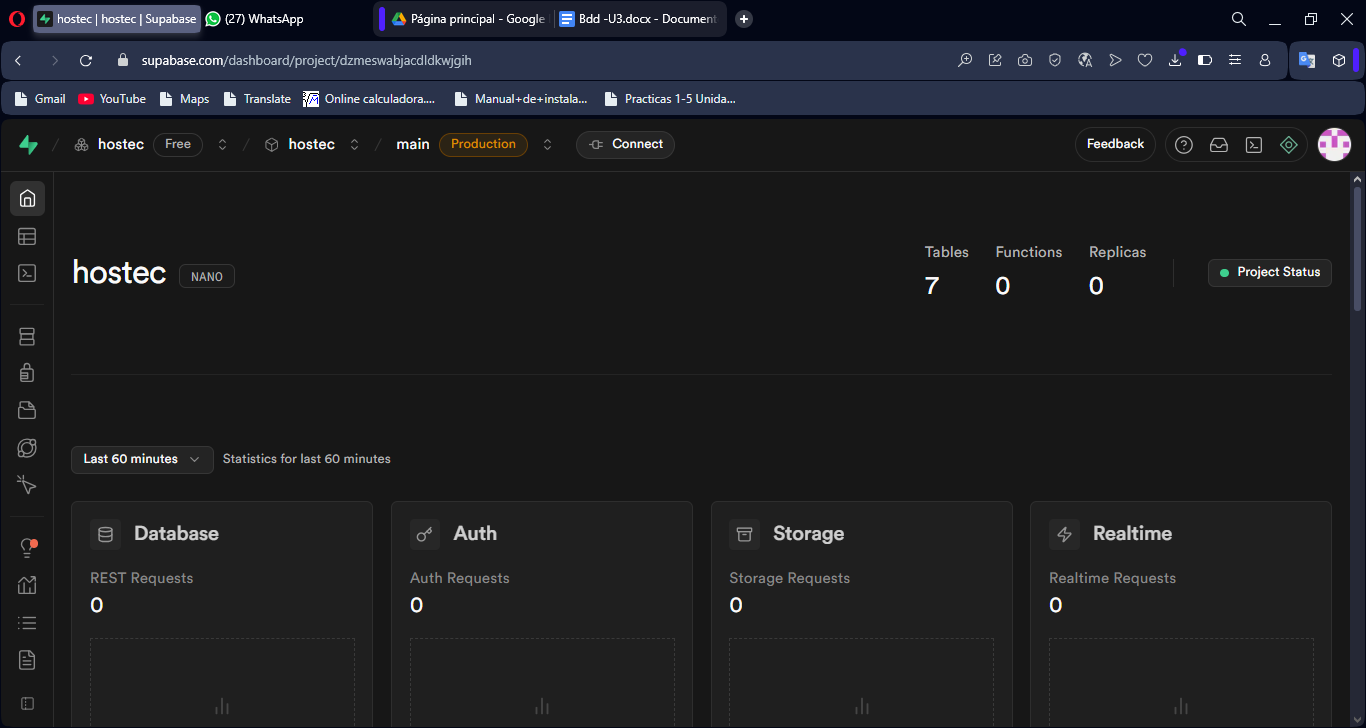
Ya creada la base nos direccionara a este menú, donde veremos desde las tabas, hasta las consultas creadas:

Ilustración 4 "Información general de la BDD"

Del lado izquierdo saldrá un submenú donde nos dará acceso a diferentes opciones:

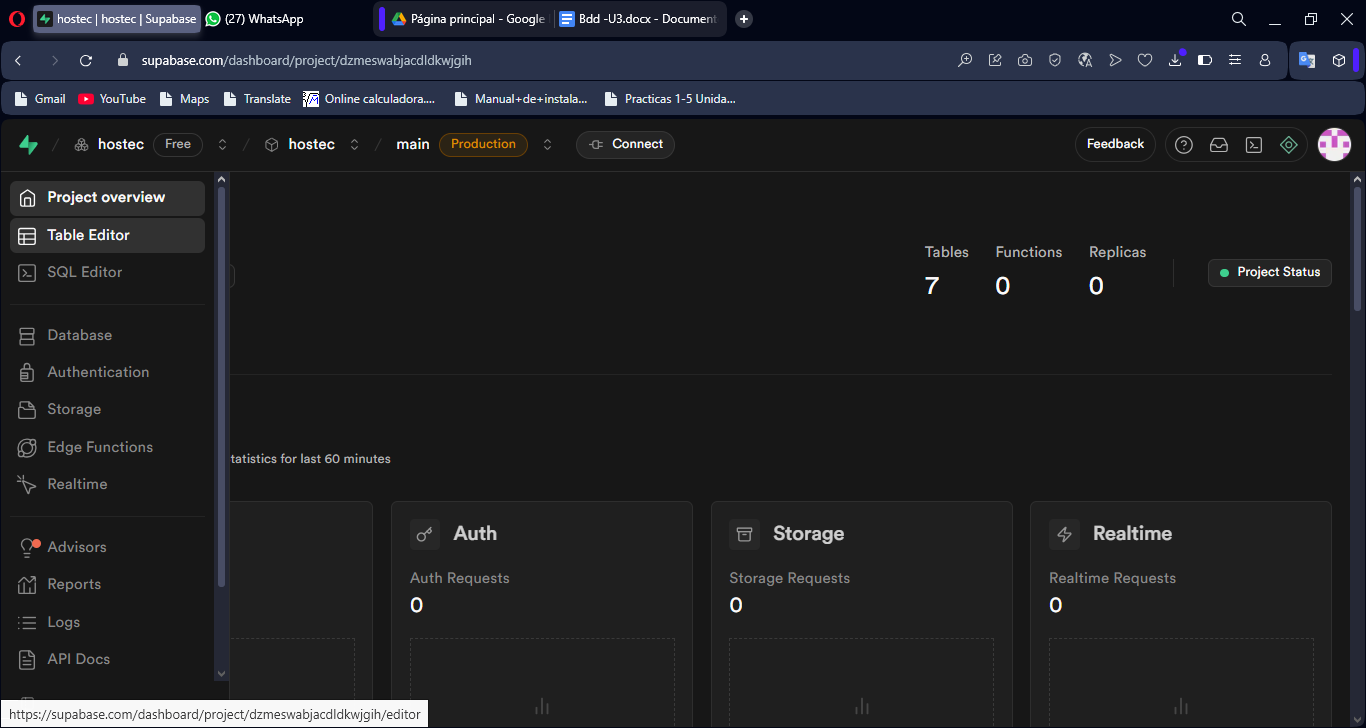


Ilustración 5 "Desglose de supabase"

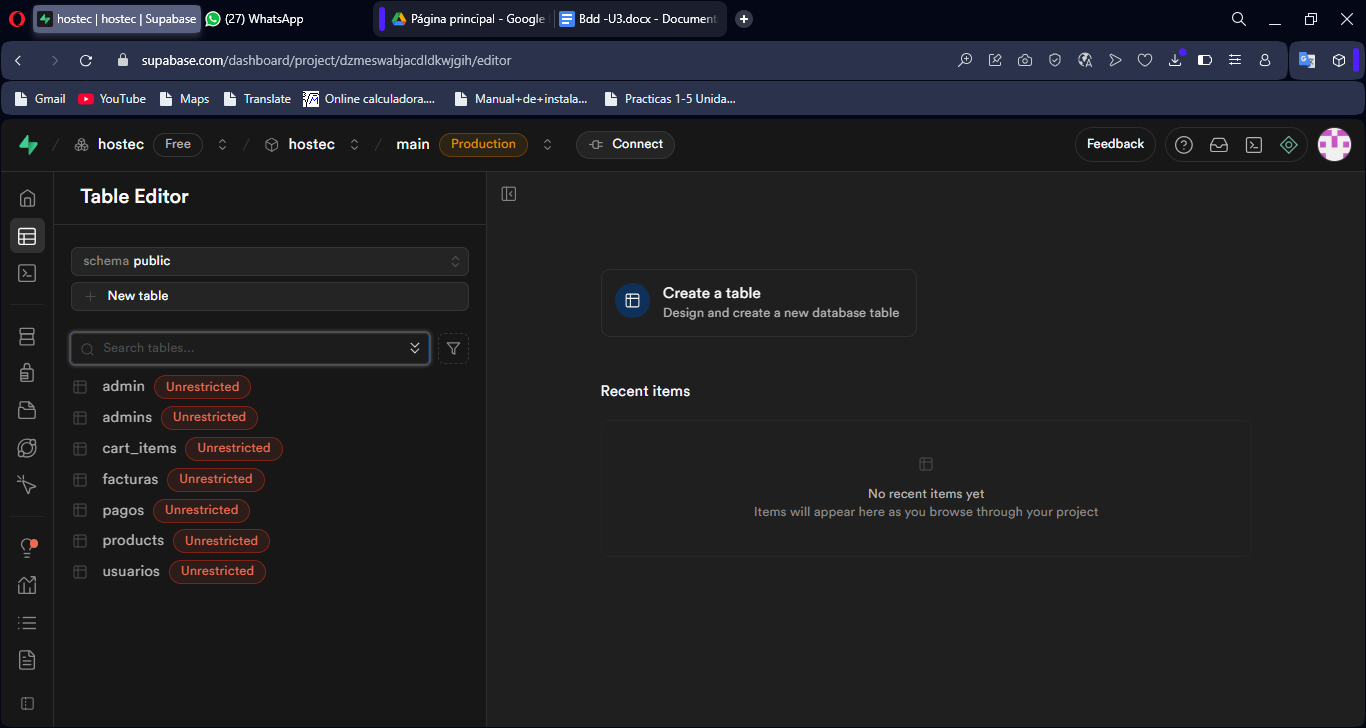
Daremos clic en “Table Editor”:  


Ilustración 6 "Visión de Table Editor"

Aquí crearemos las tablas con sus atributos, llaves primarias y foráneas.

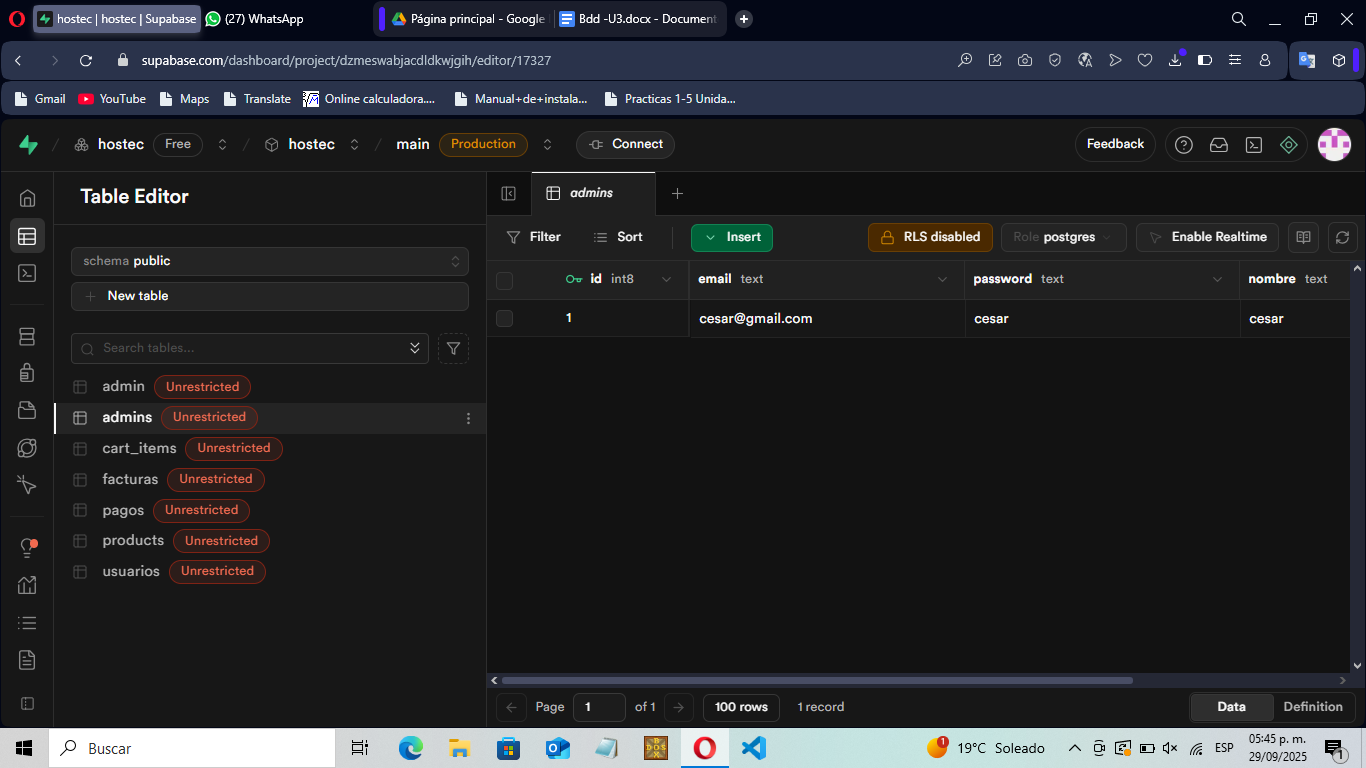
En este vamos a ver lo que contiene la tabla admins:

Ilustración 7 "Información de Tabla admins"

Como se muestra nos da un id, el email, password y nombre.

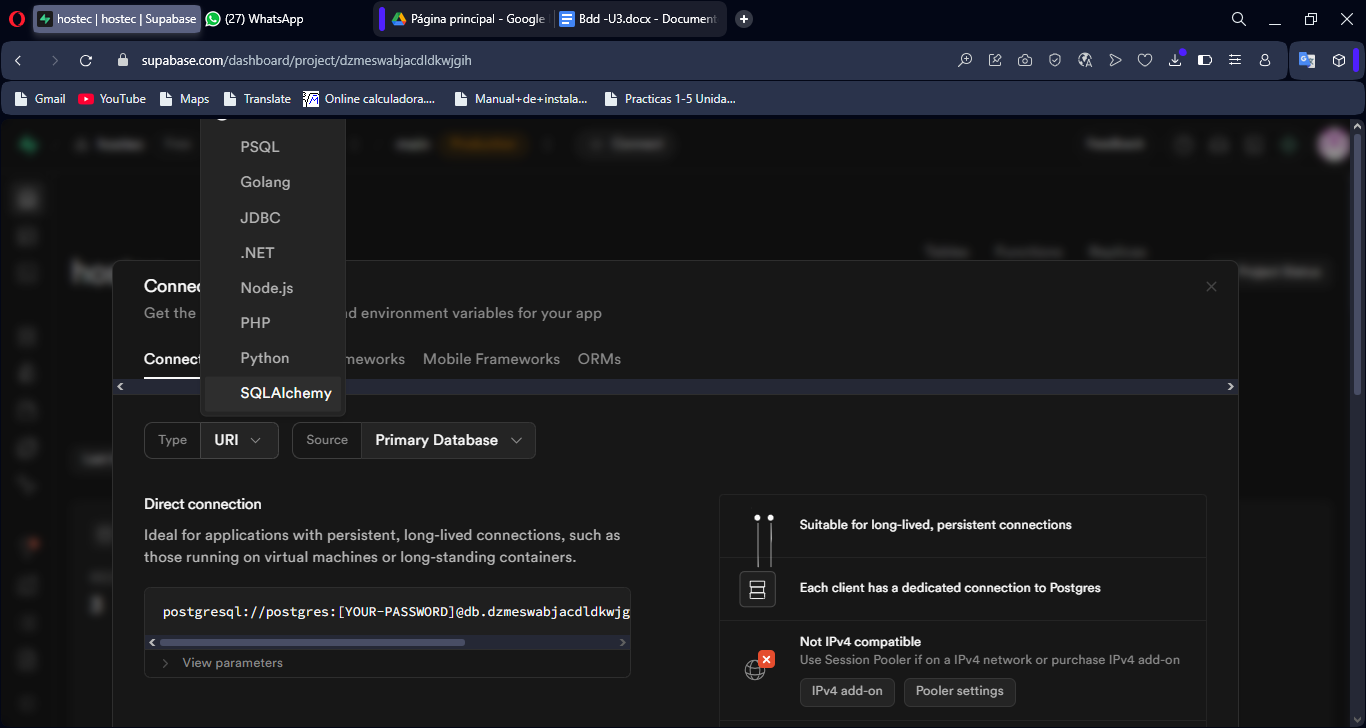
Si nos vamos a “Connect” vamos a ver el tipo de enlace al que vamos a conectar:  


Ilustración 8 "Tipos de enlaces"

En este caso estamos trabajando con URI para una conexión más rápida.

Abajo tendremos la dirección de conexión y la contraseña, esta contraseña que se da al proyecto, será el puente de conexión entre nuestro CRUD y la Data Base.

Abrimos nuestro proyecto en el archivo [supabaseClients.ts](http://supabaseclients.ts) e ingresamos las credenciales para la conexionInterfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Ilustración 9 "Conexión de la Bdd"

Podremos hacer consultas como en cualquier gestor de base de datos:

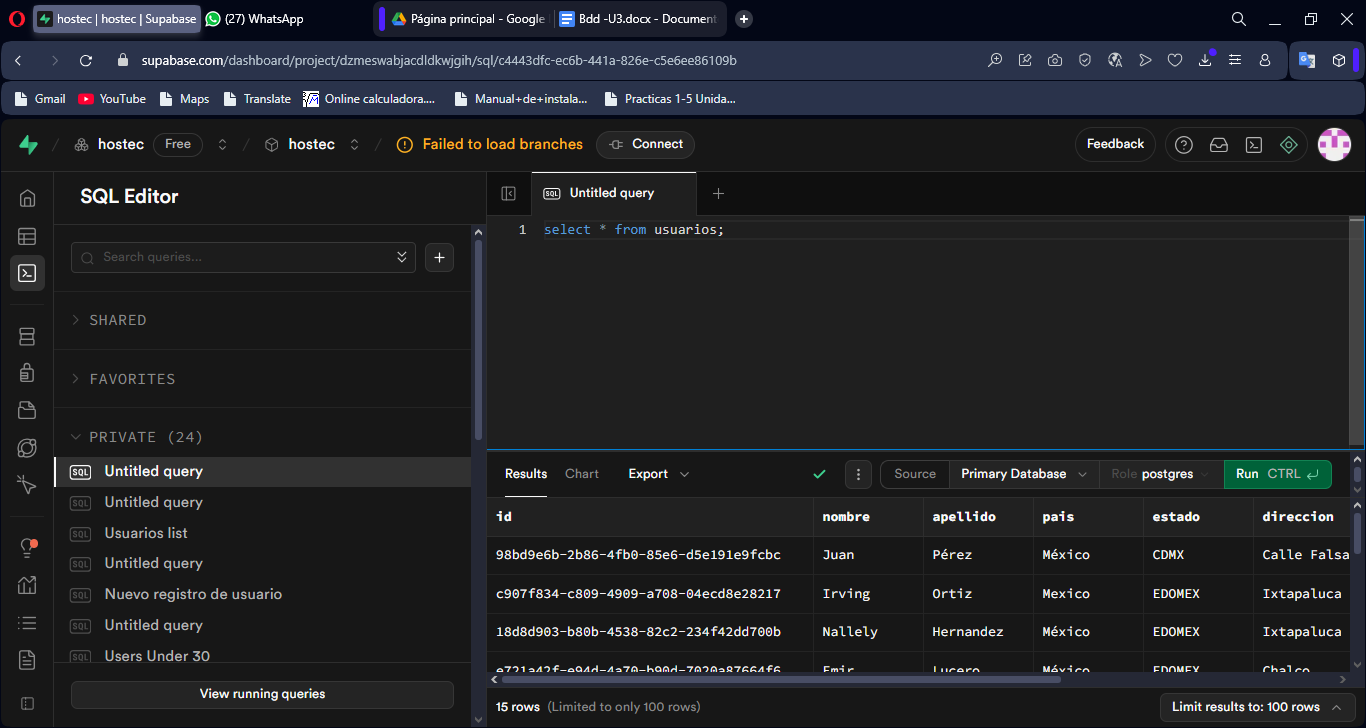


Ilustración 10 "Consulta a usuarios"

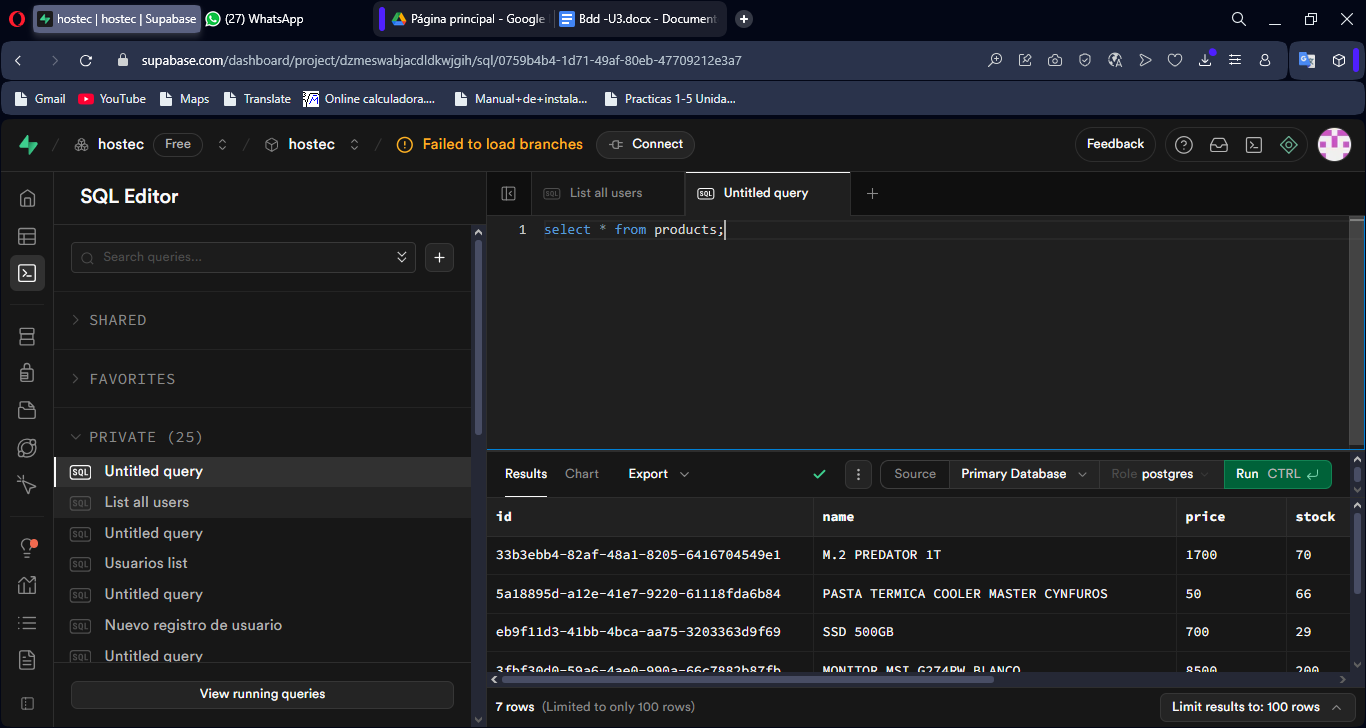


Ilustración 11 "Consulta a products"

## **¿En qué está basado?**

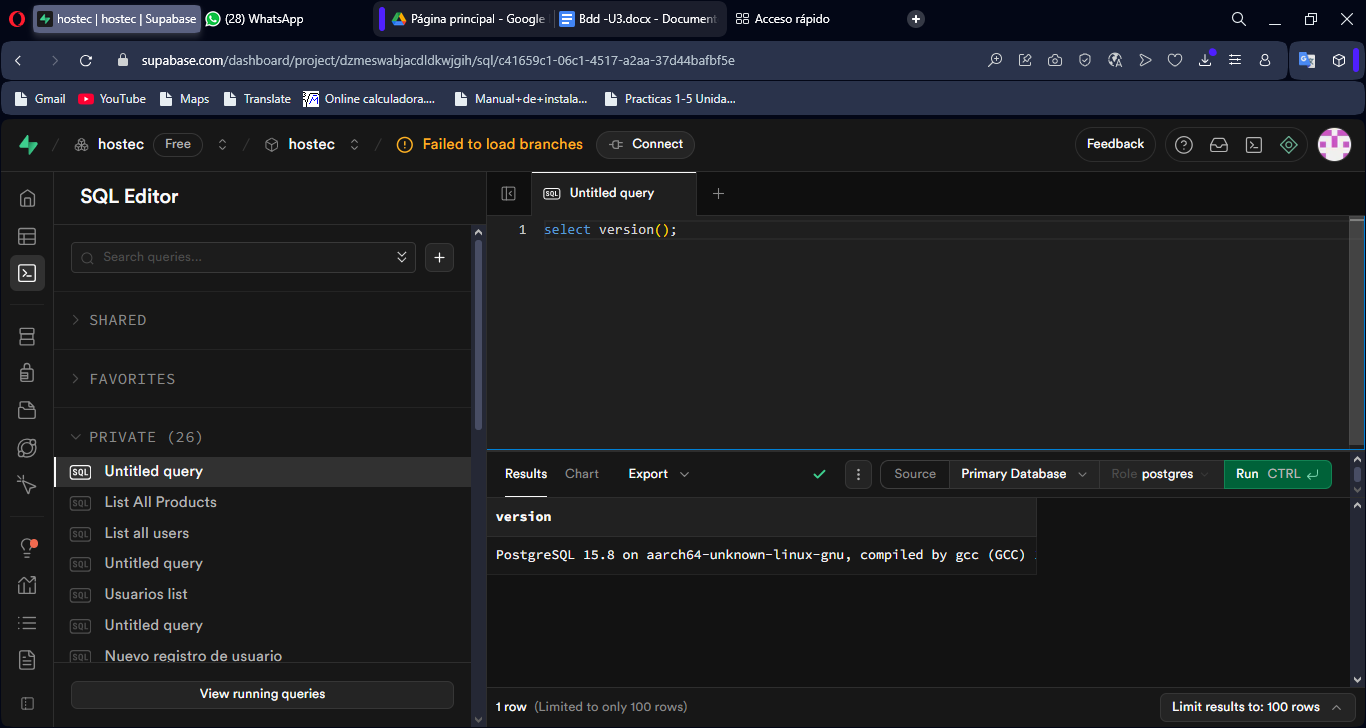
Este proyecto se adaptó a PostgresSQL 15.8, al ejecutar el comando en consola SELECT version(); nos dara como resultado la versión del gestor.

Ilustración 12 "Visualización a la versión del gestor"

# **Modelado de datos.**

El sistema cuenta con una base de datos relacional diseñada para gestionar la información de usuarios, productos, carritos de compra, pagos y facturación. El modelado se realizó a través de un Diagrama Entidad–Relación (DER) que refleja las entidades principales, sus atributos y las relaciones entre ellas.

## **Entidades principales**

Usuarios:

* Almacena los datos personales de cada usuario, incluyendo nombre, apellido, país, estado, dirección, edad, correo y contraseña.
* Se relaciona con los pagos y con las facturas que genera al adquirir productos.

Products:

* Contiene la información de los artículos disponibles en la tienda, como nombre, stock, precio y enlace a la imagen.
* Se conecta con carrito de compras y con facturas, ya que puede ser agregado por los usuarios o facturado en una compra.

Carrito de compras (cart\_items):

* Registra los productos seleccionados por cada usuario antes de confirmar una compra.
* Incluye atributos como nombre del producto, precio y cantidad.
* Se relaciona con usuarios y productos.

Pagos:

* Administra la información relacionada con el método de pago, incluyendo titular de la tarjeta, número, fecha de expiración, CVV y dirección de envío.
* Está vinculado a los usuarios, quienes realizan los pagos.

Facturas:

* Representa el comprobante de compra generado al concretar una transacción.
* Contiene fecha, dirección de envío, cantidad adquirida y el total.
* Relaciona a los usuarios con los productos adquiridos.Diagrama

  El contenido generado por IA puede ser incorrecto.

Ilustración 13 "Diagrama Entidad–Relación"

*Ilustración "Diagrama Entidad–Relación"*

## **Relaciones entre entidades**

* Un usuario puede realizar muchos pagos (1:N).
* Un usuario puede generar múltiples facturas (1:N).
* Una factura incluye uno o varios productos (N:M, resuelto mediante claves foráneas).
* Un usuario agrega productos a su carrito de compras (1:N).
* Los productos pueden estar en varios carritos y facturas.

Este modelo nos permite registrar usuarios y gestionar su información personal, controlar el inventario de productos disponibles, administrar carritos de compra de manera dinámica, procesar pagos de forma segura y generar facturas que incluyen el detalle de cada transacción realizada en el sistema**.**

## **Modelo Físico**

Se muestra la ejecución de una consulta SQL en Supabase sobre la tabla products, utilizando el comando SELECT \* FROM products;. En el resultado se observan los registros almacenados en la base de datos, donde cada producto está identificado por un UUID en el campo id y cuenta con un atributo name que describe su nombre, como por ejemplo M.2 Predator, SSD 500GB o Memoria RAM. Este ejemplo refleja la implementación del modelo físico, ya que se está accediendo directamente a los datos reales dentro del motor de base de datos PostgreSQL.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Ilustración 14 "comando SELECT \* FROM products"

Ilustración 15 "Registros almacenados

*Ilustración "comando SELECT \* FROM products"*

*Ilustración "Registros almacenados"*

Se ejecuta la consulta SELECT \* FROM usuarios; mostrando los registros almacenados en la base de datos PostgreSQL. Cada usuario está identificado por un (id) y contiene datos personales como nombre, apellido, país, estado, dirección, edad, correo y contraseña. La tabla Usuario también se relaciona con los pagos y las facturas generadas al adquirir productos.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ilustración 16 "consulta SELECT \* FROM usuarios"

*Ilustración "consulta SELECT \* FROM usuarios"*

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ilustración 17 "Registros almacenados de usuarios"

*Ilustración "Registros almacenados de usuarios"*

Se ejecuta la consulta SELECT \* FROM Products, mostrando los registros almacenados y cada producto está identificado por un (id) y contiene información como nombre, stock, precio y enlace a la imagen. La tabla Productos se conecta con el carrito de compras y con las facturas, ya que los artículos pueden ser agregados por los usuarios o facturados en una compra.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Ilustración 18 "Registros almacenados de products"

*Ilustración "Registros almacenados de products"*

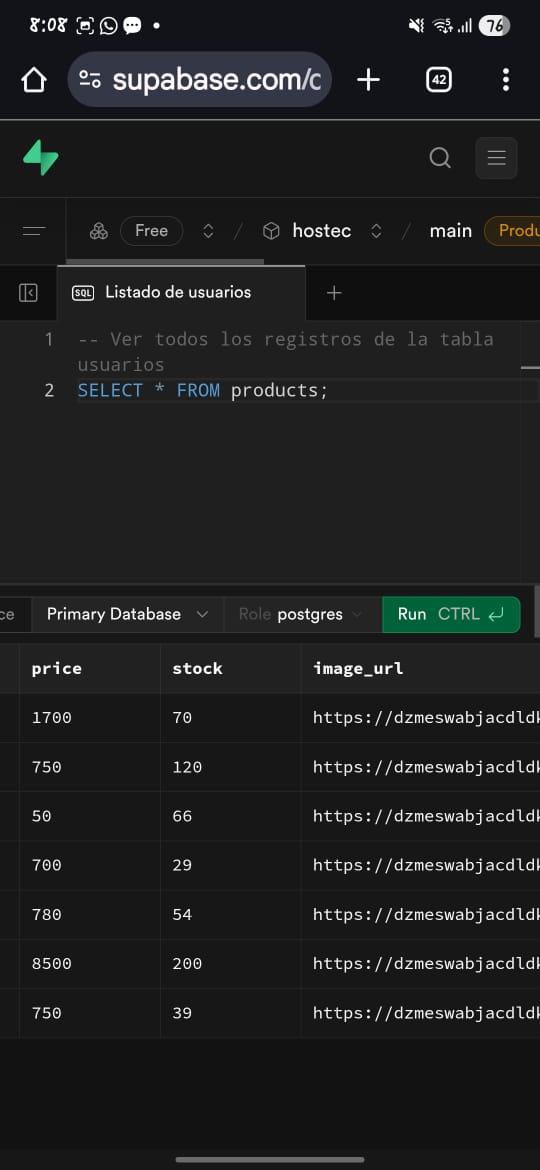


Ilustración 19 "consulta SELECT \* FROM Products"

*Ilustración "consulta SELECT \* FROM Productos"*

La imagen muestra la inserción de un registro en la tabla usuarios de la base de datos utilizando una sentencia SQL INSERT INTO. Se observan los campos de la tabla, como nombre, apellido, país, estado, dirección, edad, correo y contraseña, y los valores correspondientes de un usuario de ejemplo. En la parte inferior se muestra el resultado de la inserción, confirmando que los datos se han registrado correctamente en la tabla.

Captura de pantalla con letras y números

El contenido generado por IA puede ser incorrecto.

Ilustración 20 "Inserción de un registro en la tabla usuarios”

*Ilustración "inserción de un registro en la tabla usuarios”*

Archivo de configuración del cliente de Supabase, donde se importan las librerías necesarias, se establece la URL del proyecto y la clave de API, y se crea un cliente para interactuar con la base de datos desde la aplicación.Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Ilustración 21 "Archivo de configuración "

*Ilustración "Archivo de configuración "*

Se muestra la terminal de Windows ejecutando el proyecto con Vite mediante el comando npm run dev. El servidor se inicia correctamente en la dirección http://localhost:5173 y se observan mensajes que indican la actualización en tiempo real de los archivos del proyecto.Texto

El contenido generado por IA puede ser incorrecto.

Ilustración 22 "Ejecutando el proyecto con Vite"

*Ilustración "Ejecutando el proyecto con Vite"*

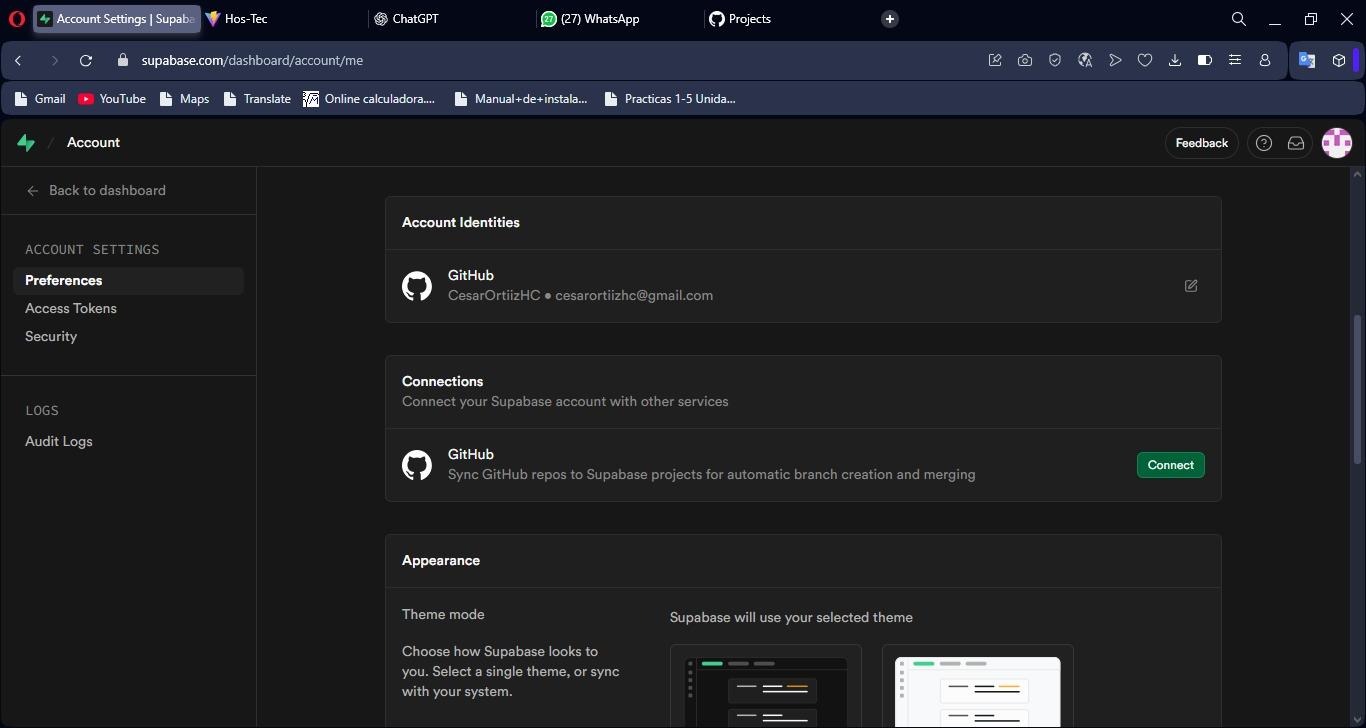
Se visualiza la página de configuración de cuenta de Supabase, donde el usuario puede gestionar su identidad vinculada a GitHub, conectar repositorios, elegir la apariencia del sistema y acceder a configuraciones adicionales de seguridad y registros.

Ilustración 23 "Repositorio en GitHub"

*Ilustración "Repositorio en GitHub"*

# **Conclusión.**

La práctica permitió comprender el proceso completo de diseño e implementación de una base de datos para un sistema de gestión de tienda en línea, desde el modelado conceptual hasta la aplicación práctica en un entorno real. A través de Supabase, se aprovecharon herramientas modernas que simplifican la creación de tablas, relaciones y consultas SQL sobre PostgreSQL, además de integrar funcionalidades como autenticación y conexión con proyectos frontend mediante Vite.

El análisis del modelo de datos, con entidades como usuarios, productos, carritos de compra, pagos y facturas, evidenció la importancia de definir correctamente las relaciones para garantizar la integridad de la información y el correcto flujo de las operaciones. Asimismo, se comprobó cómo las consultas permiten manipular y recuperar datos de manera eficiente, asegurando el control del inventario, el procesamiento de pagos y la generación de facturas.