



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



Contenido

INTRODUCCIÓN.....	3
MODELADO DE DATOS.	4
Entidades principales.....	4
Relaciones entre entidades	6
Modelo Físico	6

Figuras.

Ilustración 1 "Diagrama Entidad–Relación"	5
Ilustración 2 "Registros almacenados"	6
Ilustración 3 "comando SELECT * FROM products"	6
Ilustración 4 "consulta SELECT * FROM usuarios"	7
Ilustración 5 "Registros almacenados de usuarios"	7
Ilustración 6 "Registros almacenados de products"	8
Ilustración 7 "consulta SELECT * FROM Productos"	8
Ilustración 8 "inserción de un registro en la tabla usuarios"	8
Ilustración 9 "Archivo de configuración "	9
Ilustración 10 "Ejecutando el proyecto con Vite"	9
Ilustración 11 "Repositorio en GitHub"	10



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.



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.

Productos:

- 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.

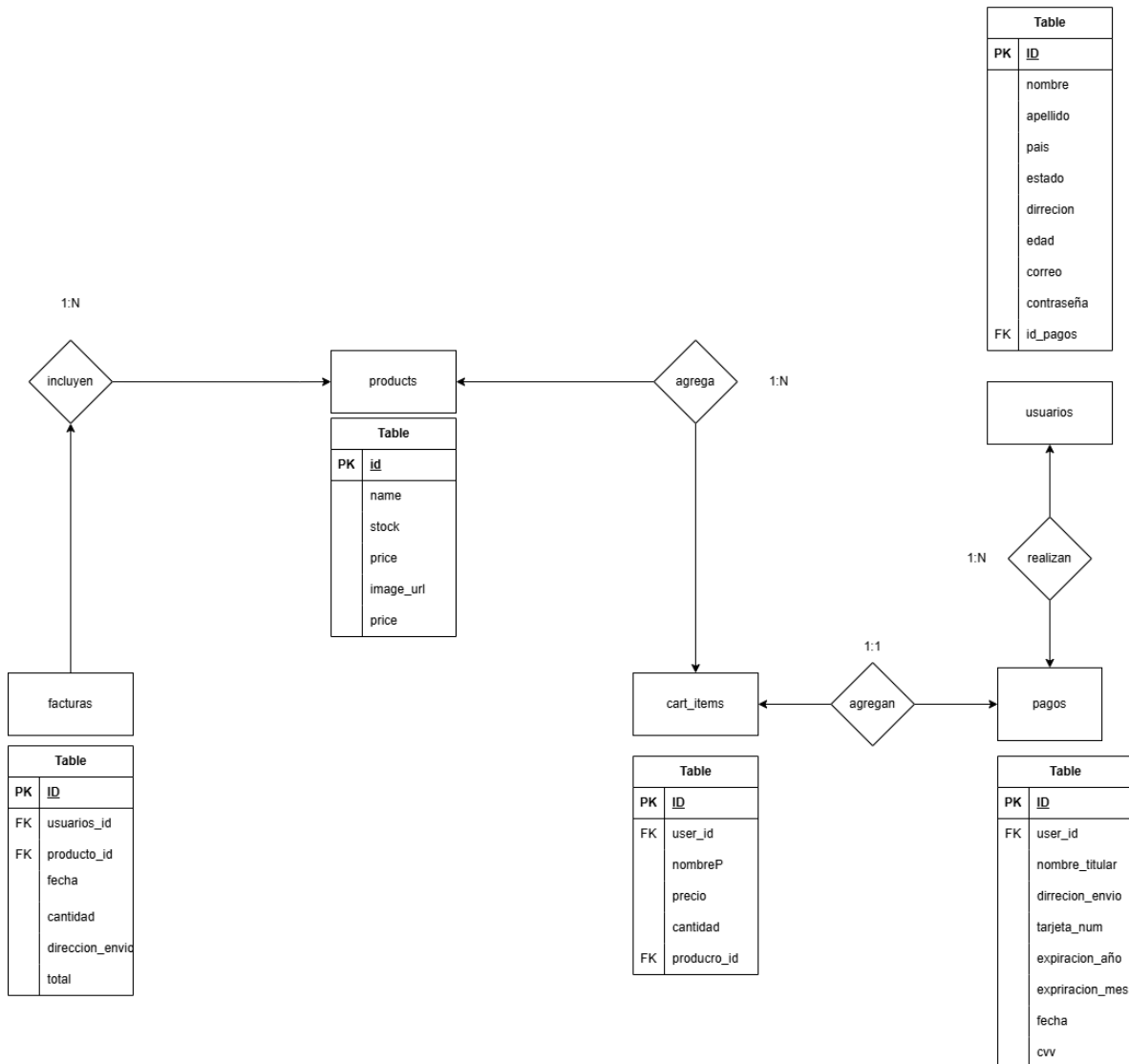


Ilustración 1 "Diagrama Entidad-Relación"

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.

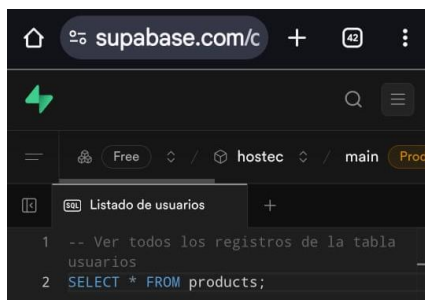
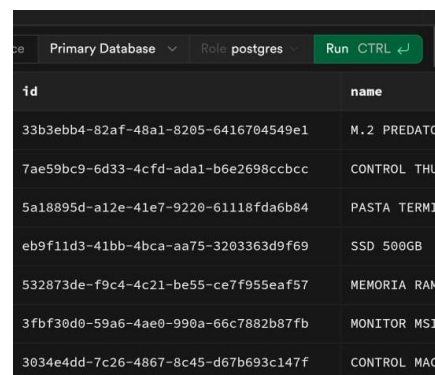


Ilustración 3 "comando `SELECT * FROM products`"



id	name
33b3ebb4-82af-48a1-8205-6416704549e1	M.2 PREDATOR
7ae59bc9-6d33-4cfd-ada1-b6e2698ccbcc	CONTROL THUNDERBOLT
5a18895d-a12e-41e7-9220-61118fda6b84	PASTA TERMINAL
eb9f11d3-41bb-4bca-aa75-3203363d9f69	SSD 500GB
532873de-f9c4-4c21-be55-ce7f955eaf57	MEMORIA RAM
3fbf30d0-59a6-4ae0-990a-66c7882b87fb	MONITOR MSI
3034e4dd-7c26-4867-8c45-d67b693c147f	CONTROL MAC

Ilustración 2 "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.

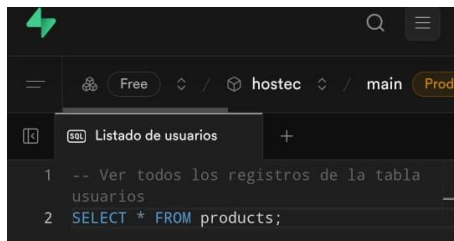
```
Free / hostec / main
Listado de usuarios
1 -- Ver todos los registros de la tabla usuarios
2 SELECT * FROM usuarios;
```

Ilustración 4 "consulta `SELECT * FROM usuarios`"

id	nombre	apellido	pais	estado	direccion	edad	correo
71daf46a-17fd-4ea6-ab96-b0c6fda05d85	Sofia	Ortiz	Mexico	EDOMEX	Ixtapalica	23	sofia1@gmail.com
e7b8b9e5-3f76-4d16-84d6-bdea44540519	Juan	Pérez	México	Estado de México	Calle 123	25	juan@example.com
5d251eaa-fc14-496c-a7cb-c57a663326e4	JOSE	LUCERO	México	EDOMEX	Chalco	22	JOSE@gmail.com
22970004-cca2-46e4-9c76-956578b7395c	FERMIN	SANABRIA	México	EDOMEX	Ixtapaluca	29	fermin@gmail.com
6c514584-2b01-4b80-b735-84df93abbb92	CARLOS	RUIZ	México	EDOMEX	Ixtapaluca	30	carlos@gmail.com
ab4d05e3-9828-4687-8acd-c8f9af3b89b9	CESAR	ORTIZ	México	EDOMEX	Ixtapaluca	21	CESAR@gmail.com
a8398bd3-55ad-4fdb-9949-5969a7b11a84	SOFIA	HERNANDEZ	México	EDOMEX	Ixtapaluca	21	sofia@gmail.com
2118c869-67ab-4b47-8ce0-7cc187e5b85f	Ferminn	Sanabria	México	EDOMEX	Ixtapaluca	23	ferminn@gmail.com
f82491cf-0ad5-405b-8ba1-d6b50a5de766	Fer	Sanabria	México	EDOMEX	Ixtapaluca	21	ferminnn@gmail.com
a69339b4-d170-4c07-a3fb-0f02de72485d	dvds	dsvsd	sdvsd	svds	sdvsd	2	fermin11@gmail.com
15 rows							

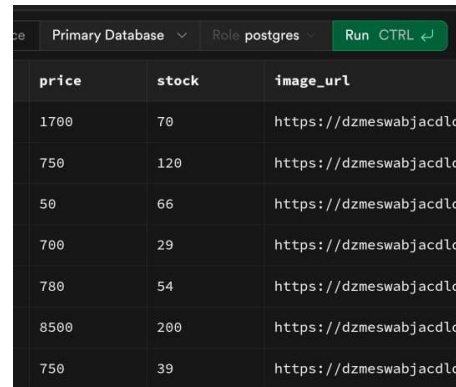
Ilustración 5 "Registros almacenados de usuarios"

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.



```
-- Ver todos los registros de la tabla usuarios
SELECT * FROM products;
```

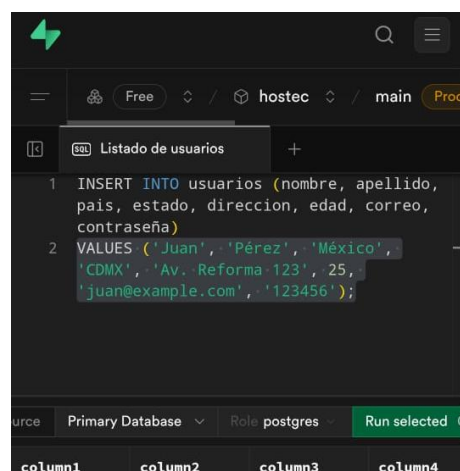
Ilustración 7 "consulta `SELECT * FROM Productos`"



price	stock	image_url
1700	70	https://dzmewabjacdld
750	120	https://dzmewabjacdld
50	66	https://dzmewabjacdld
700	29	https://dzmewabjacdld
780	54	https://dzmewabjacdld
8500	200	https://dzmewabjacdld
750	39	https://dzmewabjacdld

Ilustración 6 "Registros almacenados de `products`"

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.



```
INSERT INTO usuarios (nombre, apellido,
pais, estado, direccion, edad, correo,
contraseña)
VALUES ('Juan', 'Pérez', 'México',
'CDMX', 'Av. Reforma-123', 25,
'juan@example.com', '123456');
```

Ilustración 8 "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.

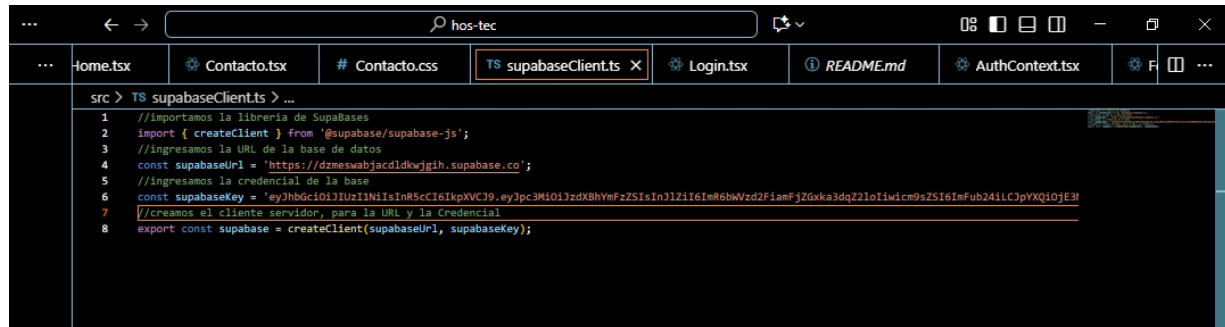


Ilustración 9 "Archivo de configuración "

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.

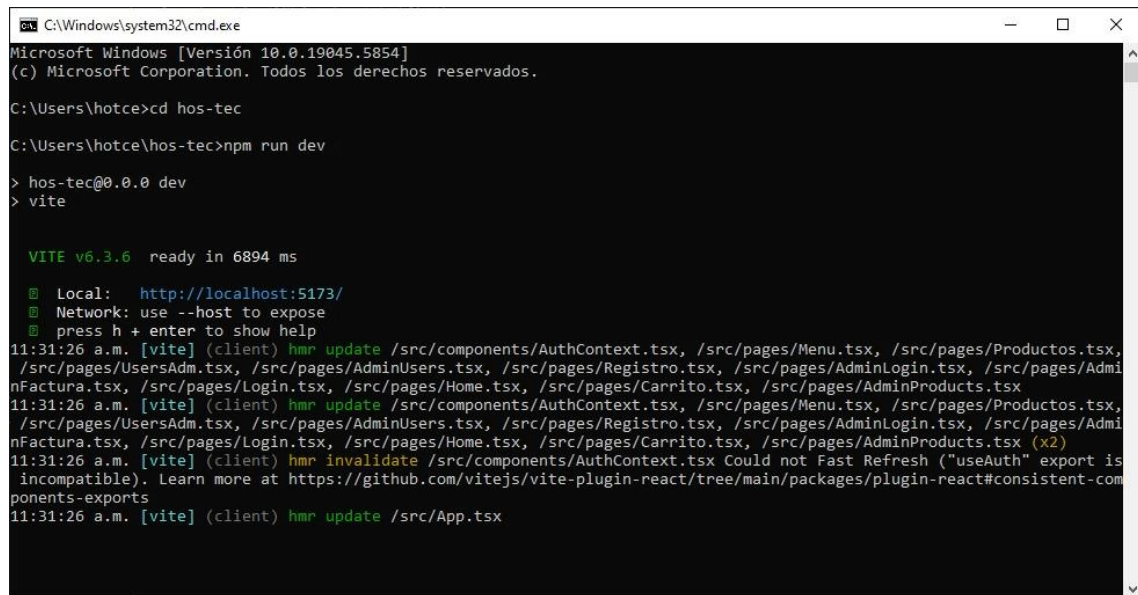


Ilustración 10 "Ejecutando el proyecto con Vite"

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.

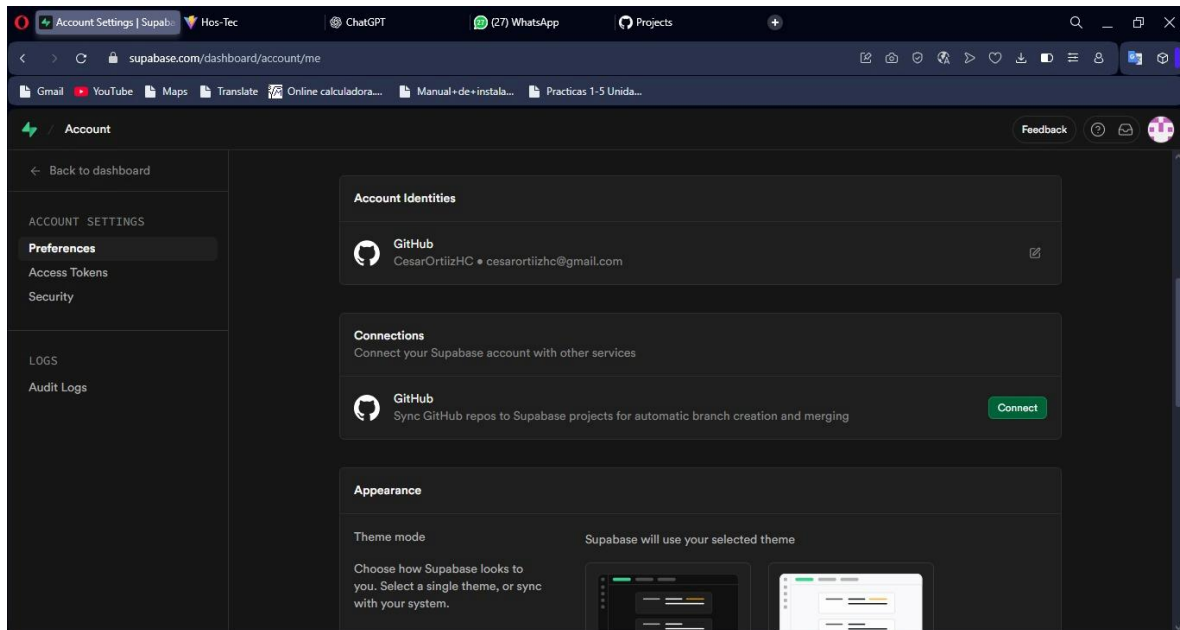


Ilustración 11 "Repositorio en GitHub"