



Universidad Nacional Autónoma de México

Facultad de Ciencias

Fundamentos de Bases de Datos

Proyecto Final

Estudiantes :

Moisés Abraham Lira Rivera
Juan Luis Peña Mata
Marco Flores Cid
Andrés Daniel López Molina
Etni Sarai Castro Sierra
Daiki Benedek Rueda Tokuhara

Profesor :

Gerardo Avilés Rosas

Ayudantes :

Luis Enrique García Gómez
Ricardo Badillo Macías

Índice

1. Introducción	2
1.1. Descripción	2
2. Modelo Entidad-Relación	3
2.1. Bazar	3
2.2. Estand	4
2.3. Agenda	5
2.4. Negocio	5
2.5. Cliente	6
2.6. Ticket	6
2.7. Mercancia	7
2.8. MétodoPago	8
2.9. Personal	9
2.10. Emprendedor	10
3. Modelo Relacional	11
3.1. Traducción al Modelo Relacional	11
4. Modelo Físico	12
4.1. Restricciones de Dominio	12
4.2. Procedimientos/Funciones (SP), Triggers y Disparadores	18
4.2.1. Procedimientos Almacenados	18
4.2.2. Procedimientos Almacenados usados para el proyecto	18
4.3. Triggers y Disparadores	22
4.3.1. Triggers usados para el proyecto	22
5. Población de la Base de Datos	23
6. Consultas	24
7. Guía de Ejecución	44
7.1. Herramientas a utilizar	44
7.2. Pasos para probar la Base de Datos desde Windows	44

1 Introducción

El bazar "**La Gatita Emprendedora**" desea tener un mayor control en el manejo de la información del mercado que estos administran, debido a que, al seguir utilizando registros en físicos a través de archivos, ha provocado múltiples problemas en el bazar.

***Contrato adjudicado a Computólogos A.C.
Solución presentada el 09 de junio de 2025.***

1.1 Descripción

El objetivo de este proyecto es diseñar una base de datos para el bazar "**La Gatita Emprendedora**". Queremos que toda la información esté en un solo lugar, bien organizada y fácil de consultar.

Para comenzar, analizamos todas las reglas del negocio que aparecen en el caso de uso. Con base en ese análisis, hicimos un diagrama Entidad-Relación (E-R) donde representamos de forma visual las entidades más importantes.

Después, adaptamos ese diagrama para convertirlo al modelo relacional, pensando ya en cómo serían las tablas en la base de datos.

A cada entidad le asignamos una llave primaria, y agregamos llaves foráneas para conectar unas tablas con otras.

Una vez que tuvimos el modelo relacional, pasamos a escribir el código DDL (Lenguaje de Definición de Datos), que es un conjunto de instrucciones que sirven para crear las tablas, sus columnas, los tipos de datos, restricciones y relaciones entre ellas. Todo esto se hizo en PostgreSQL.

Además, usamos un tipo especial de DDL para definir cómo se van a almacenar los datos y cómo se van a mantener las conexiones entre tablas, usando políticas de mantenimiento para las llaves foráneas.

Después de crear la estructura, poblamos la base con datos de prueba, usando la herramienta Mockaroo para generar registros realistas. Finalmente, escribimos las consultas en SQL.

A continuación, se presenta el desarrollo del proyecto dividido en apartados.



2 Modelo Entidad-Relación

El modelo consta de 19 entidades, de las cuales 9 son entidades heredadas, cabe mencionar que no se cuenta con entidades débiles.

2.1 Bazar

Se consideran como atributos compuestos *Duración* y *Ubicación* ya que el bazar no siempre será en el mismo lugar, y puede durar entre uno a varios días.

Las amenidades serán un tipo de atributo multivaluado ya que hay diversos tipos de este con los que contará el bazar, dependerán de si el bazar se realiza al aire libre o al cubierto.

Determinamos como llave primaria del bazar a "IDBazar" para garantizar la unicidad de la entidad.

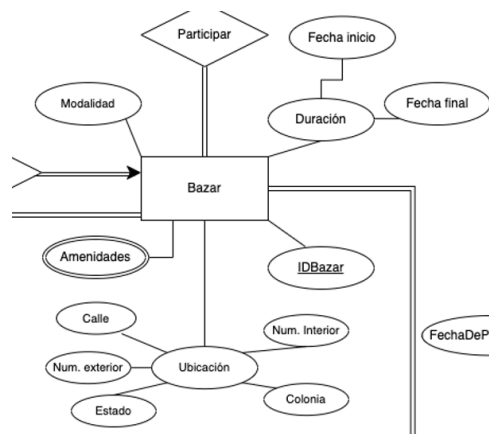


Figura 1: Entidad Bazar

Esta entidad está relacionada con las entidades:

- **Personal mediante la relación Trabajar:** Con cardinalidad N:M y participación total de ambos lados ya que varias personas pueden trabajar en varios bazares. También, un bazar requiere que siempre se tenga a disposición cierto personal y todo personal trabaja en un bazar.
- **Agenda mediante la relación Tener:** La participación será total de ambos lados ya que todo bazar necesita una agenda para realizarse y toda agenda será de un bazar. En cuanto a la cardinalidad, es 1:1 porque un solo bazar solo puede tener una agenda
- **Negocio mediante la relación Participar:** Con cardinalidad N:M ya que varios bazares pueden tener varios negocios. Ya que un negocio puede elegir a cuales bazares ir la participación es parcial de lado de Negocio, pero en todo bazar deberán participar negocios, por lo que la participación es total de lado de Bazar.
- **Cliente mediante la relación Ir:** Con cardinalidad N:M ya que varios clientes pueden ir a varios bazares. Además, un cliente puede elegir a que bazares ir, pero no tiene

porque ir a todos por lo que la participación es parcial de lado de Cliente, por otro lado todo bazar debe ser accesible a los clientes por lo que la participación es total de lado de Bazar.

2.2 Estand

Se tendrá un estand para que cada uno de los negocios que estarán participando en el bazar puedan presentar sus productos, por lo que determinaremos el número de estand y su precio, siendo "Num de estand" considerada como llave primaria para garantizar la unicidad de la entidad, siendo así un identificador natural.

Aunque por defecto se tiene el paquete básico (sin precio extra); como tal, habrá 3 diferentes paquetes de estand que pueden ser escogidos por los negocios:

- **Paquete Básico:** Cuenta con las amenidades de 1 mesa y 2 sillas.
- **Paquete Premium:** Cuenta con 2 mesas y 4 sillas, cuyo precio es un 10 % más del precio base del estand.
- **Paquete Emprendedor:** Cuenta con 3 mesas, 6 sillas, una pantalla táctil y toma de corriente para los productos, cuyo precio es un 30 % más del precio base del estand.

Los cuales, se contendrán como atributo de relación *Paquete stand*, esto ya que facilita el proceso de búsqueda.

Además, el precio total se deja como atributo calculado ya que dependerá de otra entidad para determinar su valor, y será un valor que puede irse modificando.

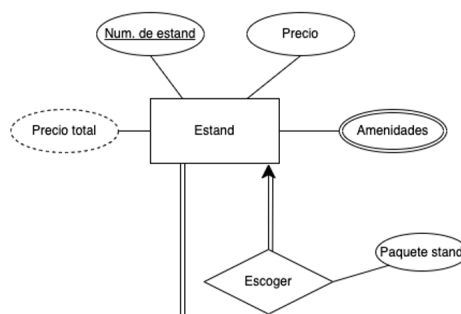


Figura 2: Entidad Estand

Esta entidad está relacionada con las entidades:

- **Agenda mediante la relación Registrar:**

Sabemos entonces que varios estands pueden tener varias agendas, por lo tanto es cardinalidad N:M. En cuanto a la participación, si se tiene un stand, para un negocio, este se registrará en alguna fecha para asistir al bazar entonces debería ser total de lado de estand.

Esta relación permite obtener de una manera más directa la información respecto a que si el estand está considerado en crearse cuando un negocio lo pide, entonces se sabe a

que negocio le pertenece junto con la información de que días estará, en cual estand y con que paquete.

En dicha relación, únicamente hay un atributo de relación que será la fecha en que el estand asistirá; se evita la información redundante como por ejemplo el paquete de estand, esto gracias a que se puede obtener a partir de la relación.

Tanto la entidad Agenda, como la relación Registrar son fuertes, ya que la agenda no depende de la existencia del negocio, pese a que este no asista.

■ **Negocio mediante la relación Escoger:**

Solo habrá un estand por negocio, esto nos indica una cardinalidad 1:1. Existirá un atributo de relación que corresponderá al paquete escogido, ya que cada vez que el negocio escoja el estand, se tiene que escoger el paquete que este posee.

En cuanto a la participación, como todo estand será escogido por algún negocio, la participación será total de lado de Estand, como no todos los negocios necesitaran un estand la participación es parcial de lado de Negocio.

2.3 Agenda

El bazar debe tener una agenda en la cual se registren los estands que se presentarán, esto para tener una correcta organización y permitir que todos los emprendedores tengan la oportunidad de vender.

Determinamos como llave primaria a “IDAgenda” para garantizar la unicidad de la entidad.

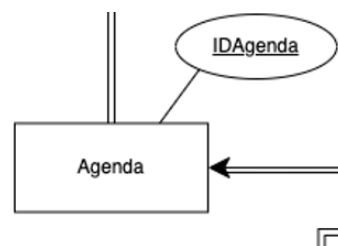


Figura 3: Entidad Agenda

2.4 Negocio

Un negocio podrá asistir varias veces al bazar mientras este dure, por lo tanto tendrá una cardinalidad N:M con participación total de un lado.

Debido a las diferentes alternativas que se brindan hoy en día, los negocios pueden tener diferentes correos, teléfonos y enlaces webs definidos como medios de contacto, por lo tanto, serán considerados como atributos multivaluados.

Para agilizar y mantener un cupo adecuado, cada negocio tendrá a lo más 2 emprendedores supervisando, lo cual nos denota una relación 1:N con participación total de ambos lados.

Determinamos como llave primaria a “IDNegocio” para garantizar la unicidad de la entidad.

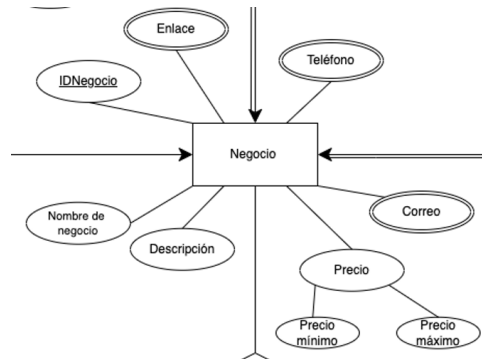


Figura 4: Entidad Negocio

2.5 Cliente

Su información fungirá para la parte de mantener un control de asistencia al bazar, como además, para la generación de tickets en el caso de haber realizado la compra en línea.

Determinamos como llave primaria a “IDCliente” para garantizar la unicidad de la entidad.

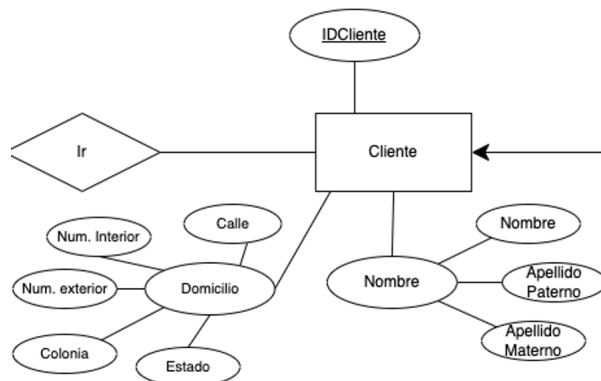


Figura 5: Entidad Cliente

2.6 Ticket

Debido a que el bazar se quedará con la ganancia del 20 % sobre el precio total a pagar (será un impuesto), el precio total a pagar y el precio final serán atributos calculados, ya que dependerá del monto de cada compra.

Determinamos como llave primaria a “IDTicket” para garantizar la unicidad de la entidad.

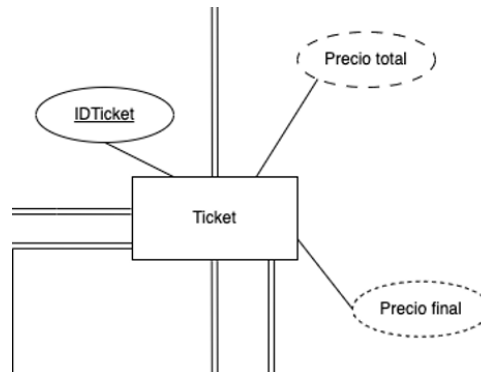


Figura 6: Entidad Ticket

Esta entidad está relacionada con las entidades:

- **Clientes mediante la relación Obtener:**

Un cliente puede tener varios tickets, por ende, la cardinalidad será 1:N. Y ya que cada ticket pertenecerá a un cliente, la participación es total de lado de Ticket, pero no todo cliente hará una compra, así la participación es parcial de lado de Cliente.

- **Emprendedor mediante la relación Imprimir:**

Un emprendedor puede realizar varios tickets, por ello, será 1:N. Además, todo ticket es impreso por un emprendedor por lo que la participación es total de lado de Ticket, sin embargo puede que un emprendedor nunca imprima un ticket, así, la participación es parcial de lado de Emprendedor.

- **Producto y Servicio mediante la relación Tener:** Su cardinalidad será N:M con participación total de lado de ticket, ya que todos los productos y servicios adquiridos en el bazar tendrán su respectivo ticket, siendo así que un ticket no puede estar vacío.

Fue determinada como una a relación binaria (aridad 2), ya que de este modo si estamos indicando que un ticket tiene servicios, tendrá sus atributos correspondientes únicamente, no tiene muchos sentido que manejemos cantidad.

2.7 Mercancia

Se considera una especialización disjunta total ya que se delimita a que un negocio solamente puede tener un giro; ya sea el de ofrecer un producto o un servicio.

A su vez, *Producto* (una de sus entidades hijas) contendrá especialización disjunta total también; puede ser perecedero (Cualquier producto que degrade su calidad con el tiempo, conllevan una fecha de preparación y fecha de caducidad) o no serlo.

Con la finalidad de que aquellos negocios que prefieran vender su mercancía en línea, se tendrá una cantidad registrada de productos que se tiene en stock, la cual es considerada como atributo calculado debido a que depende de otra entidades para su cálculo y pueden ser actualizados constantemente.

Determinamos como llave primaria a “IDMercancia” para garantizar la unicidad de la entidad, siendo identificador a la vez de sus entidades hijas.

Entidades heredadas: Producto y Servicio; Perecedero y NoPerecedero

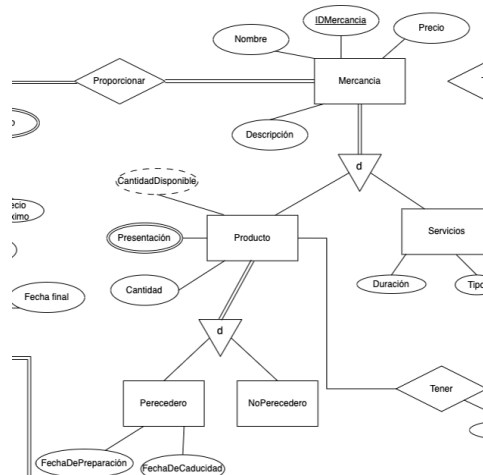


Figura 7: Entidad Mercancia

Esta entidad está relacionada con las entidades:

- **Negocio mediante la relación Proporcionar:**

Un negocio puede tener mucha mercancía, su cardinalidad será 1:N. Ya que todo negocio venderá mercancía y toda la mercancía será vendida por algún negocio, la participación es total de ambos lados.

2.8 MétodoPago

Será una especialización total con disyunción, únicamente se podrá elegir un método de pago ya que no tendría sentido cobrar dicha compra de ambas maneras, es decir, efectuar el pago 2 ocasiones.

Determinamos como llave primaria a “IDPago” para garantizar la unicidad de la entidad, el cual, será usado para los 2 tipos de pagos disponibles: Tarjeta y Efectivo.

Entidades heredadas: Tarjeta y Efectivo

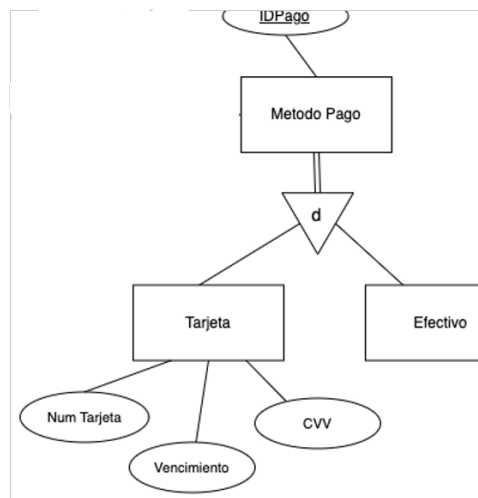


Figura 8: Entidad MétodoPago

Esta entidad está relacionada con las entidad:

- **Ticket mediante la relación Pagar:** Solo puede existir un método de pago para varios tickets, será cardinalidad N:1. Todo ticket deberá ser pagado, así, la participación es total de lado de Ticket, como no todo método de pago registrado se usará para pagar, la participación será parcial de lado de Método Pago.

2.9 Personal

Para cuando el evento sea presencial, habrá 3 tipos de personal obligatoriamente:

1. Médico
2. Seguridad
3. Limpieza

Por lo que, se considera como especialización disjunta total.

Además, el horario será considerado como atributo compuesto, ya que estará conformado por la hora en que se comienza a trabajar y también la hora en que termina su turno.

Determinamos que su RFC fungirá como identificador (ya que es bajo un contexto laboral, y es aquel que nos identifica como personas físicas ante hacienda y demás organismos).

El personal puede tener diferentes datos de contacto, ya sea que se le contacte por medio de su(s) número(s) telefónico(s) y/o correo(s), por lo cual, estos dos serán considerados como multivaluados.

Por su parte, tanto el nombre, domicilio y horario, serán considerados como compuestos, ya que hay diferentes datos que se necesitan especificar de manera más puntual para conformarlo y tenerlo correctamente registrado.

Entidades heredadas: Médico, Seguridad y Limpieza

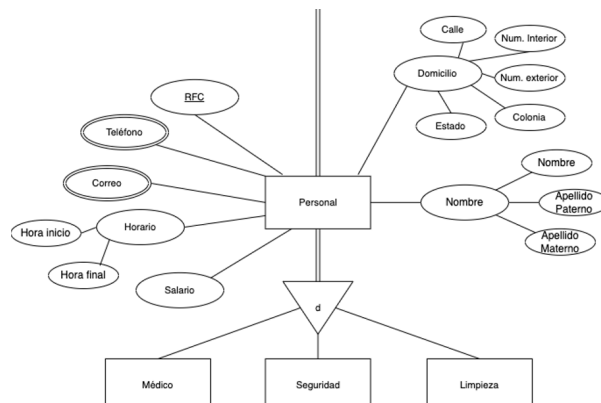


Figura 9: Entidad Personal

2.10 Emprendedor

Nuevamente, como llave primaria será el RFC tomando en cuenta el contexto laboral, y el correo y teléfono serán multivaluados porque puede tener diferentes medios de contacto.

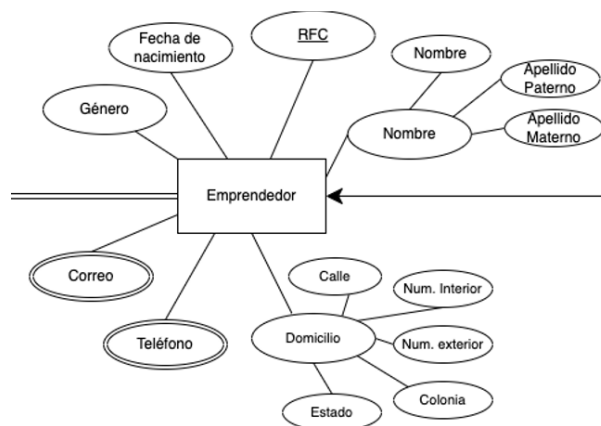


Figura 10: Entidad Emprendedor

Esta entidad está relacionada con la entidad:

- Negocio mediante la relación Supervisar: Cardinalidad:** Un negocio puede tener varios emprendedores, su cardinalidad es 1:N. La participación es total de ambos lados ya que todo negocio deberá tener emprendedores que lo supervisen y todo emprendedor supervisará a un negocio.

Actualmente es considerado como el principal modelo de datos para las aplicaciones de procesamiento de datos, esto debido a su simplicidad, ya que nos facilita el trabajo como programadores.

[illegible]

4 Modelo Físico

Usando el lenguaje para definición de datos (DDL), aseguramos la integridad referencial para obtener un esquema robusto mediante restricciones vistas en clase como llaves, default, not null, constraint, para mantener la lógica del negocio.

Recordemos que el DDL consiste en los comandos SQL que permiten definir la estructura de la base de datos.

4.1 Restricciones de Dominio

Entidad: Bazar	
modalidad	boolean NOT NULL
fecha inicio, fecha final	date NOT NULL
colonia, estado, calle	varchar(50) NOT NULL
número interior/exterior	int NOT NULL
Entidad: AmenidadBazar	
id bazar	int NOT NULL
amenidad	varchar(50) NOT NULL
Entidad: Médico	
rfc médico	varchar(13) CHECK (CHAR_LENGTH(rfc_medico)=13 AND rfc_medico ~'^[A-Z]{4}[0-9]{6}[A-Z0-9]{3}\$')
colonia, nombre paterno, materno	varchar(50) NOT NULL
estado, apellido paterno, apellido materno	
salario	money NOT NULL
hora inicio, hora final	time NOT NULL
Entidad: TeléfonoMédico / CorreoMédico	
rfc médico	varchar(13) NOT NULL
teléfono / correo	varchar(20) y varchar(100) NOT NULL

Entidad: Seguridad	
nombre, apellido paterno, apellido materno, colonia, estado, calle	varchar(50) NOT NULL
salario	money NOT NULL
hora inicio, hora final	time NOT NULL
rfc seguridad	varchar(13) NOT NULL
numero interior, numero exterior	int NOT NULL
Entidad: Teléfono/Correo Seguridad	
rfc seguridad	varchar(13) NOT NULL
teléfono / correo	varchar(20) y varchar(100) NOT NULL
Entidad: Limpieza	
nombre, ap. paterno, ap. materno, colonia, estado, calle	varchar(50) NOT NULL
salario	money NOT NULL
hora inicio, hora final	time NOT NULL
numero interior, numero exterior	int NOT NULL
Entidad: Teléfono/Correo Limpieza	
rfc limpieza	varchar(13) NOT NULL
teléfono / correo	varchar(20) y varchar(100) NOT NULL
Entidad: Agenda	
id agenda	int NOT NULL

Entidad: Negocio	
id negocio	int NOT NULL
nombre negocio	varchar(50) NOT NULL
descripción	varchar(200) NOT NULL
precio máximo, mínimo	money NOT NULL
Entidad: Estand	
id negocio	int NOT NULL
numero estand	int NOT NULL
precio / precio total	int NOT NULL
paquete estand	varchar(50) NOT NULL
Entidad: AmenidadEstand	
numero estand	int NOT NULL
amenidad	varchar(50) NOT NULL
Entidad: Enlace/Telefono/Correo Negocio	
id negocio	int NOT NULL
enlace / teléfono / correo	varchar(100), varchar(50) y varchar(100) NOT NULL
Entidad: Emprendedor	
id negocio	int NOT NULL
rfc emprendedor	varchar(13) NOT NULL
género	varchar(1), CHECK NOT NULL
fecha nacimiento	date NOT NULL
nombre, apellido paterno, apellido materno, calle, colonia, estado	varchar(50) NOT NULL
numero interior, numero exterior	int NOT NULL



Entidad: Teléfono/Correo Emprendedor

rfc emprendedor	varchar(13) NOT NULL
teléfono / correo	varchar(50) y varchar(100) NOT NULL

Entidad: Servicio

id negocio	int NOT NULL
nombre	varchar(50)
descripción	varchar(100)
precio	money
duración	int, CHECK: duración > 0
tipo	varchar(50)

Entidad: ProductoNoPerecedero

id mercancia, id negocio	int NOT NULL
nombre	varchar(50) NOT NULL
descripción	varchar(200) NOT NULL
precio	money CHECK (precio > '0')
cantidad	int CHECK (cantidad >= 0)

Entidad: PresentacionProductoNoPerecedero

id mercancia	int NOT NULL
presentacion	varchar(50) NOT NULL

Entidad: ProductoPerecedero

id mercancia, id negocio	int NOT NULL
nombre	varchar(50) NOT NULL
precio	money CHECK (precio > '0')
descripcion	varchar(200) NOT NULL
cantidad	int CHECK (cantidad >= 0)
fecha preparacion, fecha caducidad	date NOT NULL

Entidad: PresentacionProductoPerecedero	
id mercancia	int NOT NULL
presentacion	varchar(50) NOT NULL
Entidad: Cliente	
nombre / apellido paterno / apellido materno / estado / colonia / calle	varchar(50) NOT NULL
numero interior, numero exterior	int (num exterior >= 0)
Entidad: Tarjeta	
id pago	int NOT NULL
numero tarjeta	char(16) CHECK (CHAR LENGTH(numtarjeta)=16)
vencimiento	date NOT NULL
cvv	char(3) CHECK (CHAR LENGTH(cvv)=3)
Entidad: Efectivo	
id pago	int NOT NULL
Entidad: Ticket	
id ticket / id cliente	int NOT NULL
rfc emprendedor	varchar(13) CHECK (CHAR LENGTH(rfc emprendedor)=13)
id pago tarjeta / id pago de efectivo	int NOT NULL
TrabajarMedico	
id bazar	int NOT NULL
rfc medico	varchar(13) CHECK (CHAR LENGTH(rfc medico)=13)

Entidad: TrabajarSeguridad	
id bazar	int NOT NULL
rfc seguridad	varchar(13) CHECK (CHAR LENGTH(rfc seguridad)=13)
Entidad: TrabajarLimpieza	
id bazar	int NOT NULL
rfc limpieza	varchar(13) CHECK (CHAR LENGTH(rfc limpieza)=13)
Tener	
id bazar, id agenda	int NOT NULL
modalidad	boolean NOT NULL
fecha final, fecha inicio	date NOT NULL
numero interior	int CHECK (num interior >= 0)
numero exterior	int CHECK (num exterior >= 0)
colonia / calle / estado	varchar(50) NOT NULL
Entidad: Registrar	
id agenda, num estand	int NOT NULL
fecha asistencia	date NOT NULL
Entidad: Participar	
id negocio, id bazar	int NOT NULL
Entidad: Ir	
id bazar, id cliente	int NOT NULL
Entidad: TenerServicio	
id ticket, id servicio	int NOT NULL

Entidad: TenerProductoPerecedero

id ticket, id mercancia	int NOT NULL
cantidad	int CHECK (cantidad >= 0)

Entidad: TenerProductoNoPerecedero

id ticket, id mercancia	int NOT NULL
cantidad	int CHECK (cantidad >= 0)

4.2 Procedimientos/Funciones (SP), Triggers y Disparadores

Usados para apoyar el trabajo y restricciones de la base de datos.

4.2.1. Procedimientos Almacenados

Los procedimientos almacenados (SP) son un conjunto precompilado de instrucciones almacenadas bajo un solo nombre y procesadas como una unidad. Pueden recibir parámetros a partir de los cuales realizan distintas acciones y pueden devolver datos de formas distintas. pueden hacer referencia a tablas, vistas, a funciones definidas por el usuario y a otros procedimientos almacenados.

4.2.2. Procedimientos Almacenados usados para el proyecto

- **Registra una nueva venta con tarjeta:** Permite insertar un ticket nuevo para un cliente y emprendedor dado, generando también el registro de pago con tarjeta.

```

1 CREATE OR REPLACE PROCEDURE registrar_venta_tarjeta(
2     IN p_id_cliente INT,
3     IN p_rfc_emprendedor VARCHAR(13),
4     IN p_num_tarjeta CHAR(16),
5     IN p_vencimiento DATE,
6     IN p_cvv CHAR(3),
7     OUT p_id_ticket INT
8 )
9 AS $$
10 DECLARE
11     v_id_pago_tarjeta INT;
12     v_id_pago_efectivo INT := 0;
13 BEGIN
14     -- Insertar el pago con tarjeta

```

```

15      INSERT INTO Tarjeta (id_pago, num_tarjeta, vencimiento,
16                          cvv)
17      VALUES (DEFAULT, p_num_tarjeta, p_vencimiento, p_cvv)
18      RETURNING id_pago INTO v_id_pago_tarjeta;
19
20      -- Insertar el pago ficticio en efectivo (por la fk)
21      INSERT INTO Efectivo (id_pago)
22      VALUES (DEFAULT)
23      RETURNING id_pago INTO v_id_pago_efectivo;
24
25      -- Insertar el ticket
26      INSERT INTO Ticket (id_cliente, rfc_emprendedor,
27                        id_pago_tarjeta, id_pago_efectivo)
28      VALUES (p_id_cliente, p_rfc_emprendedor,
29              v_id_pago_tarjeta, v_id_pago_efectivo)
30      RETURNING id_ticket INTO p_id_ticket;
31  END;
```

- **Actualizar cantidad de un producto no perecedero:** Permite modificar la cantidad de stock de un producto no perecedero.

```

1  CREATE OR REPLACE PROCEDURE
2      actualizar_stock_producto_no_perecedero(
3      IN p_id_mercancia INT,
4      IN p_nueva_cantidad INT
5      )
6  AS $$
7  BEGIN
8      UPDATE ProductoNoPerecedero
9      SET cantidad = p_nueva_cantidad
10     WHERE id_mercancia = p_id_mercancia;
11  END;
```

- **Función que actualiza el precio máximo o mínimo de la mercancía; servicio o productos (productos perecederos y no perecederos)**

```

1  CREATE OR REPLACE FUNCTION actualiza_maximo_minimo()
2  RETURNS TRIGGER AS $$
3  DECLARE
4      maximo_actual money;
5      minimo_actual money;
6      maximo_nuevo money;
7      minimo_nuevo money;
8  BEGIN
9
10     IF TG_OP = 'DELETE' THEN
11
12         maximo_actual = (SELECT precio_maximo FROM negocio
13                         WHERE id_negocio = OLD.id_negocio);
```

```
13     minimo_actual = (SELECT precio_minimo FROM negocio WHERE
14                       id_negocio = OLD.id_negocio);
15
16     IF OLD.precio = maximo_actual THEN
17         SELECT MAX(precio) INTO maximo_nuevo FROM (
18             SELECT precio FROM servicio WHERE
19                 id_negocio = OLD.id_negocio
20             UNION ALL
21             SELECT precio FROM productonoperecedero
22                 WHERE id_negocio = OLD.id_negocio
23             UNION ALL
24             SELECT precio FROM productoperecedero
25                 WHERE id_negocio = OLD.id_negocio
26         );
27         UPDATE negocio SET precio_maximo = maximo_nuevo
28             WHERE id_negocio = OLD.id_negocio;
29     END IF;
30
31     IF OLD.precio = minimo_actual THEN
32         SELECT MIN(precio) INTO minimo_nuevo FROM (
33             SELECT precio FROM servicio WHERE
34                 id_negocio = OLD.id_negocio
35             UNION ALL
36             SELECT precio FROM productonoperecedero
37                 WHERE id_negocio = OLD.id_negocio
38             UNION ALL
39             SELECT precio FROM productoperecedero
40                 WHERE id_negocio = OLD.id_negocio
41         );
42         UPDATE negocio SET precio_minimo = minimo_nuevo
43             WHERE id_negocio = OLD.id_negocio;
44     END IF;
45
46     RETURN OLD;
47 END IF;
48
49     maximo_actual = (SELECT precio_maximo FROM negocio
50                       WHERE id_negocio = NEW.id_negocio);
51     minimo_actual = (SELECT precio_minimo FROM negocio WHERE
52                       id_negocio = NEW.id_negocio);
53
54     IF TG_OP = 'UPDATE' THEN
55
56         IF NEW.precio = maximo_actual THEN
57             SELECT MAX(precio) INTO maximo_nuevo FROM (
58                 SELECT precio FROM servicio WHERE
59                     id_negocio = NEW.id_negocio
60                 UNION ALL
61                 SELECT precio FROM productonoperecedero
```

```
50         WHERE id_negocio = NEW.id_negocio
51     UNION ALL
52     SELECT precio FROM productoperecedero
53         WHERE id_negocio = NEW.id_negocio
54 );
55 UPDATE negocio SET precio_maximo = maximo_nuevo
56     WHERE id_negocio = NEW.id_negocio;
57 END IF;
58
59 IF NEW.precio = minimo_actual THEN
60     SELECT MIN(precio) INTO minimo_nuevo FROM (
61         SELECT precio FROM servicio WHERE
62             id_negocio = NEW.id_negocio
63         UNION ALL
64         SELECT precio FROM productonoperecedero
65             WHERE id_negocio = NEW.id_negocio
66         UNION ALL
67         SELECT precio FROM productoperecedero
68             WHERE id_negocio = NEW.id_negocio
69     );
70     UPDATE negocio SET precio_maximo = minimo_nuevo
71         WHERE id_negocio = NEW.id_negocio;
72 END IF;
73
74 RETURN NEW;
75 END IF;
76
77 IF TG_OP = 'INSERT' THEN
78     IF NEW.precio > maximo_actual THEN
79         UPDATE negocio SET precio_maximo = NEW.precio
80             WHERE id_negocio = NEW.id_negocio;
81     END IF;
82
83     IF NEW.precio < minimo_actual THEN
84         UPDATE negocio SET precio_minimo = NEW.precio
85             WHERE id_negocio = NEW.id_negocio;
86     END IF;
87
88 RETURN NEW;
89 END IF;
90
91 RETURN NULL;
92 END;
```

```
1 CREATE OR REPLACE FUNCTION revisar_caducidad()
2 RETURNS TRIGGER AS $$
```

```
3 DECLARE
4     fecha_comp DATE := '05/05/2024';
5 BEGIN
6     IF NEW.fecha_caducidad < fecha_comp THEN
7         RAISE EXCEPTION 'El producto que se quiere insertar
8             caduc en %', NEW.fecha_caducidad;
9     END IF;
10    RETURN NEW;
11 END;
```

4.3 Triggers y Disparadores

Los **Triggers**, son funciones que se ejecutan de forma automática en respuesta a ciertos eventos que ocurren en la BD. Para identificar el tipo de operación que disparó el trigger empleamos `inserting`, `updating` y `deleting`.

Cuando los SP se ocupan para validar datos o controlar la integridad de una base de datos se ejecutan a través de **disparadores**.

4.3.1. Triggers usados para el proyecto

- *Triggers de inserción a nivel de fila (For Each Row)*: El trigger se dispara una vez por cada fila ingresada sobre la tabla especificada.
- **Trigger que lanza una excepción si al insertar un producto perecedero ya está caducado.** Se aplica sobre la tabla `productoperecedero`

```
1 SELECT E.id_pago AS id_pago_tarjeta, 'Tarjeta' AS
   tipo_pago
2 FROM Tarjeta E
3 UNION
4 SELECT E.id_pago AS id_pago_efectivo, 'Efectivo' AS
   tipo_pago
5 FROM Efectivo E;
```

- **Triggers que actualizan el precio máximo o mínimo de un negocio cuando se inserta un servicio, producto perecedero o producto no perecedero.**

```
1 CREATE TRIGGER actualiza_maximo_minimo_servicio
2     AFTER INSERT OR UPDATE OF precio OR DELETE
3     ON servicio
4     FOR EACH ROW
5     EXECUTE FUNCTION actualiza_maximo_minimo();
```

La herramienta utilizada fue Mockaroo.

[illegible]

```

51
52
53
54 insert into AmenidadBazar (id_bazar, amenidad) values ('1', 'Estacionamiento');
55 insert into AmenidadBazar (id_bazar, amenidad) values ('2', 'Carpas');
56 insert into AmenidadBazar (id_bazar, amenidad) values ('3', 'Toma de corrientes');
57 insert into AmenidadBazar (id_bazar, amenidad) values ('4', 'Wifi');
58 insert into AmenidadBazar (id_bazar, amenidad) values ('5', 'Baño');
59 insert into AmenidadBazar (id_bazar, amenidad) values ('6', 'Salida de emergencia');
60 insert into AmenidadBazar (id_bazar, amenidad) values ('7', 'Seguridad');
61 insert into AmenidadBazar (id_bazar, amenidad) values ('8', 'Estacionamiento');
62 insert into AmenidadBazar (id_bazar, amenidad) values ('9', 'Carpas');
63 insert into AmenidadBazar (id_bazar, amenidad) values ('10', 'Toma de corrientes');
64 insert into AmenidadBazar (id_bazar, amenidad) values ('11', 'Baño');
65 insert into AmenidadBazar (id_bazar, amenidad) values ('12', 'Baño');
66 insert into AmenidadBazar (id_bazar, amenidad) values ('13', 'Salida de emergencia');
67 insert into AmenidadBazar (id_bazar, amenidad) values ('14', 'Seguridad');
68 insert into AmenidadBazar (id_bazar, amenidad) values ('15', 'Estacionamiento');
69 insert into AmenidadBazar (id_bazar, amenidad) values ('16', 'Carpas');
70 insert into AmenidadBazar (id_bazar, amenidad) values ('17', 'Toma de corrientes');
71 insert into AmenidadBazar (id_bazar, amenidad) values ('18', 'Wifi');
72 insert into AmenidadBazar (id_bazar, amenidad) values ('19', 'Baño');
73 insert into AmenidadBazar (id_bazar, amenidad) values ('20', 'Salida de emergencia');
74 insert into AmenidadBazar (id_bazar, amenidad) values ('21', 'Seguridad');
75 insert into AmenidadBazar (id_bazar, amenidad) values ('22', 'Estacionamiento');
76 insert into AmenidadBazar (id_bazar, amenidad) values ('23', 'Carpas');
77 insert into AmenidadBazar (id_bazar, amenidad) values ('24', 'Toma de corrientes');
78 insert into AmenidadBazar (id_bazar, amenidad) values ('25', 'Wifi');
79 insert into AmenidadBazar (id_bazar, amenidad) values ('26', 'Baño');
80 insert into AmenidadBazar (id_bazar, amenidad) values ('27', 'Salida de emergencia');
81 insert into AmenidadBazar (id_bazar, amenidad) values ('28', 'Seguridad');
82 insert into AmenidadBazar (id_bazar, amenidad) values ('29', 'Estacionamiento');
83 insert into AmenidadBazar (id_bazar, amenidad) values ('30', 'Carpas');

```

[illegible]

6 Consultas

Resultado de la evaluación de las 15 consultas; indicadores y/o conclusiones

1. Bazares registrados con fechas y ubicación

```
1 SELECT id_bazar, modalidad, fecha_inicio, fecha_final,
   colonia, estado, calle
2 FROM bazar;
```

	id_bazar	modalidad	fecha_inicio	fecha_final	colonia	estado	calle
1	1	[v]	2024-06-10	2024-06-12	El Porvenir	Chiapas	Lawn
2	2	[v]	2024-06-09	2024-06-12	El Calvario	Oaxaca	Spohn
3	3	[]	2024-06-10	2024-06-13	Revolucion Verde	Tamaulipas	Annamark
4	4	[v]	2024-06-11	2024-06-12	Buenos Aires	Jalisco	Bayside
5	5	[]	2024-06-08	2024-06-13	5 de Mayo	Tamaulipas	Forest Dale
6	6	[v]	2024-06-09	2024-06-12	Magdaleno Aguilar	Tamaulipas	Oxford
7	7	[]	2024-06-08	2024-06-13	Progreso	Chihuahua	Mcbride
8	8	[]	2024-06-09	2024-06-12	Ejidal	Guerrero	Menomonie
9	9	[v]	2024-06-09	2024-06-13	Altavista	Chihuahua	New Castle
10	10	[v]	2024-06-09	2024-06-13	Niños Heroes	Jalisco	Arrowood
11	11	[]	2024-06-10	2024-06-13	Lazaro Cardenas	Guerrero	Fair Oaks
12	12	[v]	2024-06-10	2024-06-12	Lazaro Cardenas	Sinaloa	Meadow Ridge
13	13	[]	2024-06-10	2024-06-13	El Calvario	Puebla	High Crossing
14	14	[v]	2024-06-08	2024-06-12	San Isidro	Jalisco	Eastlawn
15	15	[v]	2024-06-09	2024-06-13	San Francisco	Tamaulipas	Sutteridge

2. Médicos registrados con RFC, horario y salario

```
1 SELECT rfc_medico, nombre, apellido_paterno, apellido_materno
   , salario, hora_inicio, hora_final
2 FROM medico;
```

	rfc_medico	nombre	apellido_paterno	apellido_materno	salario	hora_inicio	hora_final
1	ZOWK748301G47	Ingamar	Freeburn	Eldin	\$11,510.06	08:44:00	19:47:00
2	LTLW023831T49	Gene	Dodimead	Studd	\$5,898.28	09:50:00	18:29:00
3	NFOD925347H01	Thomasina	Goves	Twitchings	\$5,201.77	10:56:00	19:39:00
4	KFKU583116D20	Hurleigh	Dorrity	Sesons	\$15,516.91	10:31:00	18:26:00
5	EPYO246970A64	Rey	Bridger	Di Nisco	\$18,436.61	08:17:00	18:38:00
6	YETB818565K00	Else	Scafe	Hallmark	\$16,701.61	10:59:00	19:50:00
7	EYYT186317Q64	Elizabet	Garritley	Pietron	\$11,144.34	08:51:00	19:39:00
8	HMYT919756H20	Marijn	Burris	Gianelli	\$16,637.20	08:41:00	19:06:00
9	GCKV535157T94	Sybila	Yoslowitz	Sea	\$15,886.64	10:58:00	18:33:00
10	OVSU057327R11	Jacques	Halsho	Copeman	\$7,904.53	10:11:00	19:19:00
11	HXMR184611W77	Marion	Drillingcourt	Riggoll	\$7,382.27	10:20:00	18:24:00
12	UREV646465T22	Kerby	Kenninghan	Dymond	\$19,806.10	10:16:00	19:04:00
13	KBNY514327H69	Constantine	Rassell	McMenamy	\$6,894.83	09:56:00	18:16:00
14	YWAR881944U41	Nealon	Hellwing	Laible	\$8,901.74	08:59:00	18:22:00
15	WHXN419235E82	Jedd	Tocque	Liggett	\$13,968.53	09:41:00	19:50:00

3. Negocios con sus precios y descripción

```
1 SELECT nombre_negocio, descripcion, precio_minimo,
   precio_maximo
2 FROM negocio;
```

	A1 nombre_negocio	A2 descripcion	123 precio_minimo	123 precio_maximo
1	Livepath	lectus in quam fringilla rhoncus mauris enim leo rhoncus sed	\$5.97	\$1,657.30
2	Gigazoom	sollicitudin ut suscipit a feugiat et eros vestibulum ac est lacinia nisi venenatis tristique	\$4.22	\$1,041.45
3	Voonder	vestibulum velit id pretium iaculis diam erat fermentum justo nec condimentum neque	\$7.37	\$952.96
4	Gabitone	quisque erat eros viverra eget congue eget semper rutrum nulla nunc purus phasellus	\$5.77	\$1,434.04
5	Dynabox	vehicula condimentum curabitur in libero ut massa volutpat convallis morbi	\$6.11	\$917.97
6	Jabberstorm	eleifend luctus ultricies eu nibh quisque id justo sit amet sapien dignissim	\$1.11	\$1,246.36
7	Realbridge	purus aliquet at feugiat non pretium quis lectus suspendisse potenti in eleifend quam a	\$8.56	\$1,846.09
8	Jaxnation	ac nulla sed vel enim sit amet nunc viverra dapibus nulla suscipit ligula	\$3.74	\$1,328.70
9	Fivebridge	ante ipsum primis in faucibus orci luctus et ultrices posuere	\$4.01	\$1,506.75
10	Ooba	potenti nullam porttitor lacus at turpis donec posuere metus vitae ipsum aliquam non mauris morbi	\$8.37	\$1,197.84
11	Twitterbridge	leo odio condimentum id luctus nec molestie sed justo pellentesque viverra pede	\$3.63	\$770.06
12	Tazz	sem fusce consequat nulla nisl nunc nisl dui bibendum felis sed interdum venenatis turpis enim	\$4.52	\$1,015.47
13	Olooo	faucibus cursus urna ut tellus nulla ut erat id mauris vulputate elementum nullam	\$2.60	\$810.57
14	Vitz	semper sapien a libero nam dui proin leo odio porttitor id consequat in consequat ut	\$6.91	\$1,592.96
15	Izio	enim leo rhoncus sed vestibulum sit amet cursus id turpis integer	\$7.90	\$1,058.86

4. Productos no perecederos con cantidad y precio

```
1 SELECT nombre, descripcion, precio, cantidad
2 FROM productonoperecedero
3 ORDER BY precio DESC;
```

	A1 nombre	A2 descripcion	123 precio	123 cantidad
1	Organic Baby Carrots	ut massa quis augue luctus tincidunt nulla mollis molestie lorem	\$999.20	43
2	Spicy Thai Coconut Soup	magnis dis parturient montes nascetur ridiculus mus vivamus vestibulum sagittis sapien cum sociis	\$998.65	69
3	Foldable Yoga Mat	eget congue eget semper rutrum nulla nunc purus phasellus in felis donec semper	\$995.61	21
4	Vegetable Quinoa Bowl	lectus in est risus auctor sed tristique in tempus sit amet sem	\$994.42	91
5	Hiking Gaiters	sit amet cursus id turpis integer aliquet massa id lobortis convallis tortor	\$985.91	63
6	Sweet Cherry Tomatoes	felis fusce posuere felis sed lacus morbi sem mauris laoreet ut	\$984.16	26
7	Organic Cereal Bars	eros vestibulum ac est lacinia nisi venenatis tristique fusce congue diam id ornare	\$982.68	16
8	Sooji (Semolina)	turpis enim blandit mi in porttitor pede justo eu massa donec dapibus dui at velit	\$979.69	47
9	Portable Camping Shower	in porttitor pede justo eu massa donec dapibus dui at velit eu est	\$977.90	66
10	Classic Leather Wallet	at nunc commodo placerat praesent blandit nam nulla integer pede justo lacinia	\$975.17	40
11	Pistachios	curae mauris viverra diam vitae quam suspendisse potenti nullam porttitor lacus	\$974.40	45
12	Portable Hammock	pede morbi porttitor lorem id ligula suspendisse ornare consequat lectus in	\$972.93	57
13	Pet Food Storage Container	ipsum primis in faucibus orci luctus et ultrices posuere cubilia	\$970.20	60
14	Electric Egg Cooker	fusce lacus purus aliquet at feugiat non pretium quis lectus suspendisse potenti	\$970.14	57
15	Antique Style Clock	imperdiet et commodo vulputate justo in blandit ultrices enim lorem ipsum dolor sit	\$967.86	57

5. Negocios con servicios registrados

```
1 SELECT s.nombre AS nombre_servicio, s.descripcion, s.precio,
   s.duracion, s.tipo, n.nombre_negocio
2 FROM servicio s
3 JOIN negocio n ON s.id_negocio = n.id_negocio;
```

	A1 nombre_servicio	A2 descripcion	123 precio	123 duracion	A4 tipo	A5 nombre_negocio
1	Bodyweight Resistance Bands	mi integer ac neque dui bibendum morbi non quam nec dui	\$328.07	10	Renta de juegos	Chatterpoint
2	Wireless Charger	lacus at velit vivamus vel nulla eget eros elementum pellentesque quisque porta volutpat erat quisque	\$345.27	78	Renta de juegos	Jetpulse
3	Outdoor Inflatable Pool	tempus vel pede morbi porttitor lorem id ligula suspendisse ornare consequat lectus	\$481.72	55	Renta de juegos	Talane
4	Coconut Milk Yogurt	aliquet maecenas leo odio condimentum id luctus nec molestie sed justo pellentesque	\$949.38	87	Clases	Flipstorm
5	Travel Jewelry Organizer	nulla tellus in sagittis dui vel nisl dui ac nibh fusce	\$383.33	43	Renta de juegos	Podcat
6	Mango Chilli Sauce	plates dictumst aliquam augue quam sollicitudin vitae consectetur eget rutrum at lorem integer	\$177.32	63	Masajes	Jayo
7	Chocolate Dipped Fruit	vel lectus in quam fringilla rhoncus mauris enim leo rhoncus sed	\$568.49	90	Clases	Fiveclub
8	Smoky BBQ Jackfruit	ac leo pellentesque ultrices mattis odio donec vitae nisi nam ultrices libero	\$274.73	29	Renta de juegos	Vimbo
9	Sweet Chili Dipping Sauce	adipiscing elit proin interdum mauris non ligula pellentesque ultrices phasellus id sapien in	\$956.05	36	Renta de sombrillas	Meevee
10	Luxury Bath Robe	mattis odio donec vitae nisi nam ultrices libero non mattis pulvinar nulla pede	\$136.32	34	Renta de bicicleta	Oyope
11	Tennis Racket	varius nulla facilisi cras non velit nec nisi vulputate nonummy maecenas	\$411.34	83	Renta de bicicleta	Pikoboo
12	Indoor Plants	in hac habitasse platea dictumst aliquam augue quam sollicitudin vitae consectetur eget rutrum	\$768.36	95	Clases	Oodoo
13	Solar Garden Lights	diam neque vestibulum eget vulputate ut ultrices vel augue vestibulum ante	\$211.15	108	Clases	Skalith
14	Organic Almonds	gravida sem praesent id massa id nisl venenatis lacinia aenean sit	\$374.43	55	Clases	Divape
15	Non-Stick Crepe Pan	at feugiat non pretium quis lectus suspendisse potenti in eleifend quam a odio	\$855.51	13	Renta de juegos	Twitterwire



6. Emprendedores con género

```
1 SELECT rfc_emprendedor, genero
2 FROM emprendedor;
```

	A-Z rfc_emprendedor	A-Z genero
1	ZKWY728219R35	F
2	WFAV844052A19	M
3	APAV946540Y95	M
4	UUQH732122X04	F
5	NEGC796169J75	M
6	ZMKI005860X57	F
7	JYKB403989Y51	M
8	LYBI252617R58	F
9	THRI600017B52	F
10	XNFF405287W54	F
11	EKCZ112775M59	M
12	WTUH599416K14	M
13	EBXY232165F00	M
14	EEYB668635R93	F
15	JBES982631R96	F

7. Correos de médicos agrupados por RFC

```
1 SELECT rfc_medico, STRING_AGG(correo, ', ') AS correos
2 FROM correomedico
3 GROUP BY rfc_medico;
```

	A-Z rfc_medico	A-Z correos
1	HEMK569016A43	hemk569016a43@hotmail.com
2	MFHA304750R37	mfha304750r37@hotmail.com
3	TVOC717625D23	tvoc717625d23@hotmail.com
4	AYAD307881C77	ayad307881c77@gmail.com
5	XSSU446419V93	xssu446419v93@live.com
6	YETB818565K00	yetb818565k00@hotmail.com, yetb818565k00@ciencias.com
7	BNOZ977515G74	bnoz977515g74@hotmail.com
8	DMUE376369N04	dmue376369n04@gmail.com
9	VVAY438036M14	vway438036m14@hotmail.com
10	GAMS873211J62	gams873211j62@hotmail.com
11	QIRR902219I70	qirr902219i70@outlook.com
12	RBGP269023U55	rbgp269023u55@hotmail.com
13	LQXM972477I20	lqxm972477i20@live.com
14	IXDK560672J66	ixdk560672j66@hotmail.com
15	YLDJ150157M62	yldj150157m62@outlook.com

8. Número de médicos por rango salarial

```
1 SELECT CASE
2     WHEN salario < 10000 THEN 'Menos de 10k'
3     WHEN salario BETWEEN 10000 AND 20000 THEN '10k - 20k'
4     ELSE 'Mas de 20k'
5 END AS rango_salarial,
6 COUNT(*) AS total_medicos
7 FROM medico
8 GROUP BY rango_salarial;
```

	A-Z rango_salarial	123 total_medicos
1	10k - 20k	191
2	Menos de 10k	109

9. Servicios con precio mayor al promedio

```
1 SELECT nombre, precio
2 FROM servicio
3 WHERE precio > (SELECT AVG(precio) FROM servicio);
```

	A-Z nombre ▼	123 precio ▼
1	Coconut Milk Yogurt	\$949.38
2	Chocolate Dipped Fruit	\$568.49
3	Sweet Chili Dipping Sauce	\$956.05
4	Indoor Plants	\$768.36
5	Non-Stick Crepe Pan	\$855.51
6	Printed Maxi Skirt	\$680.15
7	Nature Explorer Lens Kit	\$965.32
8	Herb Drying Rack	\$962.67
9	Deep Tissue Massage Gun	\$859.51
10	Collapsible Storage Crates	\$620.82
11	Sriracha Sauce	\$719.55
12	Travel Pillow with Memory Foam	\$523.10
13	Frozen Mixed Berries	\$593.46
14	Vanilla Ice Cream	\$963.75
15	Curried Lentil Salad	\$992.89

10. Número de amenidades por bazar

```
1 SELECT id_bazar, COUNT(*) AS total_amenidades
2 FROM ameniadbazar
3 GROUP BY id_bazar;
```

	123 id_bazar	123 total_amenidades
1	42	4
2	29	4
3	4	4
4	34	4
5	41	4
6	46	4
7	40	4
8	43	4
9	32	4
10	10	4
11	9	4
12	7	4
13	35	4
14	45	4
15	38	4

11. Paquetes de estand disponibles

```
1 SELECT DISTINCT paquete_estand
2 FROM estand;
```

	A-Z paquete_estand
1	premium
2	emprendedor
3	basico

12. Negocios con sus enlaces de contacto

```
1 SELECT n.nombre_negocio, e.enlace
2 FROM negocio n
3 JOIN enlacenegocio e ON n.id_negocio = e.id_negocio;
```

	A-Z nombre_negocio	A-Z enlace
1	Livepath	https://deliciousdays.com
2	Yodoo	https://berkeley.edu
3	Jayo	http://cnn.com
4	Jabbercube	https://youtu.be
5	Roexo	https://businessinsider.com
6	Rhybox	https://oaic.gov.au
7	Gigazoom	https://weibo.com
8	Voonder	https://hexun.com
9	Gabtune	https://about.me
10	Dynabox	http://opensource.org
11	Jabberstorm	https://paypal.com
12	Brainsphere	https://sogou.com
13	Realbridge	https://liveinternet.ru
14	Jaxnation	https://networkadvertising.org
15	Fivebridge	https://gmpg.org

13. Servicios por tipo y duración promedio

```
1 SELECT tipo, AVG(duracion) AS duracion_promedio
2 FROM servicio
3 GROUP BY tipo;
```

	A-Z tipo	123 duracion_promedio
1	Clases	63.1851851852
2	Renta de sombrillas	63.4683544304
3	Renta de juegos	65.8
4	Renta de bicicleta	54.2558139535
5	Masajes	58

14. Bazar más largo en duración

```
1 SELECT id_bazar, fecha_inicio, fecha_final, (fecha_final -
2     fecha_inicio) AS dias_duracion
3 FROM bazar
4 ORDER BY dias_duracion DESC
5 LIMIT 1;
```

	123 id_bazar	fecha_inicio	fecha_final	123 dias_duracion
1	5	2024-06-08	2024-06-13	5

15. Emprendedores con más de un contacto

```

1 SELECT rfc_emprendedor
2 FROM (
3     SELECT rfc_emprendedor FROM telefonoemprendedor
4     UNION
5     SELECT rfc_emprendedor FROM correoemprendedor
6 ) t
7 GROUP BY rfc_emprendedor
8 HAVING COUNT(*) > 1;

```

	A-Z nombre_negocio	A-Z enlace
1	Livepath	https://deliciousdays.com
2	Yodoo	https://berkeley.edu
3	Jayo	http://cnn.com
4	Jabbercube	https://youtu.be
5	Roexo	https://businessinsider.com
6	Rhybox	https://oaic.gov.au
7	Gigazoom	https://weibo.com
8	Voonder	https://hexun.com
9	Gabtune	https://about.me
10	Dynabox	http://opensource.org
11	Jabberstorm	https://paypal.com
12	Brainsphere	https://sogou.com
13	Realbridge	https://liveinternet.ru
14	Jaxnation	https://networkadvertising.org
15	Fivebridge	https://gmpg.org

16. Listado completo de productos disponibles en el bazar (perecederos y no perecederos)

```

1 SELECT nombre, descripcion, precio, cantidad, 'Perecedero' AS
   tipo
2 FROM ProductoPerecedero
3 UNION
4 SELECT nombre, descripcion, precio, cantidad, 'No Perecedero'
   AS tipo
5 FROM ProductoNoPerecedero;
```

	nombre	descripcion	precio	cantidad	tipo
1	Active Racerback Tank	parturiunt montes nascetur ridiculus mus etiam vel augue vestibulum rutrum rutrum	\$743.87	46	No Perecedero
2	Adjustable Dog Harness	tristique fusce congue diam id ornare imperdiet sapien urna pretium nisl ut volutpat sapien	\$857.10	82	No Perecedero
3	Adjustable Garden Rake	congue etiam justo etiam pretium iaculis justo in hac habitasse platea	\$517.03	30	No Perecedero
4	Adjustable Garden Rake	justo morbi ut odio cras mi pede malesuada in imperdiet et commodo vulputate justo in	\$249.45	95	No Perecedero
5	Adjustable Laptop Desk	nulla justo aliquam quis turpis eget elit sodales scelerisque mauris sit amet eros	\$887.33	71	No Perecedero
6	Adjustable Resistance Bands	et ultrices posuere cubilla curae mauris viverra diam vitae quam	\$500.31	97	No Perecedero
7	Aged White Cheddar Popcorn	nibh in hac habitasse platea dictumst aliquam augue quam sollicitudin vitae consectetur eget rutrum at	\$269.52	5	No Perecedero
8	Air Fryer Oven	potenti nullam porttitor lacus at turpis donec posuere metus vitae ipsum	\$516.58	16	No Perecedero
9	Air Fryer Oven	sed accumsan felis ut at dolor quis odio consequat varius integer	\$826.02	14	No Perecedero
10	Almond Butter	magna bibendum imperdiet nullam orci pede venenatis non sodales sed tincidunt	\$293.69	69	No Perecedero
11	Almond Butter Granola Bars	quisque porta volutpat erat quisque erat eros viverra eget congue eget	\$826.44	93	No Perecedero
12	Almond Joy Bars	amet nunc viverra dapibus nulla suscipit ligula in lacus curabitur at ipsum	\$595.83	45	No Perecedero
13	Andouille Sausage	massa id lobortis convallis tortor risus dapibus augue vel accumsan tellus nisi eu	\$609.58	53	No Perecedero
14	Antique Style Clock	imperdiet et commodo vulputate justo in blandit ultrices enim lorem ipsum dolor sit	\$967.86	57	No Perecedero
15	Appetizer - Cheese Bites	ante ipsum primis in faucibus orci luctus et ultrices posuere cubilla curae dui	\$116.71	95	Perecedero

17. Historial de ventas realizadas en un periodo especifico

```

1 SELECT T.id_ticket, T.id_cliente, T.rfc_emprendedor, T.
   id_pago_tarjeta, T.id_pago_efectivo
2 FROM Ticket T
3 JOIN TenerProductoPerecedero TP ON T.id_ticket = TP.id_ticket
4 JOIN ProductoPerecedero P ON TP.id_mercancia = P.id_mercancia
5 WHERE P.fecha_preparacion BETWEEN '2024-01-01' AND '
   2024-12-31';
```

	id_ticket	id_cliente	rfc_emprendedor	id_pago_tarjeta	id_pago_efectivo
1	908	637	ZNIU619815H06	[NULL]	454
2	97	978	TPNW693575O58	49	[NULL]
3	977	439	KDKJ948229C60	489	[NULL]
4	490	778	IMLB084198T30	[NULL]	245
5	645	290	CSOL891177Q22	323	[NULL]
6	301	168	EBXY232165F00	151	[NULL]
7	813	687	IVLM787422O29	407	[NULL]
8	685	193	BXEW822870N95	343	[NULL]
9	341	193	WVIR684493H08	171	[NULL]
10	216	251	DGXO986112X99	[NULL]	108
11	192	279	VPNR329711I05	[NULL]	96
12	328	622	ZNIU619815H06	[NULL]	164
13	115	767	BXEW822870N95	58	[NULL]
14	15	33	KVWL853224M75	8	[NULL]
15	435	354	ODMQ264273O49	218	[NULL]

18. Reporte de ingresos por tipo de pago (es decir, tarjeta o efectivo)

```

1 SELECT
2     CASE
3         WHEN T.id_pago_tarjeta IS NOT NULL THEN 'Tarjeta'
4         WHEN T.id_pago_efectivo IS NOT NULL THEN 'Efectivo'
5         ELSE 'Otro'
6     END AS tipo_pago,
7     COUNT(*) AS cantidad_transacciones
8 FROM Ticket T
9 GROUP BY tipo_pago;

```

	A-Z tipo_pago ▼	123 cantidad_transacciones ▼
1	Efectivo	500
2	Tarjeta	500

19. Ventas realizadas por un emprendedor específico

```

1 SELECT T.id_ticket, C.nombre, C.apellido_paterno, T.
   rfc_emprendedor
2 FROM Ticket T
3 JOIN Cliente C ON T.id_cliente = C.id_cliente
4 WHERE T.rfc_emprendedor = 'RGVJ228467U22';

```

	123 id_ticket ▼	A-Z nombre ▼	A-Z apellido_paterno ▼	A-Z rfc_emprendedor ▼
1	226	Ciro	Kanwell	RGVJ228467U22
2	916	Freda	Hawkshaw	RGVJ228467U22
3	468	Dorian	Blunt	RGVJ228467U22
4	694	Zarla	Lafranconi	RGVJ228467U22
5	132	Kipp	Wittrington	RGVJ228467U22

20. Productos con baja rotacion (ventas menores a 5 unidades)

```

1 SELECT P.id_mercancia, P.nombre, COALESCE(SUM(TNP.cantidad),
2      0) AS total_vendido
3 FROM ProductoNoPerecedero P
4 LEFT JOIN TenerProductoNoPerecedero TNP ON P.id_mercancia =
5      TNP.id_mercancia
6 GROUP BY P.id_mercancia, P.nombre
7 HAVING SUM(TNP.cantidad) < 5;

```

	id_mercancia	nombre	total_vendido
1	92	Aged White Cheddar Popcorn	4
2	188	Scented Candle Set	3
3	91	Sriracha Sauce	3
4	105	Samsung Galaxy Smartwatch	4
5	89	Jasmine Rice	4
6	13	Woven Storage Baskets	1
7	155	Portable Dog Water Bottle	4
8	75	Harvest Grain Salad	4
9	186	Spicy Tuna Rolls	1
10	114	Apple Sauce	1
11	165	Chili Lime Corn Chips	1
12	39	Cork Yoga Mat	4
13	118	Feta Cheese Crumbles	4
14	32	Portable USB-C Monitor	4
15	150	Hiking Gaiters	4

21. Informacion de contacto de los clientes

```

1 SELECT id_cliente, nombre, apellido_paterno, apellido_materno
2      , calle, colonia, estado
3 FROM Cliente;

```

	id_cliente	nombre	apellido_paterno	apellido_materno	calle	colonia	estado
1	1	Mildrid	Iglesia	Devote	Ilene	Benito Juarez	Mexico
2	2	Erv	Hought	Chitty	Farmco	Niños Heroes	Tamaulipas
3	3	Curran	Savins	Lilian	Cambridge	El Aguacate	Guerrero
4	4	Stafford	Bescoby	Peacock	Mayfield	Independencia	Oaxaca
5	5	Zita	Dimnage	Becerro	Fairview	Benito Juarez	Tamaulipas
6	6	Alethea	Bagworth	Kelloway	Ridge Oak	Fovissste	Guerrero
7	7	Farrah	Swayton	Easterby	Clove	San Lorenzo	Mexico
8	8	Addi	Heales	Moscrop	Garrison	Santo Tomas	Mexico
9	9	Bordie	Philps	Tomowicz	Laurel	Hidalgo	Chihuahua
10	10	Meredith	Winspurr	Moules	Upham	Libertad	Sonora
11	11	Robyn	Getsham	Coveney	Jackson	Cuauhtemoc	Tamaulipas
12	12	Michel	Chastanet	Critcher	Autumn Leaf	Vicente Guerrero	Guerrero
13	13	Murdoch	Berrey	Davidovic	Kim	Independencia	Tamaulipas
14	14	Faustine	Pauly	Wheeler	Basil	Del Valle	Tamaulipas
15	15	Mag	Deere	Harradence	Union	San Sebastian	Mexico

22. Productos próximos a caducar (menos de 7 días)

```
1 SELECT nombre, fecha_caducidad
2 FROM ProductoPerecedero
3 WHERE fecha_caducidad <= CURRENT_DATE + INTERVAL '7 days';
```

	A-Z nombre	fecha_caducidad
1	Cookie Chocolate Chip With	2024-07-19
2	Cheese - Fontina	2025-01-06
3	Wine - Zonnebloem Pinotage	2025-02-12
4	Pastrami	2025-04-24
5	Shrimp - 31/40	2024-09-24
6	Corn Shoots	2024-10-21
7	Table Cloth 91x91 Colour	2024-06-20
8	Pie Filling - Cherry	2025-06-09
9	Wine - Barossa Valley Estate	2025-03-05
10	Swiss Chard - Red	2024-06-16
11	Pepper - Orange	2025-05-22
12	Contreau	2024-10-03
13	Greens Mustard	2024-10-29
14	Zucchini - Green	2024-11-04
15	Munchies Honey Sweet Trail Mix	2025-03-23

23. Servicios ofrecidos por cada negocio

```

1 SELECT N.nombre_negocio, S.nombre AS servicio, S.precio
2 FROM Negocio N
3 JOIN Servicio S ON N.id_negocio = S.id_negocio;

```

	A-Z nombre_negocio	A-Z servicio	123 precio
1	Chatterpoint	Bodyweight Resistance Bands	\$328.07
2	Jetpulse	Wireless Charger	\$345.27
3	Talane	Outdoor Inflatable Pool	\$481.72
4	Flipstorm	Coconut Milk Yogurt	\$949.38
5	Podcat	Travel Jewelry Organizer	\$383.33
6	Jayo	Mango Chilli Sauce	\$177.32
7	Fiveclub	Chocolate Dipped Fruit	\$568.49
8	Vimbo	Smoky BBQ Jackfruit	\$274.73
9	Meevee	Sweet Chili Dipping Sauce	\$956.05
10	Oyope	Luxury Bath Robe	\$136.32
11	Pixoboo	Tennis Racket	\$411.34
12	Oodoo	Indoor Plants	\$768.36
13	Skalith	Solar Garden Lights	\$211.15
14	Divape	Organic Almonds	\$374.43
15	Twitterwire	Non-Stick Crepe Pan	\$855.51

24. Emprendedores mayores de 30 años

```
1 SELECT nombre, apellido_paterno, apellido_materno,
   fecha_nacimiento
2 FROM Emprendedor
3 WHERE fecha_nacimiento <= CURRENT_DATE - INTERVAL '30 years';
```

	A-Z nombre	A-Z apellido_paterno	A-Z apellido_materno	🕒 fecha_nacimiento
1	Amity	Wilcinskis	Rany	1976-08-09
2	Magda	Borgnet	Zanuciolii	1954-04-13
3	Elva	McCheyne	Paoli	1985-10-06
4	Winston	Dansken	Novotni	1962-09-02
5	Erda	Mordey	Phipard-Shears	1969-09-25
6	Katinka	Sara	Stiling	1977-07-30
7	Connor	Bluck	Labrone	1965-11-27
8	Chadd	Carson	Verma	1995-03-26
9	Malvin	Houselee	MacIver	1981-11-24
10	Stanislas	Chapellow	Stammirs	1963-06-11
11	Karna	Nias	Heeley	1962-08-02
12	Davidson	Castelletti	Courvert	1983-06-14
13	Vale	Ferras	Boyn	1984-03-05
14	Peirce	Ringer	Wickham	1979-06-04
15	Shepperd	Rosenzveig	Hadrill	1953-07-13

25. 5 productos no perecederos más vendidos

```
1 SELECT P.nombre, SUM(TNP.cantidad) AS total_vendido
2 FROM ProductoNoPerecedero P
3 JOIN TenerProductoNoPerecedero TNP ON P.id_mercancia = TNP.
   id_mercancia
4 GROUP BY P.nombre
5 ORDER BY total_vendido DESC
6 LIMIT 5;
```

	A-Z nombre	123 total_vendido
1	Set of Decorative Storage Bins	34
2	Kids' Outdoor Explorer Kit	31
3	Tofu	29
4	Creative Puzzle Game	29
5	Satin Slip Dress	28

26. Cantidad de trabajadores de cada tipo por bazar

```

1 SELECT B.id_bazar, --cambiar
2     (SELECT COUNT(*) FROM TrabajarMedico WHERE id_bazar =
3      B.id_bazar) AS medicos,
4     (SELECT COUNT(*) FROM TrabajarSeguridad WHERE id_bazar
5      = B.id_bazar) AS seguridad,
6     (SELECT COUNT(*) FROM TrabajarLimpieza WHERE id_bazar
7      = B.id_bazar) AS limpieza
8 FROM Bazar B;

```

	id_bazar	medicos	seguridad	limpieza
1	1	7	9	10
2	2	7	7	11
3	3	6	8	5
4	4	9	10	10
5	5	9	11	10
6	6	6	13	6
7	7	5	7	12
8	8	5	8	9
9	9	6	8	11
10	10	7	9	8
11	11	9	6	5
12	12	2	14	14
13	13	10	8	10
14	14	9	11	8
15	15	6	8	13

27. Promedio de precios de productos por negocio

```

1 SELECT id_negocio, AVG(precio::numeric::float8) AS
   precio_promedio
2 FROM (
3     SELECT id_negocio, precio FROM ProductoPerecedero
4     UNION ALL
5     SELECT id_negocio, precio FROM ProductoNoPerecedero
6 ) AS productos
7 GROUP BY id_negocio;

```

	id_negocio	precio_promedio
1	58	446.288
2	8	844.95
3	116	364.3366666667
4	184	581.565
5	87	407.7766666667
6	71	566.0633333333
7	68	586.404
8	51	596.595
9	146	579.55
10	70	453.15
11	52	385.615
12	80	866.3633333333
13	190	459.63
14	162	813.72
15	132	320.6566666667

28. Negocios que participan en más de 3 bazares

```
1 SELECT id_negocio, COUNT(id_bazar) AS total_bazares
2 FROM Participar
3 GROUP BY id_negocio
4 HAVING COUNT(id_bazar) >= 3;
```

	123 id_negocio	123 total_bazares
1	58	3
2	8	3
3	87	3
4	68	3
5	51	3
6	80	3
7	70	3
8	52	3
9	84	3
10	92	3
11	69	3
12	60	3
13	22	3
14	59	3
15	65	3



29. Clientes que han realizado más de 2 compras

```
1 SELECT id_cliente, COUNT(id_ticket) AS total_compras
2 FROM Ticket
3 GROUP BY id_cliente
4 HAVING COUNT(id_ticket) > 2;
```

	123 id_cliente	123 total_compras
1	874	3
2	506	3
3	10	3
4	166	3
5	27	3
6	58	3
7	229	3
8	996	3
9	942	3
10	59	3
11	944	3
12	532	3
13	85	3
14	948	3
15	300	3

30. Resumen financiero del bazar por ingresos estimados

```

1 SELECT E.id_pago AS id_pago_tarjeta, 'Tarjeta' AS tipo_pago
2 FROM Tarjeta E
3 UNION
4 SELECT E.id_pago AS id_pago_efectivo, 'Efectivo' AS tipo_pago
5 FROM Efectivo E;

```

	123 id_pago_tarjeta ▼	A-Z tipo_pago ▼
1	405	Tarjeta
2	130	Efectivo
3	482	Efectivo
4	387	Tarjeta
5	574	Tarjeta
6	48	Efectivo
7	175	Efectivo
8	136	Efectivo
9	279	Tarjeta
10	552	Tarjeta
11	92	Tarjeta
12	650	Tarjeta
13	214	Tarjeta
14	300	Efectivo
15	73	Tarjeta

7 Guía de Ejecución

7.1 Herramientas a utilizar

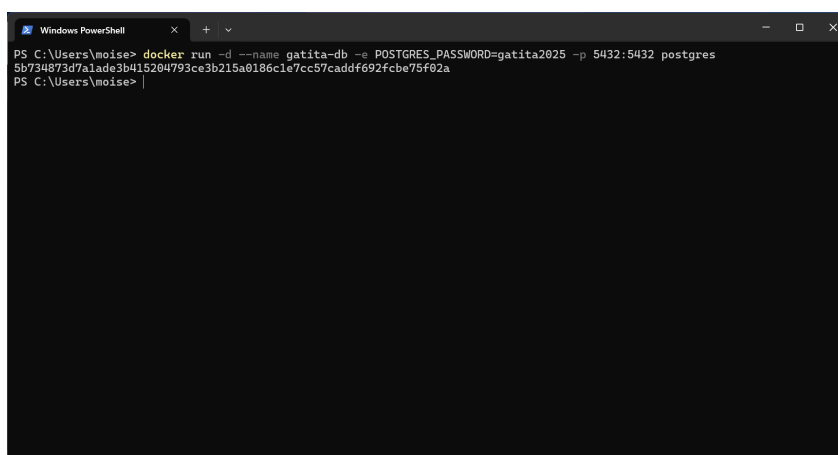
Herramienta	Descripción
PostgreSQL	Sistema Manejador de Bases de Datos.
Docker	Plataforma que permite ejecutar contenedores. Se usó para montar una instancia de PostgreSQL sin necesidad de instalación local.
psql	Línea de comandos para conectarse y ejecutar instrucciones SQL dentro de PostgreSQL.

7.2 Pasos para probar la Base de Datos desde Windows

Paso 1: Abrimos **Docker Desktop**.

Paso 2: Cuando Docker ya esté corriendo, ejecutamos el siguiente comando:

```
shell
1 docker run -d --name gatita-db -e POSTGRES_PASSWORD=gatitaEmp -p 5432:5432 postgres
```

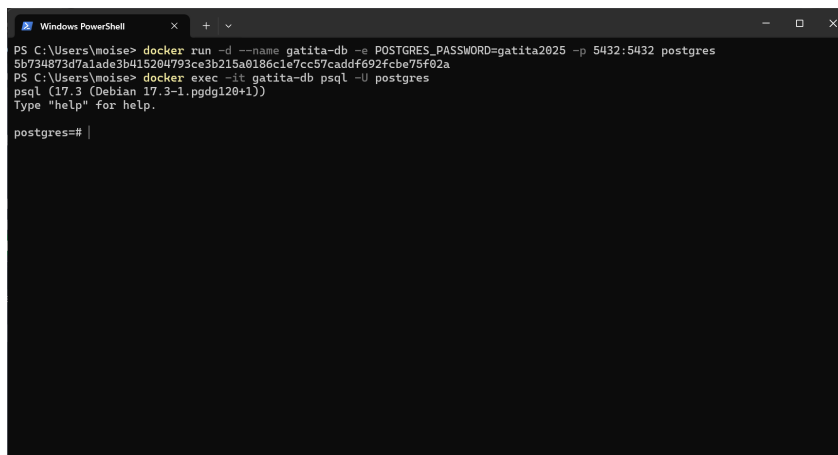


```
Windows PowerShell
PS C:\Users\moise> docker run -d --name gatita-db -e POSTGRES_PASSWORD=gatita2025 -p 5432:5432 postgres
5b734873d7alade3b415204793ce3b215a0186c1e7cc57caddf692fcbef5f02a
PS C:\Users\moise> |
```

Paso 3: Conectarse al contenedor.

Una vez que el contenedor está en ejecución, podemos acceder a PostgreSQL directamente desde la terminal usando el siguiente comando:

```
shell
1 docker exec -it gatita-db psql -U postgres
```



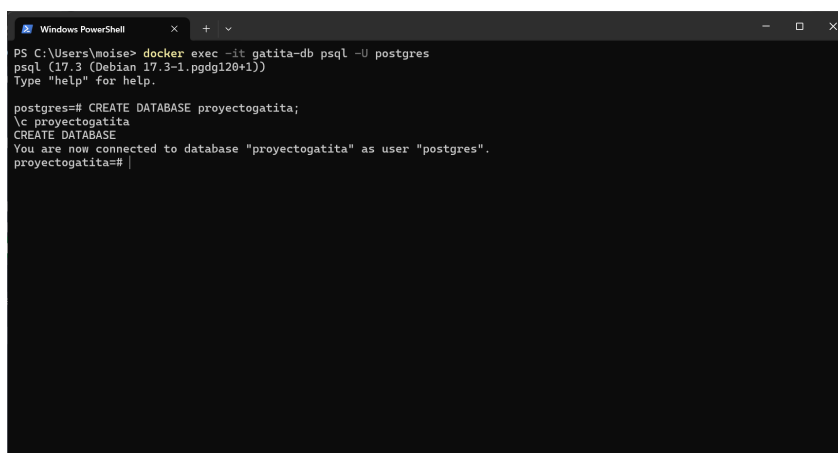
```
Windows PowerShell
PS C:\Users\moise> docker run -d --name gatita-db -e POSTGRES_PASSWORD=gatita2025 -p 5432:5432 postgres
5b734873d7alade3b415204793ce3b215a0186c1e7cc57caddf692fcbef75f02a
PS C:\Users\moise> docker exec -it gatita-db psql -U postgres
psql (17.3 (Debian 17.3-1.pgdgl20+1))
Type "help" for help.

postgres=# |
```

Paso 4: Crear la base de datos.

Una vez dentro de PostgreSQL, ejecutamos las siguientes instrucciones para crear la base de datos y conectarnos a ella:

```
shell
1 CREATE DATABASE proyectogatita;
2 \c proyectogatita
```



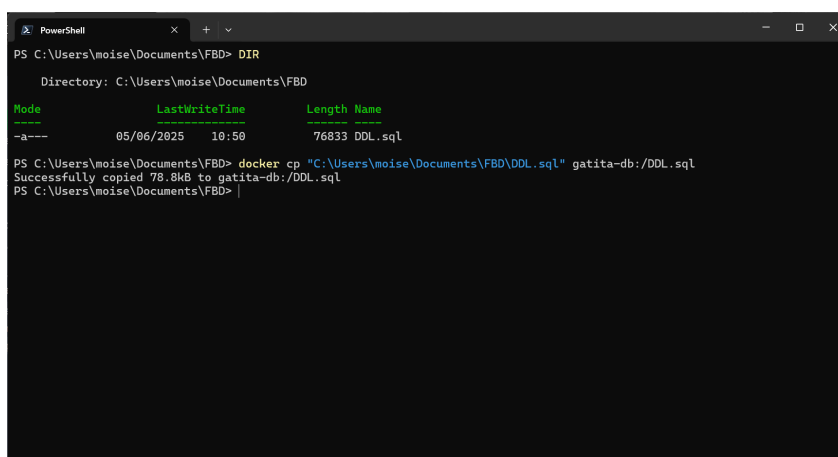
```
Windows PowerShell
PS C:\Users\moise> docker exec -it gatita-db psql -U postgres
psql (17.3 (Debian 17.3-1.pgdgl20+1))
Type "help" for help.

postgres=# CREATE DATABASE proyectogatita;
\c proyectogatita
CREATE DATABASE
You are now connected to database "proyectogatita" as user "postgres".
proyectogatita=# |
```

Paso 5: Copiamos el archivo DDL.sql.

Para ejecutar el script de creación de tablas dentro del contenedor, primero debemos copiar el archivo desde nuestro sistema local al contenedor de Docker. Usamos el siguiente comando:

```
shell
1 docker cp "C:\Users\moise\Documents\FBD\DDL.sql" gatita-db:/DDL.
  sql
```



```
PowerShell
PS C:\Users\moise\Documents\FBD> DIR

Directory: C:\Users\moise\Documents\FBD

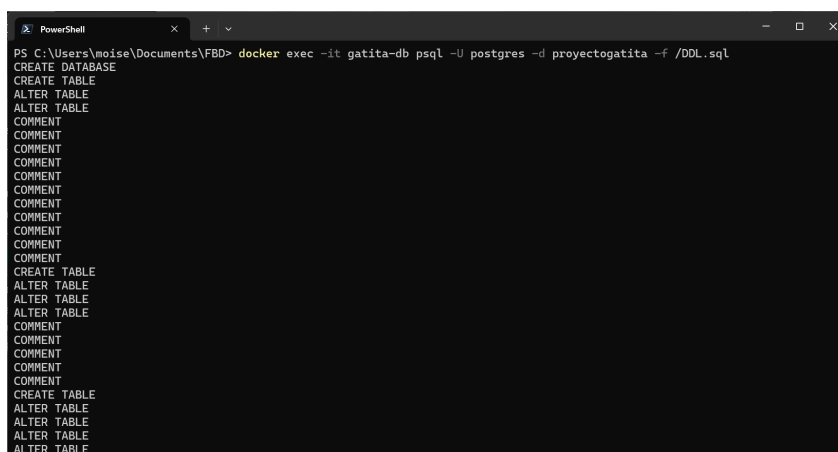
Mode                LastWriteTime         Length Name
----                -
-a----           05/06/2025   10:50             76833 DDL.sql

PS C:\Users\moise\Documents\FBD> docker cp "C:\Users\moise\Documents\FBD\DDL.sql" gatita-db:/DDL.sql
Successfully copied 78.8kB to gatita-db:/DDL.sql
PS C:\Users\moise\Documents\FBD> |
```


Paso 6: Crear las tablas con el archivo DDL.

Una vez copiado el archivo al contenedor, podemos ejecutar su contenido dentro de PostgreSQL para crear las tablas definidas. Desde PowerShell o una terminal, utilizamos el siguiente comando:

```
shell
1 docker exec -it gatita-db psql -U postgres -d proyectogatita -f /
  DDL.sql
```



```
PowerShell
PS C:\Users\moise\Documents\FBD> docker exec -it gatita-db psql -U postgres -d proyectogatita -f /DDL.sql
CREATE DATABASE
CREATE TABLE
ALTER TABLE
ALTER TABLE
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
CREATE TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
COMMENT
COMMENT
COMMENT
COMMENT
CREATE TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
```



The screenshot shows a Windows PowerShell window with the title bar 'Windows PowerShell'. The terminal content is as follows:

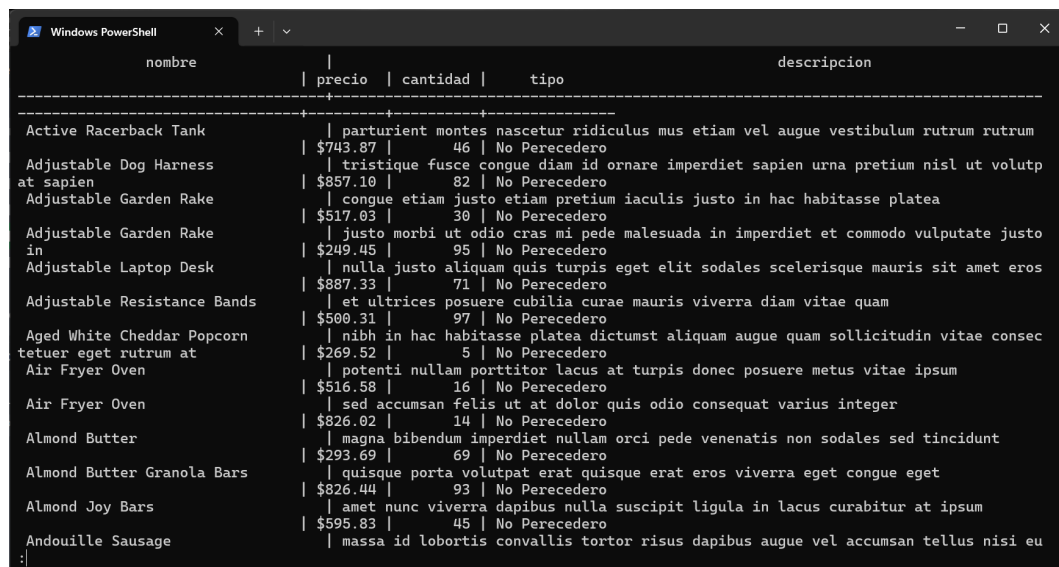
```
PS C:\Users\moise> docker cp "C:\Users\moise\Documents\FBD\DDL.sql" gitita-db:/DDL.sql
Successfully copied 2.06MB to gitita-db:/DDL.sql
PS C:\Users\moise> |
```

Usamos el siguiente comando:

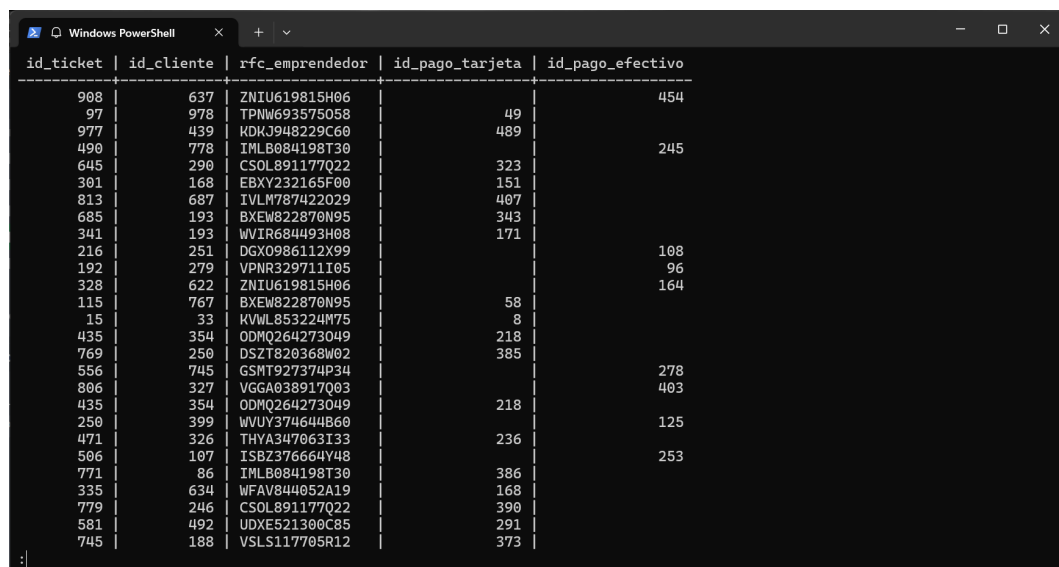
A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Windows PowerShell". The terminal background is dark gray. It displays a series of 20 identical "INSERT 0 1" commands, one per line. At the bottom, the prompt "PS C:\Users\moise>" is visible. The window has standard OS controls (minimize, maximize, close) in the top right corner.

Paso 9: Ejecutar consultas.

Se enlistan algunas consultas como ejemplo. Después de ejecutar el archivo, con el comando **q** podemos ir cambiando de consulta.



nombre	precio	cantidad	tipo	descripcion
Active Racerback Tank	\$743.87	46	No Perecedero	parturient montes nascetur ridiculus mus etiam vel augue vestibulum rutrum rutrum
Adjustable Dog Harness	\$857.10	82	No Perecedero	tristique fusce congue diam id ornare imperdiet sapien urna pretium nisl ut volutp
Adjustable Garden Rake	\$517.03	30	No Perecedero	congue etiam justo etiam pretium iaculis justo in hac habitasse platea
Adjustable Garden Rake	\$249.45	95	No Perecedero	justo morbi ut odio cras mi pede malesuada in imperdiet et commodo vulputate justo
Adjustable Laptop Desk	\$887.33	71	No Perecedero	nulla justo aliquam quis turpis eget elit sodales scelerisque mauris sit amet eros
Adjustable Resistance Bands	\$500.31	97	No Perecedero	et ultrices posuere cubilia curae mauris viverra diam vitae quam
Aged White Cheddar Popcorn	\$269.52	5	No Perecedero	nibh in hac habitasse platea dictumst aliquam augue quam sollicitudin vitae consec
tetuer eget rutrum at	\$516.58	16	No Perecedero	potenti nullam porttitor lacus at turpis donec posuere metus vitae ipsum
Air Fryer Oven	\$826.02	14	No Perecedero	sed accumsan felis ut at dolor quis odio consequat varius integer
Air Fryer Oven	\$293.69	69	No Perecedero	magna bibendum imperdiet nullam orci pede venenatis non sodales sed tincidunt
Almond Butter	\$826.44	93	No Perecedero	quisque porta volutpat erat quisque erat eros viverra eget congue eget
Almond Butter Granola Bars	\$595.83	45	No Perecedero	amet nunc viverra dapibus nulla suscipit ligula in lacus curabitur at ipsum
Almond Joy Bars				
Andouille Sausage				massa id lobortis convallis tortor risus dapibus augue vel accumsan tellus nisi eu



id_ticket	id_cliente	rfc_emprendedor	id_pago_tarjeta	id_pago_efectivo
908	637	ZNIU619815H06		454
97	978	TPNW693575058	49	
977	439	KDKJ948229C60	489	
490	778	IMLB0884198T30		245
645	290	CSOL891177Q22	323	
301	168	EBXY232165F00	151	
813	687	IVLM787422029	407	
685	193	BXEW822870N95	343	
341	193	WVIR684493H08	171	
216	251	DGX0986112X99		108
192	279	VPNR329711I05		96
328	622	ZNIU619815H06		164
115	767	BXEW822870N95	58	
15	33	KVWL853224M75	8	
435	354	ODMQ264273049	218	
769	250	DSZT820368W02	385	
556	745	GSMT927374P34		278
806	327	VGGA038917Q03		403
435	354	ODMQ264273049	218	
250	399	WVUY374644B60		125
471	326	THYA347063I33	236	
506	107	ISBZ376664Y48		253
771	86	IMLB0884198T30	386	
335	634	WFAV844052A19	168	
779	246	CSOL891177Q22	390	
581	492	UDXE521300C85	291	
745	188	VSL5117705R12	373	

```
Windows PowerShell
```

nombre	fecha_caducidad
Cookie Chocolate Chip With	2024-07-19
Cheese - Fontina	2025-01-06
Wine - Zonnebloem Pinotage	2025-02-12
Pastrami	2025-04-24
Shrimp - 31/40	2024-09-24
Corn Shoots	2024-10-21
Table Cloth 91x91 Colour	2024-06-20
Pie Filling - Cherry	2025-06-09
Wine - Barossa Valley Estate	2025-03-05
Swiss Chard - Red	2024-06-16
Pepper - Orange	2025-05-22
Contreau	2024-10-03
Greens Mustard	2024-10-29
Zucchini - Green	2024-11-04
Munchies Honey Sweet Trail Mix	2025-03-23
Sponge Cake Mix - Vanilla	2025-02-23
Veal - Knuckle	2024-11-13
Pork - European Side Bacon	2024-06-16
Coffee Cup 8oz 5338cd	2024-10-21
Chocolate Bar - Reese Pieces	2024-10-24
Soup - Campbells Beef Stew	2025-02-27
Oil - Truffle, White	2025-05-23
Pepper - White, Ground	2024-12-28
Beef - Ox Tail, Frozen	2024-07-04
Cloves - Whole	2024-11-07
Pizza Pizza Dough	2025-01-08
Eel - Smoked	2024-11-07

```
Windows PowerShell
```

id_negocio	precio_promedio
58	446.28799999999999
8	844.94999999999999
116	364.33666666666664
184	581.565
87	407.77666666666664
71	566.06333333333334
68	586.404
51	596.595
146	579.55
70	453.15
52	385.615
80	866.36333333333333
190	459.62999999999994
162	813.72
132	320.65666666666667
84	445.69333333333333
170	416.51
192	886.435
169	414.91999999999996
176	678.81666666666666
92	487.37
101	808.98666666666667
69	556.2775
180	348.32666666666665
114	497.945
115	353.5925
60	330.28

```
Windows PowerShell
```

```
PS C:\Users\moise> docker exec -it gatita-db psql -U postgres -d proyectogatita -f /Query.sql
```

tipo_pago	cantidad_transacciones
Efectivo	500
Tarjeta	500

(2 rows)

id_ticket	nombre	apellido_paterno	rfc_emprendedor
226	Ciro	Kanwell	RGVJ228467U22
916	Freda	Hawkshaw	RGVJ228467U22
468	Dorian	Blunt	RGVJ228467U22
694	Zarla	Lafranconi	RGVJ228467U22
132	Kipp	Wittrington	RGVJ228467U22

(5 rows)

id_mercancia	nombre	total_vendido
92	Aged White Cheddar Popcorn	4
188	Scented Candle Set	3
91	Sriracha Sauce	3
105	Samsung Galaxy Smartwatch	4
89	Jasmine Rice	4
13	Woven Storage Baskets	1
155	Portable Dog Water Bottle	4
75	Harvest Grain Salad	4
186	Spicy Tuna Rolls	1
114	Apple Sauce	1
165	Chili Lime Corn Chips	1
39	Cork Yoga Mat	4

Paso 10: Cargar procedimientos y triggers.

```
Windows PowerShell
PS C:\Users\moise> docker cp "C:\Users\moise\Documents\FBD\SP_Triggers.sql" gatita-db:/SP_Triggers.sql
Successfully copied 8.19kB to gatita-db:/SP_Triggers.sql
PS C:\Users\moise> docker exec -it gatita-db psql -U postgres -d proyectogatita -f /SP_Triggers.sql
CREATE PROCEDURE
CREATE PROCEDURE
CREATE FUNCTION
CREATE TRIGGER
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER
CREATE TRIGGER
CREATE TRIGGER
PS C:\Users\moise> |
```

Una vez dentro de PostgreSQL, se ejecuta la consulta para verificar si el trigger es correcto. Ejemplo:

```
Windows PowerShell
PS C:\Users\moise> docker exec -it gatita-db psql -U postgres -d proyectogatita
psql (17.3 (Debian 17.3-1.pgdgl20+1))
Type "help" for help.

proyectogatita=# SELECT tgname, tgrelid::regclass AS table_name
FROM pg_trigger
WHERE tgname = 'revisar_caducidad';
      tgname      | table_name
-----+-----
revisar_caducidad | productoperecedero
(1 row)

proyectogatita=#
```