

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

DISEÑO DE BASES DE DATOS



CREACION DE UNA BASE

Presenta

Ricardo Soberanis Romero
Carlos Alan Solorio Del Toro
Mariana Salome Garcia Gonzalez

Profesor: Gabriel Nafarrate Vázquez

Fecha 6/05/2024

Contenido

1. Nombre del proyecto	4	
2. Introducción	4	
3. Situación actual	4	
3.1 Operaciones que realiza el negocio	4	
3.2 Reglas de negocio	4	
3.3 Requerimientos de información	5	
4. Alcance del proyecto	5	
5. Diseño de la base de datos	5	
5.1 Diagrama entidad relación en su forma extendida (EER)	5	
5.2 Transformación del diagrama ER a tablas	5	
6. Tablas a normalizar	5	
6.1 Tabla inicial	5	
6.2 Evidencias de 1FN	5	
6.3 Evidencias de 2FN	5	
6.4 Evidencias de 3FN	5	6
7. Diseño de los procesos de negocio	6	
8. Diseño de las consultas	6	
9.1 Álgebra relacional	6	
9.2 SQL	6	
9. Conclusiones	7	
10. Reflexiones y o aprendizajes logrados	7	
11. Fuentes bibliográficas	7	
Se divide en 3 entregas en las siguientes fechas:		7
		8

1. Nombre del proyecto

Implementación de bases de datos para soportar las operaciones básicas de un software para punto de venta.

2. Introducción

Se busca representar a través de una base de datos las distintas operaciones que se generan al operar un software de punto de venta

- Proceso de ventas
- Gestión de inventarios
- Control de ingresos
- Catálogo de clientes
- Catálogo de productos
- Cajeros, Cuentas

Además de soportar las operaciones cotidianas en cajas registradoras, hacer operaciones de manera integral (ticket, salida almacén, registro de ingreso de dinero) deberá poder hacer consultas para obtener la información relacionada al punto de venta.

3. Situación actual

El concepto del software administrativo integral ha cobrado importancia en los últimos años, el punto de venta es una herramienta clave para conocer el potencial que tienen las bases de datos relacionales para soportar la operación de manera eficiente.

3.1 Operaciones que realiza el negocio

- Catálogo de productos con sus diversas características
- Catálogo de almacenes
- Catálogo de clientes
- Catálogo de cajeros
- Catálogo de cajas
- Catálogo de cuentas
- Asignación de cajeros a cajas y cuenta
- Registro de ventas, emisión de tickets.
- Creación automática de documentos asociados al ticket de venta.
- Entradas y salidas de mercancía a un almacén.
- Registro de ingresos y egresos
- Política de descuentos
- *Devoluciones

3.2 Reglas de negocio

- Pueden existir diferentes cajas
- Pueden existir diferentes vendedores
- Pueden existir diferentes cuentas
- Catálogo de artículos
- Características de su artículo
- Al menos dos clasificadores (departamento, categoría, tipo) - Pueden existir artículos inventariables y no inventariables (servicio) - Alta de un almacén y poder registrar entradas y salidas de mercancías.
- Catálogo de clientes
- Agregar clientes particulares y cliente público en general.
- Proceso de ventas
- La emisión de tickets puede ser a público en general o a un cliente particular previamente capturado
- Que sea integral, un ticket se inicia con estatus “En proceso”, al cambiar este estatus a “Activo” ya no es modificable, la base de datos genera la correspondiente salida de almacén, así como el ingreso de dinero de acuerdo a sus características, también debe existir el estatus de “Cancelado” que cancela las operaciones relacionadas y no se incluye en los reportes.
- El ingreso de dinero puede ser en efectivo, tarjeta o mixto.
- Definir alguna política de descuento, puede ser por producto, por departamento, por cliente, etc. Aplicar la regla en los tickets que cumplan con esa característica

3.3 Requerimientos de información

La empresa requiere:

1. Emisión de ticket
2. Reporte del producto con su cantidad y su monto en dinero en un rango de fechas determinado, incluir productos que no se vendieron en ese periodo.
3. Reporte de los 10 mejores productos vendidos por rango de fechas.
4. Reporte de ingresos de ventas por cliente.
5. Reporte de ventas por mes, filtrados por año.
6. Corte de caja, por vendedor, en una caja particular en una fecha determinada, cuánto ingresó en efectivo y cuánto en tarjeta de crédito.
7. Corte de caja detallado, incluir listado de tickets y totales del reporte anterior.
8. Reporte de existencias por producto al día de hoy.
9. Estado de cuenta de dinero, ingresos y egresos, por rango de fechas.

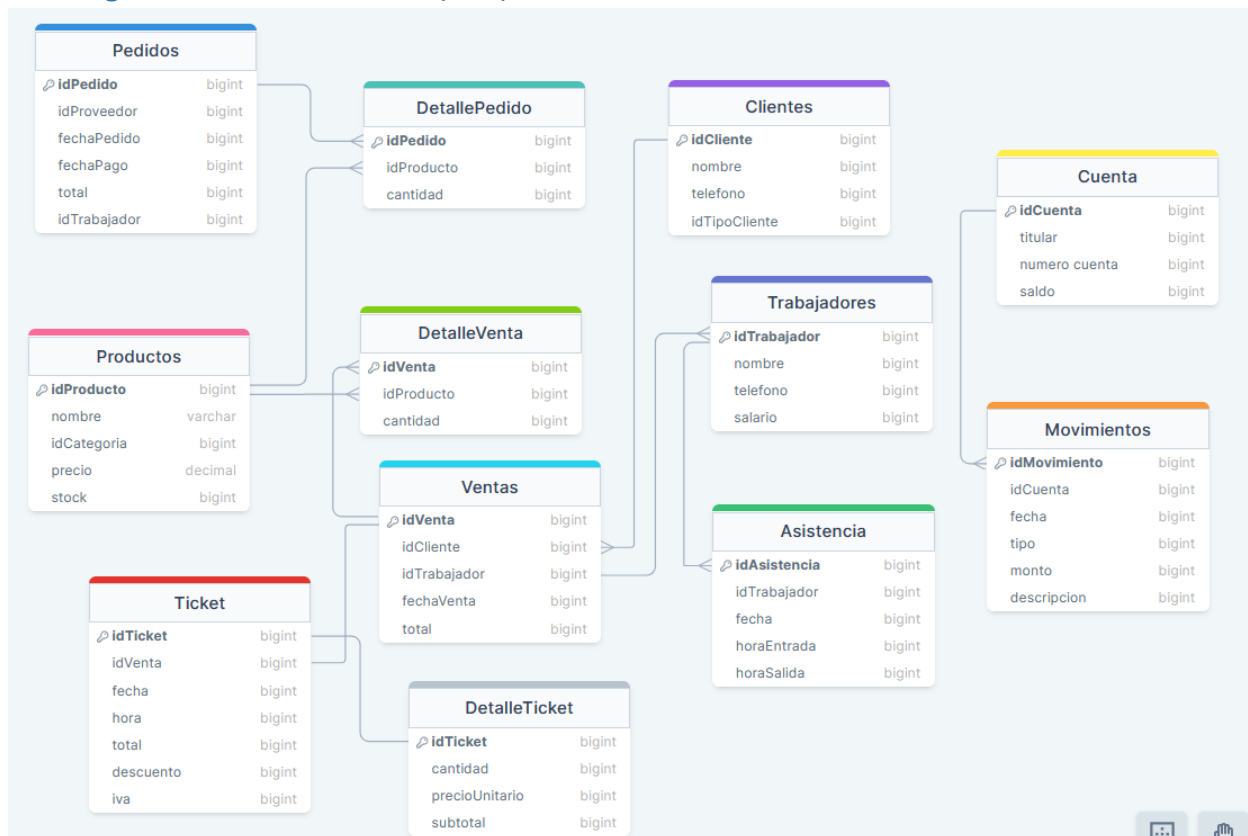
10. Reporte libre.

4. Alcance del proyecto

El alcance de este proyecto es crear la base de datos, realizar su poblado simulando la operación para después hacer las consultas que muestren la información pertinente de un punto de venta.

5. Diseño de la base de datos

5.1 Diagrama entidad relación (EER)



5.2 Transformación del diagrama ER a tablas

Código de la creación

-- Tabla Categoria

```
CREATE TABLE Categoria ( idCategoria INT PRIMARY KEY,
    nombre VARCHAR(100)
);
```

-- Tabla Productos

```
CREATE TABLE Productos (  
    idProducto INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    idCategoria INT,  
    precio DECIMAL(10,2),  
    stock INT,  
    FOREIGN KEY (idCategoria) REFERENCES Categoria(idCategoria)  
);
```

-- Tabla Clientes

```
CREATE TABLE Clientes (  
    idCliente INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    telefono VARCHAR(20),  
    idTipoCliente INT  
);
```

-- Tabla Trabajadores

```
CREATE TABLE Trabajadores (  
    idTrabajador INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    telefono VARCHAR(20),  
    salario DECIMAL(10,2)  
);
```

-- Tabla Proveedor

```
CREATE TABLE Proveedor (  
    idProveedor INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    telefono VARCHAR(20)  
);
```

-- Tabla Pedidos

```
CREATE TABLE Pedidos (  
    idPedido INT PRIMARY KEY,  
    idProveedor INT,  
    fechaPedido DATE,  
    fechaPago DATE,  
    total DECIMAL(10,2),  
    idTrabajador INT,  
    FOREIGN KEY (idProveedor) REFERENCES Proveedor(idProveedor),
```

```
FOREIGN KEY (idTrabajador) REFERENCES Trabajadores(idTrabajador)
);
```

```
-- Tabla DetallePedido
```

```
CREATE TABLE DetallePedido (
    idPedido INT,
    idProducto INT,
    cantidad INT,
    PRIMARY KEY (idPedido, idProducto),
    FOREIGN KEY (idPedido) REFERENCES Pedidos(idPedido),
    FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)
);
```

```
-- Tabla Ventas
```

```
CREATE TABLE Ventas (
    idVenta INT PRIMARY KEY,
    idCliente INT,
    idTrabajador INT,
    fechaVenta DATE,
    total DECIMAL(10,2),
    FOREIGN KEY (idCliente) REFERENCES Clientes(idCliente),
    FOREIGN KEY (idTrabajador) REFERENCES Trabajadores(idTrabajador)
);
```

```
CREATE TYPE TVP_DetalleVenta AS TABLE (
    idProducto INT,
    cantidad INT
);
```

```
CREATE TYPE TVP_DetallePedido AS TABLE (
    idProducto INT,
    cantidad INT
);
GO
```

```
-- Tabla Asistencia
```

```
CREATE TABLE Asistencia (
    idAsistencia INT PRIMARY KEY,
    idTrabajador INT,
    fecha DATE,
    horaEntrada TIME,
    horaSalida TIME,
    FOREIGN KEY (idTrabajador) REFERENCES Trabajadores(idTrabajador)
);
```

);

-- Tabla Cuenta

```
CREATE TABLE Cuenta (  
    idCuenta INT PRIMARY KEY,  
    titular VARCHAR(100),  
    numeroCuenta VARCHAR(20),  
    saldo DECIMAL(10,2)  
);
```

-- Tabla Movimientos

```
CREATE TABLE Movimientos (  
    idMovimiento INT PRIMARY KEY,  
    idCuenta INT,  
    fecha DATE,  
    tipo VARCHAR(20),  
    monto DECIMAL(10,2),  
    descripcion VARCHAR(100),  
    FOREIGN KEY (idCuenta) REFERENCES Cuenta(idCuenta)  
);
```

-- Tabla Ticket

```
CREATE TABLE Ticket (  
    idTicket INT PRIMARY KEY,  
    idVenta INT,  
    fecha DATE,  
    hora TIME,  
    total DECIMAL(10,2),  
    descuento DECIMAL(10,2),  
    iva DECIMAL(10,2),  
    FOREIGN KEY (idVenta) REFERENCES Ventas(idVenta)  
);
```

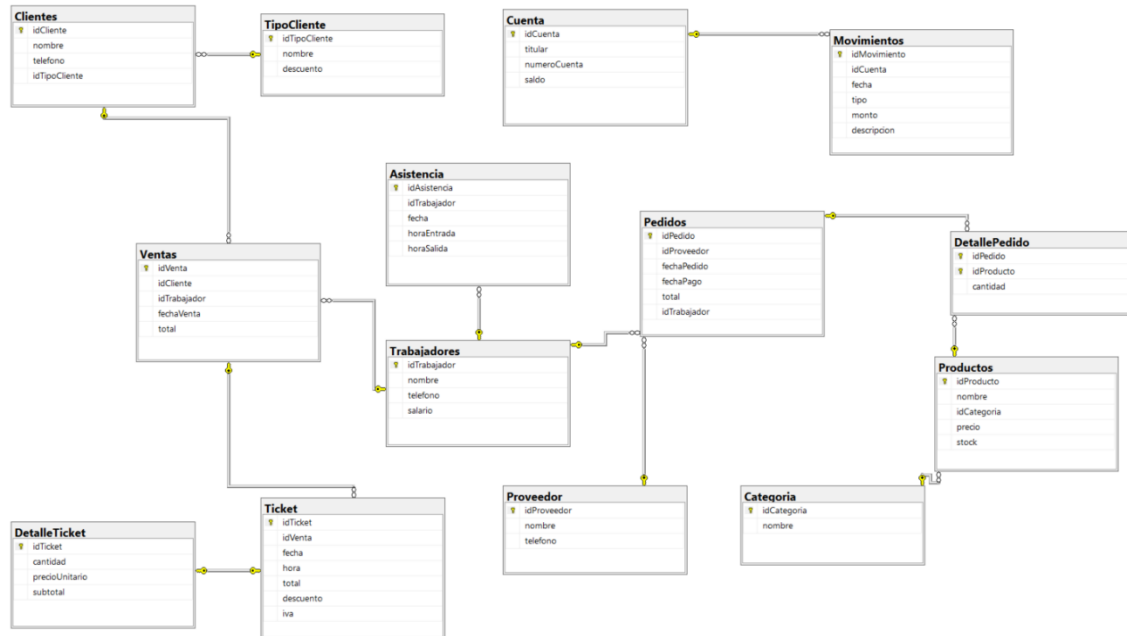
-- Tabla DetalleTicket

```
CREATE TABLE DetalleTicket (  
    idTicket INT,  
    cantidad INT,  
    precioUnitario DECIMAL(10,2),  
    subtotal DECIMAL(10,2),  
    PRIMARY KEY (idTicket),  
    FOREIGN KEY (idTicket) REFERENCES Ticket(idTicket)  
);
```



```
CREATE TABLE TipoCliente (
  idTipoCliente INT PRIMARY KEY,
  nombre VARCHAR(100),
  descuento DECIMAL(5,2)
);
```

<Captura de pantalla del diagrama de tablas que genera Ms SQL y definición de tablas en SQL lenguaje>



6. Normalización

Nuestra base de datos sigue las leyes de normalización ya que no contiene ningún tipo de dato repetido en ninguna tabla. Repasándolas en orden, sabemos que cumple con la primera forma normalizada:

Primera forma normal (1FN):

- Todas las tablas tienen una clave primaria que identifica de manera única cada registro.
- No hay campos que contengan listas o valores repetidos dentro de un mismo registro.
- Todos los campos contienen valores atómicos, es decir, un solo valor por celda.

Sabemos que también cumple con la primera forma normalizada:

Segunda forma normal (2FN):

- Cada tabla tiene una clave primaria que no depende de ninguna otra columna.
- Todos los campos no clave dependen por completo de la clave primaria, es decir, no hay dependencias parciales.

Y, por último, también sabemos que cumple con la tercera forma normalizada:

Tercera forma normal (3FN):

- No existen dependencias transitivas, es decir, no hay campos que dependan de otros campos que no son la clave primaria.

Para comprobar que los puntos anteriores se cumplen, podemos crear tablas de ejemplo con datos ficticios y mostrar que se cumplen estas tres formas normales. Por ejemplo:

Tabla 'empleado' con datos de ejemplo:

idVenta	idCliente	idTrabajador	fecha	total
1	1	1	13/01/2022	543.64
2	2	1	14/01/2022	283.39
3	3	3	15/01/2022	154.93
4	4	2	16/01/2022	378.78
5	5	1	17/01/2022	243.84
6	6	1	18/01/2022	2342.86
7	7	3	19/01/2022	454.76

idProducto	nombre	idCategoria	precio	stock
1	Paleta mango	1	20	85
2	Paleta mango-chamoy	3	35	45
3	Paleta fresa	2	25	68
4	Paleta Maracuyá	3	35	87

idCategoria	categoria
1	De Agua
2	De leche
3	Especial

idTipoCliente	tipo	Descuento
1	General	0
2	Privado	10%

idVenta	idProducto	cantidad
1	2	3
1	4	4
2	3	2
2	7	5
3	14	3
3	2	5
4	12	6
4	1	3
4	23	7
5	6	6
5	4	8
6	3	5
6	5	7
6	7	4
6	13	6
6	17	6
6	2	4
6	4	7
6	14	5
6	11	4
7	2	8
7	4	6
7	5	4

idTrabajador	nombre	telefono	salario
1	Alberto Javier	3328405384	50
2	Francisco Cordoba	3348930219	50
3	Ameli Cruz	3338990483	50

idCliente	nombre	telefono	idTipoCliente
1	Jose Cruz Romero	3338493040	1
2	Alberto Gomez Perez	3328493045	1
3	Wendy Colin	3328394587	1
4	Abigail Guzman	3338274565	2
5	Xochitl Lopez	3331874569	1
6	Lorenzo del Muro	3342429635	2
7	Rogelio Mendoza	5753566432	2

idTicket	idVenta	hora	iva	descuento
1	1	11:35	16%	0%
2	2	12:03	16%	0%
3	3	13:43	16%	0%
4	4	13:59	16%	10%
5	5	14:24	16%	0%
6	6	14:50	16%	10%
7	7	16:02	16%	10%

Tabla producto con datos de ejemplo:

Como se puede observar, en estas tablas (se podrían incluir todas las tablas, pero requeriría de demasiado espacio y sería redundante) cada registro tiene una clave primaria única (idEmpleado, idProducto), no hay campos repetidos ni dependencias parciales o transitivas. Por lo tanto, el modelo de base de datos cumple con las tres formas normales.

7. Diseño de los procesos de negocio

Se debe poder realizar entradas directas de almacén.

Se debe poder realizar salidas de dinero a las cuentas.

Nuestro negocio Palettería "Frías" sigue un modelo en el que hacemos un pedido a un proveedor y ellos nos surten, el pago se realiza 3 o 4 días después cuando el proveedor regrese. La palettería distribuye las paletas a diferentes expendios (abarrotes, tiendas, farmacias, etc.) y también tiene venta al público en general. Hay dos empleados por cada turno, pero como también está trabajando el jefe por las mañanas no está registrado como empleado y no cobra sueldo (él lo hace así). Se pueden cancelar compras y los pagos pueden ser en efectivo o en tarjeta.

Anexo a la entrega del reporte, se entregaron también dos documentos:

1. El primero se llama "Creación.sql" y es para crear una base de datos como la de nosotros
2. La segunda es una plantilla que creamos usando una plantilla que encontramos online usando nuestros queries, para el poblado de la base de datos con algunos ejemplos y simulaciones.

Proceso de poblado, ingresar datos con instrucciones SQL:

- Al menos 20 clientes, incluyendo el cliente "público en general".

-

	idCliente	nombre	telefono	idTipoCliente						
1	1	Público en general	NULL	1	11	11	Diego Mendoza	555-8901	1	
2	2	María Fernández	555-1234	2	12	12	Isabella Ortega	555-2345	2	
3	3	Juan Pérez	555-5678	1	13	13	Javier Morales	555-6789	1	
4	4	Ana Ramírez	555-9012	2	14	14	Valentina Rojas	555-0123	2	
5	5	Pedro Gómez	555-3456	1	15	15	Andrés Herrera	555-4567	1	
6	6	Laura Torres	555-7890	2	16	16	Camila Gutiérrez	555-8901	2	
7	7	Carlos Ruiz	555-2345	1	17	17	Sebastián Navarro	555-2345	1	
8	8	Marta Sánchez	555-6789	2	18	18	Daniela Jiménez	555-6789	2	
9	9	Luis Castro	555-0123	1	19	19	Mateo Paredes	555-0123	1	
10	10	Sofía Vargas	555-4567	2	20	20	Valeria Vega	555-4567	2	

- Al menos una cuenta bancaria general y las cuentas para los cajeros si es necesario.

	idCuenta	titular	numeroCuenta	saldo
1	1	Palettería La Michoacana	1234567890	149114.44

- Al menos un almacén.

	idProducto	nombre	idCategoria	precio	stock						
1	1	Paleta de limón	1	10.00	100	14	14	Paleta de chicle	3	20.00	50
2	2	Paleta de fresa	1	10.00	100	15	15	Paleta de grosella	3	20.00	50
3	3	Paleta de mango	1	10.00	100	16	16	Paleta de sandía	1	10.00	100
4	4	Paleta de tamarindo	1	10.00	100	17	17	Paleta de guayaba	1	10.00	100
5	5	Paleta de jamaica	1	10.00	100	18	18	Paleta de maracuyá	1	10.00	100
6	6	Paleta de coco	2	15.00	80	19	19	Paleta de coco con chocolate	2	15.00	80
7	7	Paleta de chocolate	2	15.00	80	20	20	Paleta de fresa con crema	2	15.00	80
8	8	Paleta de vainilla	2	15.00	80	21	21	Paleta de mango con chile	3	20.00	50
9	9	Paleta de cajeta	2	15.00	80	22	22	Paleta de tamarindo con chamoy	3	20.00	50
10	10	Paleta de queso	2	15.00	80	23	23	Paleta de pay de limón	3	20.00	50
11	11	Paleta de piña colada	3	20.00	50	24	24	Paleta de chocolate con almendras	3	20.00	50
12	12	Paleta de mojito	3	20.00	50	25	25	Paleta de yogurt con frutas	3	20.00	50
13	13	Paleta de margarita	3	20.00	50						

- Al menos 3 cajeros.

	idTrabajador	nombre	telefono	salario
1	1	Alejandro Sánchez	555-1111	8000.00
2	2	Gabriela Ramírez	555-2222	8000.00
3	3	Diego Fernández	555-3333	8000.00

- Al menos 120 operaciones de ventas en un periodo de 6 meses.

	idVenta	idCliente	idTrabajador	fechaVenta	total
356	356	7	3	2022-04-26	544.02
357	357	11	3	2022-04-27	142.68
358	358	10	1	2022-04-28	347.67
359	359	15	3	2022-04-29	447.03
360	360	6	3	2022-04-30	427.28
361	361	12	2	2024-02-08	270.00
362	362	11	3	2024-01-08	225.00
363	363	19	2	2024-03-13	95.00
364	364	19	2	2024-02-11	280.00
365	365	5	3	2024-02-10	300.00
366	366	11	3	2024-04-10	900.00
367	367	3	2	2024-01-10	135.00
368	368	19	2	2024-04-05	375.00
369	369	3	2	2024-03-07	300.00
370	370	2	2	2024-04-19	900.00
371	371	12	1	2024-03-21	144.00
372	372	14	3	2024-04-17	18.00

- Al menos 5 cancelaciones
- *Al menos 5 devoluciones (opcional)

8. Diseño de las consultas

Resolver las necesidades de información solicitadas en el punto 3.3, mostrando para cada una la instrucción SQL y el resultado de la consulta

La empresa requiere:

11. Emisión de ticket

	idTicket	idVenta	fecha	hora	total	descuento	iva
1	2	1	2023-01-01	17:34:21.2266667	117.28	0.10	0.16

12. Reporte del producto con su cantidad y su monto en dinero en un rango de fechas determinado, incluir productos que no se vendieron en ese periodo.

	Producto	CantidadVendida	MontoVendido
1	Paleta de cajeta	132	1980.00
2	Paleta de chicle	112	2240.00
3	Paleta de chocolate	116	1740.00
4	Paleta de chocolate con almendras	98	1960.00
5	Paleta de coco	112	1680.00
6	Paleta de coco con chocolate	108	1620.00
7	Paleta de fresa	86	860.00
8	Paleta de fresa con crema	117	1755.00
9	Paleta de grosella	169	3380.00
10	Paleta de guayaba	140	1400.00
11	Paleta de jamaica	115	1150.00
12	Paleta de limón	148	1480.00
13	Paleta de mango	143	1430.00
14	Paleta de mango con chile	166	3320.00
15	Paleta de maracuyá	156	1560.00
16	Paleta de margarita	134	2680.00

13. Reporte de los 10 mejores productos vendidos por rango de fechas.

	Producto	CantidadVendida
1	Paleta de queso	177
2	Paleta de grosella	169
3	Paleta de mango con chile	166
4	Paleta de vainilla	160
5	Paleta de maracuyá	156
6	Paleta de limón	148
7	Paleta de mango	143
8	Paleta de pay de limón	142
9	Paleta de guayaba	140
10	Paleta de tamarindo	135

14. B Reporte de ingresos de ventas por cliente.

	Cliente	CompraTotal
1	Público en general	5925.88

15. Reporte de ventas por mes, filtrados por año.

	Mes	VentasTotales
1	1	9672.47
2	2	8427.31
3	3	8432.97
4	4	8486.80

16. Venta total en una fecha por empleado

	VentasTotales
1	474.49

17. Corte de caja detallado, incluir listado de tickets y totales del reporte anterior.

	idTrabajador	nombre	idTicket	total
1	3	Diego Fernández	2	117.28
2	2	Gabriela Ramírez	3	136.70
3	2	Gabriela Ramírez	4	487.08
4	3	Diego Fernández	5	515.60
5	3	Diego Fernández	6	68.03
6	1	Alejandro Sánchez	7	532.64
7	1	Alejandro Sánchez	8	389.95
8	2	Gabriela Ramírez	9	160.22
9	3	Diego Fernández	10	46.67
10	3	Diego Fernández	11	386.15
11	2	Gabriela Ramírez	12	409.77
12	1	Alejandro Sánchez	13	509.00
13	3	Diego Fernández	14	385.92
14	1	Alejandro Sánchez	15	402.69
15	1	Alejandro Sánchez	16	257.70
16	1	Alejandro Sánchez	17	239.01
17	3	Diego Fernández	18	281.18
18	2	Gabriela Ramírez	19	331.56
19	3	Diego Fernández	20	195.23
20	1	Alejandro Sánchez	21	419.40
21	3	Diego Fernández	22	189.30
22	1	Alejandro Sánchez	23	350.42
23	2	Gabriela Ramírez	24	544.52

18. Reporte de existencias por producto al día de hoy.

	Producto	Existencia
1	Paleta de limón	100
2	Paleta de fresa	100
3	Paleta de mango	100
4	Paleta de tamarindo	100
5	Paleta de jamaica	100
6	Paleta de coco	80
7	Paleta de chocolate	80
8	Paleta de vainilla	80
9	Paleta de cajeta	80
10	Paleta de queso	80
11	Paleta de piña colada	50
12	Paleta de mojito	50
13	Paleta de margarita	50
14	Paleta de chicle	50
15	Paleta de grosella	50
16	Paleta de sandía	100

19. Estado de cuenta de dinero, ingresos y egresos, por rango de fechas.

	fecha	tipo	monto	descripcion
1	2023-01-01	Ingreso	117.28	Venta: #1
2	2023-01-02	Ingreso	136.70	Venta: #2
3	2023-01-03	Ingreso	487.08	Venta: #3
4	2023-01-04	Ingreso	515.60	Venta: #4
5	2023-01-05	Ingreso	68.03	Venta: #5
6	2023-01-06	Ingreso	532.64	Venta: #6
7	2023-01-07	Ingreso	389.95	Venta: #7
8	2023-01-08	Ingreso	160.22	Venta: #8
9	2023-01-09	Ingreso	46.67	Venta: #9
10	2023-01-10	Ingreso	386.15	Venta: #10
11	2023-01-11	Ingreso	409.77	Venta: #11
12	2023-01-12	Ingreso	509.00	Venta: #12
13	2023-01-13	Ingreso	385.92	Venta: #13
14	2023-01-14	Ingreso	402.69	Venta: #14
15	2023-01-15	Ingreso	257.70	Venta: #15
16	2023-01-16	Ingreso	239.01	Venta: #16
17	2023-01-17	Ingreso	281.18	Venta: #17

20. Reporte libre.

	Paleta	CantidadVendida
1	Paleta de queso	143
2	Paleta de grosella	122
3	Paleta de margarita	122

8.1 Álgebra relacional

8.1.1 Operaciones del banco

Depósito(idMovimiento, idCuenta, fecha, monto, descripcion) =

```
ρ(idMovimiento, idCuenta, fecha, monto, 'Depósito' AS descripcion)(
    σtipo='Ingreso'(Movimientos)
)
```

Retiro(idMovimiento, idCuenta, fecha, monto, descripcion) =

```
ρ(idMovimiento, idCuenta, fecha, monto, 'Retiro' AS descripcion)(
    σtipo='Egreso'(Movimientos)
)
```

SaldoCuenta(idCuenta, titular, numeroCuenta, saldo) =

```
ρ(idCuenta, titular, numeroCuenta, saldo + Σmonto)(
    Cuenta ⋈ Depósito
) - ρ(idCuenta, titular, numeroCuenta, saldo - Σmonto)(
```


Cuenta ⌘ Retiro

)

8.1.2 Reportes a obtener

VentasPorCliente(nombre, totalVentas) =

ynombre, SUM(total)(Clientes ⌘ Ventas)

VentasPorMes(año, mes, totalVentas) =

yYEAR(fechaVenta), MONTH(fechaVenta), SUM(total)(Ventas)

TopProductosVendidos(nombre, cantidadVendida) =

ynombre, SUM(cantidad)(Productos ⌘ DetalleVenta)

⌘cantidadVendida DESC LIMIT 10

TicketVenta(idTicket, fecha, hora, total, descuento, iva, producto, cantidad, precioUnitario, subtotal) =

⌘idTicket, fecha, hora, total, descuento, iva, nombre AS producto, cantidad, precioUnitario, subtotal(

Ticket ⌘ DetalleTicket ⌘ DetalleVenta ⌘ Productos

)

8.1 Cálculos relevantes

IngresosPorCliente(nombre, ingresoTotal) =

ynombre, SUM(total)(Clientes ⋈ Ventas)

GastosPorProveedor(nombre, gastoTotal) =

ynombre, SUM(total)(Proveedor ⋈ Pedidos)

VentasPorVendedor(nombre, ventasTotal) =

ynombre, SUM(total)(Trabajadores ⋈ Ventas)

8.2 SQL

9.2.1 Mostrar el código SQL los reportes solicitados en el punto 3.3 necesidades de información.

-- Emisión de ticket

SELECT

t.idTicket,
t.fecha,
t.hora,
tr.nombre **AS** trabajador,
c.nombre **AS** cliente,
p.nombre **AS** producto,
dt.cantidad,
dt.precioUnitario,
dt.subtotal,
t.descuento,
t.iva,
t.total

FROM

Ticket t
INNER JOIN Ventas v **ON** t.idVenta = v.idVenta
INNER JOIN Trabajadores tr **ON** v.idTrabajador = tr.idTrabajador

```
INNER JOIN Clientes c ON v.idCliente = c.idCliente
INNER JOIN DetalleTicket dt ON t.idTicket = dt.idTicket
INNER JOIN Productos p ON dt.idProducto = p.idProducto
WHERE
    t.idTicket = @idTicket;

-- Reporte del producto con su cantidad y su monto en dinero en un rango
de fechas determinado, incluir productos que no se vendieron en ese
periodo
SELECT
    p.idProducto,
    p.nombre,
    ISNULL(SUM(dt.cantidad), 0) AS cantidad,
    ISNULL(SUM(dt.subtotal), 0) AS monto
FROM
    Productos p
    LEFT JOIN DetalleTicket dt ON p.idProducto = dt.idProducto
    LEFT JOIN Ticket t ON dt.idTicket = t.idTicket
WHERE
    t.fecha BETWEEN @fechaInicio AND @fechaFin OR t.fecha IS NULL
GROUP BY
    p.idProducto, p.nombre;

-- Reporte de los 10 mejores productos vendidos por rango de fechas
SELECT TOP 10
    p.idProducto,
    p.nombre,
    SUM(dt.cantidad) AS cantidad,
    SUM(dt.subtotal) AS monto
FROM
    Productos p
    INNER JOIN DetalleTicket dt ON p.idProducto = dt.idProducto
    INNER JOIN Ticket t ON dt.idTicket = t.idTicket
WHERE
    t.fecha BETWEEN @fechaInicio AND @fechaFin
GROUP BY
    p.idProducto, p.nombre
ORDER BY
    cantidad DESC;

-- Reporte de ingresos de ventas por cliente
SELECT
    c.idCliente,
    c.nombre,
```

```
SUM(v.total) AS ingresos
FROM
  Clientes c
  INNER JOIN Ventas v ON c.idCliente = v.idCliente
GROUP BY
  c.idCliente, c.nombre;

-- Reporte de ventas por mes, filtrados por año
SELECT
  MONTH(v.fechaVenta) AS mes,
  SUM(v.total) AS ventas
FROM
  Ventas v
WHERE
  YEAR(v.fechaVenta) = @anio
GROUP BY
  MONTH(v.fechaVenta)
ORDER BY
  mes;

-- Corte de caja, por vendedor, en una fecha determinada, cuánto ingresó
SELECT
  t.idTrabajador,
  t.nombre,
  SUM(v.total) AS ingresos
FROM
  Trabajadores t
  INNER JOIN Ventas v ON t.idTrabajador = v.idTrabajador

GROUP BY
  t.idTrabajador, t.nombre;

-- Corte de caja detallado, incluir listado de tickets y totales del
reporte anterior
SELECT
  t.idTrabajador,
  t.nombre,
  tk.idTicket,
  tk.total
FROM
  Trabajadores t
  INNER JOIN Ventas v ON t.idTrabajador = v.idTrabajador
  INNER JOIN Ticket tk ON v.idVenta = tk.idVenta
```

```
SELECT
    SUM(tk.total) AS totalIngresos
FROM
    Trabajadores t
    INNER JOIN Ventas v ON t.idTrabajador = v.idTrabajador
    INNER JOIN Ticket tk ON v.idVenta = tk.idVenta
WHERE
    t.idTrabajador = @idTrabajador AND v.fechaVenta = @fecha;
```

-- Reporte de existencias por producto al día de hoy

```
SELECT
    idProducto,
    nombre,
    stock
FROM
    Productos;
```

-- Estado de cuenta de dinero, ingresos y egresos, por rango de fechas

```
SELECT
    m.fecha,
    m.tipo,
    m.monto,
    m.descripcion
FROM
    Movimientos m
WHERE
    m.fecha BETWEEN @fechaInicio AND @fechaFin
ORDER BY
    m.fecha;
```

-- Reporte de la paleta que más compra un cliente en un mes específico

```
SELECT TOP 1
    p.idProducto,
    p.nombre,
    SUM(dt.cantidad) AS cantidad
FROM
    Productos p
    INNER JOIN DetalleTicket dt ON p.idProducto = dt.idProducto
    INNER JOIN Ticket t ON dt.idTicket = t.idTicket
    INNER JOIN Ventas v ON t.idVenta = v.idVenta
WHERE
    v.idCliente = @idCliente AND YEAR(v.fechaVenta) = @anio AND
    MONTH(v.fechaVenta) = @mes
```

GROUP BY

p.idProducto, p.nombre

ORDER BY

cantidad **DESC**;

9. Conclusiones

- 1) El manejo de las bases de datos SQL es una habilidad muy importante para poder generar bases de datos según los requerimientos de una amplia variedad de negocios. Las bases de datos (en este caso relacionales) se usan en distintos sectores de la industria y en diversas aplicaciones para gestionar información estructurada y eficiente, por eso es el estándar para interactuar con bases de datos relacionales. Dominar este lenguaje es muy importante para cualquier desarrollador ya que es una herramienta muy usada.
- 2) El diseño de bases de datos es muy importante en el desarrollo de software de todo tipo, ya que es una habilidad muy importante a la hora de manejar grandes cantidades de información. Por lo que, una estructura bien diseñada evita redundancias innecesarias (si se implementa correctamente la normalización) y optimiza la ejecución de las operaciones, es por esto que, es de gran importancia el conocer el buen diseño de bases de datos utilizando las reglas de normalización.
- 3) Consideramos que crear una base de datos SQL correctamente normalizada ayuda a comprender cómo estructurar información de manera eficiente en una base de datos, ya que las normas de normalización ayudan a estructurar información de manera eficiente sin tener duplicados, y al seguir estas reglas, se obtiene una base de datos más organizada y con relaciones claras entre ellas. Creemos que las bases de datos han cambiado la industria de la tecnología de muchas maneras debido a la facilidad que han otorgado a la hora de trabajar con grandes cantidades de información
- 4) Un proyecto como este te ayuda a comprender la administración y toda la logística que hay detrás de un negocio, y no solo eso, también te ayuda a ver nuevas formas de estar mejorando, por decir, en el caso de las paletas sin la base de datos no podrían hacer cálculos tan rápidos como los que tenemos, y estas nuevas formas de visualizar la información trae consigo grandes ventajas, como si ya sé que esta tienda siempre me compra tantas paletas y las que más me compran son estas pues debo de estar preparado, también ver lugares donde puedes lanzar nuevos productos porque sabes que se vende

mucho, son cosas de análisis que si bien sin la base de datos claro que se pueden hacer, ya con la ayuda de esto es mucho más sencillo y preciso.

10. Reflexiones y/o aprendizajes logrados

- 1) Este proyecto nos sirvió para comprender cómo una base de datos bien diseñada puede simplificar y optimizar las operaciones de un negocio, en este caso, un punto de venta de una tienda.
- 2) Volvimos a practicar normalizar las tablas para evitar redundancias o guardar un mismo dato 2 veces, así como a establecer las relaciones apropiadas entre las diferentes tablas. Considero que ese concepto de normalización es uno de los conceptos más importantes de esta materia, ya que nos ayuda mucho a eficientar la base de datos, porque hay muchas cosas que se pueden resolver en un par de minutos con una búsqueda en google, pero el diseño de una base normalizada es algo que se tiene que hacer desde el inicio y no se puede saber el diseño exacto con una búsqueda en google por lo que es importante conocer el uso correcto de estas normas para aplicarlas desde el inicio.
- 3) Como ultima reflexión queremos recalcar la importancia de entender el funcionamiento del negocio para diseñar una buena base de datos, se tienen que tomar en cuenta muchos factores como nos menciona el profe, que si como se hacen los pagos, que sucede cuando se paga, como están clasificados los productos, que operaciones se quieren realizar, básicamente entender las demandas del negocio para poder hacer una base de datos que cubra todas las necesidades.

11. Fuentes bibliográficas

Normalization in SQL DBMS: 1NF, 2NF, 3NF, and BCNF Examples | PopSQL. (s. f.).

<https://popsql.com/blog/normalization-in-sql> Cómo citar en formato IEEE

<https://es.slideshare.net/terapeuta839/cmo-citar-ieee>

SQL stored procedures. (s. f.). https://www.w3schools.com/sql/sql_stored_procedures.asp

SQL LEFT JOIN keyword. (s. f.). https://www.w3schools.com/sql/sql_join_left.asp

SQL Stored Procedures (With Examples). (s. f.). <https://www.programiz.com/sql/stored-procedures>