

Documentación del Proyecto

Integrantes

Alejandro Gonzalez Silva

Juan Esteban Pereira

Miguel Ángel Pineda

TECNOLOGÍA DESARROLLO DE SOFTWARE BASES DE DATOS

Docente

Victor Manuel Vargas Forero

**Universidad del Valle
Sede regional Yumbo - 2023**

Proyecto

Realizar una Base de datos para gestión de inventarios de un MiniMarket (MiniMercado)

Requerimientos:

1. Control de Existencias (Inventario):

Mostrar la cantidad de existencias de los productos (Nombre Producto, Categoría, Stock, ID Producto).

2. Registro de productos (Inventario):

Registro de nuevos productos con: (Nombre Producto, Stock, Categoría, Precio, ID Producto).

3. Modificación de los productos (Inventario):

Modificación (editar, agregar, eliminar) de los productos existentes en base a los datos proporcionados anteriormente.

4. Facturación de venta:

Registra los detalles de una venta (ID venta, ID Producto, Cantidad Vendido, Precio Venta, Fecha de Venta)

5. Historial de ventas:

Registra los detalles de la facturación de la venta y se compara con el precio actual del producto (ID venta, ID Producto, Nombre Producto, Cantidad Vendida, Precio Venta, Fecha de Venta, Precio Producto)

6. Entrada (Inventario):

Registrar la entrada de productos al inventario con sus respectivos datos.

7. Salida (Inventario):

Registrar la salida de productos del inventario con sus respectivos datos.

8. Actualización de existencias (Inventario):

Se restarán los productos vendidos y se sumarán productos comprados.

9. Datos de proveedores:

Almacenar los datos de los proveedores de MiniMarket para hacer más pedidos a futuro (ID Proveedor, Nombre Proveedor, Dirección Proveedor, Teléfono Proveedor, Correo Proveedor).

10. Modificar datos de proveedores:

Se debe poder editar o eliminar los datos de los proveedores ingresados anteriormente.

11. Historial de compra a proveedores:

Registrar los detalles (ID Proveedor, Nombre Proveedor, ID Producto, ID Compra Proveedor, Stock Compra Proveedor, Fecha Compra Proveedor, Precio Compra Proveedor) de compra de un producto al proveedor.

12. Clasificación de Producto por Categorías:

Clasificar los productos en categorías para facilitar la búsqueda y análisis (enlatados, granos, verduras, frutas, carnes, licores, bebidas, miscelánea, productos de aseo, comestibles o dulces) .

13. Mínima Cantidad de Existencias:

Definir la mínima cantidad de existencias de los productos a partir de la subcategorías asignada a estos.

14. Alerta por pocas existencias de producto:

Se producirá una alerta cuando la cantidad de los productos esté por debajo de la cantidad mínima definida a partir de la SubCategoría asignada a esta.

15. Rol de empleado:

A cada empleado se le asigna un rol dependiendo de su funcionalidad dentro del MiniMarket como: (Administrador, tendero y bodega).

16. Datos de los empleados:

Se almacenarán los datos de los empleados del MiniMarket como (ID empleado, Nombre Empleado, Número Teléfono, Horario, Sueldo, Rol Empleado)

17. Modificar datos del empleado: Se debe poder eliminar o editar los datos de los empleados anteriormente ingresados.

18. Interfaz según el rol del empleado:

Una interfaz fácil e intuitiva para el uso de la base de datos, dependiendo del rol(sea administrador, tendero o almacenista) que tenga asignado cada empleado cada uno tendrá su respectivo usuario y contraseña.

19. Interfaz para el inventario:

Una interfaz donde se muestre el inventario de productos y que se pueda: Agregar, Editar y Eliminar.

20. Interfaz para el historial de ventas:

Una interfaz donde se muestre el historial de ventas realizadas.

21. Interfaz para los datos del proveedor:

Una interfaz donde se muestran los datos de los proveedores y que se pueda: Agregar, Editar y Eliminar.

22. Interfaz para los datos del empleado:

Una interfaz donde se muestran los datos de todos los empleados que trabajan en el MiniMarket y que se pueda: Agregar, Editar y Eliminar.

Esquema Relacional:

Entidades:

1. Inventario:

- ID_Producto (PK)
- Nombre_Producto
- Stock
- Precio
- ID_Categoría (FK)
- Categoría (FK)

2. Proveedores:

- ID_Proveedor (PK)
- Nombre_Proveedor
- Dirección_Proveedor
- Teléfono_Proveedor
- Correo_Proveedor

3. Rol_Empleado:

- ID_Rol (PK)
- Rol_Empleado

4. Empleados:

- ID_empleado (PK)
- Nombre_Empleado
- Teléfono_Empleado
- Horario
- Sueldo
- ID_Rol (FK)
- Rol Empleado

5. Salidas:

- ID_Salida (PK)
- ID_Producto (FK)
- Cantidad_Vendida

6. Entradas:

- ID_Entrada (PK)
- ID_Producto (FK)
- Cantidad_Compra

7. Historial_Ventas:

- ID_Venta (PK)
- ID_Producto (FK)
- Nombre_Producto
- ID_Salida (FK)
- Cantidad_Vendida
- Precio_Venta
- Fecha_Venta

8. Historial_Compra_Proveedor:

- ID_Compra_Proveedor (PK)
- ID_Proveedor (FK)
- Nombre_Proveedor
- ID_Producto (FK)
- Nombre_Producto
- ID_Entrada (FK)
- Cantidad_Compra
- Fecha_Compra
- Precio_Compra

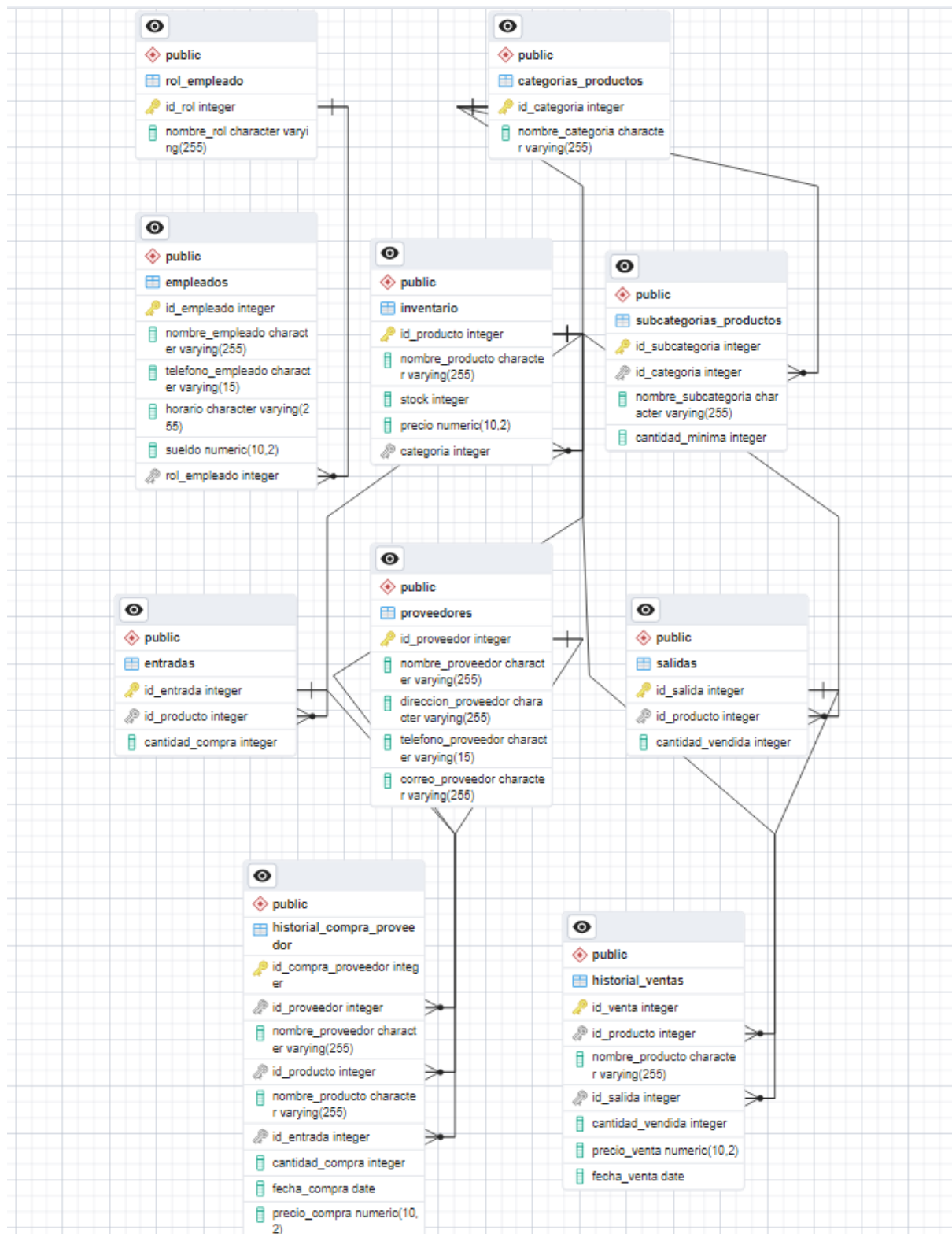
9. Categorías_Productos:

- ID_Categoría (PK)
- Nombre_Categoría
- Cantidad_Minima

Relaciones:

- SubCategoría referencia ID_SubCategoría en SubCategorías_Productos
- Rol_Empleado referencia ID_Rol en Rol_Empleado
- ID_Producto referencia ID_Producto en Inventario
- ID_Salida referencia ID_Salida en Salidas
- ID_Proveedor referencia ID_Proveedor en Proveedores
- ID_Entrada referencia ID_Entrada en Entradas
- ID_Categoría referencia ID_Categoría en Categorías_Productos

Modelo Entidad Relación



Consultas:

1. Listar todos los productos de una categoría específica:

```
SELECT * FROM Inventario
WHERE Categoria = (SELECT ID_Categoria FROM Categorías_Productos WHERE
Nombre_Categoria = 'Electrónicos');
```

2. Mostrar las categorías con la cantidad mínima de productos requerida:

```
SELECT * FROM Categorías_Productos
WHERE ID_Categoria IN (
    SELECT Categoria
    FROM Inventario
    GROUP BY Categoria
    HAVING MIN(Stock) <= 5
);
```

3. Encontrar los productos que tienen un precio superior al promedio de todos los productos:

```
SELECT * FROM Inventario
WHERE Precio > (SELECT AVG(Precio) FROM Inventario);
```

4. Obtener el nombre del proveedor que más productos ha suministrado:

```
SELECT Nombre_Proveedor FROM Proveedores
WHERE ID_Proveedor = (
    SELECT TOP 1 ID_Proveedor
    FROM Historial_Compra_Proveedor
    GROUP BY ID_Proveedor
    ORDER BY COUNT(ID_Producto) DESC
);
```

5. Mostrar los productos que han tenido al menos una venta:

```
SELECT * FROM Inventario
WHERE ID_Producto IN (SELECT DISTINCT ID_Producto FROM Historial_Ventas);
```

6. Calcular el total de ventas por empleado:

```
SELECT ID_Empleado, Nombre_Empleado, SUM(Cantidad_Vendida *
Precio_Venta) AS Total_Ventas
FROM Empleados
JOIN Historial_Ventas ON Empleados.ID_Empleado =
Historial_Ventas.ID_Empleado
GROUP BY ID_Empleado, Nombre_Empleado;
```

7. Encontrar los productos que nunca han sido vendidos:

```
SELECT * FROM Inventario
WHERE ID_Producto NOT IN (SELECT DISTINCT ID_Producto FROM
Historial_Ventas);
```

8. Calcular el total de compras realizadas por cada proveedor:

```
SELECT ID_Proveedor, Nombre_Proveedor, SUM(Cantidad_Compra *
Precio_Compra) AS Total_Compras
```

```

FROM Proveedores
JOIN Historial_Compra_Proveedor ON Proveedores.ID_Proveedor =
Historial_Compra_Proveedor.ID_Proveedor
GROUP BY ID_Proveedor, Nombre_Proveedor;

```

9. Listar las categorías que tienen productos con stock agotado

```

SELECT DISTINCT Categorías_Productos.*
FROM Categorías_Productos
JOIN Inventario ON Categorías_Productos.ID_Categoría = Inventario.Categoría
WHERE Stock = 0;

```

10. Mostrar los empleados que tienen el sueldo más alto:

```

SELECT * FROM Empleados
WHERE Sueldo = (SELECT MAX(Sueldo) FROM Empleados);

```

11. Obtener la cantidad total de productos en stock por categoría:

```

SELECT Categorías_Productos.Nombre_Categoría, SUM(Stock) AS Total_Stock
FROM Categorías_Productos
JOIN Inventario ON Categorías_Productos.ID_Categoría = Inventario.Categoría
GROUP BY Categorías_Productos.Nombre_Categoría;

```

12. Encontrar los productos que tienen un precio más bajo que el precio promedio de su categoría:

```

SELECT * FROM Inventario
WHERE Precio < (
    SELECT AVG(Precio)
    FROM Inventario AS I2
    WHERE I2.Categoría = Inventario.Categoría
);

```

13. Listar los productos que tienen un precio de venta superior al doble de su precio de compra:

```

SELECT * FROM Historial_Ventas
WHERE Precio_Venta > 2 * (SELECT Precio_Compra FROM
Historial_Compra_Proveedor WHERE ID_Producto = Historial_Ventas.ID_Producto);

```

14. Calcular el total de ventas diarias:

```

SELECT Fecha_Venta, SUM(Cantidad_Vendida * Precio_Venta) AS Total_Ventas
FROM Historial_Ventas
GROUP BY Fecha_Venta;

```

15. Mostrar los proveedores que han suministrado productos en más de una categoría:

```

SELECT DISTINCT P.*
FROM Proveedores P
JOIN Historial_Compra_Proveedor HCP ON P.ID_Proveedor = HCP.ID_Proveedor
JOIN Inventario I ON HCP.ID_Producto = I.ID_Producto
GROUP BY P.ID_Proveedor, I.Categoría
HAVING COUNT(DISTINCT I.Categoría) > 1;

```

16. Encontrar los productos más vendidos junto con la cantidad total vendida:


```

SELECT ID_Producto, Nombre_Producto, SUM(Cantidad_Vendida) AS
Total_Vendido
FROM Historial_Ventas
GROUP BY ID_Producto, Nombre_Producto
ORDER BY Total_Vendido DESC;

```

17. Listar los proveedores que no han tenido ninguna transacción de compra:

```

SELECT ID_Proveedor, Nombre_Proveedor
FROM Proveedores
WHERE ID_Proveedor NOT IN (SELECT DISTINCT ID_Proveedor FROM
Historial_Compra_Proveedor);

```

18. Calcular el promedio de sueldos por rol de empleado:

```

SELECT R.Nombre_Rol, AVG(E.Sueldo) AS Sueldo_Promedio
FROM Rol_Empleado R
JOIN Empleados E ON R.ID_Rol = E.Rol_Empleado
GROUP BY R.Nombre_Rol;

```

19. Mostrar los productos que han tenido al menos dos ventas con un intervalo de más de una semana entre ellas:

```

SELECT *
FROM Historial_Ventas H1
WHERE EXISTS (
    SELECT *
    FROM Historial_Ventas H2
    WHERE H1.ID_Producto = H2.ID_Producto
    AND H1.ID_Venta <> H2.ID_Venta
    AND ABS(DATEDIFF(day, H1.Fecha_Venta, H2.Fecha_Venta)) > 7
);

```

20. Listar los productos que han tenido una variación de stock mayor al 50% en los últimos tres meses:

```

SELECT I.*
FROM Inventario I
WHERE ABS((I.Stock - (SELECT Stock FROM Inventario WHERE ID_Producto =
I.ID_Producto AND Fecha_Venta < DATEADD(month, -3, GETDATE())) ) / I.Stock) >
0.5;

```

Link Programa:

<https://drive.google.com/drive/folders/1wBKw3AHrnsfrHymRT5k0H0Zi5jpQIUl8?usp=sharing>

