



PROYECTO FORMATIVO ADAX STORE MANAGER

Integrantes:

- Barbosa Gonzalez Brigitt Natalia
- Camacho Del Río Damián Alejandro
- Giraldo Torres Evelyn Stephanie
 - Martinez Molina Santiago
 - Trujillo Mondragon Sara Sofia



ADAX



ADAX

Elogan:
Códigos que transforma
negócios



www.sena.edu.co

@SENAComunica

TABLA DE CONTENIDO



- | | | | |
|-----------|----------------------------|-----------|----------------------|
| 01 | objetivo General | 04 | Pregunta problema |
| 02 | Objetivos específicos | 05 | Alcance del proyecto |
| 03 | Planteamiento del problema | 06 | Justificación |

TABLA DE **CONTENIDO**



Base de datos DDL



Funciones



Consultas



Control de versiones



Procedimientos



Objetivo General

Desarrollar un sistema de información software de Terminal Punto de Venta (TPV) para una tienda de un barrio, con el fin de automatizar y optimizar la gestión de ventas, inventario, comprobante de pago, clientes y pagos, incluyendo la integración de billeteras móviles como Nequi y Daviplata.



www.sena.edu.co

@SENAComunica



1. Generar un módulo de gestión de ventas que permita el registro eficiente de transacciones.
2. Implementar un módulo de control de inventario y fijación de precios.
3. Generar reportes estadísticos que permitan ver el estado de ventas
4. Garantizar la seguridad y privacidad de los datos.

Objetivos Específicos



@SENAComunica

www.sena.edu.co



La tienda de barrio “con mucho” enfrenta dificultades en sus procesos de registro de ventas y control de inventario. Actualmente, estos procesos se realizan manualmente, lo que resulta en varios problemas:

1. Registro manual de ventas
2. Falta de control de ingresos
3. Aumento de precios inconsistentes

Planteamiento del problema



Pregunta problema

¿Cómo diseñar e implementar un sistema de Terminal Punto de Venta(TPV) que permita a la tienda de barrio “Con Mucho Gusto” automatizar y optimizar los procesos de gestión de ventas, inventario, facturación, clientes y pagos, ¿incluyendo la implementación de billeteras móviles como Nequi y Daviplata?



Alcance del proyecto

Nuestro software planea ser usado en una gran parte de pequeñas tiendas sin llegar a otro tipo de mercado más grande. El software no incluye el desarrollo de una tienda en línea. Tampoco se contempla la implementación de funcionalidades adicionales más allá de las definidas



www.sena.edu.co

@SENAComunica



Justificación

Este proyecto es relevante porque aporta conocimientos sobre el diseño e implementación de software TPV, contribuyendo a la innovación tecnológica en el sector de tiendas de barrio. En Colombia, existen más de 300.000 tiendas de barrio, que representan el 40% de las ventas minoristas y requieren modernizarse para mejorar su eficiencia, rentabilidad, seguridad, satisfacción del cliente, cumplimiento de normas legales y facilitar opciones de pago digital.



www.sena.edu.co

@SENAComunica

Base de datos DDL (SQL)



```

adaxDB (1).sql
1 CREATE DATABASE IF NOT EXISTS `adaxstore` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci */;
2 USE `adaxstore`;
3 -- MySQL dump 10.13 Distrib 8.0.38, for Win64 (x86_64)
4 --
5 -- Host: 127.0.0.1 Database: adaxstore
6 --
7 -- Server version 5.5.5-10.4.32-MariaDB
8
9 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
10 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
11 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
12 /*!40053 SET NAMES utf8 */;
13 /*!40103 SET @OLD_TIME_ZONE=@TIME_ZONE */;
14 /*!40103 SET TIME_ZONE='+00:00' */;
15 /*!40001 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
16 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
17 /*!40101 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
18 /*!40111 SET @OLD_SQL_NOTES=@SQL_NOTES, SQL_NOTES=0 */;

20 --
21 -- Estructura de la tabla `cliente`
22 --
23
24 DROP TABLE IF EXISTS `cliente`;
25 /*!40101 SET @saved_cs_client      = @@character_set_client */;
26 /*!50503 SET character_set_client = utf8mb4 */;
27 CREATE TABLE `cliente` (
28   `id_Cliente` int(11) NOT NULL,
29   `Documento` int(10) NOT NULL,
30   `Nombre1_Cliente` varchar(20) NOT NULL,
31   `Nombre2_Cliente` varchar(20) NOT NULL,
32   `Apellido1_Cliente` varchar(20) NOT NULL,
33   `Apellido2_Cliente` varchar(20) NOT NULL,
34   `Tipo_documento` varchar(15) NOT NULL,
35   PRIMARY KEY (`id_Cliente`)
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
37 /*!40101 SET character_set_client = @saved_cs_client */;

40 -- Exportar datos para la tabla `cliente`
41 --
42
43 LOCK TABLES `cliente` WRITE;
44 /*!40000 ALTERAR LA TABLA `cliente` DESABILITAR CLAVES*/;
45 INSERT INTO `cliente` VALUES (1,1012345678,'Juan','Pablo','Garcia','Mendoza','CC'),(2,1012345680,'Ana','Maria','Martinez','Rodriguez','CC');
46 /*!40000 ALTERAR LA TABLA `cliente` TECLAS DE HABILITACIÓN */;
47 UNLOCK TABLES;
48
49 --
50 -- Estructura de la tabla `entregaproductos`
51 --
52
53 DROP TABLE IF EXISTS `entregaproductos`;
54 /*!40101 SET @saved_cs_client      = @@character_set_client */;
55 /*!50503 SET character_set_client = utf8mb4 */;
56 CREATE TABLE `entregaproductos` (
57   `proveedor_idproveedor` int(11) NOT NULL,
58   `producto_id_Producto` int(11) NOT NULL,
59   `fecha_Entrega` date DEFAULT NULL,
60   `cantidad` int(11) DEFAULT NULL,
61   PRIMARY KEY (`proveedor_idproveedor`, `producto_id_Producto`),
62   KEY `fk_entregaproductos_producto1_idx` (`producto_id_Producto`),
63   CONSTRAINT `fk_entregaproductos_producto1` FOREIGN KEY (`producto_id_Producto`) REFERENCES `producto` (`id_Producto`) ON DELETE NO ACTION
64   CONSTRAINT `fk_entregaproductos_proveedor1` FOREIGN KEY (`proveedor_idproveedor`) REFERENCES `proveedor` (`idproveedor`) ON DELETE NO ACTION
65 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
66 /*!40101 SET character_set_client = @saved_cs_client */;

68 --
69 -- Exportar datos para la tabla `entregaproductos`
70 --
71
72 LOCK TABLES `entregaproductos` WRITE;
73 /*!40000 ALTERAR LA TABLA `entregaproductos` DESHABILITAR TECLAS*/;
74 INSERT INTO `entregaproductos` VALUES (1,1,'2024-08-01',100),(2,2,'2024-08-02',150),(3,3,'2024-08-03',200),(4,4,'2024-08-04',250),(5,5,'2024-08-05',300);
75 /*!40000 ALTERAR LA TABLA `entregaproductos` ENABLE KEYS */;
76 UNLOCK TABLES;

```



```
79 -- Estructura de la tabla 'factura'
80 --
81
82 DROP TABLE IF EXISTS `factura`;
83 /*!40101 SET @saved_cs_client      = @@character_set_client */;
84 /*!50503 SET character_set_client = utf8mb4 */;
85 CREATE TABLE `factura` (
86     `venta_id_Venta` int(11) NOT NULL,
87     `producto_id_Producto` int(11) NOT NULL,
88     `Cantidad` int(11) NOT NULL,
89     `Precio` double NOT NULL,
90     `Estado` varchar(45) NOT NULL,
91     PRIMARY KEY (`venta_id_Venta`, `producto_id_Producto`),
92     KEY `fk_detalles_venta_producto_idx` (`producto_id_Producto`),
93     CONSTRAINT `fk_detalles_venta_producto` FOREIGN KEY (`producto_id_Producto`) REFERENCES `producto`(`id_Producto`) ON DELETE NO ACTION
94     CONSTRAINT `fk_detalles_venta_venta` FOREIGN KEY (`venta_id_Venta`) REFERENCES `venta`(`id_Venta`) ON DELETE NO ACTION ON UPDATE NO A
95 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
96 /*!40101 SET character_set_client = @saved_cs_client */;
97
98 --
99 -- Exporta datos de la tabla 'factura'
100 --
101
102 LOCK TABLES `factura` WRITE;
103 /*!40000 ALTER TABLE `factura` DESHABILITAR TECLAS */;
104 INSERT INTO `factura` VALUES (1,1,2950,'Pagado'),(1,2,1750,'Pagado'),(2,3,1,1200,'Pendiente'),(2,4,1,800,'Pendiente'),(3,5,3,150,'Pagad
105 /*!40000 ALTER TABLE `factura` TECLAS DE HABILITACIÓN */;
106 UNLOCK TABLES;
107
```



www.sena.edu.co

@SENAComunica

Base de datos DDL (SQL)

```
--Estructura de la tabla 'inventario'
--
--DROP TABLE IF EXISTS `inventario`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `inventario` (
    `id_Inventario` int(11) NOT NULL,
    `cantidadInventario` int(11) DEFAULT NULL,
    `fechaModificacion` date DEFAULT NULL,
    `estado_revision` varchar(15) NOT NULL,
    `tienda_idtienda` int(11) NOT NULL,
    PRIMARY KEY (`id_Inventario`),
    KEY `fk_inventario_idtienda` (`tienda_idtienda`),
    CONSTRAINT `fk_inventario_idtienda` FOREIGN KEY (`tienda_idtienda`) REFERENCES `tienda`(`idtienda`) ON DELETE NO ACTION ON UPDATE NO AC
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Exportar datos de la tabla 'inventario'
--

LOCK TABLES `inventario` WRITE;
/*!40000 ALTER TABLE `inventario` DESHABILITAR TECLAS */;
INSERT INTO `inventario` VALUES (1,150,'2024-08-10','Revisado'),(2,80,'2024-08-11','Revisado'),(3,200,'2024-08-12','Revisado'),(4,0
/*!40000 ALTER TABLE `inventario` HABILITAR TECLAS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results     = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client  = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_general_ci */;
/*!50003 SET @saved_sql_mode       = @@sql_mode */;
/*!50003 SET sql_mode            = 'NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;
/*!50003 CREATE*/ /*!50017 DEFINER='root'@'localhost'*/ /*!50003 trigger registrar_movimiento
```



```
/*!50003 CREATE*/ /*!50017 DEFINER='root'@'localhost' */ /*!50003 trigger registrar_movimiento
after insert on inventario
for each row
begin
insert into movimientos(cantidad_despues,fecha_movimiento,fecha_modificacion,estado_despues,inventario_id_Inventario)
values(new.cantidadInventario,current_date(),new.fechaModificacion,new.estado_revision,new.id_Inventario);
end */;;
DELIMITER ;
/*!50003 SET sql_mode      = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;

-- 
-- Estructura de la tabla `metododepago`
-- 

DROP TABLE IF EXISTS `metododepago`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `metododepago` (
  `ID_Met_pago` int(11) NOT NULL,
  `Nombre` varchar(45) NOT NULL,
  PRIMARY KEY (`ID_Met_pago`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Exportar datos de la tabla `metododepago`
-- 

LOCK TABLES `metododepago` WRITE;
/*!40000 ALTER TABLA `metododepago` DESHABILITAR TECLAS*/;
INSERT INTO `metododepago` VALUES (1,'tarjeta de credito'),(2,'tarjeta de debito'),(3,'nequi'),(4,'Daviplata'),(5,'Efectivo');
/*!40000 ALTER TABLA `metododepago` HABILITAR TECLAS*/;
UNLOCK TABLES;
```



Consulta para obtener el valor total de ventas por cliente y producto comprado

```
1 •  SELECT c.Nombre1_Cliente, c.Apellido1_Cliente, p.Nombre, p.Descripción, SUM(f.Cantidad * f.Precio) AS Total  
2   FROM factura f  
3   JOIN producto p ON f.producto_id_Producto=p.id_Producto  
4   JOIN venta v ON f.venta_id_Venta=v.id_Venta  
5   JOIN cliente c ON v.cliente_id_Cliente=c.id_Cliente  
6   GROUP BY c.Nombre1_Cliente, c.Apellido1_Cliente;  
7
```

Nombre1_Cliente	Apellido1_Cliente	Nombre	Descripción	Total
Fernando	Zapata	Plancha T-fal	Plancha de vapor antiadherente	410
Isabella	Acosta	Televisor LG OLED	Televisor con pantalla OLED de 55 pulgadas	3200
Javier	Cordero	Plancha T-fal	Plancha de vapor antiadherente	590
Jorge	Ramírez	Plancha T-fal	Plancha de vapor antiadherente	410
Juan	García	Laptop Dell Inspiron	Laptop con procesador Intel i5	2650
Juliana	Gómez	Aspiradora Rowenta	Aspiradora de 1500W con bolsa	530
Laura	López	Cámara Canon EOS	Cámara réflex digital con lente de 18-55mm	1320
Lina	Ríos	Televisor LG OLED	Televisor con pantalla OLED de 55 pulgadas	2000
Marta	Morales	Laptop Dell Inspiron	Laptop con procesador Intel i5	2450
Mauricio	Salazar	Cámara Canon EOS	Cámara réflex digital con lente de 18-55mm	1320
Natalia	Herrera	Plancha T-fal	Plancha de vapor antiadherente	460
Óscar	Cárdenas	Aspiradora Rowenta	Aspiradora de 1500W con bolsa	460
Paola	Reyes	Aspiradora Rowenta	Aspiradora de 1500W con bolsa	460
Sandra	Guerrero	Plancha T-fal	Plancha de vapor antiadherente	280
Sergio	Castillo	Laptop Dell Inspiron	Laptop con procesador Intel i5	3600
Verónica	García	Laptop Dell Inspiron	Laptop con procesador Intel i5	2450



Consulta para obtener el detalle de una venta específica

```
1 • SELECT v.id_Venta, v.FechaVenta, c.Nombre1_Cliente, c.Nombre2_Cliente, c.Documento, p.Nombre, f.Cantidad, SUM(f.Precio) as total  
2   FROM factura f  
3   JOIN producto p ON f.producto_id_Producto=p.id_Producto  
4   JOIN venta v ON f.venta_id_Venta=v.id_Venta  
5   JOIN cliente c ON v.cliente_id_Cliente=c.id_Cliente  
6   WHERE v.id_Venta=1  
7   ORDER by total;
```

	id_Venta	FechaVenta	Nombre1_Cliente	Nombre2_Cliente	Documento	Nombre	Cantidad	total
1	2024-05-12	Juan	Pablo	1012345678	Laptop Dell Inspiron	2	1700	



Consulta que le permita al cliente ganar una promoción si hace una compra mayor a 200.000

```
1 •  SELECT c.id_Cliente, c.Nombre1_Cliente, c.Nombre2_Cliente, c.Documento, p.Nombre, SUM(f.Precio) AS total_venta,
2   CASE
3     WHEN SUM(f.Precio) > 200 THEN 'El cliente se ganó una promoción en uno de sus productos'
4     ELSE 'no hay promocion'
5   END AS promocion
6   FROM factura f
7   JOIN producto p ON f.producto_id_Producto=p.id_Producto
8   JOIN venta v ON f.venta_id_Venta=v.id_Venta
9   JOIN cliente c ON v.cliente_id_Cliente=c.id_Cliente
10  GROUP BY c.id_Cliente, c.Nombre1_Cliente, p.Nombre;
```

id_Cliente	Nombre1_Cliente	Nombre2_Cliente	Documento	Nombre	total_venta	promocion
1	Juan	Pablo	1012345678	Laptop Dell Inspiron	950	El cliente se ganó una promoción en uno de sus
1	Juan	Pablo	1012345678	Smartphone Samsung Galaxy	750	El cliente se ganó una promoción en uno de sus
2	Ana	Maria	1012345680	Refrigerador Samsung	800	El cliente se ganó una promoción en uno de sus
2	Ana	Maria	1012345680	Televisor LG OLED	1200	El cliente se ganó una promoción en uno de sus
3	Carlos	Andrés	1023456789	Aspiradora Rowenta	150	no hay promocion
3	Carlos	Andrés	1023456789	Cafetera Philips	80	no hay promocion
4	Laura	Alejandra	1034567801	Cámara Canon EOS	600	El cliente se ganó una promoción en uno de sus
4	Laura	Alejandra	1034567801	Microwave Panasonic	120	no hay promocion
5	Jorge	Luis	1045678902	Air Fryer Philips	180	no hay promocion
5	Jorge	Luis	1045678902	Plancha T-fal	50	no hay promocion
6	Marta	Lucía	1056789013	Laptop Dell Inspiron	950	El cliente se ganó una promoción en uno de sus
6	Marta	Lucía	1056789013	Smartphone Samsung Galaxy	750	El cliente se ganó una promoción en uno de sus
7	David	Alejandro	1067890124	Refrigerador Samsung	800	El cliente se ganó una promoción en uno de sus
7	David	Alejandro	1067890124	Televisor LG OLED	1200	El cliente se ganó una promoción en uno de sus
8	Paola	Andrea	1078901235	Aspiradora Rowenta	150	no hay promocion
8	Paola	Andrea	1078901235	Cafetera Philips	80	no hay promocion
9	Felipe	Javier	1089012346	Cámara Canon EOS	600	El cliente se ganó una promoción en uno de sus
9	Felipe	Javier	1089012346	Microwave Panasonic	120	no hay promocion
10	Sandra	Paola	1090123457	Air Fryer Philips	180	no hay promocion
10	Sandra	Paola	1090123457	Plancha T-fal	50	no hay promocion



Consulta para obtener el nombre del producto, la fecha de venta, el método de pago y el total vendido por producto en un mes específico

```
1 •  SELECT p.Nombre, v.FechaVenta, mp.Nombre, SUM(f.Cantidad) AS total_vendido
2   FROM factura f
3   JOIN producto p ON f.producto_id_Producto=p.id_Producto
4   JOIN venta v ON f.venta_id_Venta=f.venta_id_Venta
5   JOIN metododepago mp ON v.metododepago_ID_Met_pago=mp.ID_Met_pago
6   WHERE month(v.FechaVenta)= '06'
7   GROUP BY p.Nombre, v.FechaVenta
8   ORDER BY total_vendido DESC;
```

Nombre	FechaVenta	Nombre	total_vendido
Aspiradora Rowenta	2024-06-07	tarjeta de debito	11
Aspiradora Rowenta	2024-06-04	Daviplata	11
Aspiradora Rowenta	2024-06-06	tarjeta de credito	11
Aspiradora Rowenta	2024-06-10	Efectivo	11
Aspiradora Rowenta	2024-06-03	nequi	11
Aspiradora Rowenta	2024-06-09	Daviplata	11
Aspiradora Rowenta	2024-06-02	tarjeta de debito	11
Aspiradora Rowenta	2024-06-08	nequi	11
Aspiradora Rowenta	2024-06-01	tarjeta de credito	11
Aspiradora Rowenta	2024-06-05	Efectivo	11
Air Fryer Philips	2024-06-06	tarjeta de credito	10
Air Fryer Philips	2024-06-10	Efectivo	10
Air Fryer Philips	2024-06-03	nequi	10
Air Fryer Philips	2024-06-09	Daviplata	10
Air Fryer Philips	2024-06-02	tarjeta de debito	10
Air Fryer Philips	2024-06-08	nequi	10
Air Fryer Philips	2024-06-01	tarjeta de credito	10
Air Fryer Philips	2024-06-05	Efectivo	10
Air Fryer Philips	2024-06-07	tarjeta de debito	10
Air Fryer Philips	2024-06-04	Daviplata	10
Laptop Dell Inspiron	2024-06-07	tarjeta de debito	9



Consulta para obtener los detalles de las facturas junto con los nombres de los clientes y los métodos de pago utilizados

```
1 •  SELECT f.venta_id_Venta,concat(c.Nombre1_Cliente,' ',c.Apellido2_Cliente) as Nombre,f.Cantidad,f.precio,f.estado,mp.Nombre  
2 From factura f  
3 inner join venta v on f.venta_id_Venta = v.id_Venta  
4 inner join cliente c on v.cliente_id_Cliente = c.id_Cliente  
5 inner join metododepago mp on v.metododepago_ID_Net_pago = mp.ID_Net_pago  
6 group by venta_id_Venta,Nombre,mp.Nombre  
7 order by venta_id_Venta,Nombre,mp.Nombre
```

	venta_id_Venta	Nombre	Cantidad	precio	estado	Nombre
▶	1	JuanMendoza	2	950	Pagado	tarjeta de credito
2	AnaRodríguez	1	1200	Pendiente	tarjeta de debito	
3	CarlosPérez	3	150	Pagado	nequi	
4	LauraCastro	2	600	Pendiente	Daviplata	
5	JorgeSerrano	1	50	Pagado	Efectivo	
6	MartaOspina	1	950	Pagado	tarjeta de credito	
7	DavidCórdoba	1	1200	Pendiente	tarjeta de debito	
8	PaolaSuárez	2	150	Pagado	nequi	
9	FelipeCano	1	600	Pendiente	Daviplata	
10	SandraNaranjo	2	50	Pagado	Efectivo	
11	SergioVargas	3	950	Pagado	tarjeta de credito	
12	IsabellaMolina	2	1200	Pendiente	tarjeta de debito	
13	AndrésCastro	1	150	Pagado	nequi	
14	DanielaSalazar	2	600	Pendiente	Daviplata	
15	JavierJaramillo	1	50	Pagado	Efectivo	
16	VerónicaHurtado	1	950	Pagado	tarjeta de credito	
17	EstebanTorres	1	1200	Pendiente	tarjeta de debito	
18	JulianaValencia	3	150	Pagado	nequi	
19	MauricioPineda	2	600	Pendiente	Daviplata	
20	NataliaPalacio	2	50	Pagado	Efectivo	
21	AlejandroOrd...	2	950	Pagado	tarjeta de credito	
22	LinaCastillo	1	1200	Pendiente	tarjeta de debito	
23	ÁlvaroMárquez	?	150	Demandado	nequi	



Consulta para obtener el nombre del cliente y la cantidad de cada producto que se ha comprado en una tienda específica

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0436 segundos.)

```
SELECT c.Nombre1_Cliente, c.Nombre2_Cliente, p.Nombre, f.Cantidad FROM cliente c INNER JOIN venta v ON c.id_Cliente = v.cliente_id_Cliente INNER JOIN factura f ON v.id_Venta = f.venta_id_Venta INNER JOIN producto p ON f.producto_id_Producto = p.id_Producto WHERE v.tienda_idTienda = (SELECT idTienda FROM tienda WHERE nombreTienda = 'Supermercado El Barrio');
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

Nombre1_Cliente	Nombre2_Cliente	Nombre	Cantidad
Juan	Pablo	Laptop Dell Inspiron	2
Juan	Pablo	Smartphone Samsung Galaxy	1
Marta	Lucía	Laptop Dell Inspiron	1
Marta	Lucía	Smartphone Samsung Galaxy	2
Sergio	Iván	Laptop Dell Inspiron	3
Sergio	Iván	Smartphone Samsung Galaxy	1
Verónica	Sánchez	Laptop Dell Inspiron	1
Verónica	Sánchez	Smartphone Samsung Galaxy	2
Alejandro	José	Laptop Dell Inspiron	2
Alejandro	José	Smartphone Samsung Galaxy	3



Consulta para obtener el nombre del producto y la cantidad vendida por cada proveedor

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0386 segundos.)

```
SELECT p.Nombre, SUM(f.Cantidad) AS CantidadVendida FROM producto p INNER JOIN factura f ON p.id_Producto = f.producto_id_Producto INNER JOIN venta v ON f.venta_id_Venta = v.id_Venta INNER JOIN entregaproductos ep ON p.id_Producto = ep.producto_id_Producto INNER JOIN proveedor prov ON ep.proveedor_idproveedor = prov.idproveedor WHERE prov.nombre = 'Electrodomésticos del Norte' GROUP BY p.Nombre;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

Nombre	CantidadVendida
Refrigerador Samsung	5



Consulta para revisar el estado del inventario de una venta y productos que están por agotarse

Mostrando filas 0 - 24 (total de 30, La consulta tardó 0.0020 segundos.)

```
SELECT t.nombreTienda AS Nombre_Tienda, p.Nombre AS Nombre_Producto, i.cantidadInventario AS Cantidad_Disponible, p.Stock_Min AS Stock_Minimo, CASE WHEN i.cantidadInventario <= p.Stock_Min THEN 'Producto por agotarse' ELSE 'Suficiente inventario' END AS Estado_Inventory FROM movimientos m JOIN tienda t ON m.id_tienda = t.idtienda JOIN inventario i ON t.idtienda = i.tienda_idtienda JOIN producto p ON i.id_Inventario = p.inventario_id_Inventario WHERE i.cantidadInventario IS NOT NULL;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 < > >> | Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

Nombre_Tienda	Nombre_Producto	Cantidad_Disponible	Stock_Minimo	Estado_Inventory
Supermercado El Barrio	Laptop Dell Inspiron	150	10	Suficiente inventario
Supermercado El Barrio	Laptop Dell Inspiron	150	10	Suficiente inventario
ElectroShop Bogotá	Smartphone Samsung Galaxy	80	20	Suficiente inventario
ElectroShop Bogotá	Smartphone Samsung Galaxy	80	20	Suficiente inventario
Librería El Saber	Televisor LG OLED	200	5	Suficiente inventario
Librería El Saber	Televisor LG OLED	200	5	Suficiente inventario
Farmacia La Salud	Refrigerador Samsung	80	8	Suficiente inventario
Moda y Estilo	Aspiradora Rowenta	120	15	Suficiente inventario
Moda y Estilo	Aspiradora Rowenta	120	15	Suficiente inventario
Deportes y Más	Cafetera Philips	90	20	Suficiente inventario
Deportes y Más	Cafetera Philips	90	20	Suficiente inventario
Deportes y Más	Cafetera Philips	90	20	Suficiente inventario
Tienda de Abarrotes La Familia	Cámara Canon EOS	100	5	Suficiente inventario
Muebles y Decoración	Microwave Panasonic	75	10	Suficiente inventario
Muebles y Decoración	Microwave Panasonic	75	10	Suficiente inventario
Café Gourmet	Teclado Logitech	85	25	Suficiente inventario
Ropa y Moda	Ratón Logitech	95	30	Suficiente inventario
Ropa y Moda	Ratón Logitech	95	30	Suficiente inventario
Ropa y Moda	Ratón Logitech	95	30	Suficiente inventario



procedimiento para buscar un producto por su popularidad

✓ Su consulta se ejecutó con éxito.
10 filas afectadas por la última sentencia del procedimiento.

```
SET @p0='no popular'; CALL `BucarProductospopularidad`(@p0);
```

Resultados de la ejecución de la rutina 'BucarProductospopularidad'

id_Producto	nombre	cantidad	categoria_popularidad
5	Aspiradora Rowenta	11	No Popular
10	Air Fryer Philips	10	No Popular
2	Smartphone Samsung Galaxy	9	No Popular
1	Laptop Dell Inspiron	9	No Popular
7	Cámara Canon EOS	8	No Popular
6	Cafetera Philips	8	No Popular
9	Plancha T-fal	7	No Popular
3	Televisor LG OLED	6	No Popular
4	Refrigerador Samsung	5	No Popular
8	Microwave Panasonic	5	No Popular

Nombre de rutina: BucarProductospopularidad

Tipo: PROCEDURE

Dirección	Nombre	Tipo	Longitud/Valores	Opciones
IN	Nivel_popu	VARCHAR	20	Juego de caracteres

Parámetros:

Definición:

```
1 BEGIN
2   SELECT p.id_Producto, p.nombre, SUM(f.cantidad) AS cantidad,
3   CASE
4     WHEN cantidad > 20 THEN "popular"
5     WHEN cantidad BETWEEN 10 AND 20 THEN 'Medio Popular'
6     ELSE 'No Popular'
7   END AS categoria_popularidad
8   FROM factura f
9   JOIN producto p ON f.producto_id_Producto = p.id_Producto
10  GROUP BY p.id_Producto, p.nombre
11  HAVING categoria_popularidad = Nivel_popu
12  ORDER BY cantidad DESC;
13 END
```



procedimiento para buscar factura por el id

Estructura SQL Buscar Generar una const...

Su consulta se ejecutó con éxito.
2 filas afectadas por la última sentencia del procedimiento.

```
SET @p0='3'; CALL `BuscarFactura`(@p0);
```

Resultados de la ejecución de la rutina 'BuscarFactura'

venta_id_Venta	producto_id_producto	estado
3	5	Pagado
3	6	Pagado

Nombre de rutina: BuscarFactura
Tipo: PROCEDURE

Dirección	Nombre	Tipo	Longitud/Valores
IN	id_fac	INT	11

Parámetros:

Agregar parámetro

```
1 BEGIN
2 SELECT venta_id_Venta, producto_id_producto, estado FROM factura
3 WHERE venta_id_Venta=id_fac;
4 END
```



procedimiento para buscar las ventas entre dos fechas

Detalles

Nombre de rutina	Fechasdesdehasta		
Tipo	PROCEDURE ▾		
Dirección	Nombre	Tipo	Longitud/V.
Parámetros	IN	fecha1	DATE
	IN	fecha2	DATE

Agregar parámetro

```

1 BEGIN
2 SELECT fecha1,fecha2
3 FROM venta
4 WHERE FechaVenta between fecha1 AND fecha2;
5 END
    
```



procedimiento que permita mostrar el proveedor de un producto

Detalles

Nombre de rutina: monstrar_proveedor_de_p

Tipo: PROCEDURE

Dirección	Nombre	Tipo	Longitud/Valores
IN	p_id_Producto	INT	11

Parámetros:

Agregar parámetro

```
1 BEGIN
2 select producto.nombre as nombre_producto,
3 proveedor.nombre as nombre_proveedor
4 from producto
5 join entregaproductos on producto.id_Producto = entregaproductos.producto_id_Producto
6 join proveedor on entregaproductos.proveedor_idproveedor = proveedor.idproveedor
7 where id_producto = p_id_producto;
8 END
```

Continuar Cerrar

✓ Su consulta se ejecutó con éxito.
1 fila afectada por la última sentencia del procedimiento.

```
SET @p0='3'; CALL `monstrar_proveedor_de_producto`(@p0);
```

Resultados de la ejecución de la rutina 'monstrar_proveedor_de_producto'

nombre_producto	nombre_proveedor
Televisor LG OLED	TechnoGadgets



función que permita ver el total de venta por dia

✓ Su consulta se ejecutó con éxito.

```
SET @p0='2024-05-15'; SELECT `Total_venta`(@p0) AS `Total_venta`;
```

Resultados de la ejecución de la rutina 'Total_venta'

Total_venta
720.00

Detalles

Nombre de rutina	Total_venta
Tipo	FUNCTION
Parámetros	
Nombre	dia_venta
Tipo	DATE
Longitud/Valores	---
Opciones	<input checked="" type="checkbox"/> Eliminar

Agregar parámetro

Retornar el tipo	DECIMAL
Retornar longitud/valores	10,2
Retornar opciones	

Definición

```
1 BEGIN
2 DECLARE total_ventas DECIMAL(10, 2);
3 DECLARE mensaje VARCHAR(30);
4 SELECT SUM(factura.Precio) INTO total_ventas
5 FROM factura
6 JOIN venta ON factura.venta_id_Venta=venta.id_Venta
7 WHERE DATE(venta.FechaVenta)= dia_venta;
8 IF total_ventas IS NULL THEN
9 SET mensaje = 'no se hicieron ventas ese dia';
10 RETURN 0;
11 ELSE
12 RETURN total_ventas;
13 END IF;
14 END
```

Continuar **Cerrar**



función para obtener el historial de movimiento

✓ Su consulta se ejecutó con éxito.

```
SET @p0='2'; SELECT `obtener_historial_movimientos`(@p0) AS `obtener_historial_movimientos`
```

Resultados de la ejecución de la rutina 'obtener_historial_movimientos'

obtener_historial_movimientos
2000.00

Detalles

Nombre de rutina: obtener_historial_movimie
Tipo: FUNCTION

Nombre	Tipo	Longitud/Valores	Opcional
id_Venta	INT	11	

Parámetros: id_Venta

Retornar el tipo: DECIMAL
Retornar longitud/valores: 10,2
Retornar opciones:

```
1 BEGIN
2     DECLARE costo_total DECIMAL(10,2);
3     SELECT SUM(Cantidad * Precio) INTO costo_total
4     FROM Factura
5     WHERE venta_id_Venta = id_Venta;
6     RETURN costo_total;
7 END
```

Continuar Cerrar



función para obtener el producto más vendido

✓ Su consulta se ejecutó con éxito.

```
SELECT `producto_mas_vendido`() AS `producto_mas_vendido`;
```

Resultados de la ejecución de la rutina 'producto_mas_vendido'

producto_mas_vendido
5



www.sena.edu.co

@SENAComunica

Detalles

Nombre	Tipo	Longitud/Valores	Opciones
producto_mas_vendido	FUNCTION		

Parámetros

Retornar el tipo: INT
Retornar longitud/valores: 11
Retornar opciones:

Definición:

```
1 BEGIN
2     DECLARE id INT;
3     SELECT
4         f.producto_id_Producto
5         INTO
6             id
7         FROM
8             factura f
9             GROUP BY
10                f.producto_id_Producto
11             ORDER BY
12                SUM(f.Cantidad) DESC
13             LIMIT 1;
14             RETURN id;
15 END
```



función para obtener stock de un producto

Estructura SQL Buscar Generar una consulta Exportar

✓ Su consulta se ejecutó con éxito.

```
SET @p0='3'; SELECT `obtener_stock_producto`(@p0) AS `obtener_stock_producto`;
```

Resultados de la ejecución de la rutina 'obtener_stock_producto'

obtener_stock_producto
30

Detalles

Nombre de rutina: obtener_stock_producto
Tipo: FUNCTION

Nombre	Tipo	Longitud/Valores
p_id_producto	INT	11

Parámetros: Agregar parámetro

Retornar el tipo: INT
Retornar longitud/valores: 11
Retornar opciones:

```
1 BEGIN
2     DECLARE p_stock INT;
3     SELECT stock INTO p_stock FROM producto WHERE id_producto = p_id_producto;
4     RETURN p_stock;
5 END
```

Continuar Cerrar



función para ver el total de clientes que han realizado compras

✓ Su consulta se ejecutó con éxito.

```
SELECT `total_cliente`() AS `total_cliente`;
```

Resultados de la ejecución de la rutina 'total_cliente'

total_cliente
30

Detalles

Nombre	Tipo	Longitud/Valores	Opciones
total_cliente	FUNCTION		

Parámetros

Retornar el tipo: INT

Retornar longitud/valores: 11

Retornar opciones:

```
1 BEGIN
2 DECLARE total_clientes INT;
3 SELECT COUNT(id_cliente) INTO total_clientes
4 FROM cliente;
5 RETURN total_clientes;
```



Control de versiones

Nuestro software tendrá un control de versiones administrado en la plataforma Git y GitHub.



<https://github.com/Xfezor/ADAX-Store-Manager>



G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co