

BASE DE DATOS 1

HITO 5



Integrantes:

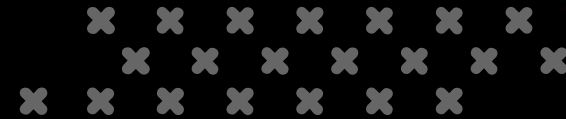
Jhonatan David Alanoca Blanco

Mijail Oliver Choque Amaro

Karen Alisson Chuquimia Mamani

Edson Iver Condori Condori

Jhanpol Machaca Aranibar



INTRODUCCION

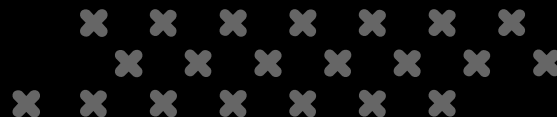
Como parte introductoria de la Defensa Final del Hito 5, nos centraremos en el caso de como las tiendas requieren de una buena administración para el control de la venta de sus productos, para lo cual se aplicara todos los conocimientos adquiridos en la materia de Base de Datos 1, para de esa manera crear un Sistema de Ventas con el objetivo de dar solución a esta necesidad.



ANALISIS Y DISEÑO DE LA BASE DE DATOS


Contexto: Dado a lo descrito anteriormente en la introducción **Spartan Game Shop (Nombre Ficticio)**, es un sistema de ventas creado para una tienda de Tecnología, este sistema permite facilitar la Administración de la misma además de que la hace eficiente al momento de resguardar los datos.

Spartan Game Shop




IDENTIFICANDO LAS TABLAS NECESARIAS

LoginGerente

	Id_Login_Gerente
	Contraseña
	Usuario

LoginEmpleado

	id_Login_Empleado
	Contraseña
	Usuario

PersonalGerente

	Id_Gerente
	Nombre_Gerente
	Apellido_Gerente
	Sueldo_Gerente
	Turno_Gerente

PersonalEmpleado

	Id_Empleado
	Nombre_Empleado
	Apellido_Empleado
	Sueldo_Empleado
	Turno_Empleado

IDENTIFICANDO LAS TABLAS NECESARIAS

Inventario

Id_Inventario
Id_Productos
Productos
Marcas

Ventas

Id_Ventas
Id_Cliente
Id_Productos
Cuenta_a_Pagar
Cuenta_Pagada
Total_a_Pagar

Clientes

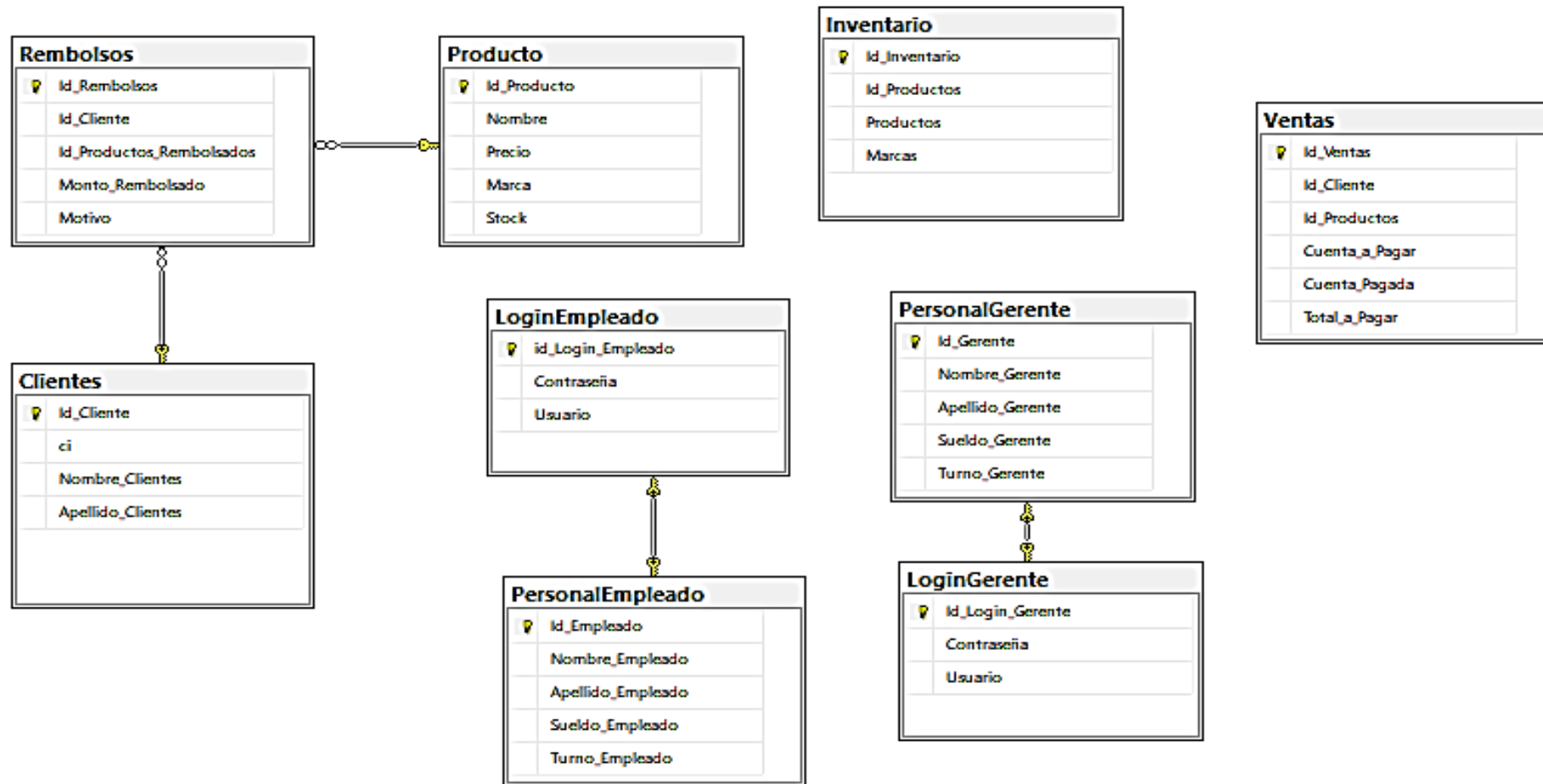
Id_Cliente
ci
Nombre_Clientes
Apellido_Clientes

Rembolsos

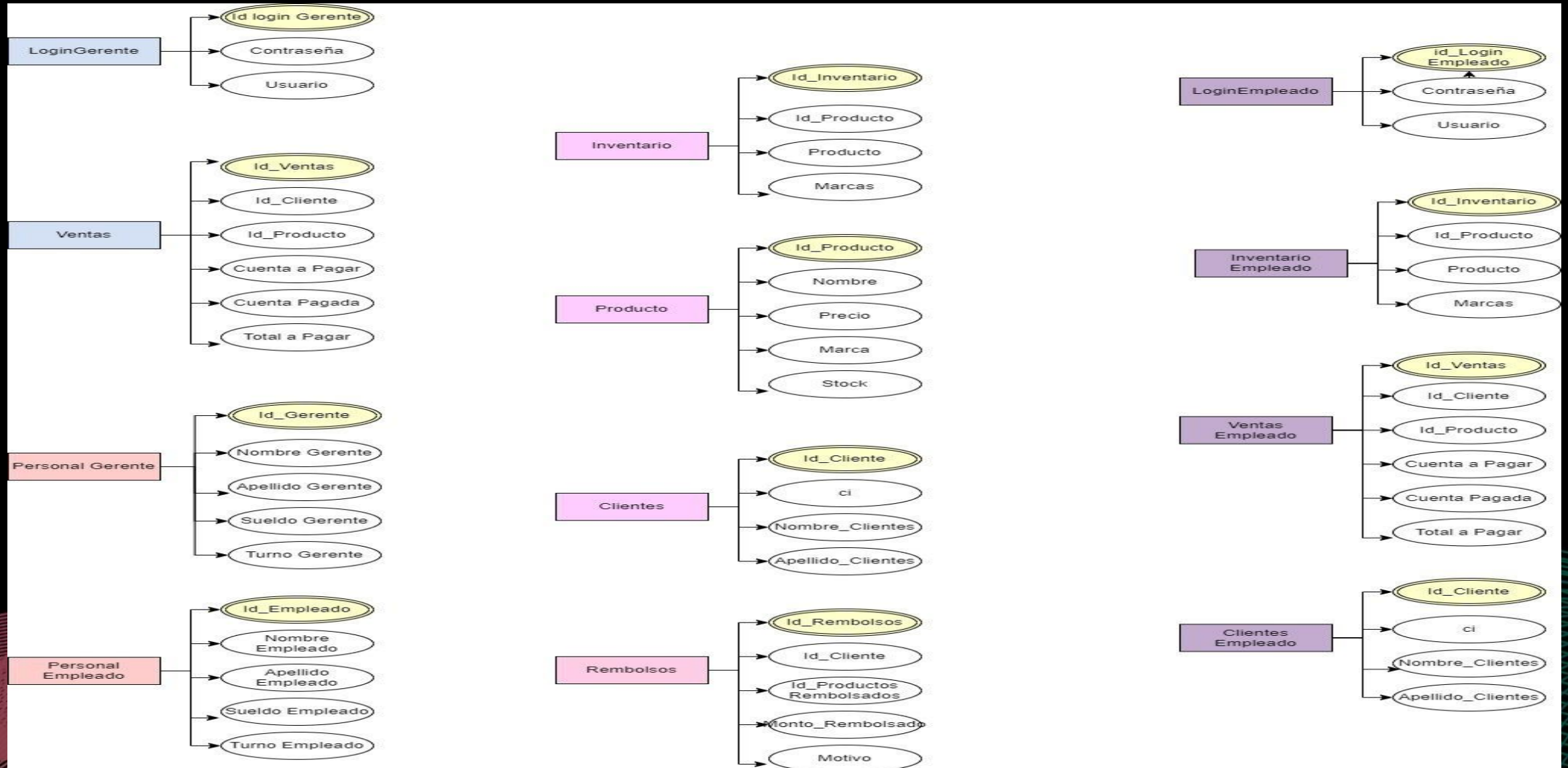
Id_Rembolsos
Id_Cliente
Id_Productos_Rembolsados
Monto_Rembolsado
Motivo



IDENTIFICANDO LAS TABLAS NECESARIAS



DISEÑO DE LA BASE DE DATOS



ARCHIVO SQL

```
/*      TABLAS PARA EL ROL GERENTE      */
-----

CREATE TABLE LoginGerente
(
    Id_Login_Gerente VARCHAR(100) PRIMARY KEY,
    Contraseña VARCHAR(100),
    Usuario VARCHAR(100),
    FOREIGN KEY (Id_Login_Gerente) REFERENCES PersonalGerente(Id_Gerente)
);
```

```
/* TABLAS PARA EL ROL EMPLEADO */
-----

CREATE TABLE LoginEmpleado
(
    id_Login_Empleado VARCHAR(100) PRIMARY KEY,
    Contraseña VARCHAR(100),
    Usuario VARCHAR(100),
    FOREIGN KEY (id_Login_Empleado) REFERENCES PersonalEmpleado(Id_Empleado)
);
```


ARCHIVO SQL

```
-----  
/* TABLAS PARA CLIENTE */  
-----
```

```
CREATE TABLE Clientes
```

```
(
```

```
    Id_Cliente VARCHAR(100) PRIMARY KEY,
```

```
    ci INTEGER,
```

```
    Nombre_Clientes VARCHAR(100),
```

```
    Apellido_Clientes VARCHAR(100)
```

```
);
```

```
-----  
/* TABLAS PARA PRODUCTO */  
-----
```

```
CREATE TABLE Producto
```

```
(
```

```
    Id_Producto VARCHAR(100) PRIMARY KEY,
```

```
    Nombre VARCHAR(100),
```

```
    Precio VARCHAR(100),
```

```
    Marca VARCHAR(100),
```

```
    Stock VARCHAR(100)
```

```
);
```

ARCHIVO SQL

```
-----  
/* TABLAS PARA INVENTARIO */  
-----
```

```
CREATE TABLE Inventario
```

```
(  
    Id_Inventario VARCHAR(100) PRIMARY KEY,  
    Id_Productos VARCHAR(100),  
    Productos VARCHAR(100),  
    Marcas VARCHAR(100),  
);
```

```
-----  
/* TABLAS PARA VENTAS */  
-----
```

```
CREATE TABLE Ventas
```

```
(  
    Id_Ventas VARCHAR(100) PRIMARY KEY,  
    Id_Cliente VARCHAR(100),  
    Id_Productos VARCHAR(100),  
    Cuenta_a_Pagar VARCHAR(100),  
    Cuenta_Pagada VARCHAR(100),  
    Total_a_Pagar VARCHAR(100)  
);
```


ARCHIVO SQL

```
-----  
/* TABLAS PARA REMBOLSO */  
-----
```

```
CREATE TABLE Rembolsos
```

```
(  
  Id_Rembolsos VARCHAR(100) PRIMARY KEY,  
  Id_Cliente VARCHAR (100),  
  Id_Productos_Rembolsados VARCHAR(100),  
  Monto_Rembolsado VARCHAR (100),  
  Motivo VARCHAR (200),  
  FOREIGN KEY (Id_Cliente) REFERENCES Clientes(Id_Cliente),  
  FOREIGN KEY (Id_Productos_Rembolsados) REFERENCES Producto(Id_Producto)  
);
```

```
CREATE TABLE PersonalGerente
```

```
  Id_Gerente VARCHAR(100) PRIMARY KEY,  
  Nombre_Gerente VARCHAR(100),  
  Apellido_Gerente VARCHAR(100),  
  Sueldo_Gerente VARCHAR(100),  
  Turno_Gerente VARCHAR(100),  
);
```

```
CREATE TABLE PersonalEmpleado
```

```
(  
  Id_Empleado VARCHAR(100) PRIMARY KEY,  
  Nombre_Empleado VARCHAR(100),  
  Apellido_Empleado VARCHAR(100),  
  Sueldo_Empleado VARCHAR(100),  
  Turno_Empleado VARCHAR(100)  
);
```

ARCHIVO SQL

```
INSERT INTO Producto(Id_Producto,Nombre,Precio,Marca,Stock)
VALUES ('P-02','Celular Redmi Not 8','1200$','Xiaomi',15),
('P-05','Celular Redmi 8','800$','Xiaomi',20),
('P-06','Pc 780 ','7000$','Dell Computer',2),
('P-07','Pc 3020','9000$','Dell Computer',10),
('P-08','Pc 7010','15000$','Dell Computer',15),
('P-09','Laptop Pavillon 14','13000$','Hp',30),
('P-10','Laptop Spectre 360','10000$','Hp',10)|
```


ARCHIVO SQL

- + System Stored Procedures
- + dbo.PA_AgregarCliente
- + dbo.PA_AgregarInventario
- + dbo.PA_AgregarPersonalEmpleado
- + dbo.PA_AgregarPersonalGerente
- + dbo.PA_AgregarProducto
- + dbo.PA_AgregarRembolso
- + dbo.PA_AgregarVenta
- + dbo.PA_BuscarApellido_Clientes
- + dbo.PA_BuscarApellidoEmpleado
- + dbo.PA_BuscarApellidoGerente
- + dbo.PA_Buscarci
- + dbo.PA_BuscarCuenta_a_Pagar
- + dbo.PA_BuscarCuenta_Pagada
- + dbo.PA_BuscarId_Cliente
- + dbo.PA_BuscarId_Productos
- + dbo.PA_BuscarId_Productos_Rembolsados
- + dbo.PA_BuscarId_Rembolsos
- + dbo.PA_BuscarIdCliente
- + dbo.PA_BuscarIdEmpleado
- + dbo.PA_BuscarIdGerente
- + dbo.PA_BuscarIdInventario
- + dbo.PA_BuscarIdProducto
- + dbo.PA_BuscarIdVentas
- + dbo.PA_BuscarMarcaProducto
- + dbo.PA_BuscarMarcas
- + dbo.PA_BuscarMonto_Rembolsado
- + dbo.PA_BuscarMotivo

- + dbo.PA_BuscarNombre_Clientes
- + dbo.PA_BuscarNombreEmpleado
- + dbo.PA_BuscarNombreGerente
- + dbo.PA_BuscarNombreProducto
- + dbo.PA_BuscarPrecioProducto
- + dbo.PA_BuscarProducto
- + dbo.PA_BuscarRemId_Cliente
- + dbo.PA_BuscarStockProducto
- + dbo.PA_BuscarSueldoEmpleado
- + dbo.PA_BuscarSueldoGerente
- + dbo.PA_BuscarTIdProducto
- + dbo.PA_BuscarTotal_a_Pagar
- + dbo.PA_BuscarTurnoEmpleado
- + dbo.PA_BuscarTurnoGerente
- + dbo.PA_CantidadClientes
- + dbo.PA_CantidadProductos
- + dbo.PA_CantidadProductosInventario
- + dbo.PA_CantidadVentas
- + dbo.PA_Cuenta_a_PagarMenor5000
- + dbo.PA_Cuenta_PagadaMayor1000
- + dbo.Pa_EliminarCliente
- + dbo.PA_EliminarInventario
- + dbo.PA_EliminarPersonalEmpleado
- + dbo.PA_EliminarPersonalGerente
- + dbo.PA_EliminarProducto
- + dbo.Pa_EliminarRembolso

- + dbo.Pa_EliminarVenta
- + dbo.PA_ModificarCliente
- + dbo.PA_ModificarInventario
- + dbo.PA_ModificarPersonalEmpleado
- + dbo.PA_ModificarPersonalGerente
- + dbo.PA_ModificarProducto
- + dbo.PA_ModificarRembolso
- + dbo.PA_ModificarVenta
- + dbo.PA_MostrarCliente
- + dbo.PA_MostrarInventario
- + dbo.PA_MostrarPersonalEmpleado
- + dbo.PA_MostrarPersonalGerente
- + dbo.PA_MostrarProducto
- + dbo.PA_MostrarRembolsos
- + dbo.PA_MostrarVenta
- + dbo.PA_TerminaContraseña12PE
- + dbo.PA_TerminaContraseña12PG

USO INNER JOIN

```
SELECT PG.*  
FROM PersonalGerente AS PG  
    INNER JOIN LoginGerente AS LG ON PG.Id_Gerente = LG.Id_Login_Gerente  
WHERE LG.Contraseña LIKE '12%'
```

```
SELECT PE.*  
FROM PersonalEmpleado AS PE  
    INNER JOIN LoginEmpleado AS LE ON PE.Id_Empleado = LE.id_Login_Empleado  
WHERE LE.Contraseña LIKE '12%'
```

```
SELECT V.*  
FROM Ventas AS V  
    INNER JOIN Clientes AS C ON V.Id_Cliente = C.Id_Cliente  
WHERE CAST(V.Cuenta_a_Pagar AS INTEGER) < 5000
```

```
SELECT V.*  
FROM Ventas AS V  
    INNER JOIN Clientes AS C ON V.Id_Cliente = C.Id_Cliente  
WHERE CAST(V.Cuenta_Pagada AS INTEGER) > 1000
```


USO FUNCIONES

```
CREATE FUNCTION CantidadProductos()  
RETURNS VARCHAR(100) AS  
BEGIN  
    DECLARE @Cantiad VARCHAR(100);  
    SELECT @Cantiad = COUNT(P.Id_Producto)  
    FROM Producto AS P  
    RETURN @Cantiad;  
END;
```

```
CREATE FUNCTION CantidadProductosInventario()  
  
RETURNS VARCHAR(100) AS  
BEGIN  
    DECLARE @Cantiad VARCHAR(100);  
    SELECT @Cantiad = COUNT(CAST(CAST(Precio AS VARCHAR(4)) AS INTEGER))  
    FROM Producto AS P  
    RETURN @Cantiad;  
END;  
  
SELECT dbo.CantidadProductosInventario() 'CANTIDAD DE PRODUCTOS EN EL INVENTARIO'
```

USO FUNCIONES

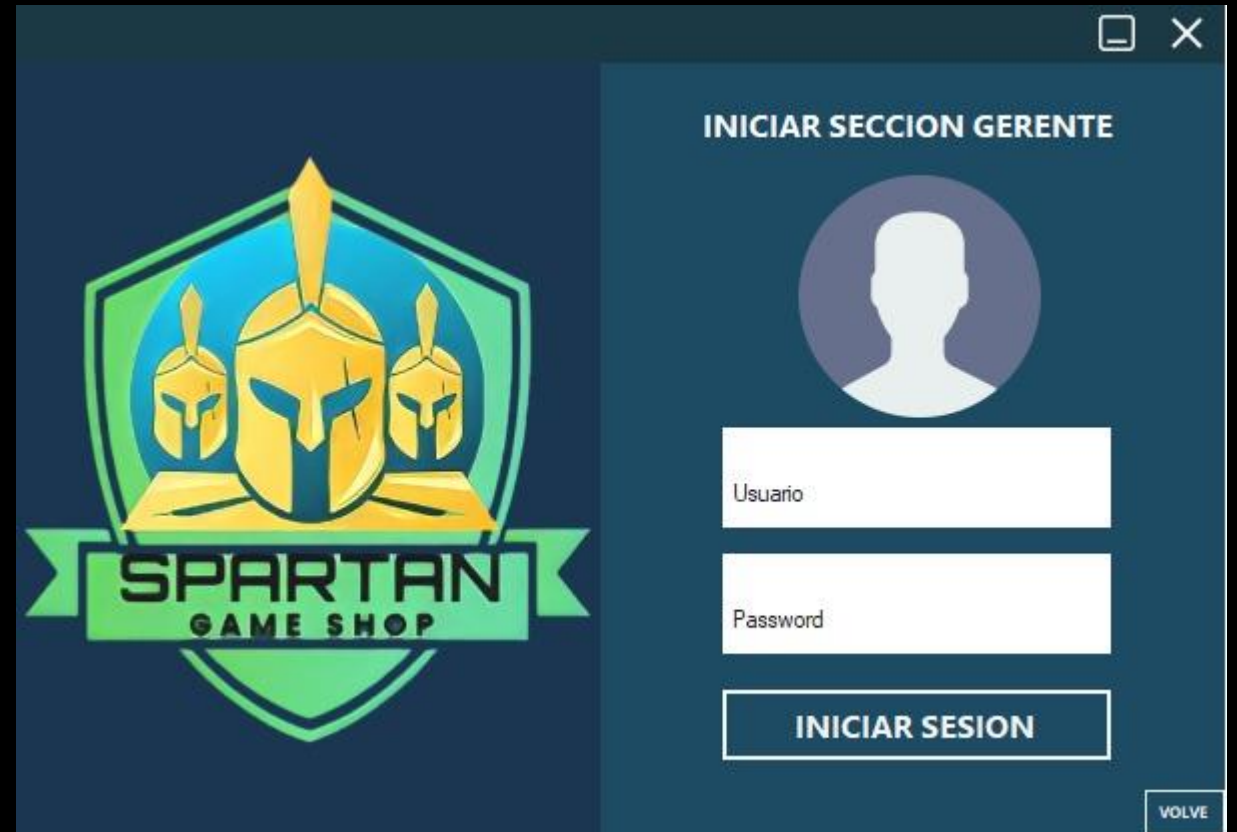
```
CREATE FUNCTION CantidadClientes()  
RETURNS VARCHAR(100) AS  
BEGIN  
    DECLARE @Cantiad VARCHAR(100);  
    SELECT @Cantiad = COUNT(C.Id_Cliente)  
    FROM Clientes AS C  
    RETURN @Cantiad;  
END;  
  
SELECT dbo.CantidadClientes() 'CANTIDAD DE CLIENTES'
```

```
CREATE FUNCTION CantidadVentas()  
RETURNS VARCHAR(100) AS  
BEGIN  
    DECLARE @Cantiad VARCHAR(100);  
    SELECT @Cantiad = COUNT(V.Id_Cliente)  
    FROM Ventas AS V  
    RETURN @Cantiad;  
END;  
  
SELECT dbo.CantidadVentas() AS 'CANTIDAD DE VENTAS'
```

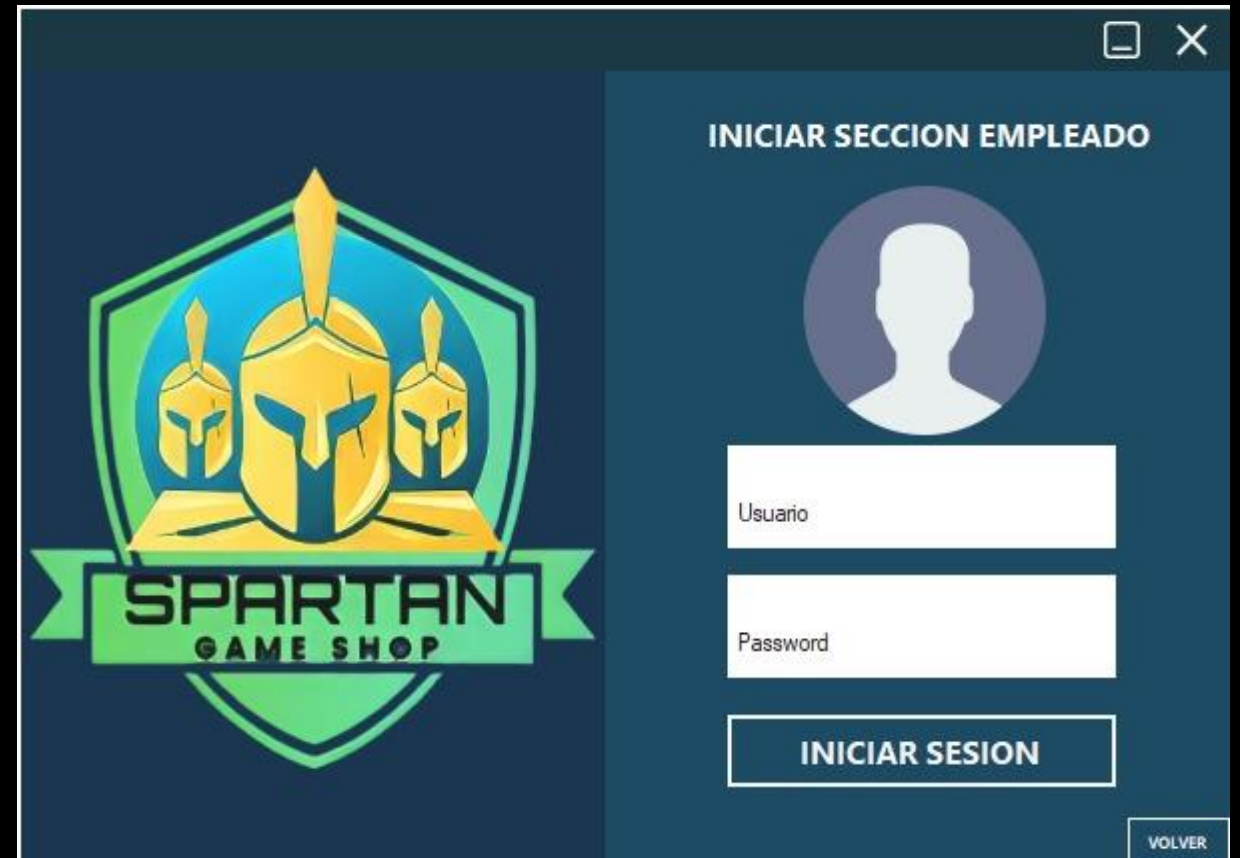

USABILIDAD



USABILIDAD



USABILIDAD



USABILIDAD



USABILIDAD



USABILIDAD

Menu

Inicio

Inventario

Ventas

Clientes

Reembolso

Personal

Cerrar seccion

SPARTAN GAME SHOP

PRODUCTOS

BD PRODUCTO

ELIMINAR

AGREGAR

MODIFICAR

ACTUALIZAR

	Id_Producto	Nombre	Precio	Marca	Stock
▶	P-02	Celular Redmi ...	1200\$	Xiaomi	15
	P-05	Celular Redmi 8	800\$	Xiaomi	20
	P-06	Pc 780	7000\$	Dell Computer	2
	P-07	Pc 3020	9000\$	Dell Computer	10
	P-08	Pc 7010	15000\$	Dell Computer	15
	P-09	Laptop Pavillon...	13000\$	Hp	30
	P-10	Laptop Spectre ...	10000\$	Hp	10
*					

NUMERO DEREGLISTROS 7

VOLVER

USABILIDAD

 Menu

 Inicio

 Inventario

 Ventas

 Clientes

 Reembolso

 Personal

 Cerrar seccion

 SPARTAN GAME SHOP

VENTAS

Lista de Ventas

-- Buscar Por --

AGREGAR

MODIFICAR

ELIMINAR

	Id_Ventas	Id_Cliente	Id_Productos	Cuenta_a_Pagar
	V-1	C-1	P-02	600\$
	V-2	C-2	P-08	3500\$
▶	V-3	C-1	P-05	0\$
*				


<


>


NUMERO DEREGISTROS 3


ACTUALIZAR


USABILIDAD


 Menu


 Inicio


 Inventario


 Ventas



 Clientes

 Reembolso

 Personal


 Cerrar seccion

 SPARTAN GAME SHOP



CLIENTES

LISTA DE CLIENTES

-- Buscar Por --

AGREGAR

MODIFICAR

ELIMINAR

	Id_Cliente	ci	Nombre_Clientes	Apellido_Clientes
	C-1	129585851	Mijail	Choque
	C-2	125874536	Jorge	Manzano
▶	C-3	236666	Luis	Mendoza
•				

NUMERO DEREGISTROS 3

ACTUALIZAR

USABILIDAD

Menu

Inicio

Inventario

Ventas

Clientes

Reembolso

Personal

Cerrar seccion

SPARTAN GAME SHOP

REMBOLSOS

LISTA DE REEMBOLSOS

AGREGAR

MODIFICAR

ELIMINAR

	Id_Reembolsos	Id_Cliente	Id_Productos_Ren	Monto_Rebols	Motivo
▶	Rem-1	C-2	P-06	800\$	Mala Funcionalidad
	Rem-2	C-3	P-06,P-08	21000\$	Mal estado del equipo(falla
•					

NUMERO DEREGISTROS 2

ACTUALIZAR

USABILIDAD

Menu

Inicio

Inventario

Ventas

Clientes

Reembolso

Personal

Cerrar seccion

SPARTAN GAME SHOP

REMBOLSOS

LISTA DE REMBOLSOS

-- Buscar Por --

AGREGAR

MODIFICAR

ELIMINAR


	Id_Reembolsos	Id_Cliente	Id_Productos_Ren	Monto_Rebols	Motivo
▶	Rem-1	C-2	P-06	800\$	Mala Funcionalidad
	Rem-2	C-3	P-06,P-08	21000\$	Mal estado del equipo(falla
•					

NUMERO DEREGISTROS 2

ACTUALIZAR

USABILIDAD


 Menu

 SPARTAN GAME SHOP






 Inicio

 Inventario

 Ventas

 Clientes

 Reembolso

 Personal

 Cerrar sesion

PERSONAL EMPLEADO

BD PESONAL EMPLEADO

ELIMINAR

AGREGAR

MODIFICAR

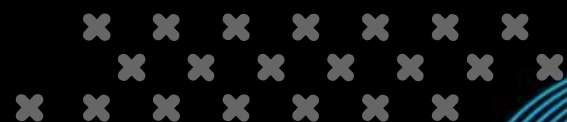
ACTUALIZAR

	Id_Empleado	Nombre_Emplead	Apellido_Empleac	Sueldo_Empleadc	Turno_Empleado
	PEmp-1	Jorge	Perez	1000\$	Mañana
	Pemp-2	Edgar	Mamani Copa	2500\$	Mañana-Tarde
	PEmp-3	Julian	Mendizabal	5000\$	Mañana-Tarde-...
▶▶					

NUMERO DEREGISTROS 3

CONCLUSIONES

- 1. Se logro realizar el sistema de ventas mediante los conocimientos adquiridos en la materia de Base de Datos 1.**
- 2. Como recomendación a futuro se podría añadir mas tablas para el gerente que mejoraría la administración de la tienda.**
- 3. Se recomendaría que a futuro se pueda montar este Sistema de Venta en la Nube y a la par realizar diferentes mejoras.**



*GRACIAS
POR SU
ATENCIÓN*

