

BASE DE DATOS 1

HITO 5



Integrantes:

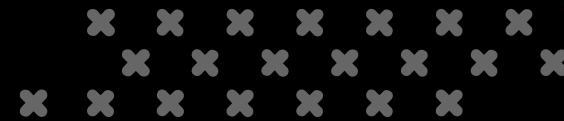
Jhonatan David Alanoca Blanco

Mijail Oliver Choque Amaro

Karen Alisson Chuquimia Mamani

Edson Iver Condori Condori

Jhanpol Machaca Aranibar



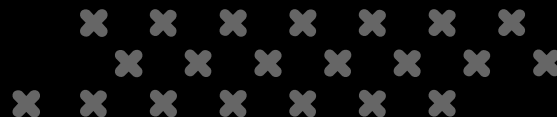
INTRODUCCION

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


LoginPersonal

 Id_Login_Personal
Contraseña
Usuario


Inventario

 Id_Inventario
Id_Productos
Id_Gerente_Del_Inventario
Productos
Marcas

Personal

 Id_Personal
Nombre
Apellido
Roll
Sueldo
Tipo_Moneda
Turno

Productos


 Id_Producto
Nombre
Precio
Tipo_Moenda
Marca
Stock
Ubicacion_del_Producto

IDENTIFICANDO LAS TABLAS NECESARIAS


Cientes

 Id_Cliente
Id_Vendedor
CI
Nombre_Clientes
Apellido_Clientes

Rembolsos

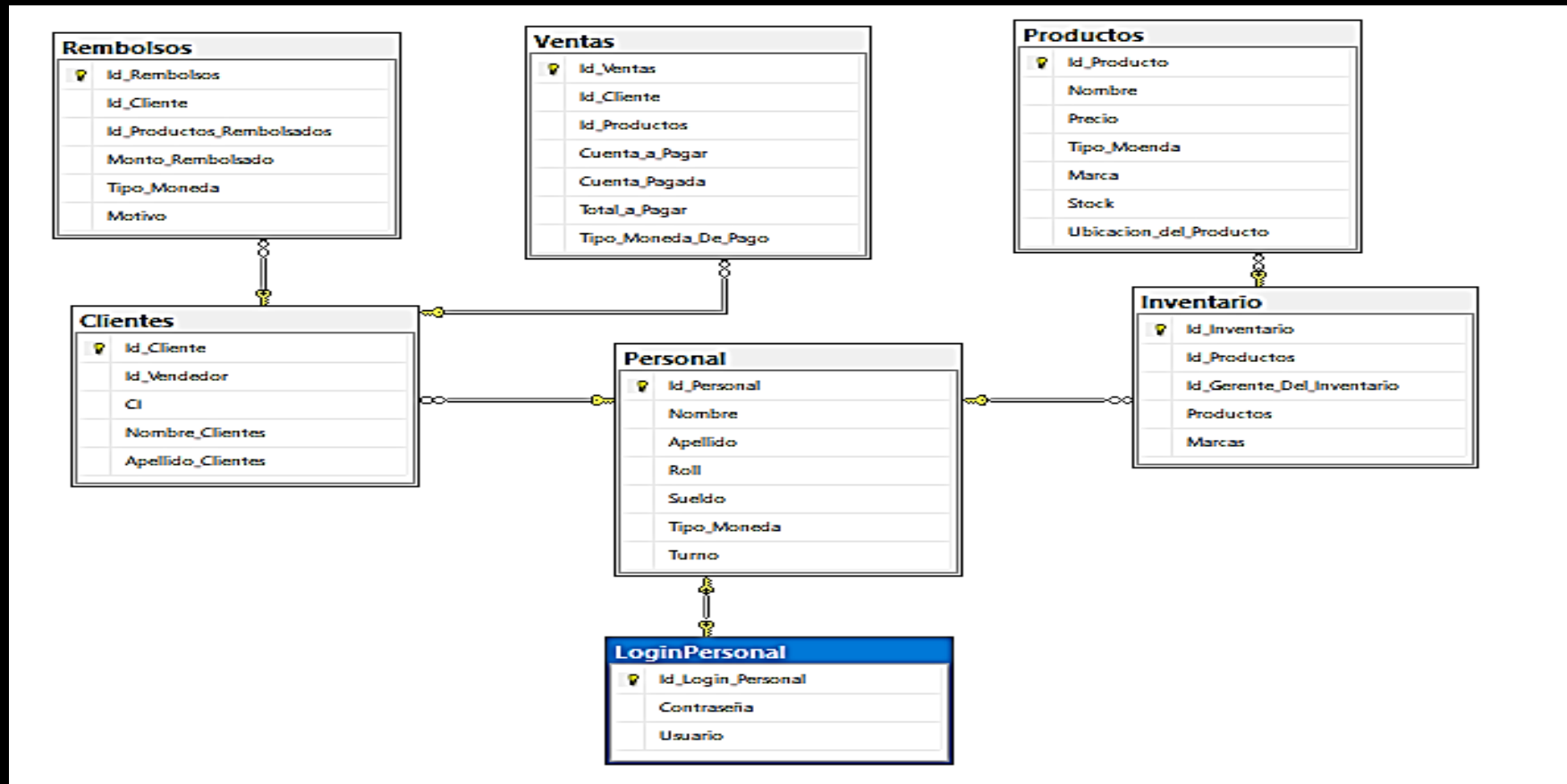
 Id_Rembolsos
Id_Cliente
Id_Productos_Rembolsados
Monto_Rembolsado
Tipo_Moneda
Motivo

Ventas

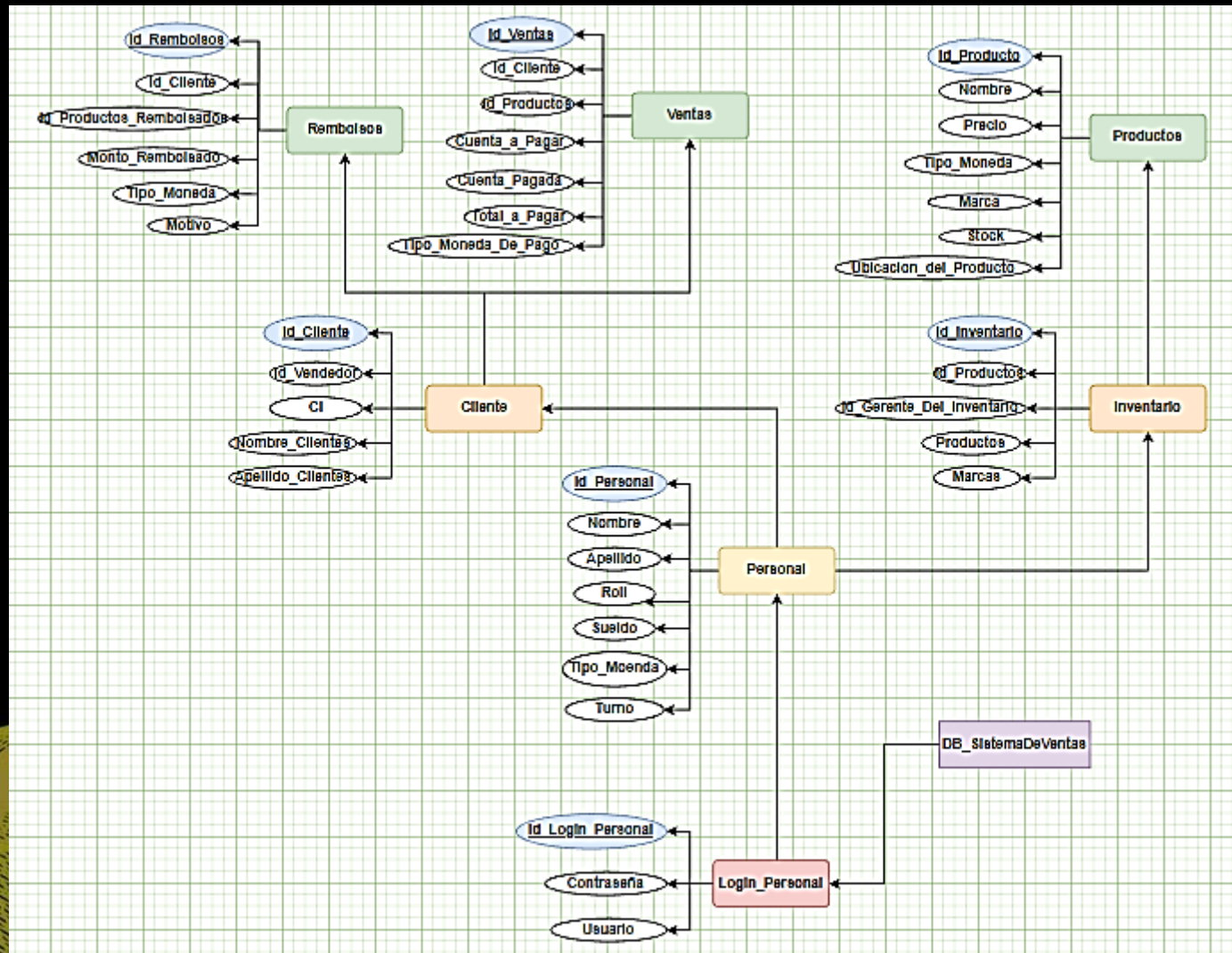
 Id_Ventas
Id_Cliente
Id_Productos
Cuenta_a_Pagar
Cuenta_Pagada
Total_a_Pagar
Tipo_Moneda_De_Pago



IDENTIFICANDO LAS TABLAS NECESARIAS



DISEÑO DE LA BASE DE DATOS



ARCHIVO SQL

```
CREATE TABLE LoginPersonal
(
    Id_Login_Personal VARCHAR(100) PRIMARY KEY,
    Contraseña VARCHAR(100),
    Usuario VARCHAR(100)

    FOREIGN KEY (Id_Login_Personal) REFERENCES Personal(Id_Personal)
);
```


ARCHIVO SQL

```
CREATE TABLE Personal
(
    Id_Personal VARCHAR(100) PRIMARY KEY,
    Nombre VARCHAR(100),
    Apellido VARCHAR(100),
    Roll VARCHAR (100),
    Sueldo FLOAT,
    Tipo_Moenda VARCHAR(100),
    Turno VARCHAR(100)
);
```

ARCHIVO SQL

```
CREATE TABLE Inventario
(
    Id_Inventario VARCHAR(100) PRIMARY KEY,
    Id_Productos VARCHAR(100),
    Id_Gerente_Del_Inventario VARCHAR(100),
    Productos VARCHAR(100),
    Marcas VARCHAR(100),

    FOREIGN KEY (Id_Gerente_Del_Inventario) REFERENCES Personal(Id_Personal)
);
```

ARCHIVO SQL

```
CREATE TABLE Productos
(
    Id_Producto VARCHAR(100) PRIMARY KEY,
    Nombre VARCHAR(100),
    Precio FLOAT,
    Tipo_Moenda VARCHAR(100),
    Marca VARCHAR(100),
    Stock VARCHAR(100),
    Ubicacion_del_Producto VARCHAR(100),

    FOREIGN KEY (Ubicacion_del_Producto) REFERENCES Inventario(Id_Inventario)
);
```


ARCHIVO SQL

```
CREATE TABLE Clientes
(
    Id_Cliente VARCHAR(100) PRIMARY KEY,
    Id_Vendedor VARCHAR(100),
    CI INTEGER,
    Nombre_Clientes VARCHAR(100),
    Apellido_Clientes VARCHAR(100),

    FOREIGN KEY (Id_Vendedor) REFERENCES Personal(Id_Personal)
);
```

ARCHIVO SQL

```
CREATE TABLE Rembolsos
(
    Id_Rembolsos VARCHAR(100) PRIMARY KEY,
    Id_Cliente VARCHAR (100),
    Id_Productos_Rembolsados VARCHAR(100),
    Monto_Rembolsado FLOAT,
    Tipo_Moneda VARCHAR(100),
    Motivo VARCHAR (200),

    FOREIGN KEY (Id_Cliente) REFERENCES Clientes(Id_Cliente)
);
```

ARCHIVO SQL

```
CREATE TABLE Ventas
(
    Id_Ventas VARCHAR(100) PRIMARY KEY,
    Id_Cliente VARCHAR(100),
    Id_Productos VARCHAR(100),
    Cuenta_a_Pagar FLOAT,
    Cuenta_Pagada FLOAT,
    Total_a_Pagar FLOAT,
    Tipo_Moneda_De_Pago VARCHAR(100),

    FOREIGN KEY (Id_Cliente) REFERENCES Clientes(Id_Cliente)
);
```


PROCEDIMIENTOS ALMACENADOS

+	dbo.PA_AdicionarCliente
+	dbo.PA_AdicionarInventario
+	dbo.PA_AdicionarLoginPersonal
+	dbo.PA_AdicionarPersonal
+	dbo.PA_AdicionarProductos
+	dbo.PA_AdicionarRembolso
+	dbo.PA_AdicionarVenta
+	dbo.PA_EliminarCliente
+	dbo.PA_EliminarInventario
+	dbo.PA_EliminarLoginPersonal
+	dbo.PA_EliminarPersonal
+	dbo.PA_EliminarProductos
+	dbo.PA_EliminarRembolso
+	dbo.PA_EliminarVenta
+	dbo.PA_ModificarCliente
+	dbo.PA_ModificarInventario
+	dbo.PA_ModificarLoginPersonal
+	dbo.PA_ModificarPersonal
+	dbo.PA_ModificarProductos
+	dbo.PA_ModificarRembolso
+	dbo.PA_ModificarVenta
+	dbo.PA_MostrarCliente
+	dbo.PA_MostrarInventario
+	dbo.PA_MostrarLoginPersonal
+	dbo.PA_MostrarPersonal
+	dbo.PA_MostrarProductos
+	dbo.PA_MostrarRembolsos
+	dbo.PA_MostrarVenta

USO INNER JOIN

```
SELECT LP.*  
FROM Personal AS P  
INNER JOIN LoginPersonal AS LP ON P.Id_Personal = LP.Id_Login_Personal  
WHERE P.Roll LIKE '%Gerente%'
```

```
SELECT R.*  
FROM Clientes AS C  
INNER JOIN Rembolso AS R ON C.Id_Cliente = R.Id_Cliente  
WHERE C.Id_Vendedor BETWEEN 'P-4' AND 'P-10'
```

USO INNER JOIN

```
SELECT COUNT(C.Id_Vendedor)
FROM Clientes AS C
INNER JOIN Rembolso AS R ON C.Id_Cliente = R.Id_Cliente
INNER JOIN Personal AS P ON P.Id_Personal = C.Id_Vendedor
WHERE P.Roll = '%Gerente%'
```

```
SELECT PR.*
FROM Inventario AS I
INNER JOIN Productos AS PR ON I.Id_Inventario = PR.Ubicacion_del_Producto
INNER JOIN Personal AS P ON I.Id_Gerente_Del_Inventario = P.Id_Personal
WHERE P.Roll = 'Gerente' AND I.Id_Inventario = 'Inv-5' AND P.Id_Personal = 'P-1'
```


USO FUNCIONES

```
CREATE FUNCTION Sueldo_maximo(@roll VARCHAR(100))  
RETURNS FLOAT AS  
BEGIN  
    DECLARE @R FLOAT  
  
    SELECT @R = MAX(P.Sueldo)  
    FROM Personal AS P  
    WHERE P.Roll LIKE '%' + @roll + '%'  
  
    RETURN @R  
END
```

```
CREATE FUNCTION Cantidad_Personal(@roll VARCHAR(100))  
RETURNS FLOAT AS  
BEGIN  
    DECLARE @R FLOAT  
  
    SELECT @R = COUNT(P.Id_Personal)  
    FROM Personal AS P  
    WHERE P.Roll LIKE '%' + @roll + '%'  
  
    RETURN @R  
END
```

USO FUNCIONES

```
CREATE FUNCTION Cantidad_Productos_Precio_May(@precio FLOAT)
RETURNS FLOAT AS
BEGIN
    DECLARE @R FLOAT

    SELECT @R = COUNT(PR.Id_Producto)
    FROM Productos AS PR
    WHERE PR.Precio > @precio

    RETURN @R
END
```

```
CREATE FUNCTION Cantidad_Productos_Precio_Men(@precio FLOAT)
RETURNS FLOAT AS
BEGIN
    DECLARE @R FLOAT

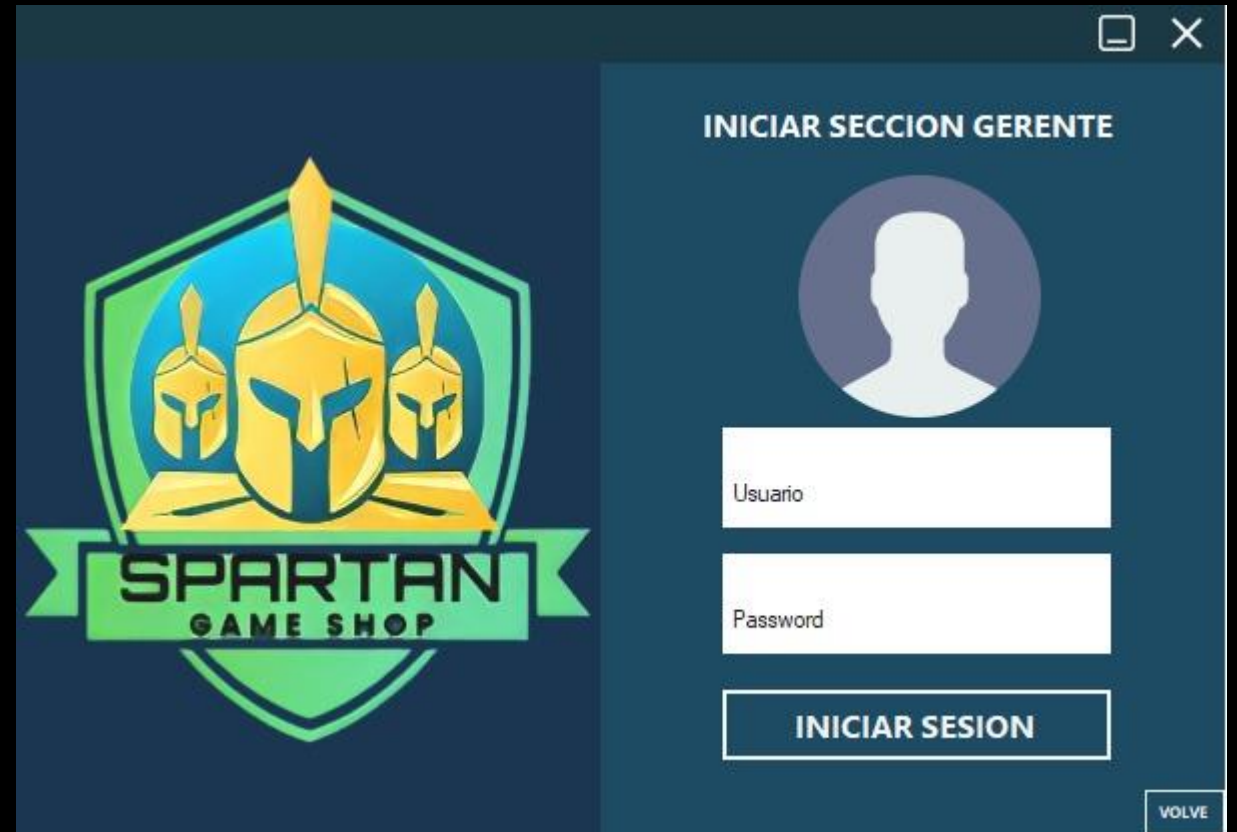
    SELECT @R = COUNT(PR.Id_Producto)
    FROM Productos AS PR
    WHERE PR.Precio < @precio

    RETURN @R
END
```

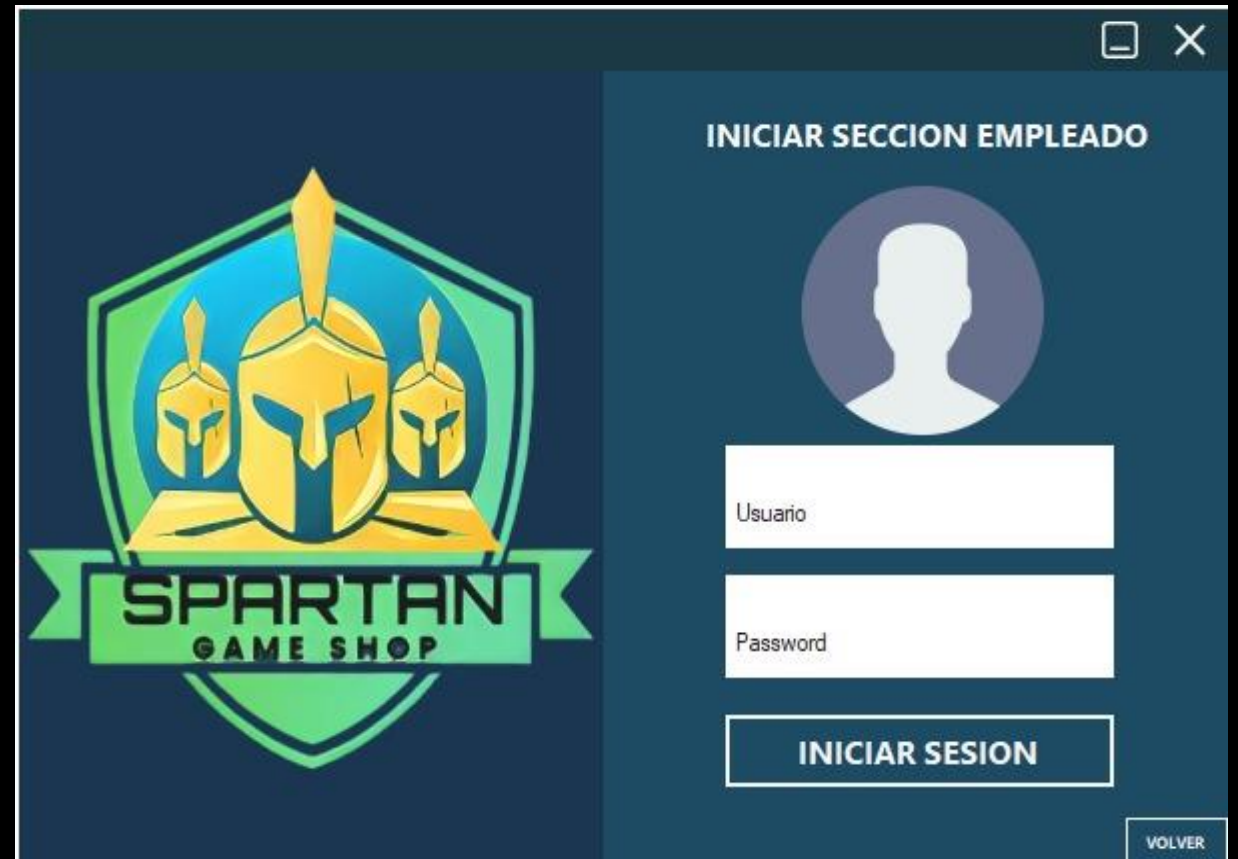

USABILIDAD



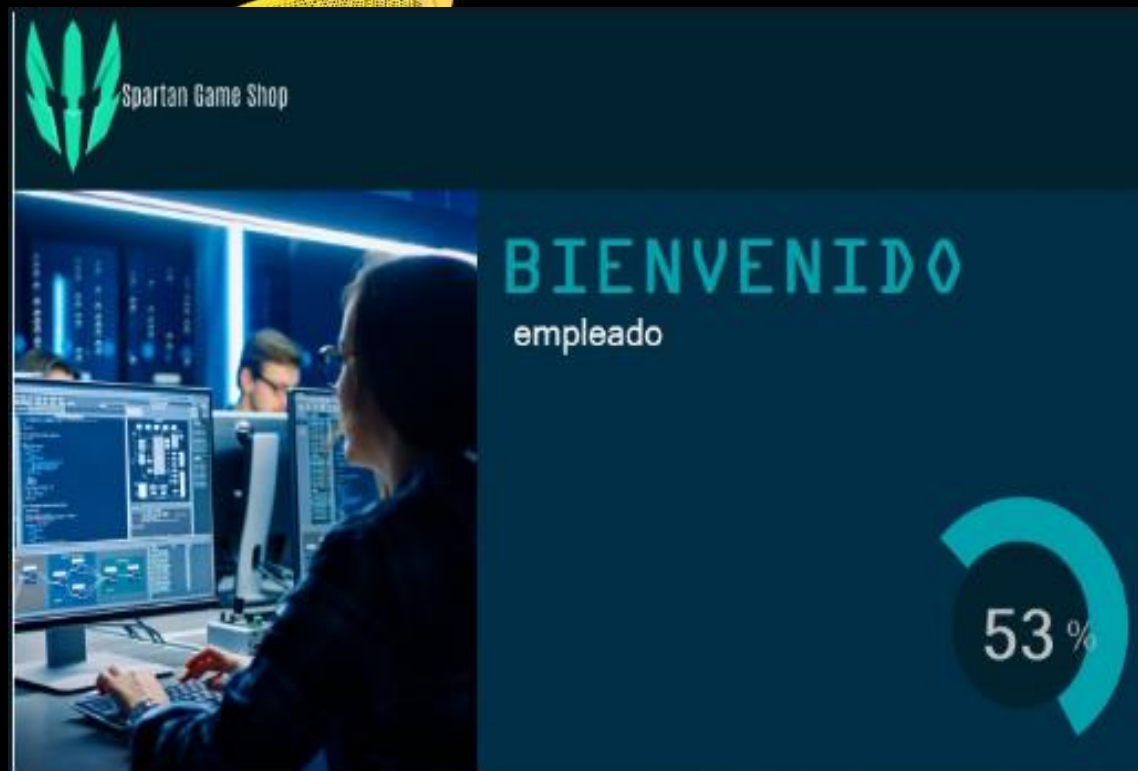
USABILIDAD



USABILIDAD

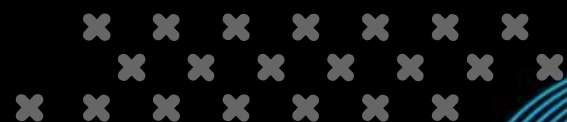


USABILIDAD



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*

