

La aplicación tendrá una base de datos con dos tablas. Una para registrar el empleado y otra para registrar los pagos y descuentos que el empleado reciba. En la tabla de empleados, tendremos un campo para registrar y acumular los pagos que reciba cada empleado. Deben hacer upsert en la tabla de empleados y un insert en la tabla de nómina. En el upsert se incrementará en pagoacumulado que el empleado reciba. Crear stored Procedures. A continuación, las tablas a definir en la base de datos:

tblClientes

- id
- tipodocumento
- documento
- nombres
- apellidos
- fechanacimiento
- sexo
- fechaingreso
- sueldo
- pagosacumulados

Nota: Usar Stored Procedure.

tblnomina

- id
- tipodocumento
- documento
- tipoPago
- monto
- fechapago
- mes
- ano

Desarrollo:

Program.cs

```
using System;
using System.Data.SqlClient;

class Program
{
    static void Main()
    {
        string connectionString = "Server=localhost; Database=BDProject;
User Id=sa; Password=20186947Ismael";
```

```
using (SqlConnection connection = new
SqlConnection(connectionString))
        {
            connection.Open();
            Console.WriteLine("Conexión exitosa");
            string salir = "N", tipoDocumento, documento, nombres,
apellidos, fechaNacimiento, sexo, fechaIngreso;
            decimal sueldo:
            while (salir == "N")
                Console.Write("Tipo de Documento: ");
                tipoDocumento = Console.ReadLine();
                Console.Write("Documento: "):
                documento = Console.ReadLine();
                Console.Write("Nombres: ");
                nombres = Console.ReadLine();
                Console.Write("Apellidos: ");
                apellidos = Console.ReadLine();
                Console.Write("Fecha de Nacimiento (YYYY-MM-DD): ");
                fechaNacimiento = Console.ReadLine();
                Console.Write("Sexo: "):
                sexo = Console.ReadLine();
                Console.Write("Fecha de Ingreso (YYYY-MM-DD): ");
                fechaIngreso = Console.ReadLine();
                Console.Write("Sueldo: ");
                sueldo = Convert.ToDecimal(Console.ReadLine());
                // Usar el procedimiento almacenado para realizar upsert
en la tabla de clientes
                using (SqlCommand upsertClienteCommand = new
SqlCommand("UpsertCliente", connection))
                {
                    upsertClienteCommand.CommandType =
System.Data.CommandType.StoredProcedure;
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@TipoDocumento", tipoDocumento));
```

```
upsertClienteCommand.Parameters.Add(new
SqlParameter("@Documento", documento));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@Nombres", nombres));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@Apellidos", apellidos));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@FechaNacimiento", fechaNacimiento));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@Sexo", sexo));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@FechaIngreso", fechaIngreso));
                    upsertClienteCommand.Parameters.Add(new
SqlParameter("@Sueldo", sueldo));
                    upsertClienteCommand.ExecuteNonQuery();
                    Console.WriteLine("Datos de cliente
insertados/actualizados exitosamente");
                    // Ahora, insertar en la tabla de nómina
                    Console.Write("Tipo de Pago: ");
                    string tipoPago = Console.ReadLine();
                    Console.Write("Monto: "):
                    decimal monto = Convert.ToDecimal(Console.ReadLine());
                    Console.Write("Fecha de Pago (YYYY-MM-DD): ");
                    string fechaPago = Console.ReadLine();
                    Console.Write("Mes: ");
                    int mes = Convert.ToInt32(Console.ReadLine());
                    Console.Write("Año: ");
                    int ano = Convert.ToInt32(Console.ReadLine());
                    // Usar el procedimiento almacenado para insertar
datos en la tabla de nómina
                    using (SqlCommand insertNominaCommand = new
SqlCommand("InsertNomina", connection))
                        insertNominaCommand.CommandType =
System.Data.CommandType.StoredProcedure;
```

```
insertNominaCommand.Parameters.Add(new
SqlParameter("@TipoDocumento", tipoDocumento));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@Documento", documento));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@TipoPago", tipoPago));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@Monto", monto));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@FechaPago", fechaPago));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@Mes", mes));
                        insertNominaCommand.Parameters.Add(new
SqlParameter("@Año", ano));
                        insertNominaCommand.ExecuteNonQuery();
                        Console.WriteLine("Datos de nómina insertados
exitosamente");
                    }
                    Console.WriteLine("¿Desea salir? (Y/N)");
                    salir = Console.ReadLine().ToUpper();
            }
        }
```

appsettings.json

```
{
   "ConnectionStrings": {
      "Default": "Server=172.17.0.2,1401; Database=BDProject; User Id=sa;
Password=20186947Ismael"
   }
}
```

Create Table

```
-- Crear tabla de clientes
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME =
'tblClientes')
```

```
BEGIN

CREATE TABLE tblClientes (

Id INT PRIMARY KEY IDENTITY(1,1),

TipoDocumento VARCHAR(20),

Documento VARCHAR(20),

Nombres VARCHAR(100),

Apellidos VARCHAR(100),

FechaNacimiento DATE,

Sexo VARCHAR(10),

FechaIngreso DATE,

Sueldo DECIMAL(10, 2),

PagosAcumulados DECIMAL(10, 2) DEFAULT 0
)

END
```

```
-- Crear tabla de nómina

IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'tblNomina')

BEGIN

CREATE TABLE tblNomina (
    Id INT PRIMARY KEY IDENTITY(1,1),
    TipoDocumento VARCHAR(20),
    Documento VARCHAR(20),
    TipoPago VARCHAR(50),
    Monto DECIMAL(10, 2),
    FechaPago DATE,
    Mes INT,
    Ano INT
   )

END
```

Stored Procedured

```
--- Procedimiento almacenado UpsertCliente

IF NOT EXISTS (SELECT * FROM sys.procedures WHERE name = 'UpsertCliente')

BEGIN

EXEC('

CREATE PROCEDURE UpsertCliente

@TipoDocumento VARCHAR(20),

@Documento VARCHAR(20),

@Nombres VARCHAR(100),

@Apellidos VARCHAR(100),

@FechaNacimiento DATE,

@Sexo VARCHAR(10),

@FechaIngreso DATE,

@Sueldo DECIMAL(10, 2)

AS
```

```
BEGIN

UPDATE tblClientes

SET PagosAcumulados = PagosAcumulados + @Sueldo

WHERE TipoDocumento = @TipoDocumento AND Documento = @Documento;

IF @@ROWCOUNT = 0

BEGIN

INSERT INTO tblClientes (TipoDocumento, Documento, Nombres, Apellidos,
FechaNacimiento, Sexo, FechaIngreso, Sueldo)

VALUES (@TipoDocumento, @Documento, @Nombres, @Apellidos,
@FechaNacimiento, @Sexo, @FechaIngreso, @Sueldo);

END;

END

')

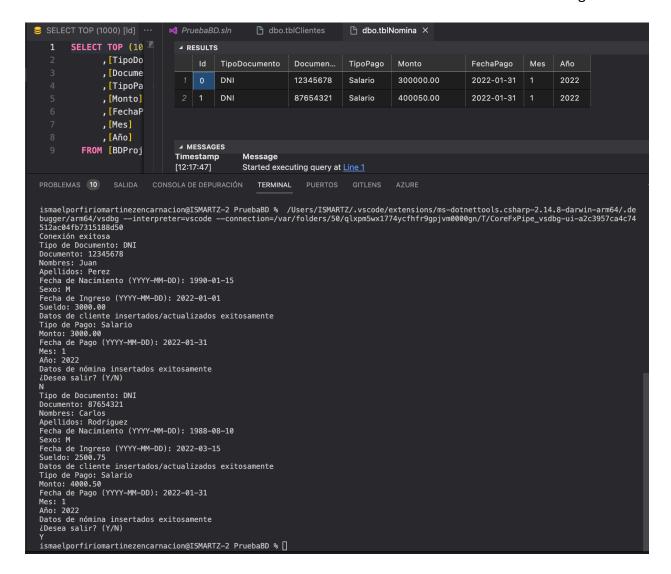
END

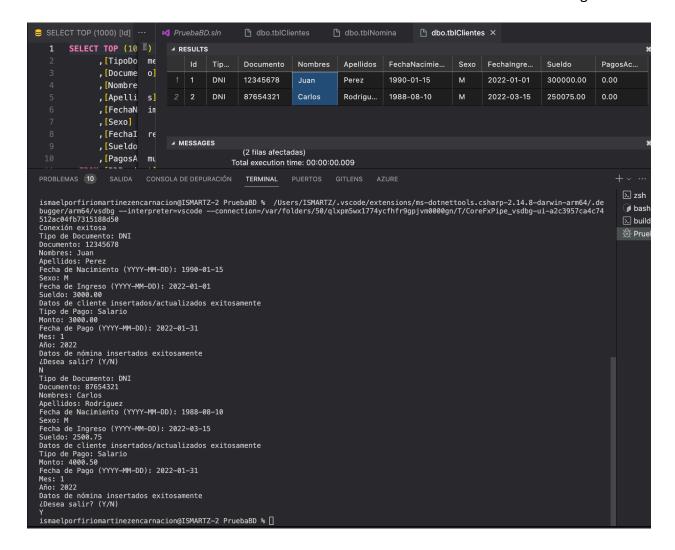
END
```

```
    Procedimiento almacenado InsertNomina

IF NOT EXISTS (SELECT * FROM sys.procedures WHERE name = 'InsertNomina')
BEGIN
    EXEC('
        CREATE PROCEDURE InsertNomina
            @TipoDocumento VARCHAR(20),
            @Documento VARCHAR(20),
            @TipoPago VARCHAR(50),
            @Monto DECIMAL(10, 2),
            @FechaPago DATE,
            @Mes INT,
            @Año INT
        BEGIN
            INSERT INTO tblNomina (TipoDocumento, Documento, TipoPago, Monto,
FechaPago, Mes, Año)
            VALUES (@TipoDocumento, @Documento, @TipoPago, @Monto, @FechaPago, @Mes,
@Año);
        END
END
```

Evidencias





Archivos Zip y Documentos

https://github.com/ISMARZDEV/Desarrollo-de-Software-3---Semana-2---Tarea-3.git