

Ejercicio 1 - Desarrollar una aplicación para registrar los pasajeros del arca de NOE. A continuación los campos:

```
ID
Tipo
Nombres
FechaNacimiento
Sexo
Estado
Nota
```

Desarrollo:

Program.cs

```
using System;
using System.Data;
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";
            while (salir.ToUpper() == "N")
            {
                // Insertar un pasajero
                InsertarPasajero(connection);
                Console.Write("¿Desea salir? (Y/N): ");
```

```
salir = Console.ReadLine().ToUpper();
            }
       }
    static void InsertarPasajero(SqlConnection connection)
        using (SqlCommand insertCommand = new
SqlCommand("InsertarPasajeroArca", connection))
        {
            insertCommand.CommandType = CommandType.StoredProcedure;
            // Parámetros del pasajero
            Console.Write("Tipo: ");
            string tipo = Console.ReadLine();
            Console.Write("Nombres: ");
            string nombres = Console.ReadLine();
            Console.Write("Fecha de Nacimiento (YYYY-MM-DD): ");
            DateTime fechaNacimiento =
Convert.ToDateTime(Console.ReadLine()):
            Console.Write("Sexo: "):
            string sexo = Console.ReadLine();
            Console.Write("Estado: ");
            string estado = Console.ReadLine();
            Console.Write("Nota: ");
            string nota = Console.ReadLine();
            insertCommand.Parameters.AddWithValue("@Tipo", tipo);
            insertCommand.Parameters.AddWithValue("@Nombres", nombres);
            insertCommand.Parameters.AddWithValue("@FechaNacimiento",
fechaNacimiento);
            insertCommand.Parameters.AddWithValue("@Sexo", sexo);
            insertCommand.Parameters.AddWithValue("@Estado", estado);
            insertCommand.Parameters.AddWithValue("@Nota", nota);
            insertCommand.ExecuteNonQuery();
            Console.WriteLine("Pasajero registrado exitosamente");
        }
```

appsettings.json

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

Create Table

```
CREATE TABLE PasajerosArca (
    ID INT PRIMARY KEY IDENTITY(1,1),
    Tipo NVARCHAR(50),
    Nombres NVARCHAR(100),
    FechaNacimiento DATE,
    Sexo NVARCHAR(10),
    Estado NVARCHAR(50),
    Nota NVARCHAR(500)
);
```

Stored Procedured

```
CREATE PROCEDURE InsertarPasajeroArca

@Tipo NVARCHAR(50),

@Nombres NVARCHAR(100),

@FechaNacimiento DATE,

@Sexo NVARCHAR(10),

@Estado NVARCHAR(50),

@Nota NVARCHAR(500)

AS

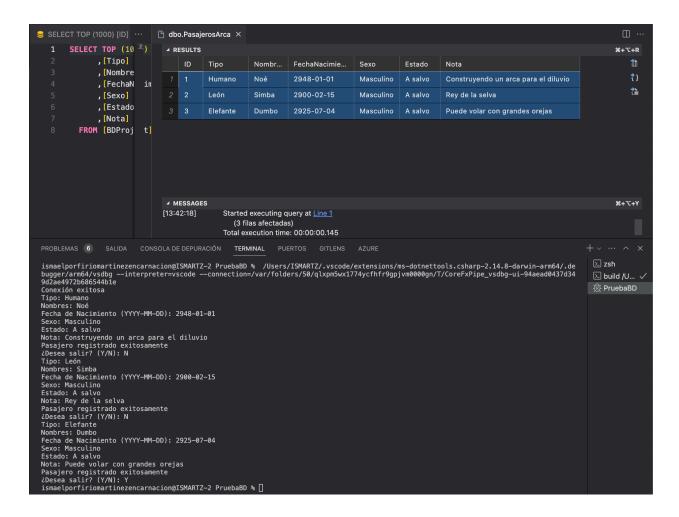
BEGIN

INSERT INTO PasajerosArca (Tipo, Nombres, FechaNacimiento, Sexo, Estado, Nota)

VALUES (@Tipo, @Nombres, @FechaNacimiento, @Sexo, @Estado, @Nota);

END;
```

Evidencias



Ejercicio 2 - Desarrollar una aplicación para registrar los consumos de cada tripulante

- idConsumo
- Id
- Comentario
- Fechalngreso
- Estado

Ejercicio 3 - Listar lo que ha consumido cada tripulante

Nombres--- Comentario --- Fecha

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");
            // Registrar un nuevo tripulante (utilizando stored procedure)
            Console.Write("Nombre: ");
            int idTripulante = InsertarNuevoTripulante(connection,
Console.ReadLine())
            // Insertar consumo de manera interactiva
            InsertarConsumoInteractivo(connection, idTripulante);
            // Listar los consumos del tripulante (utilizando stored
procedure)
            ListarTodosConsumos(connection);
        }
    static void InsertarConsumoInteractivo(SqlConnection connection, int
idTripulante)
        Console.Write("Comentario de Consumo: ");
```

```
string comentario = Console.ReadLine();
        Console.Write("Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): ");
        DateTime fechaIngreso = Convert.ToDateTime(Console.ReadLine());
        Console.Write("Estado: ");
        string estado = Console.ReadLine();
        // Utilizar stored procedure para insertar el consumo
        InsertarConsumo(connection, idTripulante, comentario,
fechaIngreso, estado);
        Console.WriteLine("Consumo registrado exitosamente");
    static void InsertarConsumo(SqlConnection connection, int
idTripulante, string comentario, DateTime fechaIngreso, string estado)
        using (SqlCommand command = new SqlCommand("InsertarConsumo",
connection))
        {
            command.CommandType = System.Data.CommandType.StoredProcedure;
            command.Parameters.Add(new SqlParameter("@IdTripulante",
idTripulante));
            command.Parameters.Add(new SqlParameter("@Comentario",
comentario)):
            command.Parameters.Add(new SqlParameter("@FechaIngreso",
fechaIngreso));
            command.Parameters.Add(new SqlParameter("@Estado", estado));
            command.ExecuteNonQuery();
        }
    }
    static int InsertarNuevoTripulante(SqlConnection connection, string
nombres)
    {
        using (SqlCommand command = new SqlCommand("InsertarTripulante",
connection))
```

```
command.CommandType = System.Data.CommandType.StoredProcedure;
            command.Parameters.Add(new SqlParameter("@Nombres", nombres));
            return Convert.ToInt32(command.ExecuteScalar());
        }
    static void ListarTodosConsumos(SqlConnection connection)
        using (SqlCommand command = new SqlCommand("ListarTodosConsumos",
connection))
        {
            command.CommandType = System.Data.CommandType.StoredProcedure
            using (SqlDataReader reader = command.ExecuteReader())
            {
                Console.WriteLine("Nombres\t\tComentario\t\tFecha");
                while (reader.Read())
                    string nombres = reader["Nombres"].ToString();
                    string comentario = reader["Comentario"].ToString();
                    DateTime fechaIngreso =
Convert.ToDateTime(reader["FechaIngreso"]);
Console.WriteLine($"{nombres}\t\t{comentario}\t\t{fechaIngreso}");
            }
        }
```

appsettings.json

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

Create Table

```
-- Tabla para los tripulantes
CREATE TABLE Tripulantes (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Nombres VARCHAR(100) NOT NULL
);
```

```
-- Tabla para los consumos
CREATE TABLE Consumos (
    Id INT PRIMARY KEY IDENTITY(1,1),
    IdTripulante INT FOREIGN KEY REFERENCES Tripulantes(Id),
    Comentario VARCHAR(255),
    FechaIngreso DATETIME,
    Estado VARCHAR(50)
);
```

Stored Procedured

```
-- Stored procedure para registrar un consumo

CREATE PROCEDURE InsertarConsumo

@IdTripulante INT,

@Comentario VARCHAR(255),

@FechaIngreso DATETIME,

@Estado VARCHAR(50)

AS

BEGIN

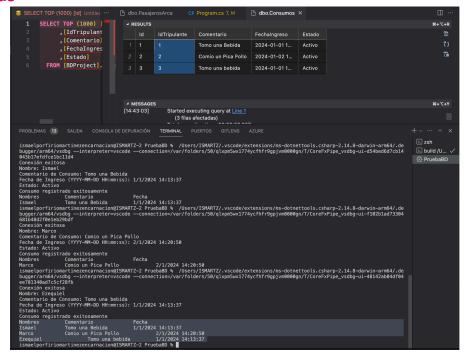
INSERT INTO Consumos (IdTripulante, Comentario, FechaIngreso, Estado)

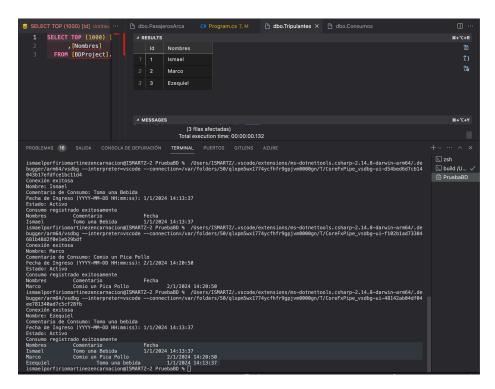
VALUES (@IdTripulante, @Comentario, @FechaIngreso, @Estado);

END;
```

```
-- Stored procedure para listar consumos de tripulantes
CREATE PROCEDURE ListarTodosConsumos
AS
BEGIN
    SELECT Nombres, Comentario, FechaIngreso
    FROM Tripulantes
    INNER JOIN Consumos ON Tripulantes.Id = Consumos.IdTripulante;
END;
```

Evidencias





Archivos Zip y Documentos

https://github.com/ISMARZDEV/IDS-3---Semana-4---Tarea-6.git