



ASIGNATURA

IDS345L-1 LAB. DESARROLLO DE SOFTWARE III

DOCENTE

CARLOS RAFAEL MÉNDEZ COMAS

carlos.mendez@intec.edu.do

SECCIÓN 01 - MARTES

ALUMNO: Ismael Porfirio Martínez Encarnación **ID:** 1077546

CARRERA

INGENIERIA DE SOFTWARE

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

The screenshot shows a SQL Server Enterprise Manager window with a query executed in the 'dbo.PasajerosArca' database. The query is a SELECT TOP (10) statement. The results are displayed in a table with 8 columns: ID, Tipo, Nombre, FechaNacimiento, Sexo, Estado, and Nota. The results show three rows of data for different types of passengers: Humano, León, and Elefante.

ID	Tipo	Nombre	FechaNacimiento	Sexo	Estado	Nota
1	Humano	Noé	2948-01-01	Masculino	A salvo	Construyendo un arca para el diluvio
2	León	Simba	2900-02-15	Masculino	A salvo	Rey de la selva
3	Elefante	Dumbo	2925-07-04	Masculino	A salvo	Puede volar con grandes orejas

The messages pane shows the execution details: Started executing query at Line 1, (3 filas afectadas), Total execution time: 00:00:00.145.

The terminal pane shows the output of the application, including the connection status, the data retrieved for each passenger, and the prompt to register the passenger.

```
ismaelporfiomartinezencarnacion@ISMARTZ-2 PruebaBD % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.debugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qxpm5wx1774ycfhr9gpjvm0000gn/T/CoreFxPipe_vsdg-u1-94aead0437d34
9d2ae4972b686544bie
Conexión exitosa
Tipo: Humano
Nombres: Noé
Fecha de Nacimiento (YYYY-MM-DD): 2948-01-01
Sexo: Masculino
Estado: A salvo
Nota: Construyendo un arca para el diluvio
Pasajero registrado exitosamente
¿Desea salir? (Y/N): N
Tipo: León
Nombres: Simba
Fecha de Nacimiento (YYYY-MM-DD): 2900-02-15
Sexo: Masculino
Estado: A salvo
Nota: Rey de la selva
Pasajero registrado exitosamente
¿Desea salir? (Y/N): N
Tipo: Elefante
Nombres: Dumbo
Fecha de Nacimiento (YYYY-MM-DD): 2925-07-04
Sexo: Masculino
Estado: A salvo
Nota: Puede volar con grandes orejas
Pasajero registrado exitosamente
¿Desea salir? (Y/N): Y
ismaelporfiomartinezencarnacion@ISMARTZ-2 PruebaBD %
```

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

- idConsumo
- Id
- Comentario
- FechaIngreso
- 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

SELECT TOP (1000) [IdTripulante], [Comentario], [FechaIngreso], [Estado] FROM [BDProject].

Id	IdTripulante	Comentario	FechaIngreso	Estado
1	1	Tomo una Bebida	2024-01-01 1...	Activo
2	2	Comio un Pica Pollo	2024-01-02 1...	Activo
3	3	Tomo una bebida	2024-01-01 1...	Activo

Started executing query at Line 1 (3 filas afectadas)

israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-d54bed6d7cb14
043b17efdfc1b11d4
Conexión exitosa
Nombre: Israel
Comentario de Consumo: Tomo una Bebida
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 1/1/2024 14:13:37
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Israel Tomo una Bebida 1/1/2024 14:13:37
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-f182blad73384
081d48d2f0e1eb29bdf
Conexión exitosa
Nombre: Marco
Comentario de Consumo: Comio un Pica Pollo
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 2/1/2024 14:20:50
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Marco Comio un Pica Pollo 2/1/2024 14:20:50
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-48142ab04df04
ee78134bad7c5cf28fb
Conexión exitosa
Nombre: Ezequiel
Comentario de Consumo: Tomo una bebida
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 1/1/2024 14:13:37
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Israel Tomo una Bebida 1/1/2024 14:13:37
Marco Comio un Pica Pollo 2/1/2024 14:20:50
Ezequiel Tomo una bebida 1/1/2024 14:13:37
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 %

SELECT TOP (1000) [Nombres] FROM [BDProject].

Id	Nombres
1	Israel
2	Marco
3	Ezequiel

(3 filas afectadas)
Total execution time: 00:00:00.132

israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-d54bed6d7cb14
043b17efdfc1b11d4
Conexión exitosa
Nombre: Israel
Comentario de Consumo: Tomo una Bebida
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 1/1/2024 14:13:37
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Israel Tomo una Bebida 1/1/2024 14:13:37
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-f182blad73384
081d48d2f0e1eb29bdf
Conexión exitosa
Nombre: Marco
Comentario de Consumo: Comio un Pica Pollo
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 2/1/2024 14:20:50
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Marco Comio un Pica Pollo 2/1/2024 14:20:50
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 % /Users/ISMARTZ/.vscode/extensions/ms-dotnettools.csharp-2.14.8-darwin-arm64/.de
bugger/arm64/vsdbg --interpreter=vscode --connection=/var/folders/58/qlxpm5wx1774yfhfr9gpgjvm000gn/T/CoreFxPipe_vsdg-ui-48142ab04df04
ee78134bad7c5cf28fb
Conexión exitosa
Nombre: Ezequiel
Comentario de Consumo: Tomo una bebida
Fecha de Ingreso (YYYY-MM-DD HH:mm:ss): 1/1/2024 14:13:37
Estado: Activo
Consumo registrado exitosamente
Nombres Comentario Fecha
Israel Tomo una Bebida 1/1/2024 14:13:37
Marco Comio un Pica Pollo 2/1/2024 14:20:50
Ezequiel Tomo una bebida 1/1/2024 14:13:37
israelporfirioarmarteizencarnacion@ISMARTZ-2 PruebaB0 %

Archivos Zip y Documentos

<https://github.com/ISMARTZDEV/IDS-3---Semana-4---Tarea-6.git>