

Crear un archivo Excel con C# y .NET8, conectando a una Base de Datos realizada con MySQL, exportando estos datos al archivo Excel y con la utilización de Variables de Entorno

Hacer un Programa en C# con la utilización del Framework .NET8 que realice lo siguiente:

1.- Conexión a la Base de Datos MySQL

¿Cómo se establece la conexión a la base de datos usando MySqlConnection?

¿Qué hace el comando connection.Open()?

2.- Ejecución de la Consulta SQL

¿Cómo se crea y ejecuta un comando SQL con MySqlCommand?

¿Cuál es el propósito del MySqlDataAdapter y cómo se usa para llenar un DataTable?

3.- Creación del Archivo Excel

¿Cómo se utiliza ClosedXML para crear un archivo Excel?

¿Cómo se añade una hoja de trabajo y se insertan los datos del DataTable en la hoja de trabajo?

4.- Manejo de Excepciones

¿Cómo se maneja cualquier error que pueda ocurrir durante la ejecución del código?

```
using System;
using System.Data;
using MySql.Data.MySqlClient;
using ClosedXML.Excel;

namespace ExportarDatosAExcel
{
    class Program
    {
        static void Main(string[] args)
        {
            //CADENA DE CONEXIÓN
            string connectionString =
"Server=localhost;Port=3306;Database=Import_Tech;User Id=root;Password=3001Epica**";

            //CONSULTA SQL
            string query = "SELECT * FROM producto";

            //RUTA DEL ARCHIVO EXCEL
            string filePath = "productoExcel.xlsx";

            //BLOQUE DE CONTROL DE ERRORES O EXCEPCIONES
            try
            {
                //CREAR CONEXIÓN
                using (MySqlConnection connection = new MySqlConnection(connectionString))
```


CORRIDA DEL PROGRAMA

1.- Salida de Resultados en Consola:

C:\WINDOWS\system32\cmd.exe

```
Conexión exitosa.  
Datos exportados exitosamente a productoExcel.xlsx  
Presione una tecla para continuar . . .
```

2.- Dónde ver el Archivo Excel en el que se guardaron los datos exportados de la Base de Datos

C:\Users\epica\CSharp_Project\Conection_DataBase\bin\Debug

2.1.- Inicialmente creé una carpeta como repositorio de mis proyectos en C#, la carpeta **CSharp_Project**; este repositorio lo alojé en la dirección de mi disco duro **C:\Users\epica**

2.2.- Cuando creamos el proyecto con Visual Studio llamamos al proyecto **Conection_DataBase**, automáticamente Visual Studio creó una carpeta con ese mismo nombre **Conection_DataBase** en la dirección de mi disco duro asignada a ese proyecto.

2.3.- En esta última carpeta Visual Studio creó todos los componentes, archivos y subcarpetas requeridos para el correcto funcionamiento de mi proyecto.

2.4.- Quedando la ubicación definitiva de mi proyecto adentro de mi computador en la dirección: **C:\Users\epica\CSharp_Project\Conection_DataBase**

a.- En donde **C:** es: Raíz de mi disco duro.

b.- En donde **Users** es: El directorio de todos los usuarios de mi equipo.

c.- En donde **epica** es: Mi usuario.

d.- En donde **CSharp_Project** es: El repositorio o carpeta de mis proyectos en C#.

e.- En donde **Conection_DataBase** es: La carpeta creada por VS con el nombre de mi proyecto.

2.5.- Para conseguir el archivo Excel creado con las peticiones especificadas dentro del código, debemos navegar hasta las carpetas creadas por VS adentro de mi proyecto **Conection_DataBase**, que son: **bin\Debug**

2.6.- Quedando como la ruta de búsqueda del archivo Excel con los datos exportados:

C:\Users\epica\CSharp_Project\Conection_DataBase\bin\Debug

2.7.- El archivo Excel lo llamamos en nuestro código: **productoExcel.xlsx**, hacemos doble Click en este y se abrirá el archivo con Excel, mostrando todos los Datos exportados desde la Base de Datos en SQL (En mi caso MySQL).

Archivo Excel en el que se guardaron los datos exportados de la Base de Datos conectada

ID_Producto	Nombre_Producto	Descripción_Producto	PrecioCompra	PrecioVenta	Descuento_Producto	Stock	ID_Categoria	NIP_Proveedor
1	Televisor Samsung 55"	Televisor LED 4K UHD	500	700	10	50	1	1
2	Televisor LG 65"	Televisor OLED 4K UHD	1000	1300	10	30	1	2
3	Televisor Sony 75"	Televisor LED 4K UHD	1200	1500	10	20	1	3
4	Televisor Panasonic 50"	Televisor LED Full HD	400	600	10	40	1	4
5	Televisor Philips 43"	Televisor LED Full HD	300	500	10	60	1	5
6	Laptop Dell XPS 13	Laptop con procesador Intel i7	800	1000	5	25	2	6
7	Laptop HP Spectre x360	Laptop convertible con procesador Intel i5	900	1100	5	20	2	7
8	Laptop MacBook Pro	Laptop con procesador Apple M1	1200	1500	5	15	2	8
9	Laptop Lenovo ThinkPad	Laptop con procesador Intel i7	700	900	5	30	2	9
10	Laptop Asus ZenBook	Laptop con procesador AMD Ryzen 7	750	950	5	20	2	10
11	iPhone 13	Teléfono móvil con pantalla OLED	800	1000	15	50	3	11
12	Samsung Galaxy S21	Teléfono móvil con pantalla AMOLED	700	900	15	40	3	12
13	Google Pixel 6	Teléfono móvil con pantalla OLED	600	800	15	30	3	13
14	OnePlus 9	Teléfono móvil con pantalla AMOLED	500	700	15	35	3	14
15	Xiaomi Mi 11	Teléfono móvil con pantalla AMOLED	450	650	15	45	3	15
16	iPad Pro	Tableta con pantalla Liquid Retina	900	1100	5	25	4	1
17	Samsung Galaxy Tab S7	Tableta con pantalla AMOLED	700	900	5	30	4	2
18	Microsoft Surface Pro 7	Tableta con pantalla PixelSense	800	1000	5	20	4	3
19	Lenovo Tab P11	Tableta con pantalla IPS	300	500	5	40	4	4
20	Amazon Fire HD 10	Tableta con pantalla Full HD	150	300	5	50	4	5
21	Canon EOS R5	Cámara digital sin espejo	3000	3500	12	10	5	6
22	Nikon Z7 II	Cámara digital sin espejo	2500	3000	12	15	5	7
23	Sony Alpha a7 III	Cámara digital sin espejo	2000	2500	12	20	5	8
24	Fujifilm X-T4	Cámara digital sin espejo	1500	2000	12	25	5	9
25	Panasonic Lumix GH5	Cámara digital sin espejo	1200	1700	12	30	5	10

CODIGO SIN EL USO DE CLASES EN C#

```
using System;
using System.Data;
using MySql.Data.MySqlClient;
using ClosedXML.Excel;

namespace ExportarDatosAExcel
{
    class Program
    {
        static void Main(string[] args)
        {
            //CADENA DE CONEXIÓN
            string connectionString =
"Server=localhost;Port=3306;Database=Import_Tech;User Id=root;Password=3001Epica**";

            //CONSULTA SQL
            string query = "SELECT * FROM producto";

            //RUTA DEL ARCHIVO EXCEL
            string filePath = "productoExcel.xlsx";
        }
    }
}
```

```

        //BLOQUE DE CONTROL DE ERRORES O EXCEPCIONES
        try
        {
            //CREAR CONEXIÓN
            using (MySqlConnection connection = new
MySqlConnection(connectionString))
            {
                connection.Open();
                Console.WriteLine("Conexión exitosa.");

                //EJECUTAR LA CONSULTA
                using (MySqlCommand command = new MySqlCommand(query, connection))
                {
                    //CREAR UN ADAPTADOR DE DATOS
                    using (MySqlDataAdapter dataAdapter = new
MySqlDataAdapter(command))
                    {
                        DataTable dataTable = new DataTable();
                        dataAdapter.Fill(dataTable);

                        //CREAR EL ARCHIVO EXCEL
                        using (var workbook = new XLWorkbook())
                        {
                            var worksheet = workbook.Worksheets.Add("Producto");
                            worksheet.Cell(1, 1).InsertTable(dataTable);
                            workbook.SaveAs(filePath);
                        }

                        //MENSAJE DE EXITO
                        Console.WriteLine("Datos exportados exitosamente a " +
filePath);
                    }
                }
            }
        }

        //BLOQUE DE MANEJO DE EXCEPCIONES
        catch (Exception ex)
        {
            Console.WriteLine("Error: " + ex.Message);
        }
    }
}
}

```

CODIGO EN C# EXPLICADO

```
using System; //Importar la Librería System: Librería base en .NET, que contiene
               clases fundamentales, tipos de datos básicos como String, Int32,
               DateTime, y muchas otras. Permite el uso de clases como Console para
               imprimir mensajes en la consola.

using System.Data; //Importar la Librería System.Data: Proporciona clases para
                   trabajar con datos y bases de datos. Incluye clases como
                   DataTable, DataSet, DataRow, y DataColumn.
                   Permite la creación y manipulación de tablas de datos en memoria.

using MySql.Data.MySqlClient; //Cambiado a MySql.Data.MySqlClient.
                               Importar la Librería MySql.Data.MySqlClient, que
                               contiene clases específicas para trabajar con bases de
                               datos MySQL. Incluye clases como MySqlConnection,
                               MySqlCommand, y MySqlDataAdapter.
                               Permite la conexión y ejecución de comandos en una
                               base de datos MySQL.

using ClosedXML.Excel; //Importar la Librería ClosedXML.Excel: Proporciona clases para
                       trabajar con archivos Excel de manera sencilla. Incluye
                       clases como XLWorkbook y IXLWorksheet.
                       Permite la creación y manipulación de archivos Excel.

namespace ExportarDatosAExcel
{
    class Program
    {
        static void Main(string[] args)
        {
            //CADENA DE CONEXIÓN
            string connectionString =
"Server=localhost;Port=3306;Database=Import_Tech;User Id=root;Password=3001Epica**";
            //Propósito: Esta línea define una cadena de conexión que se utiliza para
            establecer una conexión con la base de datos MySQL.
            //Componentes:
            //Server = localhost: Especifica que el servidor de la base de datos está
            en la misma máquina(localhost).
            //Port = 3306: Indica el puerto en el que MySQL está escuchando(3306 es el
            puerto predeterminado para MySQL).
            //Database = Import_Tech: Especifica el nombre de la base de datos a la
            que se desea conectar(Import_Tech).
            //User Id = root: Indica el nombre de usuario que se utilizará para la
            conexión(root).
            //Password = 3001Epica: Proporciona la contraseña del usuario para la
            autenticación.
```

```

//CONSULTA SQL
string query = "SELECT * FROM producto";
//Propósito: Esta línea define una consulta SQL que se utilizará para
                obtener datos de la base de datos.
//Componentes:
//SELECT *: Indica que se desean seleccionar todas las columnas de la
                tabla.
//FROM producto: Especifica que los datos se obtendrán de la tabla llamada
                producto.

//RUTA DEL ARCHIVO EXCEL
string filePath = "productoExcel.xlsx";
//Propósito: Esta línea define la ruta y el nombre del archivo Excel donde
                se guardarán los datos exportados.
//Componentes:
//productoExcel.xlsx: Especifica el nombre del archivo Excel.
                        Si no se proporciona una ruta completa, el archivo
                        se guardará en el directorio de trabajo actual.

//BLOQUE DE CONTROL DE ERRORES O EXCEPCIONES
try //Propósito: Se utiliza para encapsular el código que puede generar
                excepciones.
                Si ocurre una excepción, el control se transfiere al
                bloque catch, donde se maneja el error.
                Permite manejar errores de manera controlada y mostrar
                mensajes de error en la consola.

{
    //CREAR CONEXIÓN
    using (MySqlConnection connection = new
MySqlConnection(connectionString))
    {
        //Propósito: Establece una conexión con la base de datos MySQL
                utilizando la cadena de conexión connectionString.
        //Componentes:
        //MySqlConnection connection = new MySqlConnection(connectionString):
                Crea una nueva instancia de MySqlConnection con la cadena de
                conexión proporcionada.

        connection.Open(); //Abre la conexión a la base de datos.
        Console.WriteLine("Conexión exitosa."); //Imprime un mensaje en la
                consola indicando que la
                conexión fue exitosa.
    }
}

```

```
//EJECUTAR LA CONSULTA
using (MySqlCommand command = new MySqlCommand(query, connection))
{
    //Propósito: Ejecuta la consulta SQL y llena un DataTable con los
        resultados.
    //Componentes:
    //using: Asegura que el libro de Excel se cierre y se liberen los
        recursos al final del bloque.
    //MySqlCommand command = new MySqlCommand(query, connection):
        Crea una nueva instancia de MySqlCommand con la consulta SQL y
        la conexión abierta.
```

```

        //CREAR UN ADAPTADOR DE DATOS
        using (MySqlDataAdapter dataAdapter = new
MySqlDataAdapter(command))
        {
            //Propósito: Actuar como puente entre la base de datos y el
                DataTable.
            {
                DataTable dataTable = new DataTable(); //Crea una nueva
                    instancia de DataTable para almacenar los datos.

                dataAdapter.Fill(dataTable); //Llena el DataTable con los
                    datos obtenidos de la
                    consulta Fila por Fila.

            //CREAR EL ARCHIVO EXCEL
            using (var workbook = new XLWorkbook())
            {
                var worksheet = workbook.Worksheets.Add("Producto");
                //Agrega una nueva hoja de cálculo llamada "Producto"
                    al libro de Excel.
            //Propósito:
                Crea un archivo Excel y guarda los datos del
                DataTable en una hoja de cálculo.
            //Componentes:
                var workbook = new XLWorkbook(): Crea una nueva
                    instancia de
                    XLWorkbook, que
                    Representa un libro de
                    Excel.

                worksheet.Cell(1, 1).InsertTable(dataTable);
                //Inserta los datos del DataTable en la celda A1 de la
                    hoja de cálculo como una tabla.
```



```

        workbook.SaveAs(filePath); //Guarda el libro de Excel
                                    en la ruta especificada
                                    por filePath.
    }

    //MENSAJE DE ÉXITO
    Console.WriteLine("Datos exportados exitosamente a " +
filePath);

    //Propósito: Imprime un mensaje en la consola indicando
    que los datos se exportaron exitosamente al archivo
    Excel en la ruta especificada por filePath.
    }
}

//BLOQUE DE MANEJO DE EXCEPCIONES
catch (Exception ex)
{
    //Propósito: El bloque catch se utiliza para manejar excepciones que
    puedan ocurrir dentro del bloque try.
    Si se lanza una excepción en el bloque try, el control se transfiere al
    bloque catch, donde se puede manejar el error de manera adecuada.
    //Componentes:
    Exception: Exception es la clase base para todas las excepciones
    en.NET.
    ex: Es una variable que contiene la instancia de la excepción que fue
    lanzada. Esta variable proporciona información sobre el error que
    ocurrió.

    Console.WriteLine("Error: " + ex.Message);
    //Console.WriteLine("Error: " + ex.Message):Esta línea imprime un
    mensaje en la consola que
    incluye el texto "Error: "
    seguido del mensaje de la
    excepción
    (ex.Message). //El mensaje de la excepción proporciona detalles
    acerca de lo que
    causó el error.
}
}
}
}
}

```

EN BASE AL CÓDIGO ANTERIOR REALIZAR UN CÓDIGO QUE HAGA LA MISMAS FUNCIONES, PERO QUE ESTE CÓDIGO TENGA ESTAS NUEVAS CARACTERÍSTICAS QUE LO MEJOREN, SEPARÁNDOLO EN VARIAS CLASES Y FUNCIONES PARA UNA MEJOR ORGANIZACIÓN Y SEGURIDAD.

PROYECTO: ExportarDatosAExcel

- 1.- Separar en varios archivos o funciones por responsabilidades creando Clases.
- 2.- La Función Principal Main debe llamar a todas estas clases.
- 3.- Configurar, Crear y leer aquello que se requiera con Variables de Entorno, para tener una Seguridad de Datos.

Paso 1: Crear las Clases

Clase: DatabaseHelper

Esta clase manejará la conexión a la base de datos y la ejecución de la consulta.

```
using System;
using System.Data;
using MySql.Data.MySqlClient;

namespace ExportarDatosAExcelClasesUserPassworMySQL
{
    public class DatabaseHelper
    {
        private string connectionString;

        public DatabaseHelper(string connectionString)
        {
            this.connectionString = connectionString;
        }

        public DataTable ExecuteQuery(string query)
        {
            DataTable dataTable = new DataTable();
            try
            {
                using (MySqlConnection connection = new MySqlConnection(connectionString))
                {
                    connection.Open();
                    using (MySqlCommand command = new MySqlCommand(query, connection))
                    {
                        using (MySqlDataAdapter dataAdapter = new MySqlDataAdapter(command))
                        {
                            dataAdapter.Fill(dataTable);
                        }
                    }
                }
            }
        }
    }
}
```

```

        catch (Exception ex)
        {
            Console.WriteLine("Error: " + ex.Message);
        }
        return dataTable;
    }
}
}

```

Clase: DatabaseConnetion

```

using System;
using MySql.Data.MySqlClient;

namespace ExportarDatosAExcelClasesUserPassworMySQL
{
    public class DatabaseConnection
    {
        public string GetConnectionString()
        {
            string connectionString = null;
            bool isConnected = false;

            while (!isConnected)
            {
                Console.Write("Ingrese el usuario de la base de datos: ");
                string user = Console.ReadLine();

                Console.Write("Ingrese la contraseña de la base de datos: ");
                string password = ReadPassword();

                Console.Write("Ingrese Nombre del Servidor: ");
                string server = Console.ReadLine();

                Console.Write("Ingrese Nombre de la Base de Datos: ");
                string database = Console.ReadLine();

                Console.Write("Ingrese Número de Puerto: ");
                string port = Console.ReadLine();

                connectionString =
                $"Server={server};Database={database};Port={port};UserId={user};Password={password}";

                // Intentar conectar a la base de datos
                try
                {

```

```

        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            isConnected = true;
        }
    }
    catch (MySqlException)
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.BackgroundColor = ConsoleColor.Yellow;
        Console.WriteLine("ERROR EN VALIDACION DE DATOS / Inténtelo de nuevo");
        Console.WriteLine();
        Console.ResetColor();
    }
}
return connectionString;
}
private string ReadPassword()
{
    string password = string.Empty;
    ConsoleKeyInfo key;
    do
    {
        key = Console.ReadKey(true);
        if (key.Key != ConsoleKey.Backspace && key.Key != ConsoleKey.Enter)
        {
            password += key.KeyChar;
            Console.Write("*");
        }
        else
        {
            if (key.Key == ConsoleKey.Backspace && password.Length > 0)
            {
                password = password.Substring(0, (password.Length - 1));
                Console.Write("\b \b");
            }
        }
    }
    while (key.Key != ConsoleKey.Enter);
    Console.WriteLine();
    return password;
}
}
}

```

Clase **ExcelExporter**

Esta clase manejará la exportación de datos a un archivo Excel.

```
using System;
using System.Data;
using ClosedXML.Excel;

namespace ExportarDatosAExcelClasesUserPassworMySQL
{
    public class ExcelExporter
    {
        public void ExportToExcel(DataTable dataTable, string filePath, string table)
        {
            using (var workbook = new XLWorkbook())
            {
                var worksheet = workbook.Worksheets.Add(table);
                //var worksheet = workbook.Worksheets.Add("Producto");
                worksheet.Cell(1, 1).InsertTable(dataTable);
                workbook.SaveAs(filePath);
            }
        }
    }
}
```

Paso 2: Modificar la Clase **Program** o **Main**

La clase **Program** ahora llamará a las clases **DatabaseHelper** y **ExcelExporter**.

```
using System;
using System.Data;
using Spectre.Console;
using DotNetEnv;
using System.IO;

namespace ExportarDatosAExcelClasesUserPassworMySQL
{
    class Program
    {
        static void Main(string[] args)
        {
            // CONFIGURACIÓN DE VARIABLE DE ENTORNO
            string envFilePath = Path.Combine(Directory.GetCurrentDirectory(), ".env");
            Env.Load(envFilePath);
            while (true) // Bucle para reiniciar el programa en caso de error
            {
                Try
                {

```

```

// LEER VARIABLE DE ENTORNO
string? connectionString = Environment.GetEnvironmentVariable("DB_CONNECTION_STRING");
// IMPRIMIR ESTADO DE LA VARIABLE DE ENTORNO
Console.WriteLine($"Valor Inicial de DB_CONNECTION_STRING:
{connectionString ?? "NULL"}");

// VERIFICAR SI ES NECESARIO PEDIR LOS DATOS DE CONEXIÓN
if (string.IsNullOrEmpty(connectionString))
{
    var dbConnection = new DatabaseConnection();
    connectionString = dbConnection.GetConnectionString() ?? throw
new InvalidOperationException("No se pudo obtener una cadena de conexión válida.");
}

// SOLICITAR TABLA A EXPORTAR
Console.Write("Ingrese Nombre de Tabla a Exportar: ");
string? table = Console.ReadLine();

if (string.IsNullOrEmpty(table))
{
    throw new ArgumentException("El nombre de la tabla no puede estar vacío.");
}

// CONSTRUIR CONSULTA Y ARCHIVO DE SALIDA
string query = $"SELECT * FROM `{table}`"; //Usar backticks para proteger nombres de tablas
string filePath = $"{table}_Export.xlsx";

// CREAR INSTANCIAS DE LAS CLASES NECESARIAS
var dbHelper = new DatabaseHelper(connectionString);
var excelExporter = new ExcelExporter();
var dataTable = dbHelper.ExecuteQuery(query);

// VALIDAR RESULTADOS DE LA CONSULTA
if (dataTable.Rows.Count > 0)
{
    excelExporter.ExportToExcel(dataTable, filePath, table);
    AnsiConsole.MarkupLine($"[bold white on blue]Datos exportados
exitosamente a {filePath}[/]");
}
else
{
    AnsiConsole.MarkupLine("[bold yellow on red]No se encontraron
datos para exportar.[/]");
}

```


3.2.1.- EN WINDOWS:

3.2.1.1.- Desde la línea de comandos (CMD):

`setx NOMBRE_VARIABLE_ENTORNO "Server=localhost; Database=Nombre_Base_de_Datos; Port=Número_Puerto; User Id=Nombre_Usuario; Password=Contraseña;"`

a.- Configurar Variable de Entorno de Usuario

`setx DB_CONNECTION_STRING "Server=localhost;Database=Import_Tech;Port=3306;User Id=root;Password=3001Epica**;"`

b.- Configurar Variable de Entorno de Sistema

`setx DB_CONNECTION_STRING "Server=localhost;Database=Import_Tech;Port=3306;User Id=root;Password=3001Epica**;" /M`

3.2.1.2.- Desde PowerShell:

`[System.Environment]::SetEnvironmentVariable("DB_CONNECTION_STRING", "Server=localhost; Database=Nombre_Base_de_Datos; Port=Número_Puerto; User Id=Nombre_Usuario; Password=Contraseña;", "Machine")`

3.2.1.3.- Desde la Interfaz Gráfica:

- a.- Abrir el Panel de Control.
- b.- Ir a Sistema y Seguridad > Sistema.
- c.- Hacer clic en Configuración avanzada del sistema.
- d.- En la pestaña Opciones avanzadas, hacer clic en Variables de entorno.
- e.- En Variables del sistema, hacer clic en Nueva y añadir DB_CONNECTION_STRING con el valor correspondiente.

3.2.2.- EN LINUX:

a.- `export DB_CONNECTION_STRING="Server=localhost; Database=Nombre_Base_de_Datos; Port=Número_Puerto; User Id=Nombre_Usuario; Password=Contraseña;"`

b.- Para que esta configuración sea permanente: Añadir la línea anterior al archivo `~/.bashrc` o al archivo `~/.bash_profile`.

3.2.3.- EN AZURE:

3.2.3.1.- Configuración de aplicaciones web:

- a.- Ir al portal de Azure.
- b.- Seleccionar la aplicación web.
- c.- Ir a Configuración > Configuración de la aplicación.
- d.- Añadir una nueva configuración con el nombre DB_CONNECTION_STRING y el valor correspondiente.

3.2.4.- EN AWS (Elastic Beanstalk):

3.2.4.1.- Configuración de entorno:

- a.- Ir a la consola de Elastic Beanstalk.
- b.- Seleccionar el entorno.
- c.- Ir a Configuration > Software.
- d.- Añadir una nueva variable de entorno con el nombre DB_CONNECTION_STRING y el valor correspondiente.

3.2.5.- CONFIGURACIÓN EN ARCHIVO .env:

Un archivo **.env** es un archivo de texto plano que se utiliza para almacenar variables de entorno en formato clave-valor. Es muy utilizado en aplicaciones para gestionar configuraciones sin exponerlas directamente en el código.

Reglas importantes:

- 1.- Cada línea representa una variable en el formato **CLAVE=VALOR**.
- 2.- No debe haber espacios alrededor del **=**. Ejemplo correcto: **APP_PORT=5000**.
- 3.- Los valores con espacios deben estar entre comillas (" " o ' ').

Ejemplo de archivo .env:

```
DB_CONNECTION_STRING=Server=localhost;Database=mi_base_de_datos;User;Id=mi_usuario;Password=mi_contraseña;  
API_KEY=123456789abcdef  
APP_PORT=5000
```

Pasos para usar un archivo .env en un proyecto:

1.- Crear el archivo .env

- a.- En el proyecto, crear un archivo llamado **.env** en el directorio raíz.
- b.- Añade los variables de entorno dentro de este archivo.

En nuestro caso este es el archivo .env:

```
DB_CONNECTION_STRING=Server=Nombre_Servidor;Database=Nombre_BaseDatos;Port=Numero_Puerto  
;UserId=Tu_Usuario;Password=Tu_Clave_Usuario;
```

2.- Asegurarse de ignorar el archivo .env

- a.- El archivo **.env** no debe subirse a **Git** ni a ningún repositorio público, ya que puede contener datos sensibles.
- b.- Incluir el archivo **.env** en el archivo **.gitignore** previo a la utilización de **Git** para realizar el control de versiones del proyecto.

Ejemplo de archivo .gitignore:

```
# Ignorar el archivo .env  
.env
```

3.- Leer las variables de entorno en el código

Para leer las variables almacenadas en el archivo **.env**, se necesitará una biblioteca específica según el lenguaje que se use; en nuestro caso lo haremos en C# y con **.NET** según lo siguiente:

a.- Instalar la biblioteca **DotNetEnv**.

Usar el siguiente comando en la terminal para instalarla:

En la terminal:

```
dotnet add package DotNetEnv
```

b.- Cargar y usar el archivo .env

Añadir el siguiente código en el proyecto:

```
using DotNetEnv;  
//CARGAR EL ARCHIVO .env  
Env.Load();
```

4.- Buenas prácticas con archivos .env

4.1.- No incluir datos sensibles en el código fuente:

- a.- Usar **.env** para almacenar información como contraseñas o claves API.

4.2.- Utilizar valores por defecto:

- a.- Si una variable de entorno no está definida, se debe proporcionar valores predeterminados en el código.

4.3.- Usar múltiples archivos **.env** para diferentes entornos:

- a.- **.env.development** : Para desarrollo.
- b.- **.env.production** : Para producción.
- c.- Podría cargarse un archivo específico según el entorno de ejecución.

Ejemplo de carga condicional en C#:

En C#:

```
using DotNetEnv;  
  
string environment = Environment.GetEnvironmentVariable("ASPNETCORE_ENVIRONMENT") ??  
"development";  
  
if (environment == "production")  
{  
    Env.Load(".env.production");  
}  
else  
{  
    Env.Load(".env.development");  
}
```

4.- CONFIGURACIÓN Temporal vs. Permanente

4.1.- Configuración Temporal:

- a.- Desde la Línea de Comandos (CMD o Terminal): Si se configura una variable de entorno en una sesión de terminal o línea de comandos, esta configuración solo dura mientras la sesión esté activa.
- b.- Una vez que cierras la terminal o línea de comandos, la configuración desaparece.

4.2.- Configuración Permanente:

4.2.1.- Configuración del Sistema:

- a.- Si se configura una variable de entorno a nivel del sistema (por ejemplo, a través de la interfaz gráfica en Windows o añadiéndola a archivos de configuración en Linux), la configuración permanece hasta que se cambie o se elimine manualmente.
- b.- Ejemplo en Windows (a través de la interfaz gráfica): Esta configuración permanece hasta que la elimines manualmente desde el Panel de Control.
- c.- Ejemplo en Linux (añadiendo al archivo ~/.bashrc o ~/.bash_profile): Esta configuración se carga cada vez que inicias una nueva sesión de terminal.

4.3.- Resumen:

4.3.1.- **Temporal:** Configurada en una sesión de terminal, desaparece al cerrar la sesión.

4.3.2.- **Permanente:** Configurada a nivel del sistema o en archivos de configuración, permanece hasta que se elimine manualmente.

5.- OPERACIONES de Variables de Entorno con la Terminal CMD de Windows

5.1.- Listar Variables de Entorno de Usuario

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para listar variables de entorno de usuario:

Escribir el comando y presiona Enter:

reg query "HKCU\Environment"

5.2.- Listar Variables de Entorno del Sistema

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para listar variables de entorno del sistema:

Escribir el comando y presiona Enter:

reg query "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"

5.3.- Listar todas las Variables de Entorno

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para listar todas las variables de entorno existentes:

Escribir el comando y presiona Enter: **set**

5.4.- Mostrar una Variable de Entorno en el caso de que esté configurada

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para listar el contenido de una Variables de Entorno específica:

Escribir el comando y presiona Enter: **set Nombre_Variable_Entorno**

5.5.- Mostrar solamente el contenido de una Variables de Entorno

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para mostrar el contenido de una Variables de Entorno específica:

Escribir el comando y presiona Enter: **echo %Nombre_Variable_Entorno%**

5.6.- Configurar una Variables de Entorno de Usuario

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para configurar una Variables de Entorno específica:

Escribir el comando y presiona Enter: **setx Nombre_Variable_Entorno "Mi_Valor"**

Ejemplo: **setx Nombre_Variable "Server=localhost;Database=Import_Tech;Port=3306;User Id=root;Psw=A1b2C3d;"**

5.7.- Configurar una Variables de Entorno del Sistema

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para configurar una Variables de Entorno específica:

Escribir el comando y presiona Enter: **setx Nombre_Variable_Entorno "Mi_Valor" /M**

Ejemplo: **setx Nombre_Variable "Server=localhost;Database=Import_Tech;Port=3306;User Id=root;Psw=A1b2C3d;" /M**

5.8.- Borrar una Variables de Entorno de Usuario

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para borrar una Variables de Entorno de Usuario específica:

Escribir el comando y presiona Enter:

reg delete "HKCU\Environment" /F /V *Nombre_Variable_Entorno*

5.9.- Borrar una Variables de Entorno del Sistema

a.- Abrir Terminal CMD como Administrador

b.- Ejecutar el comando para borrar una Variables de Entorno del Sistema específica:

Escribir el comando y presiona Enter:

reg delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment" /F /V *Nombre_Variable*

6.- VENTAJAS de Variables de Entorno Usuario Vs. Variables de Entorno del sistema

6.1.- Ventajas de las Variable de Entorno de Usuario

a.- Es específica para el usuario actual.

b.- No requiere privilegios de administrador para configurarla.

c.- Es ideal si solo un usuario necesita acceder a esta variable en su entorno de desarrollo.

6.1.- Ventajas de las Variable de Entorno del Sistema

a.- Disponible para todos los usuarios del sistema.

b.- Útil si varios usuarios o servicios en el mismo sistema necesitan acceder a esta variable.

c.- Requiere privilegios de administrador para configurarla.

7.- Posibles problemas al correr un código con Variables de Entorno y sus Soluciones

7.1.- Verificar la Lectura de la Variable de Entorno:

Asegurarse que la línea que lee la variable con un proceso interno adicional, como el de solicitar la impresión de dicha la variable, un ejemplo en C# sería:

Console.WriteLine("Connection String from Environment: " + *Nombre_Variable*);

Con esto se puede verificar que esté imprimiendo el valor correcto. Si no imprime nada, significa que la variable de entorno no se está leyendo correctamente.

7.2.- Reiniciar Visual Studio:

A veces, las variables de entorno configuradas recientemente no se reconocen hasta que se reinicia Visual Studio; se debe intentar cerrar y volver a abrir Visual Studio.

7.3.- Permisos de Visual Studio:

Asegurarse de que Visual Studio se esté ejecutando con los permisos necesarios para leer las variables de entorno del sistema. Podemos intentar ejecutar Visual Studio como administrador.

7.4.- Verificar el Nombre de la Variable de Entorno:

Asegurarse de que no haya errores tipográficos en el nombre de la variable de entorno.

Documentación del Proyecto

Nombre del Proyecto: ExportarDatosAExcelClasesUserPassworMySQL

Descripción General

Este proyecto tiene como objetivo exportar datos de una base de datos MySQL a un archivo Excel utilizando C# y las bibliotecas MySql.Data y ClosedXML. El código está organizado en varias clases para mejorar la mantenibilidad y la seguridad.

Estructura del Proyecto

El proyecto está compuesto por las siguientes clases:

DatabaseConnection: Obtiene y verifica la cadena de conexión a la base de datos MySQL.

DatabaseHelper: Ejecuta consultas SQL y maneja la conexión a la base de datos.

ExcelExporter: Exporta datos desde un DataTable a un archivo Excel.

Program: Coordina la ejecución de las clases anteriores y es el punto de entrada del programa.

Clases y Métodos

Clase DatabaseHelper

Propósito:

Esta clase se encarga de ejecutar consultas SQL en una base de datos MySQL y devolver los resultados en un objeto DataTable.

Explicación:

1.- Constructor:

1.1.- Inicialización: El constructor recibe una cadena de conexión (connectionString) y la almacena en una variable privada para su uso posterior.

2.- Método ExecuteQuery:

2.1.- Inicialización: Crea un nuevo objeto DataTable para almacenar los resultados de la consulta.

2.2.- Conexión a la Base de Datos: Utiliza la cadena de conexión para abrir una conexión a la base de datos MySQL.

2.3.- Ejecución de la Consulta: Crea un comando SQL (MySqlCommand) con la consulta proporcionada y lo ejecuta.

2.4.- Rellenar el DataTable: Utiliza un adaptador de datos (MySqlDataAdapter) para ejecutar la consulta y rellenar el DataTable con los resultados.

2.5.- Manejo de Errores: Captura cualquier excepción que ocurra durante la ejecución de la consulta y muestra un mensaje de error en la consola.

Razón de Uso:

1.- **Abstracción:** Simplifica la ejecución de consultas SQL al encapsular la lógica de conexión y ejecución en una clase reutilizable.

2.- **Reusabilidad:** Permite ejecutar diferentes consultas SQL sin duplicar el código de conexión y manejo de resultados.

3.- **Manejo de Errores:** Proporciona un manejo centralizado de errores, lo que facilita la depuración y el mantenimiento del código.

Clase **ExcelExporter**

Propósito: Exportar los datos de un objeto DataTable a un archivo Excel utilizando la biblioteca ClosedXML.

Explicación:

1.- Método **ExportToExcel**:

- 1.1.- Inicialización del Libro de Trabajo: Crea una nueva instancia de XLWorkbook, que representa un libro de trabajo de Excel.
- 1.2.- Creación de la Hoja de Trabajo: Agrega una nueva hoja de trabajo al libro de trabajo y la nombra "Producto".
- 1.3.- Inserción de Datos: Inserta los datos del DataTable en la hoja de trabajo comenzando desde la celda (1, 1).
- 1.4.- Guardado del Archivo: Guarda el libro de trabajo en la ruta especificada por filePath.

Razón de Uso:

- 1.- **Facilidad de Uso:** Simplifica el proceso de exportar datos a Excel al encapsular toda la lógica necesaria en un solo método.
- 2.- **Reusabilidad:** Permite reutilizar el mismo método para exportar diferentes conjuntos de datos a archivos Excel.
- 3.- **Integración con ClosedXML:** Aprovecha la biblioteca ClosedXML, que proporciona una API fácil de usar para trabajar con archivos Excel en .NET.

Clase **DatabaseConnetion**

Propósito:

Se encarga proporcionar una forma de obtener una cadena de conexión a una base de datos MySQL. Si no se puede leer la cadena de conexión desde una variable de entorno, solicita al usuario que ingrese las credenciales de la base de datos y verifica la conexión antes de devolver la cadena de conexión.

Explicación:

1.- Método **GetConnectionString**:

- 1.1.- **Inicialización:** Declara una variable connectionString y un booleano isConnected para controlar el ciclo de conexión.
- 1.2.- **Bucle de Conexión:** Mientras isConnected sea false, solicita al usuario que ingrese el nombre de usuario y la contraseña de la base de datos.
- 1.3.- **Construcción de la Cadena de Conexión:** Construye la cadena de conexión usando los datos ingresados por el usuario.
- 1.4.- **Intento de Conexión:** Intenta abrir una conexión a la base de datos usando MySqlConnection. Si la conexión es exitosa, isConnected se establece en true.
- 1.5.- **Manejo de Errores:** Si la conexión falla, muestra un mensaje de error y solicita al usuario que intente nuevamente.

2.- Método ReadPassword:

2.1.- Lectura Segura de Contraseña: Lee la contraseña ingresada por el usuario sin mostrarla en la consola, reemplazándola con asteriscos (*).

Razón de Uso:

- 1.- Seguridad:** Permite ingresar credenciales de manera segura y verifica la conexión antes de proceder, asegurando que las credenciales sean correctas.
- 2.- Flexibilidad:** Si la cadena de conexión no está disponible como variable de entorno, permite al usuario ingresar las credenciales manualmente.
- 3.- Reusabilidad:** Centraliza la lógica de obtención de la cadena de conexión, facilitando su reutilización en diferentes partes del programa.

Clase Program

Propósito: El propósito del método Main en la clase Program es coordinar el flujo principal de la aplicación, que incluye la lectura de una cadena de conexión desde una variable de entorno, la ejecución de una consulta SQL para obtener datos de una base de datos MySQL, y la exportación de esos datos a un archivo Excel.

Explicación:

1.- Leer la Variable de Entorno:

- 1.1.-** Intenta leer la cadena de conexión DB_CONNECTION_STRING desde las variables de entorno del sistema.
- 1.2.-** Si la variable de entorno no está configurada, solicita al usuario que ingrese las credenciales de la base de datos utilizando la clase DatabaseConnection.

2.- Definir la Consulta y el Archivo de Exportación:

- 2.1.-** Define la consulta SQL (SELECT * FROM producto) y la ruta del archivo Excel (productoExcel.xlsx).

3.- Crear Instancias de las Clases:

- 3.1.-** Crea instancias de DatabaseHelper y ExcelExporter utilizando la cadena de conexión obtenida.

4.- Ejecutar la Consulta y Exportar los Datos:

- 4.1.-** Utiliza DatabaseHelper para ejecutar la consulta SQL y obtener los datos en un DataTable.
- 4.2.-** Si se encuentran datos, utiliza ExcelExporter para exportarlos a un archivo Excel.
- 4.3.-** Muestra un mensaje en la consola indicando si la exportación fue exitosa o si no se encontraron datos.

Razón de Uso:

- 1.- Automatización del Flujo de Trabajo:** Coordina el proceso completo de obtención de datos y exportación a Excel en un solo lugar.
- 2.- Reusabilidad y Modularidad:** Utiliza clases separadas (DatabaseConnection, DatabaseHelper, ExcelExporter) para manejar diferentes responsabilidades, lo que facilita la reutilización y el mantenimiento del código.
- 3.- Interactividad:** Permite la interacción con el usuario para obtener credenciales de la base de datos si no están configuradas como variables de entorno.

Variables de Entorno

Para mejorar la seguridad, la cadena de conexión a la base de datos se almacena en una variable de entorno llamada **DB_CONNECTION_STRING**. Esto evita que información sensible esté disponible en el código fuente.

a.- En nuestro caso hemos creado un archivo **.env**

b.- En caso que el archivo **.env** no exista, o la variable de entorno no esté configurado por otro método, el código le pedirá al usuario mediante la consola los valores de dichas variables para poder acceder a la Base de Datos.

Conceptos Clave

Separación de Responsabilidades: Cada clase tiene una responsabilidad única, lo que mejora la mantenibilidad y la claridad del código.

Seguridad: Uso de variables de entorno para manejar información sensible.

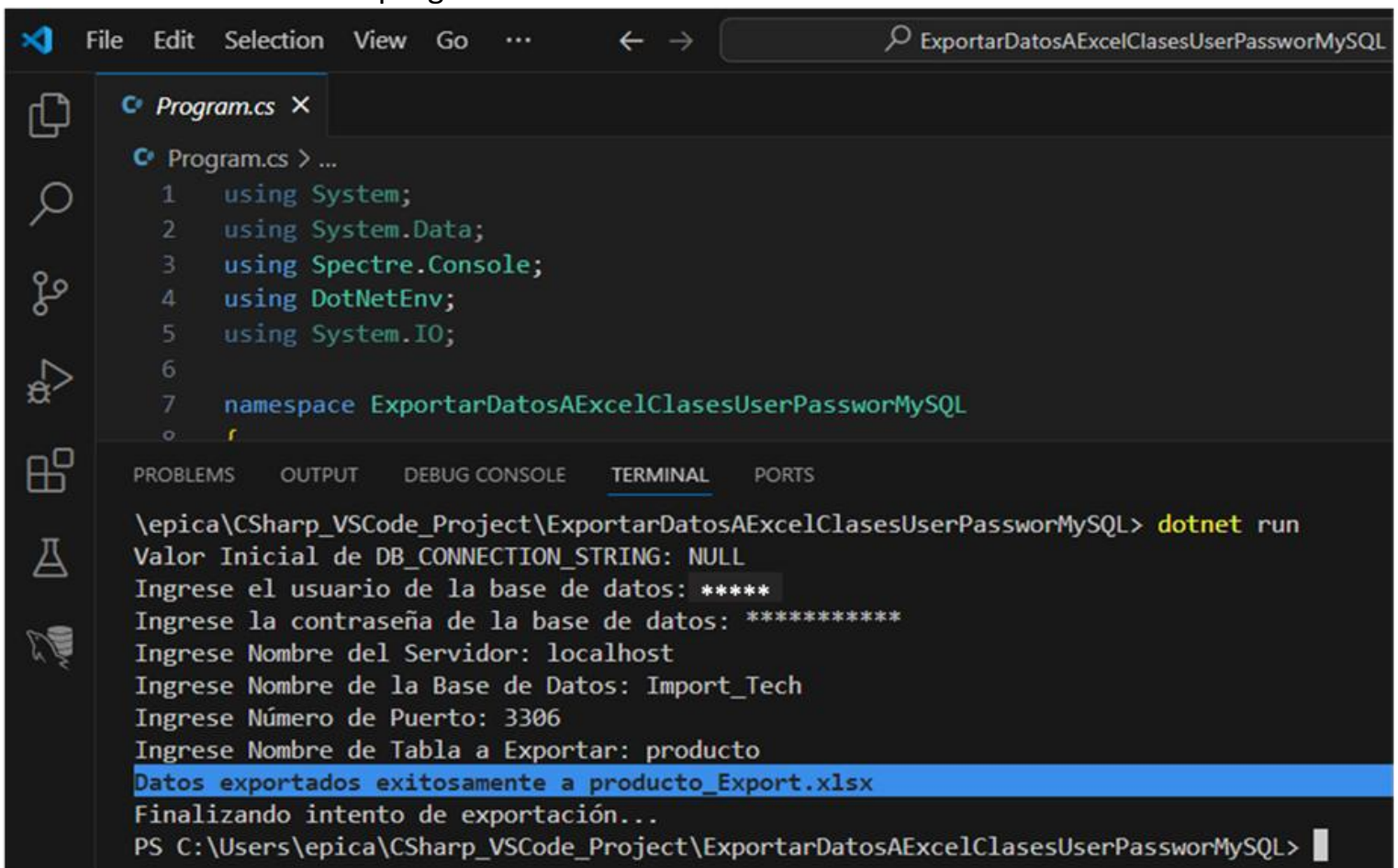
Reutilización de Código: Las clases DatabaseHelper y ExcelExporter pueden ser reutilizadas en otros proyectos que requieran funcionalidades similares.

Conclusión

Este proyecto está diseñado siguiendo buenas prácticas de programación, como la separación de responsabilidades y la seguridad de datos. La estructura modular facilita la mantenibilidad y la escalabilidad del código.

Corriendo el Proyecto en Visual Studio Code

1.- En este caso se corre el programa cuando **NO** se tiene el archivo **.env** con la variable de entorno



```
File Edit Selection View Go ... ExportarDatosAExcelClasesUserPassworMySQL
Program.cs x
Program.cs > ...
1 using System;
2 using System.Data;
3 using Spectre.Console;
4 using DotNetEnv;
5 using System.IO;
6
7 namespace ExportarDatosAExcelClasesUserPassworMySQL
8 {
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
25
```


BASE DE DATOS EXPORTADA AL ARCHIVO EXCEL PARA ESTE CASO

producto_Export.xlsx - Microsoft Excel

ID_Producto	Nombre_Producto	Descripcion_Producto	PrecioCompra_Produ	PrecioVenta_Produ	Descuento_Pro	Stock	ID_Categoria	NIP_Proveedor
1	1 Televisor Samsung 55"	Televisor LED 4K UHD	500	700	10	50	1	1
2	2 Televisor LG 65"	Televisor OLED 4K UHD	1000	1300	10	30	1	2
3	3 Televisor Sony 75"	Televisor LED 4K UHD	1200	1500	10	20	1	3
4	4 Televisor Panasonic 50"	Televisor LED Full HD	400	600	10	40	1	4
5	5 Televisor Philips 43"	Televisor LED Full HD	300	500	10	60	1	5
6	6 Laptop Dell XPS 13	Laptop con procesador Intel i7	800	1000	5	25	2	6
7	7 Laptop HP Spectre x360	Laptop convertible con procesador Intel i5	900	1100	5	20	2	7
8	8 Laptop MacBook Pro	Laptop con procesador Apple M1	1200	1500	5	15	2	8
9	9 Laptop Lenovo ThinkPad	Laptop con procesador Intel i7	700	900	5	30	2	9
10	10 Laptop Asus ZenBook	Laptop con procesador AMD Ryzen 7	750	950	5	20	2	10
11	11 iPhone 13	Teléfono móvil con pantalla OLED	800	1000	15	50	3	11
12	12 Samsung Galaxy S21	Teléfono móvil con pantalla AMOLED	700	900	15	40	3	12
13	13 Google Pixel 6	Teléfono móvil con pantalla OLED	600	800	15	30	3	13
14	14 OnePlus 9	Teléfono móvil con pantalla AMOLED	500	700	15	35	3	14
15	15 Xiaomi Mi 11	Teléfono móvil con pantalla AMOLED	450	650	15	45	3	15
16	16 iPad Pro	Tableta con pantalla Liquid Retina	900	1100	5	25	4	1
17	17 Samsung Galaxy Tab S7	Tableta con pantalla AMOLED	700	900	5	30	4	2
18	18 Microsoft Surface Pro 7	Tableta con pantalla PixelSense	800	1000	5	20	4	3
19	19 Lenovo Tab P11	Tableta con pantalla IPS	300	500	5	40	4	4
20	20 Amazon Fire HD 10	Tableta con pantalla Full HD	150	300	5	50	4	5
21	21 Canon EOS R5	Cámara digital sin espejo	3000	3500	12	10	5	6
22	22 Nikon Z7 II	Cámara digital sin espejo	2500	3000	12	15	5	7
23	23 Sony Alpha a7 III	Cámara digital sin espejo	2000	2500	12	20	5	8
24	24 Fujifilm X-T4	Cámara digital sin espejo	1500	2000	12	25	5	9

2.- En este caso se corre el programa cuando **SÍ** se tiene el archivo **.env** con la variable de entorno

```

File Edit Selection View Go ... ExportarDatosAExcelClasesUserPassworMySQL

Program.cs X
Program.cs > ...
1 using System;
2 using System.Data;
3 using Spectre.Console;
4 using DotNetEnv;
5 using System.IO;
6
7 namespace ExportarDatosAExcelClasesUserPassworMySQL
8 {
9
10 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS C:\Users\epica\CSharp_VSCode_Project\ExportarDatosAExcelClasesUserPassworMySQL> dotnet run
Valor Inicial de DB_CONNECTION_STRING: Server=localhost;Database=Import_Tech;Port=3306;UserId= ... ;Password= ***** ;
Ingrese Nombre de Tabla a Exportar: Proveedor
Datos exportados exitosamente a Proveedor_Export.xlsx
Finalizando intento de exportación...
PS C:\Users\epica\CSharp_VSCode_Project\ExportarDatosAExcelClasesUserPassworMySQL>

```

BASE DE DATOS EXPORTADA AL ARCHIVO EXCEL PARA ESTE CASO

Proveedor_Export.xlsx - Microsoft Excel

<

Rafael J. Rivas R. (epica3001.csharp@gmail.com)