

Enunciado:

Tema: Bases de datos y ADO.

Nuestro objetivo será migrar la aplicación de biblioteca para que en lugar de funcionar con una base de datos **Access** lo haga con una del tipo **Sql Server** y de esta manera demostrar que el impacto sobre el código es mínimo.

Para esto copiaremos los archivos de la aplicación del directorio **“03-Biblioteca”** al **“05-BibliotecaSQLServer”**, luego solo debemos modificar los persistidores (DAL data ccess layer), estos se encuentran en los archivos **“LibroPersistidor.cs”** y **“LectorPersistidor.cs”** tal como se muestra en la **Figura 1** y **Figura 2**, el resto de la aplicación debería seguir funcionando sin problemas.

La **Figura 3**, muestra la comparación de las cadenas de conexión a la base de datos o connectionstring.

Figura 1

Figure 1 shows a side-by-side comparison of the `LibroPersistidor.cs` file. The left pane shows the code for a SQL Server connection, and the right pane shows the code for an OleDb connection. The code is identical except for the database connection and data adapter classes.

```

// Left Pane (SQL Server)
public List<Libro> GetAll()
{
    List<Libro> libros = new List<Libro>();
    string connectionString = ConfigurationManager.ConnectionStrings["cnString"];
    using (SqlConnection cnx = new SqlConnection(connectionString))
    {
        cnx.Open();

        const string sqlQuery = "SELECT * FROM Libro ORDER BY id_libro ASC";
        using (SqlCommand cmd = new SqlCommand(sqlQuery, cnx))
        {
            DataTable table = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(cmd);
            adapter.Fill(table);
            foreach (DataRow row in table.Rows)
            {
                Libro libro = new Libro
                {
                    Id = Convert.ToInt32(row["id_libro"]),
                    CodigoIdentificacionUnico = Convert.ToString(row["codigo_id"]),
                    Titulo = Convert.ToString(row["titulo"]),
                    ISBN = Convert.ToString(row["isbn"]),
                    IdLector = (row["id_lector"] == DBNull.Value ? null : (int)
                );
                libros.Add(libro);
            }
        }
    }
}

// Right Pane (OleDb)
public List<Libro> GetAll()
{
    List<Libro> libros = new List<Libro>();
    string connectionString = ConfigurationManager.ConnectionStrings["cnString"];
    using (OleDbConnection cnx = new OleDbConnection(connectionString))
    {
        cnx.Open();

        const string sqlQuery = "SELECT * FROM Libro ORDER BY id_libro ASC";
        using (OleDbCommand cmd = new OleDbCommand(sqlQuery, cnx))
        {
            DataTable table = new DataTable();
            OleDbDataAdapter adapter = new OleDbDataAdapter(cmd);
            adapter.Fill(table);
            foreach (DataRow row in table.Rows)
            {
                Libro libro = new Libro
                {
                    Id = Convert.ToInt32(row["id_libro"]),
                    CodigoIdentificacionUnico = Convert.ToString(row["codigo_id"]),
                    Titulo = Convert.ToString(row["titulo"]),
                    ISBN = Convert.ToString(row["isbn"]),
                    IdLector = (row["id_lector"] == DBNull.Value ? null : (int)
                );
                libros.Add(libro);
            }
        }
    }
}

```

Figura 2

Figure 2 shows a side-by-side comparison of the `LibroPersistidor.cs` file, focusing on the `using` statements. The left pane shows the code for a SQL Server connection, and the right pane shows the code for an OleDb connection.

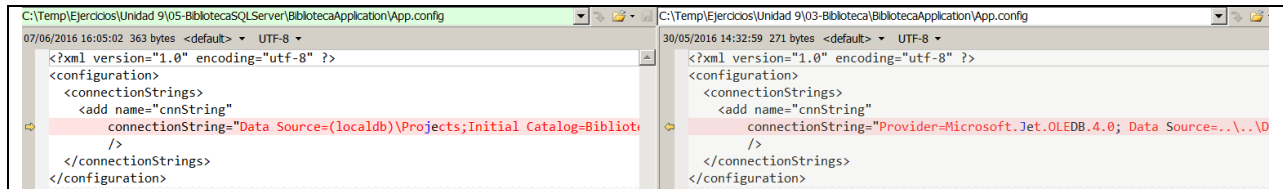
```

// Left Pane (SQL Server)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

// Right Pane (OleDb)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Configuration;
using System.Data;
using System.Data.OleDb;

```

Figura 3



La **Figura 4** muestra las diferencias a nivel de código fuente de la aplicación, existentes entre el manejo de registros de las bases de datos Sql Server y Access, cambiando unos por otros la aplicación puede utilizar una u otra indistintamente y seguir funcionando con éxito sin ser necesaria otra modificación.

Figura 4	
Sql Server	Access
"Data Source=(localdb)\Projects;Initial Catalog=BibliotecaDB;Integrated Security=True;Connect Timeout=15;Encrypt=False;TrustServerCertificate=False"	"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=..\..\Database\BibliotecaDB.mdb"
using System.Data.SqlClient;	using System.Data.OleDb;
SqlConnection	OleDbConnection
SqlCommand	OleDbCommand
SqlDataAdapter	OleDbDataAdapter
SqlDataReader	OleDbDataReader