

```
using Entities;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AccesoDatos
{
    public class DataPlanes : DataConnection
    {
        public DataTable GetPlanes(Planes planes)
        {
            string query = @"exec sp_cargar_planes";

            SqlCommand cmd = new SqlCommand(query, conexion);
            DataTable dt = new DataTable();
            SqlDataAdapter da = new SqlDataAdapter();

            try
            {
                conexion.Open();
                cmd.ExecuteNonQuery();
                da.SelectCommand = cmd;
                da.Fill(dt);
            }
            catch (Exception e)
            {
                throw new Exception("Error al listar tipos de documentos", e);
            }
            finally
            {
                conexion.Close();
                cmd.Dispose();
            }
            return dt;
        }
        public DataTable GetPlanesParaAsistencia(Planes planes, string diaDeLaSemana)
        {
            string query = @"select Planes.Plan_ID, Planes.Nombre, Jornadas_Planes.Dia
                            from Planes
                            inner join Jornadas_Planes
                            on Jornadas_Planes.Jornada_Plan_ID = Planes.Plan_ID
                            where Planes.Estado = @Estado
                            and Jornadas_Planes.Dia = @Todos
                            or Jornadas_Planes.Dia = @diaDeLaSemana";

            SqlParameter estado = new SqlParameter("@Estado", "A");
            SqlParameter todos = new SqlParameter("@Todos", "Todos");
            SqlParameter DiaDeLaSemana = new SqlParameter("@diaDeLaSemana", diaDeLaSemana);

            SqlCommand cmd = new SqlCommand(query, conexion);
            cmd.Parameters.Add(estado);
            cmd.Parameters.Add(todos);
            cmd.Parameters.Add(DiaDeLaSemana);

            DataTable dt = new DataTable();
            SqlDataAdapter da = new SqlDataAdapter();
```

```

        try
        {
            conexion.Open();
            cmd.ExecuteNonQuery();
            da.SelectCommand = cmd;
            da.Fill(dt);
        }
        catch (Exception e)
        {
            throw new Exception("Error al listar tipos de documentos", e);
        }
        finally
        {
            conexion.Close();
            cmd.Dispose();
        }
        return dt;
    }
    public int EliminarPlan(Planes planes)
    {
        int resultado = -1;

        string query = @"update Planes set Estado = @Estado where Plan_ID = @Plan_ID;
                        update Planes_Asignados set Estado = @Estado where Plan_ID =
@Plan_ID;
                        update Jornadas_Planes set Estado = @Estado where Plan_ID =
@Plan_ID"
        ;

        SqlParameter estado = new SqlParameter("@Estado", "I");
        SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);

        SqlCommand cmd = new SqlCommand(query, conexion);

        cmd.Parameters.Add(estado);
        cmd.Parameters.Add(plan_ID);

        try
        {
            OpenConnection();
            resultado = cmd.ExecuteNonQuery();
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            CloseConnection();
            cmd.Dispose();
        }
        return resultado;
    }
    public DataTable GetPlanesConFecha(Planes planes, string diaDeLaSemana, string fecha)
    {
        string query = @"
        SqlParameter estado = new SqlParameter("@Estado", "A");
        SqlParameter todos = new SqlParameter("@Todos", fecha);
        SqlParameter DiaDeLaSemana = new SqlParameter("@diaDeLaSemana", diaDeLaSemana);
    
```

```
SqlCommand cmd = new SqlCommand(query, conexion);
cmd.Parameters.Add(estado);
cmd.Parameters.Add(todos);
cmd.Parameters.Add(DiaDeLaSemana);

DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter();

try
{
    conexion.Open();
    cmd.ExecuteNonQuery();
    da.SelectCommand = cmd;
    da.Fill(dt);
}
catch (Exception e)
{
    throw new Exception("Error al listar alumnos", e);
}
finally
{
    conexion.Close();
    cmd.Dispose();
}
return dt;
}

public DataTable GetAlumnoPresentes(Planes planes, DateTime fechaPresente)
{
    string query = @"exec sp_Cargar_Presentes @Plan_ID, @Fecha";

    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);
    SqlParameter fecha = new SqlParameter("@Fecha", fechaPresente);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);
    cmd.Parameters.Add(fecha);

    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter();

    try
    {
        conexion.Open();
        cmd.ExecuteNonQuery();
        da.SelectCommand = cmd;
        da.Fill(dt);
    }
    catch (Exception e)
    {
        throw new Exception("Error al listar alumnos presentes", e);
    }
    finally
    {
        conexion.Close();
        cmd.Dispose();
    }
    return dt;
}

public DataSet GetPlanesActuales(string buscar)
```

```
{
    string query;

    if (string.IsNullOrEmpty(buscar))
    {
        query = "exec sp_cargar_planes_Actuales";
    }
    else
    {
        query = @"exec sp_cargar_Plan_unico @Estado, @Profesor, @Parametro";
    }

    SqlParameter estado = new SqlParameter("@Estado", "A");
    SqlParameter profesor = new SqlParameter("@Profesor", "Profesor");
    //SqlParameter profesor = new SqlParameter("@Profesor", "Profesor");

    SqlCommand cmd = new SqlCommand(query, conexion)
    {
        CommandType = CommandType.Text
    };

    cmd.Parameters.Add(estado);
    cmd.Parameters.Add(profesor);

    cmd.Parameters.Add(new SqlParameter()
    {
        ParameterName = "@Parametro",
        SqlDbType = SqlDbType.NVarChar,
        Value = string.Format("%{0}%", buscar)
    });

    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter();

    try
    {
        conexion.Open();
        cmd.ExecuteNonQuery();
        da.SelectCommand = cmd;
        da.Fill(ds);
    }
    catch (Exception e)
    {
        throw new Exception("Error al listar planes", e);
    }
    finally
    {
        conexion.Close();
        cmd.Dispose();
    }
    return ds;
}

public DataTable GetAlumnoPorClase(Planes planes, DateTime fechaPresente)
{
    string query = "exec sp_Cargar_Ausentes @Plan_ID, @Fecha";

    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);
    SqlParameter fecha = new SqlParameter("@Fecha", fechaPresente);

    SqlCommand cmd = new SqlCommand(query, conexion);
```

```
cmd.Parameters.Add(plan_ID);
cmd.Parameters.Add(fecha);

DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter();

try
{
    conexion.Open();
    cmd.ExecuteNonQuery();
    da.SelectCommand = cmd;
    da.Fill(dt);
}
catch (Exception e)
{
    throw new Exception("Error al listar tipos de documentos", e);
}
finally
{
    conexion.Close();
    cmd.Dispose();
}
return dt;
}

public Planes GetDatoPlan(Planes planes)
{
    string query = @"select Plan_ID, Importe_Plan,
                        Cupo_Total,
                        Cupo_Restante
                    from Planes
                    where Estado = @Estado
                    and Plan_ID = @Plan_ID";

    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);
    SqlParameter estado = new SqlParameter("@Estado", "A");

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);
    cmd.Parameters.Add(estado);

    try
    {
        OpenConnection();
        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            planes.Plan_ID = int.Parse(reader["Plan_ID"].ToString());
            planes.Importe_Plan = int.Parse(reader["Importe_Plan"].ToString());
            if (string.IsNullOrEmpty(reader["Cupo_Total"].ToString()))
            {
                planes.Cupo_Total = 0;
            }
            else
            {
                planes.Cupo_Total = int.Parse(reader["Cupo_Total"].ToString());
            }
            if (string.IsNullOrEmpty(reader["Cupo_Restante"].ToString()))
            {
                planes.Cupo_Restante = 0;
            }
        }
    }
}
```

```

        }
        else
        {
            planes.Cupo_Restante = int.Parse(reader["Cupo_Restante"].ToString());
        }
    }
    reader.Close();
    cmd.ExecuteReader();
}
catch (Exception)
{
    throw;
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return planes;
}
public int RegistrarNuevoPlan(Planes planes)
{
    int resultado = -1;
    string query = @"insert into Planes (Nombre,
                                         Importe_Plan,
                                         Duracion,
                                         Fecha_Inicio,
                                         Fecha_Alta_Plan,
                                         Cupo_Total,
                                         Estado,
                                         Persona_ID,
                                         Empleado_ID,
                                         Cupo_Restante)
                    values (@Nombre,
                            @Importe_Plan,
                            @Duracion,
                            @Fecha_Inicio,
                            @Fecha_Alta_Plan,
                            @Cupo_Total,
                            @Estado,
                            @Persona_ID,
                            @Empleado_ID,
                            @Cupo_Restante)"
    ;

    SqlParameter nombre = new SqlParameter("@Nombre", planes.Nombre);
    SqlParameter importe_Plan = new SqlParameter("@Importe_Plan",
planes.Importe_Plan);
    SqlParameter duracion = new SqlParameter("@Duracion", planes.Duracion);
    SqlParameter fecha_Inicio = new SqlParameter("@Fecha_Inicio",
planes.Fecha_Inicio);
    SqlParameter fecha_Alta_Plan = new SqlParameter("@Fecha_Alta_Plan",
planes.Fecha_Alta_Plan);
    SqlParameter cupo_Total = new SqlParameter("@Cupo_Total", planes.Cupo_Total);
    SqlParameter estado = new SqlParameter("@Estado", planes.Estado);
    SqlParameter persona_ID = new SqlParameter("@Persona_ID", planes.Persona_ID);
    SqlParameter empleado_ID = new SqlParameter("@Empleado_ID", planes.Empleado_ID);
    SqlParameter cupo_Restante = new SqlParameter("@Cupo_Restante",
planes.Cupo_Restante);

    SqlCommand cmd = new SqlCommand(query, conexion);
    cmd.Parameters.Add(nombre);

```

```

cmd.Parameters.Add(importe_Plan);
cmd.Parameters.Add(duracion);
cmd.Parameters.Add(fecha_Inicio);
cmd.Parameters.Add(fecha_Alta_Plan);
cmd.Parameters.Add(cupo_Total);
cmd.Parameters.Add(estado);
cmd.Parameters.Add(persona_ID);
cmd.Parameters.Add(empleado_ID);
cmd.Parameters.Add(cupo_Restante);

try
{
    OpenConnection();
    resultado = cmd.ExecuteNonQuery();
}
catch (Exception)
{
    throw;
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return resultado;
}

public int EditarCupoRestante(Planes planes)
{
    int resultado = -1;
    string query = @"if exists (select Cupo_Restante from Planes where Cupo_Restante >
0 and Plan_ID = @Plan_ID)
        begin
            update Planes set Cupo_Restante = Cupo_Restante - 1
            where Plan_ID = @Plan_ID
        end"
    ;

    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);

    try
    {
        OpenConnection();
        resultado = cmd.ExecuteNonQuery();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return resultado;
}

public Planes GetLastID(Planes planes)
{

```

```
string query = @"select Plan_ID from Planes where Plan_ID = (select max(Plan_ID)
from Planes)";

SqlCommand cmd = new SqlCommand(query, conexion);

try
{
    OpenConnection();
    SqlDataReader reader = cmd.ExecuteReader();

    if (reader.Read())
    {
        planes.Plan_ID = int.Parse(reader["Plan_ID"].ToString());
    }
    reader.Close();
    cmd.ExecuteReader();
}
catch (Exception)
{
    throw;
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return planes;
}

public Planes GetPlanUnico(Planes planes)
{
    string query = @"select * from Planes where Plan_ID = @Plan_ID";

    SqlParameter plan_Id = new SqlParameter("@Plan_ID", planes.Plan_ID);
    SqlCommand cmd = new SqlCommand(query, conexion);
    cmd.Parameters.Add(plan_Id);

    try
    {
        OpenConnection();
        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            planes.Plan_ID = int.Parse(reader["Plan_ID"].ToString());
            planes.Nombre = reader["Nombre"].ToString();
            planes.Importe_Plan = decimal.Parse(reader["Importe_Plan"].ToString());
            planes.Duracion = int.Parse(reader["Duracion"].ToString());
            planes.Cupo_Total = int.Parse(reader["Cupo_Total"].ToString());
            planes.Fecha_Inicio = Convert.ToDateTime(reader["Fecha_Inicio"]);
            planes.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
        }
        reader.Close();
        cmd.ExecuteReader();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {

```



```

        CloseConnection();
        cmd.Dispose();
    }
    return planes;
}

public int EditarPlan(Planes planes)
{
    int resultado = -1;
    string query = @"update Planes set Nombre = @Nombre,
                                Importe_Plan = @Importe_Plan,
                                Duracion = @Duracion,
                                Cupo_Total = @Cupo_Total,
                                Fecha_Inicio = @Fecha_Inicio,
                                Empleado_ID = @Empleado_ID
                                where Plan_ID = @Plan_ID"
    ;

    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);
    SqlParameter nombre = new SqlParameter("@Nombre", planes.Nombre);
    SqlParameter importe_Plan = new SqlParameter("@Importe_Plan",
planes.Importe_Plan);
    SqlParameter duracion = new SqlParameter("@Duracion", planes.Duracion);
    SqlParameter cupo_Total = new SqlParameter("@Cupo_Total", planes.Cupo_Total);
    SqlParameter fecha_Inicio = new SqlParameter("@Fecha_Inicio",
planes.Fecha_Inicio);
    SqlParameter empleado_ID = new SqlParameter("@Empleado_ID", planes.Empleado_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);
    cmd.Parameters.Add(nombre);
    cmd.Parameters.Add(importe_Plan);
    cmd.Parameters.Add(duracion);
    cmd.Parameters.Add(cupo_Total);
    cmd.Parameters.Add(fecha_Inicio);
    cmd.Parameters.Add(empleado_ID);

    try
    {
        OpenConnection();
        resultado = cmd.ExecuteNonQuery();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return resultado;
}

public DataTable GetPlanesParaPago(int Cliente_ID)
{
    string query = @"exec sp_Buscar_Planes_Asignados_Cliente @Estado, @Cliente_ID";

    SqlParameter estado = new SqlParameter("@Estado", "A");
    SqlParameter cliente_ID = new SqlParameter("@Cliente_ID", Cliente_ID);

```

```
SqlCommand cmd = new SqlCommand(query, conexion);

cmd.Parameters.Add(estado);
cmd.Parameters.Add(cliente_ID);

DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter();

try
{
    OpenConnection();
    cmd.ExecuteNonQuery();
    da.SelectCommand = cmd;
    da.Fill(dt);
}
catch (Exception e)
{
    throw new Exception("Error al listar Planes", e);
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return dt;
}

public Planes GetCostoPlan(Planes planes)
{
    string query = "select Importe_Plan from Planes where Plan_ID = @Plan_ID";
    SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes.Plan_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);
    cmd.Parameters.Add(plan_ID);

    try
    {
        OpenConnection();
        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            planes.Importe_Plan = decimal.Parse(reader["Importe_Plan"].ToString());
        }
        reader.Close();
        cmd.ExecuteReader();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return planes;
}
}
```

