

```

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 DataPlanesAsignados : DataConnection
    {
        public DataTable VerClasesQueTieneElCliente(Planes_Asignados planes_Asignados)
        {
            string query = @"select Planes_Asignados.Plan_Asignado_ID, Planes.Nombre
                              from Planes_Asignados
                              inner join Planes
                              on Planes_Asignados.Plan_ID = Planes.Plan_ID
                              where Planes_Asignados.Cliente_ID = @Cliente_ID"

            ;

            SqlParameter cliente_ID = new SqlParameter("@Cliente_ID",
planes_Asignados.Cliente_ID);
            SqlCommand cmd = new SqlCommand(query, conexion);
            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)
            {
                throw;
            }
            finally
            {
                CloseConnection();
            }
            return dt;
        }

        public int AsginarPlanAlCliente(Planes_Asignados planes_Asignados)
        {
            int resultado = -1;

            string query = @"insert into Planes_Asignados
                              (Plan_ID, Cliente_ID, Empleado_ID, Fecha_Inscripcion, Estado)
                              values (@Plan_ID, @Cliente_ID, @Empleado_ID, @Fecha_Inscripcion,
@Estado)"

            ;

            SqlParameter plan_ID = new SqlParameter("@Plan_ID", planes_Asignados.Plan_ID);
            SqlParameter cliente_ID = new SqlParameter("@Cliente_ID",
planes_Asignados.Cliente_ID);
            SqlParameter empleado_ID = new SqlParameter("@Empleado_ID",

```

```
planes_Asignados.Empleado_ID);
    SqlParameter fecha_Inscripcion = new SqlParameter("@Fecha_Inscripcion",
planes_Asignados.Fecha_Inscripcion);
    SqlParameter estado = new SqlParameter("@Estado", planes_Asignados.Estado);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);
    cmd.Parameters.Add(cliente_ID);
    cmd.Parameters.Add(Empleado_ID);
    cmd.Parameters.Add(fecha_Inscripcion);
    cmd.Parameters.Add(estado);

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

public int EliminarAsignacion(Planes_Asignados planes_Asignados, string NombrePlan)
{
    int resultado = -1;
    string query = @"update Planes_Asignados set Estado = @Estado
                    inner join Planes
                    on Planes_Asignados.Plan_ID = Planes.Plan_ID
                    where Planes.Nombre = @NombrePlan"
    ;

    SqlParameter estado = new SqlParameter("@Estado", planes_Asignados.Estado);
    SqlParameter nombrePlan = new SqlParameter("@NombrePlan", NombrePlan);

    SqlCommand cmd = new SqlCommand(query, conexion);
    cmd.Parameters.Add(estado);
    cmd.Parameters.Add(nombrePlan);
    try
    {
        OpenConnection();
        resultado = cmd.ExecuteNonQuery();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return resultado;
}

public DataSet BuscarPlanesAsignados(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);

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

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

public bool BuscarDuplicidad(int idplan, int Cliente_ID)
{
    bool duplicidad = false;
    string query = @"select Estado from Planes_Asignados where Plan_ID = @Plan_ID and
Cliente_ID = @Cliente_ID and Estado = @Estado";

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

    SqlCommand cmd = new SqlCommand(query, conexion);

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

    try
    {
        OpenConnection();
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            //Si el importe aparece, significa que la ultima caja abierta
            //está cerrada
            string resultado = (reader["Estado"]).ToString();
            if (resultado == "A")
            {
                duplicidad = true;
            }
        }
    }
}
```

```
        else
        {
            duplicidad = false;
        }
    }
    reader.Close();
    cmd.ExecuteNonQuery();
}
catch (Exception)
{
    throw;
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return duplicidad;
}
}
```