

```

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 DataClientes : DataConnection
    {
        public int AltaCliente(Clientes clientes)
        {
            //Como cada alta, hecha en entero, para recibir el resultado de que se hizo
            correctamente.
            int resultado = -1;
            string query = @"Insert into Clientes (Persona_ID,
                                                Estado)

                                                values (@Persona_ID,
                                                @Estado)"

            ;
            SqlParameter persona_ID = new SqlParameter("@Persona_ID", clientes.Persona_ID);
            SqlParameter Estado = new SqlParameter("@Estado", clientes.Estado);

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

            try
            {
                OpenConnection();
                resultado = cmd.ExecuteNonQuery();
            }
            catch (Exception e)
            {
                throw new Exception("Error al tratar de registrar nuevo empleado", e);
            }
            finally
            {
                CloseConnection();
                cmd.Dispose();
            }

            return resultado;
        }
        public DataSet GetClientes(string buscar)
        {
            //Esto es un filtro de búsqueda, tomando como parámetro el argumento recibido en
            el método.
            string getClientes;
            if (string.IsNullOrEmpty(buscar))
            {
                getClientes = @"sp_Cargar_Clientes_Desc"; /*es asc*/
            }
            else
            {
                getClientes = @"sp_Buscar_Listado_Clientes @query";
            }
        }
    }
}

```

```
SqlCommand cmd = new SqlCommand(getClientes, conexion)
{
    CommandType = CommandType.Text
};
cmd.Parameters.Add(new SqlParameter()
{
    ParameterName = "@query",
    SqlDbType = SqlDbType.NVarChar,
    Value = string.Format("%{0}%", buscar)
});

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 Clientes", e);
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return ds;
}

public Clientes GetCliente(Clientes clientes)
{
    /*En este método traigo los datos de un cliente particularmente
    pasando como parámetro de búsqueda el ID del cliente.*/
    string query = @"select * from Clientes where Cliente_ID = @Cliente_ID";

    SqlParameter clienteID = new SqlParameter("@Cliente_ID", clientes.Cliente_ID);

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

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

        if (reader.Read())
        {
            clientes.Cliente_ID = int.Parse(reader["Cliente_ID"].ToString());
            clientes.Persona_ID = int.Parse(reader["Persona_ID"].ToString());
            clientes.Estado = reader["Estado"].ToString();
        }
        reader.Close();
        cmd.ExecuteReader();
    }
    catch (Exception e)
    {
        throw new Exception ("No se pudo traer los datos del cliente", e);
    }
}
```

```

    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return clientes;
}
public int BajaCliente(Clientes clientes)
{
    /*
    Las bajas son solo modificaciones de las tablas, por lo que acá se realiza un
update
    para cambiar el estado de un cliente. Siempre de forma parametrizada.
    */
    int resultado = -1;
    string query = @"Update Clientes set Estado = @Estado where Cliente_ID =
@Cliente_ID";
    SqlParameter estado = new SqlParameter("@Estado", clientes.Estado);
    SqlParameter cliente_ID = new SqlParameter("@Cliente_ID", clientes.Cliente_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

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

    try
    {
        OpenConnection();
        resultado = cmd.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw new Exception("Error al tratar de editar datos del cliente", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return resultado;
}
public DataSet BuscarClienteAsistencia(string buscar)
{
    /*Acá se filtran las asistencias diarias de los clientes
    También como parámetro de búsqueda, el argumento recibido.
    */
    string query = @"sp_Buscar_Cliente_Ausentismo @buscar";
    SqlCommand cmd = new SqlCommand(query, conexion);

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

    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 Empleados", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return ds;
}

public DataSet BuscarClienteAsistenciaById(Planes_Asignados _planesAsignados)
{
    /*Acá se filtran las asistencias diarias de los clientes
    También como parámetro de búsqueda, el argumento recibido.
    */
    string query = @"exec sp_Buscar_Cliente_Por_ID @Plan_ID, @Cliente_ID"
    ;
    SqlParameter plan_ID = new SqlParameter("@Plan_ID", _planesAsignados.Plan_ID);
    SqlParameter cliente_ID = new SqlParameter("@Cliente_ID",
    _planesAsignados.Cliente_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(plan_ID);
    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 Empleados", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return ds;
}

public int EditarCliente(Clientes clientes)
{
    /*Nuevamente otra edición pero en este caso es directamente del estado únicamente.
    */
    int resultado = -1;
    string query = @"update Clientes set Estado = @Estado where Cliente_ID =
    @Cliente_ID";

```

```
SqlParameter cliente_Id = new SqlParameter("@Cliente_ID", clientes.Cliente_ID);
SqlParameter estado = new SqlParameter("@Estado", clientes.Estado);

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