

```
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 DataEmpleados : DataConnection
    {
        #region Alta de Empleados
        public int AltaEmpleado(Empleados empleados)
        {
            //como su nombre lo especifica es el alta. Totalmente igual a cualquier alta que
            //se verá en este proyecto.
            //Siempre con la parametrización correspondiente a la query para la bdd.
            int resultado = -1;
            string query = @"Insert into Empleados (Persona_ID,
                                                    Usuario,
                                                    Clave,
                                                    Tipo_Empleado_ID,
                                                    Estado_Empleado_ID)

                                                    values (@Persona_ID,
                                                            @Usuario,
                                                            @Clave,
                                                            @Tipo_Empleado_ID,
                                                            @Estado_Empleado_ID)"

            ;

            SqlParameter persona_ID = new SqlParameter("@Persona_ID", empleados.Persona_ID);
            SqlParameter usuario = new SqlParameter("@Usuario", empleados.Usuario);
            SqlParameter clave = new SqlParameter("@Clave", empleados.Clave);
            SqlParameter tipo_Empleado_ID = new SqlParameter("@Tipo_Empleado_ID",
empleados.Tipo_Empleado_ID);
            SqlParameter estado_Empleado_ID = new SqlParameter("@Estado_Empleado_ID",
empleados.Estado_Empleado_ID);

            SqlCommand cmd = new SqlCommand(query, conexion);

            cmd.Parameters.Add(persona_ID);
            cmd.Parameters.Add(usuario);
            cmd.Parameters.Add(clave);
            cmd.Parameters.Add(tipo_Empleado_ID);
            cmd.Parameters.Add(estado_Empleado_ID);

            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;
}

#endregion

#region Consultas de ID
public Empleados GetLastEmpleadoID(Empleados empleados)
{
    /*Obtendremos con este método, el último ID resgitrado de la tabla de empleados*/
    string query = @"sp_cargar_ultimo_ID";
    SqlCommand cmd = new SqlCommand(query, conexion);

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

        if (reader.Read())
        {
            empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
        }
        reader.Close();
    }
    catch (Exception e)
    {
        throw new Exception("No se pudo concretar la solicitud requerida", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }

    return empleados;
}

public Empleados GetLastID(int personaJor, Empleados empleados)
{
    /* Realiza una búsqueda similar a la anterior, solo que acá vamos a pasar
    * un argumento como parámetro de búsqueda específico*/
    string query = @"sp_cargar_ultimo_ID";
    SqlCommand cmd = new SqlCommand(query, conexion);

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

        if (reader.Read())
        {
            empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
        }
        reader.Close();
    }
    catch (Exception e)
    {
        throw new Exception("No se pudo concretar la solicitud requerida", e);
    }
}
```

```
        finally
        {
            CloseConnection();
            cmd.Dispose();
        }

        return empleados;
    }

    public DataTable GetEstadosEmpleados(Tipos_Empleados tipos_Empleados)
    {
        /*
        Los empleados tienen distintos tipos de Estados. ACTivo, baja, vacaciones, etc
        Esos son los estados que traemos para que el usuario pueda utilizarlos.
        */
        string query = @"select * from Estados_Empleados";
        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 Movimientos", e);
        }
        finally
        {
            CloseConnection();
            cmd.Dispose();
        }
        return dt;
    }

    public Empleados GetEmleadoID(Empleados empleados)
    {
        /*Acá vamos a solicitar el id de un empleado específico, ayudado con otro campo de
        la misma tabla.*/
        string query = @"select Empleado_ID from Empleados
                        where Persona_ID = @Persona_ID"
        ;
        SqlParameter personaId = new SqlParameter("@Persona_ID", empleados.Persona_ID);

        SqlCommand cmd = new SqlCommand(query, conexion);

        cmd.Parameters.Add(personaId);

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

            if (reader.Read())
            {
                empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
            }
            reader.Close();
            cmd.ExecuteNonQuery();
        }
    }
}
```

```

    }
    catch (Exception e)
    {
        throw new Exception("No se pudo realizar el cálculo requerido", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }

    return empleados;
}

#endregion

#region Traer Empleados Registrados en la bdd
public DataSet GetEmpleados(string buscar)
{
    //Solicitamos la lista de TODOS los empleados
    string getBusqueda;

    if (string.IsNullOrEmpty(buscar))
    {
        getBusqueda = @"sp_Cargar_Empleados_Desc";
    }
    else
    {
        getBusqueda = @"SELECT Personas.Persona_ID, Personas.Nombre,
Personas.Apellido, Personas.Nro_Documento,
                        Tipos_Empleados.Tipo,
                        Estados_Empleados.Estado
FROM Personas

INNER JOIN Empleados
ON Empleados.Persona_ID = Personas.Persona_ID
and Empleados.Tipo_Empleado_ID != 0
and Empleados.Tipo_Empleado_ID != 1

INNER JOIN Tipos_Empleados
ON Tipos_Empleados.Tipo_Empleado_ID =
Empleados.Tipo_Empleado_ID

INNER JOIN Estados_Empleados
ON Estados_Empleados.Estado_Empleado_ID =
Empleados.Estado_Empleado_ID

where Personas.Nombre LIKE @query
   or Personas.Apellido LIKE @query
   or Personas.Nro_Documento LIKE @query
   or Tipos_Empleados.Tipo LIKE @query
   or Estados_Empleados.Estado LIKE @query
Order by Personas.Fecha_Alta desc"

        ;
    }

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

public Empleados GetTipoEmpleado(Empleados empleados)
{
    /*Acá solicitamos los datos de 1 solo empleado, dado un parámetro de búsqueda.
    */
    string query = @"select * from Empleados
                    where Persona_ID = @Persona_ID"
    ;

    SqlParameter personaID = new SqlParameter("@Persona_ID", empleados.Persona_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(personaID);

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

        if (reader.Read())
        {
            empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
            empleados.Persona_ID = int.Parse(reader["Persona_ID"].ToString());
            empleados.Usuario = (reader["Usuario"].ToString());
            empleados.Clave = (reader["Clave"].ToString());
            empleados.Tipo_Empleado_ID =
int.Parse(reader["Tipo_Empleado_ID"].ToString());
            empleados.Estado_Empleado_ID =
int.Parse(reader["Estado_Empleado_ID"].ToString());
        }
        reader.Close();
        cmd.ExecuteNonQuery();
    }
    catch (Exception e)
    {

```

```

        throw new Exception("No se pudo traer los datos requeridos", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }

    return empleados;
}
public int EditarEmpleado(Empleados empleados)
{
    //Edición de datos como usuario y clave del empleado.
    int resultado = -1;
    string orden = @"update Empleados set Usuario = @Usuario,
                                                Clave = @Clave,
                                                Tipo_Empleado_ID = @Tipo_Empleado_ID
Where Persona_ID = @Persona_ID and Empleado_ID
= @Empleado_ID"
        ;

    SqlParameter empleado_ID = new SqlParameter("@Empleado_ID",
empleados.Empleado_ID);
    SqlParameter persona_ID = new SqlParameter("@Persona_ID", empleados.Persona_ID);
    SqlParameter usuario = new SqlParameter("@Usuario", empleados.Usuario);
    SqlParameter clave = new SqlParameter("@Clave", empleados.Clave);
    SqlParameter tipo_Empleado_ID = new SqlParameter("@Tipo_Empleado_ID",
empleados.Tipo_Empleado_ID);

    SqlCommand cmd = new SqlCommand(orden, conexion);

    cmd.Parameters.Add(empleado_ID);
    cmd.Parameters.Add(persona_ID);
    cmd.Parameters.Add(usuario);
    cmd.Parameters.Add(clave);
    cmd.Parameters.Add(tipo_Empleado_ID);

    try
    {
        OpenConnection();
        resultado = cmd.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw new Exception("Error al tratar de editar datos de empleado", e);
    }
    finally
    {
        CloseConnection();
        cmd.Dispose();
    }
    return resultado;
}
public Empleados GetLastEmpleado(int PersonaId, Empleados _empleados)
{
    /*Acá vamos a buscar el ID del último empleado dado un parámetro x.*/
    string query = @"select Empleado_ID from Empleados where Persona_ID =
@Persona_ID";

    SqlParameter personaId = new SqlParameter("@Persona_ID", PersonaId);

    SqlCommand cmd = new SqlCommand(query, conexion);

```

```
cmd.Parameters.Add(personaId);

try
{
    OpenConnection();
    SqlDataReader reader = cmd.ExecuteReader();
    if (reader.Read())
    {
        _empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
    }
    reader.Close();
    cmd.ExecuteNonQuery();
}
catch (Exception e)
{
    throw new Exception("No se ha podido traer el último ID de empleado.", e);
}
finally
{
    CloseConnection();
    cmd.Dispose();
}
return _empleados;
}

#endregion

#region Sesión de Usuarios

public bool ConsultarRegistrosLogin(bool primerLogueo)
{
    /*Acá hacemos un método booleano que nos va a decir si es la primera vez que se
loguean o no*/
    string query = @"select count(distinct Empleado_ID) as Cuenta from Empleados";
    SqlCommand cmd = new SqlCommand(query, conexion);

    try
    {
        OpenConnection();
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            if (int.Parse(reader["Cuenta"].ToString()) > 0 )
            {
                primerLogueo = false;
            }
            else
            {
                primerLogueo = true;
            }
        }
        reader.Close();
        cmd.ExecuteNonQuery();
    }
    catch (Exception)
    {
        throw;
    }
    finally

```

```

        {
            CloseConnection();
            cmd.Dispose();
        }
        return primerLogueo;
    }
    public Empleados VerificarClaveEnBdd(Tipos_Empleados _tiposEmpleados, Empleados
    _empleados)
    {
        /*Datos los parámetros de búsqueda, usuario y clave, vamos a verificar si existen
        en la bdd.*/
        string query = @"select Empleados.Empleado_ID,Empleados.Persona_ID,
Empleados.Usuario, Empleados.Clave,
                                Tipos_Empleados.Tipo, Tipos_Empleados.Acceso_Clave,
                                Tipos_Empleados.Estado
                                from Empleados
                                inner join Tipos_Empleados
                                on Empleados.Tipo_Empleado_ID = Tipos_Empleados.Tipo_Empleado_ID
                                where Empleados.Usuario = @Usuario and Empleados.Clave = @Clave"

        ;

        SqlParameter usuario = new SqlParameter("@Usuario", _empleados.Usuario);
        SqlParameter clave = new SqlParameter("@Clave", _empleados.Clave);

        SqlCommand cmd = new SqlCommand(query, conexion);

        cmd.Parameters.Add(usuario);
        cmd.Parameters.Add(clave);

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

            if (reader.Read())
            {
                _tiposEmpleados.Estado = Convert.ToString(reader["Estado"]);
                _tiposEmpleados.Acceso_Clave = Convert.ToString(reader["Acceso_Clave"]);
                _tiposEmpleados.Tipo = Convert.ToString(reader["Tipo"]);
                _empleados.Clave = Convert.ToString(reader["Clave"]);
                _empleados.Usuario = Convert.ToString(reader["Usuario"]);
                _empleados.Persona_ID = int.Parse(reader["Persona_ID"].ToString());
                _empleados.Empleado_ID = int.Parse(reader["Empleado_ID"].ToString());
            }
        }
        catch (Exception e)
        {
            throw new Exception("Error al tratar de verificar datos", e);
        }
        finally
        {
            CloseConnection();
            cmd.Dispose();
        }
        return _empleados;
    }
}

#endregion

#region Edición de clave y usuario

```



```
public int EditarClave(Empleados empleados)
{
    //Edición de clave y usuario
    int resultado = -1;
    string query = @"update Empleados set Usuario = @Usuario,
                        Clave = @Clave,
                        Where Persona_ID = @Persona_ID"
    ;

    SqlParameter usuario = new SqlParameter("@Usuario", empleados.Usuario);
    SqlParameter clave = new SqlParameter("@Clave", empleados.Clave);
    SqlParameter persona_ID = new SqlParameter("@Persona_ID", empleados.Persona_ID);

    SqlCommand cmd = new SqlCommand(query, conexion);

    cmd.Parameters.Add(usuario);
    cmd.Parameters.Add(clave);
    cmd.Parameters.Add(persona_ID);

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

    return resultado;
}

#endregion
}
```