

```
1 using Entities;
2 using System;
3 using System.Data;
4 using System.Data.SqlClient;
5
6 namespace AccesoDatos
7 {
8     public class DataAsistencia : DataConnection
9     {
10         public int PutAsistencia(Asistencias _asistencias)
11         {
12             //A lo largo de todas las clases de este proyecto, se van a observar 4 tipos
13             //distintos de métodos
14             //que sirven para el CRUD.
15             //Este es el primero.
16             //Siempre que sea un método tipo INT, se deducirá, de ahora en adelante, que
17             //es un create o insert
18             //para la base de datos.
19             /*Se crea una variable tipo entero, con valor -1 porque necesito que me
20             devuelve un 0 al finalizar la gestión
21             Si no me devuelve un 0, es porque la gestión no se llevó a cabo.
22             */
23             int resultado = -1;
24
25             //Query que se va a ejecutar en la bdd. En esta caso SQL.
26             string query = @"insert into Asistencias (Cliente_ID, Fecha, Estado,
27 Empleado_ID, Plan_Asignado_ID)
28 values (@Cliente_ID, @Fecha, @Estado, @Empleado_ID,
29 @Plan_Asignado_ID)";
30
31             /*
32             Se utilizan parámetros por seguridad. Ya que con parametrizaciones, evitamos
33             problemas de injection.
34             */
35             SqlParameter cliente_ID = new SqlParameter("@Cliente_ID",
36 _asistencias.Cliente_ID);
37             SqlParameter fecha = new SqlParameter("@Fecha", _asistencias.Fecha);
38             SqlParameter estado = new SqlParameter("@Estado", _asistencias.Estado);
39             SqlParameter empleado_ID = new SqlParameter("@Empleado_ID",
40 _asistencias.Empleado_ID);
41             SqlParameter plan_Asignado_ID = new SqlParameter("@Plan_Asignado_ID",
42 _asistencias.Plan_Asignado_ID);
43
44             /*Creamos un objeto de tipo SQLCOMMAND que es con el que vamos a realizar la
45             inserción de la que a
46             la bdd sql.*/
47             SqlCommand cmd = new SqlCommand(query, conexion);
48             /* A este mismo comando creado, le agregamos los parámetros que creamos
49             arriba*/
50             cmd.Parameters.Add(cliente_ID);
51             cmd.Parameters.Add(fecha);
52             cmd.Parameters.Add(estado);
53             cmd.Parameters.Add(empleado_ID);
54             cmd.Parameters.Add(plan_Asignado_ID);
55
56             /*Acá encerramos toda la consulta en un try catch por cualquier error que nos
57             de*/
58             try
59             {
60                 //primero abrimos la conexión
61                 OpenConnection();
62
63                 //luego enviamos la consulta y el resultado se lo asignamos a la variable
64                 //tiene que devolver un 0, siempre y cuando todo haya salido OK.
```

```

52         resultado = cmd.ExecuteNonQuery();
53     }
54     catch (Exception)
55     {
56         throw;
57     }
58     finally
59     {
60         //Por último, tenemos este finally. Que se va a ejecutar, haya o no una
exception encontrada
61         CloseConnection();
62
63         //Y cerraremos también el commando que enviamos, para que no haya futuros
errores
64         cmd.Dispose();
65     }
66
67     //por último devolvemos el entero que necesitamos
68     return resultado;
69 }
70
71 public DataSet GetAsistenciasDiarias(string buscar)
72 {
73     string query;
74     if (string.IsNullOrEmpty(buscar))
75     {
76         query = @"sp_get_Asistencias_diarias";
77     }
78     else
79     {
80         query = @"sp_Buscar_Listado_Asistencia_Total @Parametro";
81     }
82
83     SqlCommand cmd = new SqlCommand(query, conexion)
84     {
85         CommandType = CommandType.Text
86     };
87     cmd.Parameters.Add(new SqlParameter()
88     {
89         ParameterName = "@Parametro",
90         SqlDbType = SqlDbType.NVarChar,
91         Value = string.Format("{0}%", buscar)
92     });
93
94     DataSet ds = new DataSet();
95     SqlDataAdapter da = new SqlDataAdapter();
96
97     try
98     {
99         OpenConnection();
100        cmd.ExecuteNonQuery();
101        da.SelectCommand = cmd;
102        da.Fill(ds);
103    }
104    catch (Exception e)
105    {
106        throw new Exception("Error al listar asistencias", e);
107    }
108    finally
109    {
110        CloseConnection();
111        cmd.Dispose();

```

```
112     }  
113     return ds;  
114 }  
115 }  
116 }
```