

ADO

App.Config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <connectionStrings>
    <add name="InstitutoConnection"
      connectionString="Data Source=DESKTOP-0F4MFSE;Initial Catalog=InstitutoTich;User ID=sa;password=Pass2017"
      providerName="System.Data.SqlClient"
    />
    <add name="EmpresaConnection"
      connectionString="Data Source=DESKTOP-0F4MFSE;Initial Catalog=InstitutoTich;User ID=sa;password=Pass2017"
      providerName="System.Data.SqlClient"
    />
    <add name="LocalConnection"
      connectionString="Data Source=(LocalDb)\MSSQLLocalDB;Initial Catalog=InstitutoTich;Integrated Security=True"
      providerName="System.Data.SqlClient"
    />
  </connectionStrings>
</configuration>
```

Leer cadena de conexión desde app.config

```
string _cnnString = ConfigurationManager.ConnectionStrings["InstitutoConnection"].ConnectionString;

//=====
// Cargar una lista de Estados usando SqlDataReader
//=====
lstEstados = new List<Estado>();
query = $"select id,nombre from Estados where id>{idEstado}";
using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.Text;
    con.Open();
    SqlDataReader reader = comando.ExecuteReader();
    while (reader.Read())
    {
        lstEstados.Add(
            new Estado()
            {
                id = Convert.ToInt32(reader["id"]),
                nombre = reader["nombre"].ToString(),
            }
        );
    }
    con.Close();
}
```

Agregar parámetros al objeto command para pasárselos a un store procedure

```

comando.Parameters.AddWithValue("Nombre", "Nuevo Estado");

comando.Parameters.Add(new SqlParameter("Nombre", "Nuevo Estado"));

SqlParameter parametro = new SqlParameter();
parametro.ParameterName = "Nombre";
parametro.SqlDbType = SqlDbType.NVarChar;
parametro.Direction = ParameterDirection.Input;
parametro.Value = "Nuevo Estado";

//=====
// Agregar un registro a la Tabla de Estados
// A través de un procedimiento almacenado
//=====
query = $"agregarEstados";
using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.StoredProcedure;

    comando.Parameters.AddWithValue("Nombre", "Nuevo Estado");

    con.Open();
    idEstado = (Int32)comando.ExecuteScalar();
    con.Close();
}

//=====
// Actualizar un registro a la Tabla de Estados
//=====
nombreEstado = "Nuevo Nombre ADO Actualizado";
query = $"update Estados set Nombre='{nombreEstado}' where id={idEstado}";

using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.Text;
    con.Open();
    comando.ExecuteNonQuery();
    con.Close();
}

```

```

//=====
// Eliminar un registro a la Tabla de Estados
//=====
query = $"delete Estados where id={idEstado}";

using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.Text;
    con.Open();
    comando.ExecuteNonQuery();
    con.Close();
}

//=====
// Llenar un DataTable con una consulta a la Tabla de Estados
// Usando SqlDataAdapter
//=====
query = "select id, nombre from Estados";
using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.Text;
    con.Open();
    SqlDataAdapter adapter = new SqlDataAdapter(comando);
    adapter.Fill(dtEstados);
    con.Close();
}

//=====
// Llenar un DataTable con una consulta a la Tabla de Estados
// Usando SqlDataAdapter
//=====
DataTable dtEstados = new DataTable();
query = "select id, nombre from Estados";
using (SqlConnection con = new SqlConnection(_cnnString))
{
    comando = new SqlCommand(query, con);
    comando.CommandType = CommandType.Text;
    con.Open();
    SqlDataAdapter adapter = new SqlDataAdapter(comando);
    adapter.Fill(dtEstados);
    con.Close();
}

```

```
//=====
// Convertir un DataTable en una lista de Estados
// Barriendo los renglones de la tabla
//=====
lstEstados = new List<Estado>();
foreach (DataRow renglon in dtEstados.Rows)
{
    Estado estado = new Estado();
    estado.id = Convert.ToInt32(renglon["id"]);
    estado.nombre = renglon["nombre"].ToString();
    lstEstados.Add(estado);
}
..
```

```
//=====
// Convertir un DataTable en una lista de Estados
// Mediante LINQ
//=====
lstEstados = new List<Estado>();
lstEstados = (from DataRow dr in dtEstados.Rows
    select new Estado()
    {
        id = Convert.ToInt32(dr["id"]),
        nombre = dr["nombre"].ToString(),
    }).ToList();
..
```