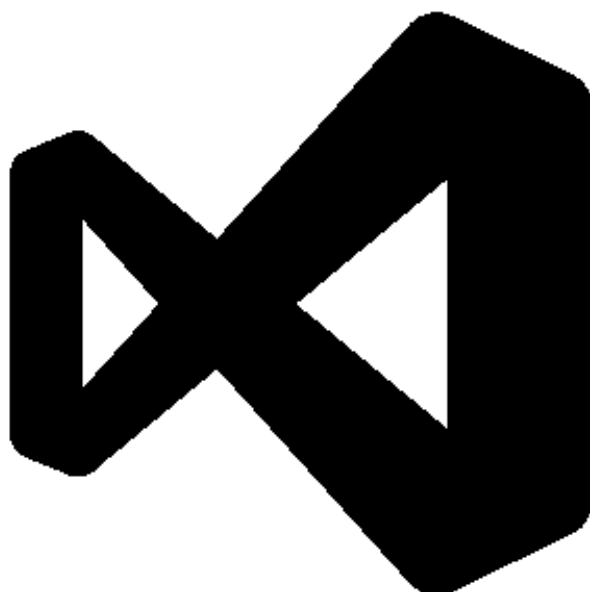


TÉCNICO DESARROLLADOR WEB



MÓDULO 7

Ejercicio Libro

Revisión No. 3

Vigente a partir del: 24 de abril del 2020

Creación de Proyecto

Procedimientos Almacenados

```
CREATE PROCEDURE [dbo].[Alumno_Agrega]
    @Matricula int,
    @Nombre nvarchar(100),
    @direccion nvarchar(100),
    @generoId nchar(1),
    @telefono nvarchar(15),
    @correo nvarchar(50),
    @fechanacimiento date,
    @curp nvarchar(18),
    @estatus nchar(1)
AS

If not exists (SELECT Matricula from Alumno where Matricula = @Matricula)

Begin
Insert into [dbo].[Alumno]( Matricula, Nombre , direccion, generoId , telefono , correo , fechanacimiento ,curp, estatus)
values (@Matricula, @Nombre , @direccion , @generoId ,@telefono , @correo , @fechanacimiento, @curp , @estatus )
End
RETURN 0
```

```
CREATE PROCEDURE [dbo].[Alumno_Consulta]
AS

    SELECT [AlumnoId]
        ,[Matricula]
        ,[Nombre]
        ,[CURP]
        ,[FechaNacimiento]
        ,[Direccion]
        ,[Telefono]
        ,[GeneroId]
        ,[Correo]
        ,[Estatus]
    FROM [Escuela].[dbo].[Alumno]
    WHERE Matricula = @Matricula

FND
```

```
CREATE PROCEDURE Alumno_ConsultaXMat
    @Matricula int
AS
BEGIN
    SELECT [AlumnoId]
        ,[Matricula]
        ,[Nombre]
        ,[CURP]
        ,[FechaNacimiento]
        ,[Direccion]
        ,[Telefono]
        ,[GeneroId]
        ,[Correo]
        ,[Estatus]
    FROM [Proyecto].[dbo].[Alumno]
    WHERE Matricula = @Matricula

END
```

Creación de proyecto ASP. Net Web Forms

Configuración de Proyecto

Configure su nuevo proyecto

Aplicación web ASP.NET (.NET Framework) C# Windows Nube Web

Nombre del proyecto

Ubicación

 ...

Nombre de la solución ⓘ


☐ Colocar la solución y el proyecto en el mismo directorio


Framework


Atrás Crear


Tipo de Proyecto


Create a new ASP.NET Web Application

**Empty**
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

**Single Page Application**
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

Add folders & core references

☐ Web Forms
☐ MVC
☐ Web API

Advanced

☒ Configure for HTTPS

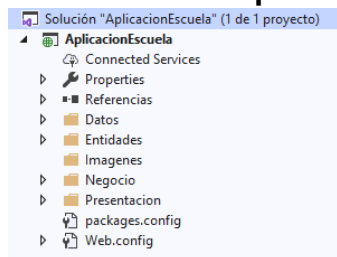
☐ Docker support
(Requires Docker Desktop)

☐ Also create a project for unit tests

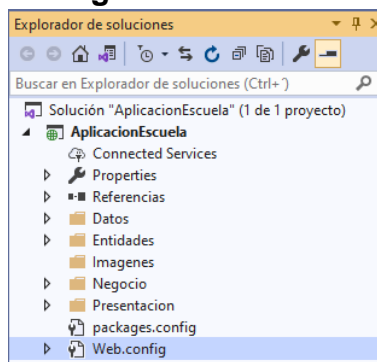
ProyectoEscuela.Tests

Back Create

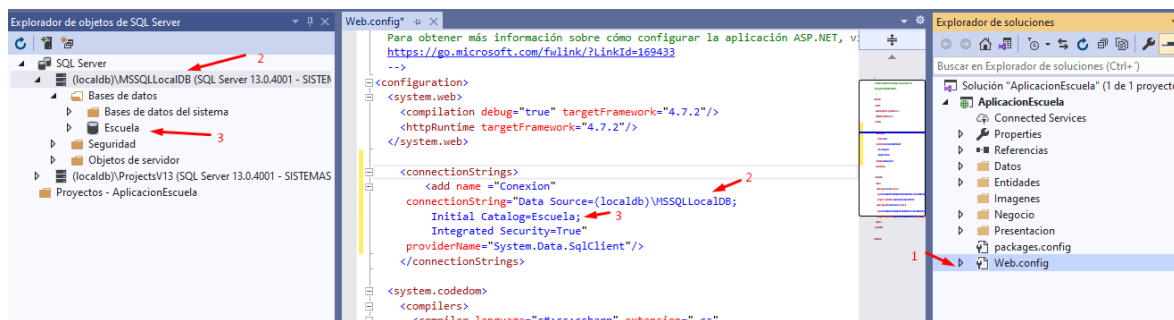
Creación de Carpetas de Proyecto



Configuración de Web Config



El código que aparece en el archivo Web.Config puede variar dependiendo de la versión de Visual Studio, sin embargo en cualquiera de las versiones hay que agregar el código ConfigurationStrings que se encuentra dentro de configuración

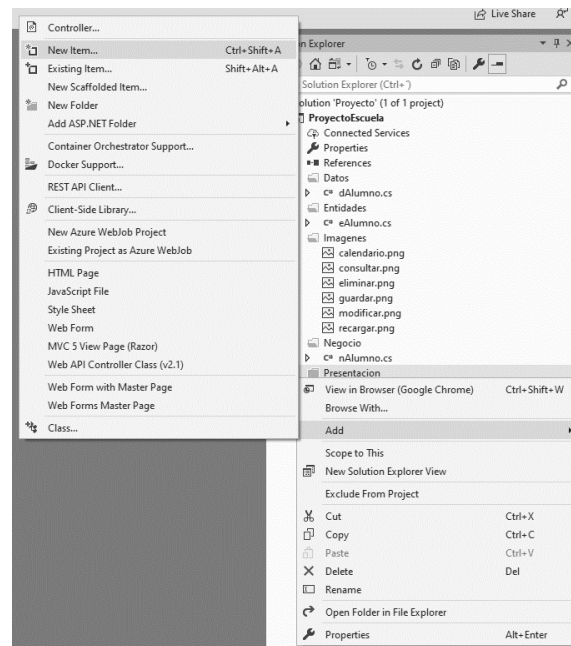


Capa Presentación

Creación de MasterPage

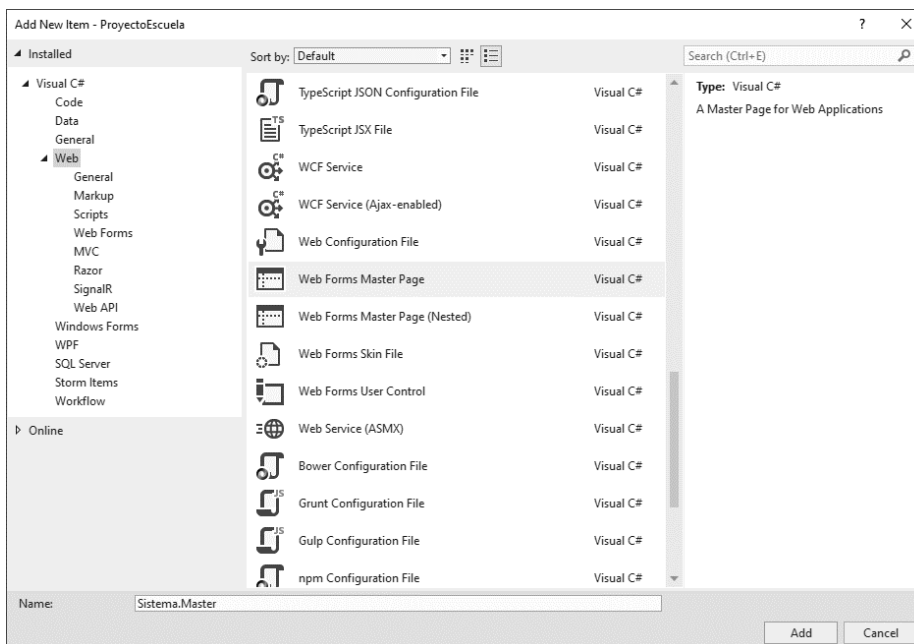
Para el entorno de nuestro proyecto necesitaremos una página maestra la cual se hará de la siguiente manera:

1. Presionar botón derecho del mouse sobre la carpeta Presentacion
2. Del menú desplegable seleccionar Agregar
3. Del menú desplegable seleccionar NewItem



En la siguiente ventana

1. Seleccionar Web Forms Master Page
2. Indicar el nombre de la MasterPage: Sistema.Master.



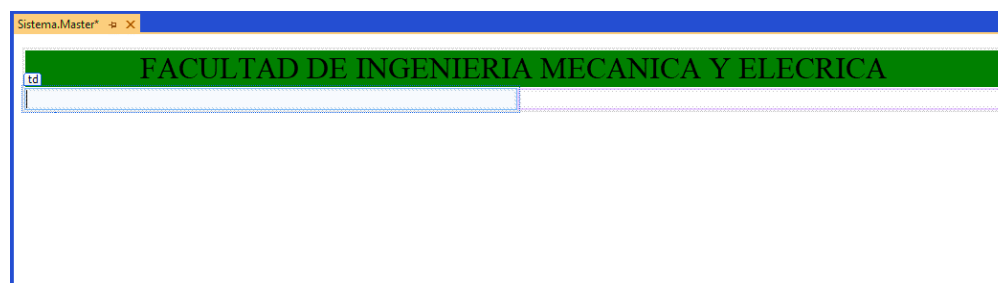
Código de Sistema. Máster

```

1 <%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Sistema.master.cs" Inherits="AplicacionEscuela.Presentacion.Sistema" %>
2
3 <!DOCTYPE html>
4
5 <html>
6 <head runat="server">
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
8 <title></title>
9 <asp:ContentPlaceholder ID="head" runat="server">
10 </asp:ContentPlaceholder>
11 <style type="text/css">
12 .auto-style1 {
13     width: 159px;
14 }
15 </style>
16 </head>
17 <body>
18 <form id="form1" runat="server">
19 <div>
20 <table>
21 <tr>
22 <td colspan = "2" style="font-size: xx-large; background-color: #000000;align-content :center;
23 text-align: center;"> FACULTAD DE INGENIERIA MECANICA Y ELECERICA </td>
24 </tr>
25 <tr>
26 <td class="auto-style1">
27
28 </td>
29 <td>
30 <asp:ContentPlaceholder ID="ContentPlaceholder1" runat="server">
31 </asp:ContentPlaceholder>
32 </td>
33 </tr>
34 </table>
35 </div>
36 </form>
37 </body>
38 </html>
39

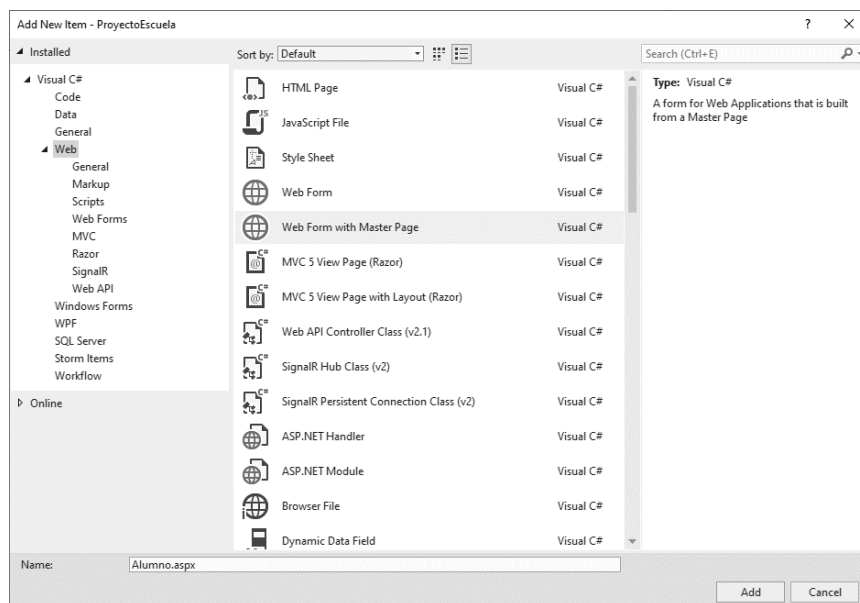
```

Vista Diseño



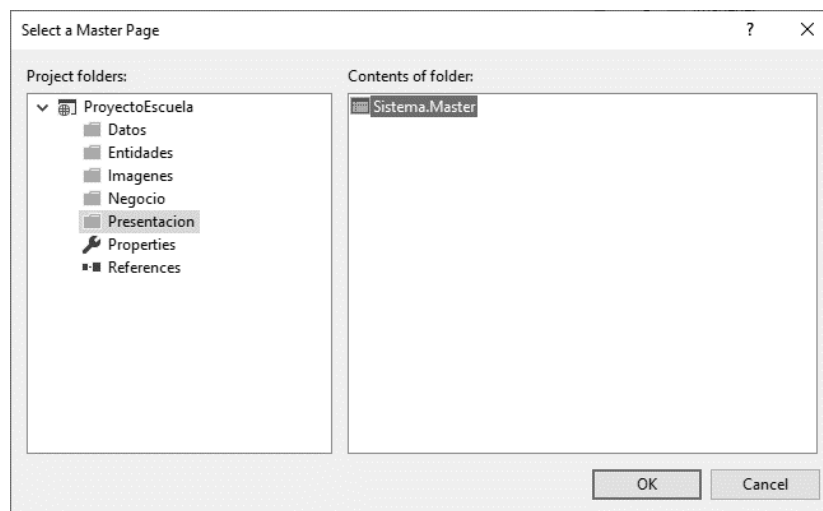
Creación de Alumno

1. Presionar botón derecho del mouse sobre la carpeta Presentación
2. Del menú desplegable seleccionar Agregar
3. Del menú desplegable seleccionar NewItem



En la siguiente ventana

1. Seleccionar WebForm with Master Page
2. Indicar el nombre del web Form with Master Page: Alumno.
3. Seleccionar la Pagina Maestra que contendrá esta página



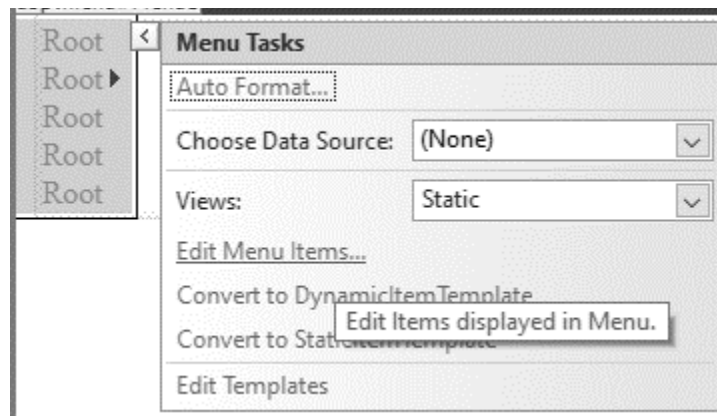
Agregar Menú a Master Page

Agregar un Control Menú dentro de la sección de Sistema.Master donde se presentará



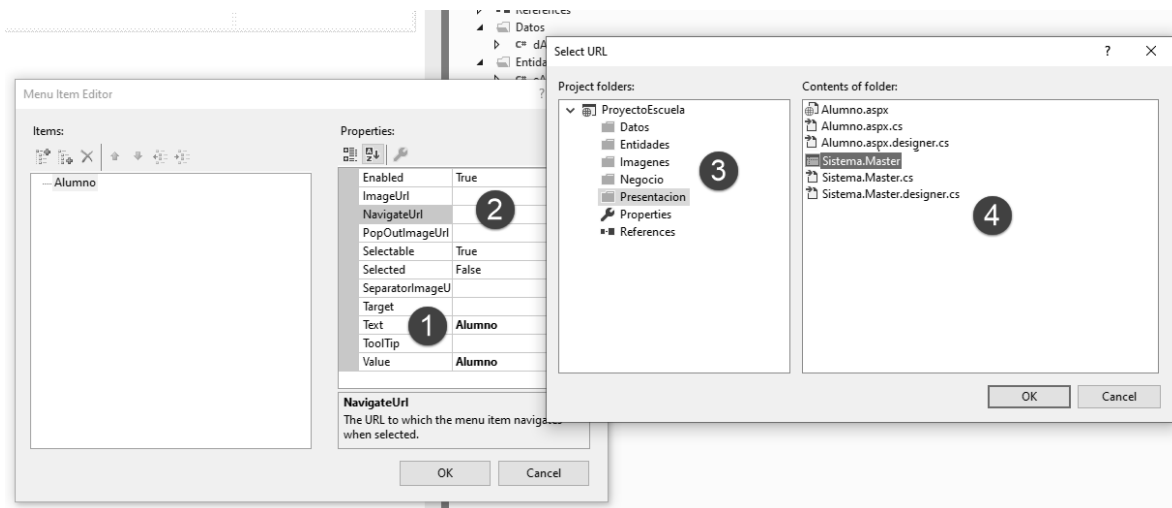
Configuración de Menú

Seleccionar Edit Menu Items



Edición de contenido de ítem

1. Texto: Alumno
2. NavigateUrl:
3. Presentacion
4. Sistema.Master

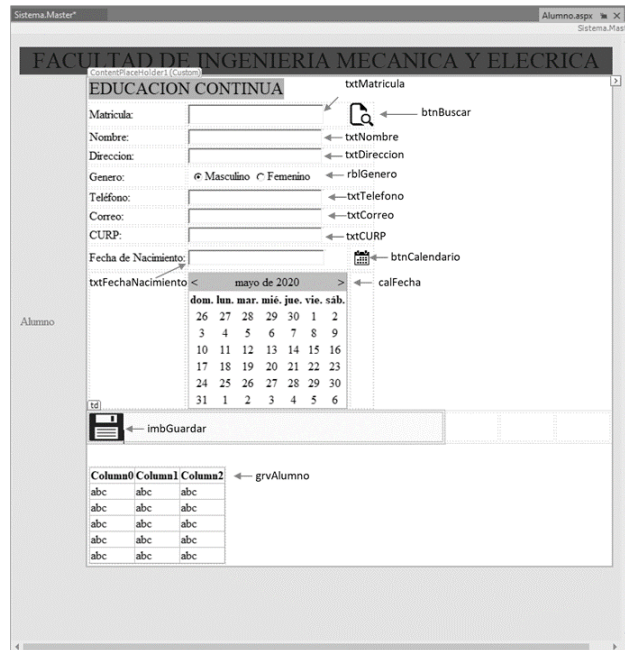


Vista previa de Menú



Creación de Esquema del Web Form Alumno

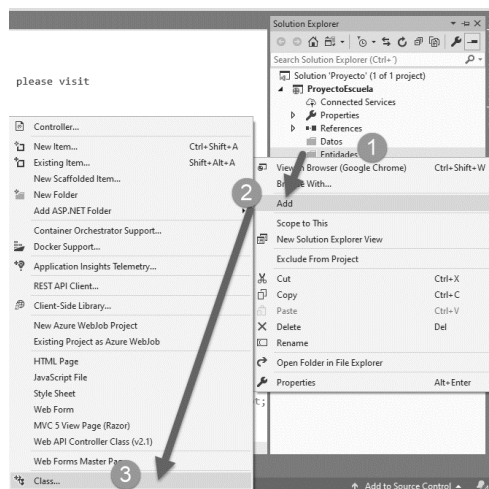
Dentro de la página alumno se agregarán solamente los controles necesarios pero no se les configurarán los eventos, esto es solo para la presentación.



Entidades o Modelos

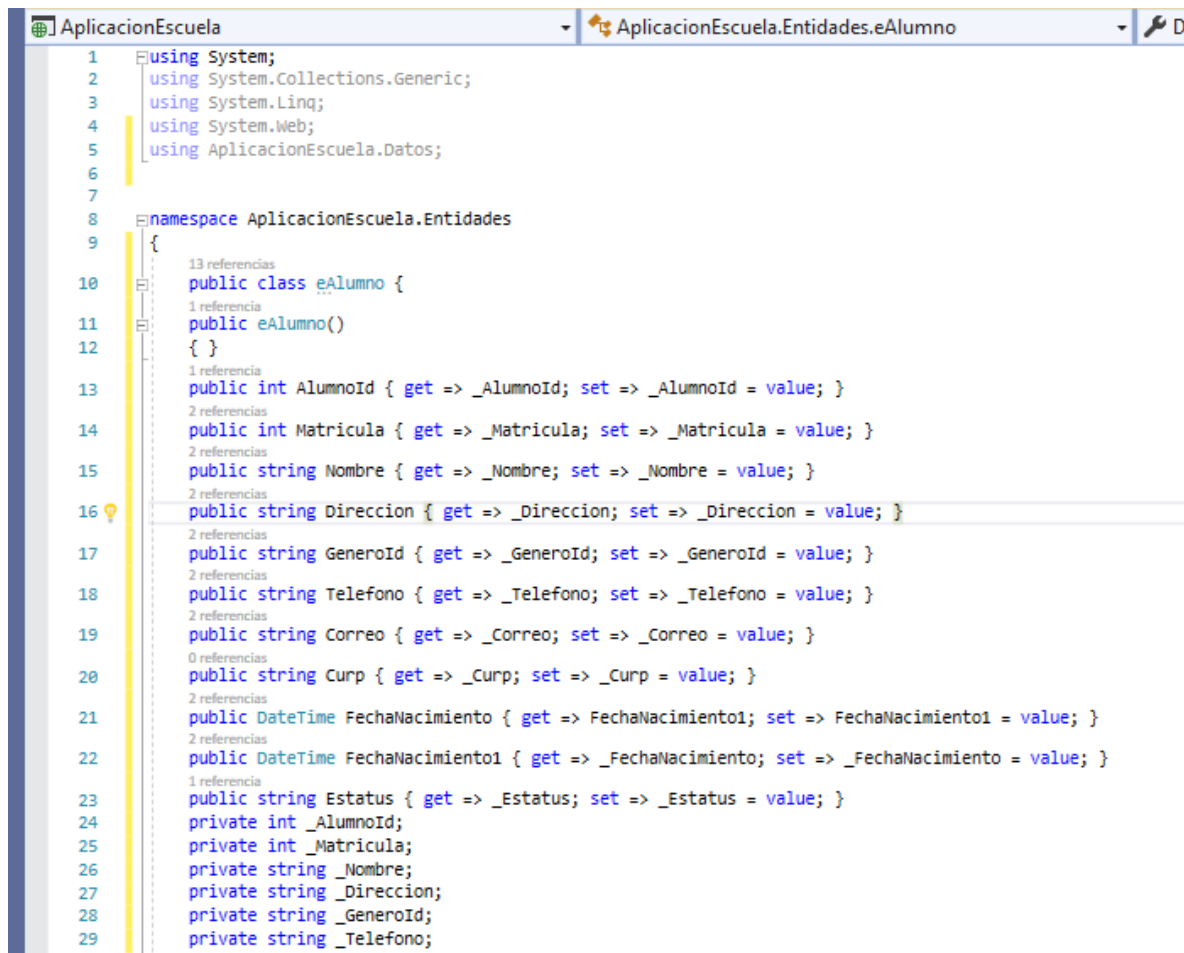
Creación de Clases Entidades

1. Presionar botón derecho del mouse sobre la carpeta Entidades
2. Del menú desplegable seleccionar Agregar
3. Del menú desplegable seleccionar Clase
4. Indicar el nombre de la clase (eAlumno)



Entidades: Clase eAlumno

Dentro de la clase eAlumno deberá ir el siguiente código:



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using AplicacionEscuela.Datos;
6
7
8  namespace AplicacionEscuela.Entities
9  {
10     public class eAlumno {
11         public eAlumno()
12         { }
13         public int AlumnoId { get => _AlumnoId; set => _AlumnoId = value; }
14         public int Matricula { get => _Matricula; set => _Matricula = value; }
15         public string Nombre { get => _Nombre; set => _Nombre = value; }
16         public string Direccion { get => _Direccion; set => _Direccion = value; }
17         public string GeneroId { get => _GeneroId; set => _GeneroId = value; }
18         public string Telefono { get => _Telefono; set => _Telefono = value; }
19         public string Correo { get => _Correo; set => _Correo = value; }
20         public string Curp { get => _Curp; set => _Curp = value; }
21         public DateTime FechaNacimiento { get => FechaNacimiento1; set => FechaNacimiento1 = value; }
22         public DateTime FechaNacimiento1 { get => _FechaNacimiento; set => _FechaNacimiento = value; }
23         public string Estatus { get => _Estatus; set => _Estatus = value; }
24         private int _AlumnoId;
25         private int _Matricula;
26         private string _Nombre;
27         private string _Direccion;
28         private string _GeneroId;
29         private string _Telefono;
```

```

30     private string _Correo;
31     private string _Curp;
32     private DateTime _FechaNacimiento;
33     private string _Estatus;
34     1 referencia
35     public eAlumno(int alumnoId, int matricula, string nombre, string curp,
36         string direccion, string generoId, string telefono, string
37         correo, DateTime fechaNacimiento, string estatus)
38     {
39         _AlumnoId = alumnoId;
40         _Matricula = matricula;
41         _Nombre = nombre;
42         _Curp = curp;
43         _FechaNacimiento = fechaNacimiento;
44         _Direccion = direccion;
45         _Telefono = telefono;
46         _GeneroId = generoId;
47         _Correo = correo;
48         _Estatus = estatus;
49     }
50 }

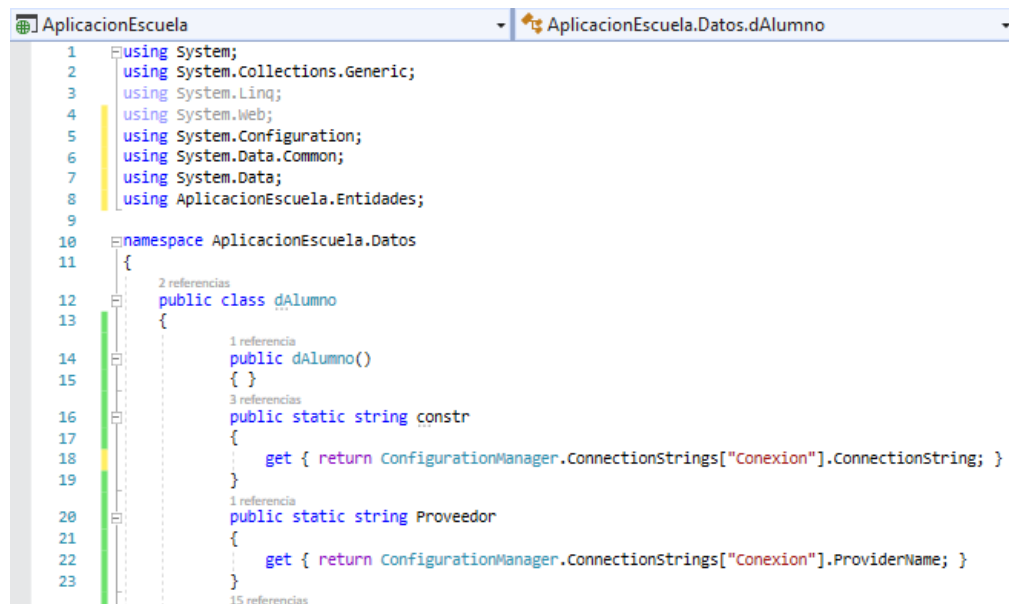
```

Capa Datos

Similar a la forma en que se agregó la clase eAlumno en la carpeta Entidades, se agregará una clase en la carpeta Datos con el nombre dAlumno

1. Presionar botón derecho del mouse sobre la carpeta Datos
2. Del menú desplegable seleccionar Agregar
3. Del menú desplegable seleccionar Clase
4. Indicar el nombre de la clase (dAlumno)

Clase de dAlumno



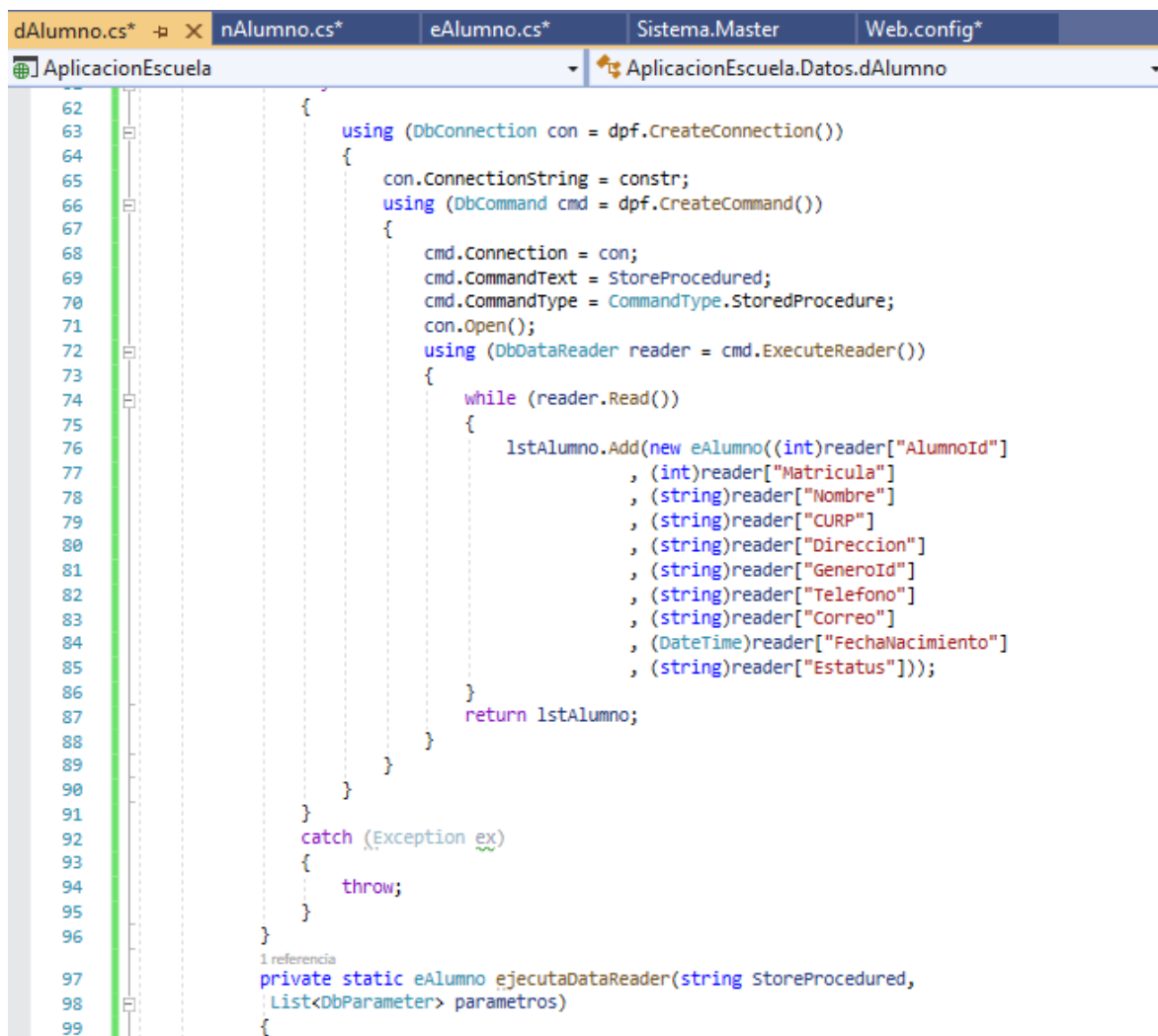
```

AplicacionEscuela
└─ AplicacionEscuela.Datos.dAlumno
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Configuration;
6  using System.Data.Common;
7  using System.Data;
8  using AplicacionEscuela.Entidades;
9
10 namespace AplicacionEscuela.Datos
11 {
12     2 referencias
13     public class dAlumno
14     {
15         1 referencia
16         public dAlumno()
17         { }
18         3 referencias
19         public static string constr
20         {
21             get { return ConfigurationManager.ConnectionStrings["Conexion"].ConnectionString; }
22         }
23         1 referencia
24         public static string Proveedor
25         {
26             get { return ConfigurationManager.ConnectionStrings["Conexion"].ProviderName; }
27         }
28     }
29     15 referencias

```

AplicacionEscuela | AplicacionEscuela.Datos.dAlumno

```
24 15 referencias
25 public static DbProviderFactory dpf
26 {
27     get { return DbProviderFactories.GetFactory(Proveedor); }
28 }
29 1 referencia
30 private static int ejecutaNonQuery(string StoreProcedured,
31     List<DbParameter> parametros)
32 {
33     int Id = 0;
34     try
35     {
36         using (DbConnection con = dpf.CreateConnection())
37         {
38             con.ConnectionString = constr;
39             using (DbCommand cmd = dpf.CreateCommand())
40             {
41                 cmd.Connection = con;
42                 cmd.CommandText = StoreProcedured;
43                 cmd.CommandType = CommandType.StoredProcedure;
44                 foreach (DbParameter parametro in parametros)
45                 {
46                     cmd.Parameters.Add(parametro);
47                 }
48                 con.Open();
49                 Id = cmd.ExecuteNonQuery();
50             }
51         }
52     }
53     catch (Exception ex)
54     {
55         throw;
56     }
57     return Id;
58 }
59 1 referencia
60 private static List<Alumno> ejecutaDataReader(string StoreProcedured)
61 {
62     DataTable dt = new DataTable();
63     List<Alumno> lstAlumno = new List<Alumno>();
64     try
```



```
62     {
63         using (DbConnection con = dpf.CreateConnection())
64         {
65             con.ConnectionString = constr;
66             using (DbCommand cmd = dpf.CreateCommand())
67             {
68                 cmd.Connection = con;
69                 cmd.CommandText = StoreProcedured;
70                 cmd.CommandType = CommandType.StoredProcedure;
71                 con.Open();
72                 using (DbDataReader reader = cmd.ExecuteReader())
73                 {
74                     while (reader.Read())
75                     {
76                         lstAlumno.Add(new eAlumno((int)reader["AlumnoId"]
77                                                 , (int)reader["Matricula"]
78                                                 , (string)reader["Nombre"]
79                                                 , (string)reader["CURP"]
80                                                 , (string)reader["Direccion"]
81                                                 , (string)reader["GeneroId"]
82                                                 , (string)reader["Telefono"]
83                                                 , (string)reader["Correo"]
84                                                 , (DateTime)reader["FechaNacimiento"]
85                                                 , (string)reader["Estatus"]));
86                     }
87                     return lstAlumno;
88                 }
89             }
90         }
91     }
92     catch (Exception ex)
93     {
94         throw;
95     }
96 }
97 1 referencia
98 private static eAlumno ejecutaDataReader(string StoreProcedured,
99 List<DbParameter> parametros)
```

```
AplicacionEscuela  AplicacionEscuela.Datos.dAlumno
100  DataTable dt = new DataTable();
101  eAlumno Alumno = new eAlumno();
102  try
103  {
104      using (DbConnection con = dpf.CreateConnection())
105      {
106          con.ConnectionString = constr;
107          using (DbCommand cmd = dpf.CreateCommand())
108          {
109              cmd.Connection = con;
110              cmd.CommandText = StoreProcedured;
111              cmd.CommandType = CommandType.StoredProcedure;
112              foreach (DbParameter parametro in parametros)
113              {
114                  cmd.Parameters.Add(parametro);
115              }
116              con.Open();
117              using (DbDataReader reader = cmd.ExecuteReader())
118              {
119                  while (reader.Read())
120                  {
121                      Alumno.AlumnoId = (int)reader["AlumnoId"];
122                      Alumno.Matricula = (int)reader["Matricula"];
123                      Alumno.Nombre = (string)reader["Nombre"];
124                      Alumno.Direccion = (string)reader["Direccion"];
125                      Alumno.GeneroId = (string)reader["GeneroId"];
126                      Alumno.Telefono = (string)reader["Telefono"];
127                      Alumno.Correo = (string)reader["Correo"];
128                      Alumno.FechaNacimiento = (DateTime)reader["FechaNacimiento"];
129                      Alumno.Curp = (string)reader["CURP"];
130                      Alumno.Estatus = (string)reader["Estatus"];
131                  }
132                  return Alumno;
133              }
134          }
135      }
136  }
137  catch (Exception ex)
138  {
139      throw;
```

```

AplicacionEscuela
AplicacionEscuela.Datos.dAlumno

140     }
141     }
142     1 referencia
143     public int AlumnoAgregar(eAlumno alumno)
144     {
145         List<DbParameter> parametros = new List<DbParameter>();
146         DbParameter param01 = dpf.CreateParameter();
147         param01.Value = alumno.Matricula;
148         param01.ParameterName = "Matricula";
149         parametros.Add(param01);
150         DbParameter param02 = dpf.CreateParameter();
151         param02.Value = alumno.Nombre;
152         param02.ParameterName = "Nombre";
153         parametros.Add(param02);
154         DbParameter param03 = dpf.CreateParameter();
155         param03.Value = alumno.Direccion;
156         param03.ParameterName = "Direccion";
157         parametros.Add(param03);
158         DbParameter param04 = dpf.CreateParameter();
159         param04.Value = alumno.GeneroId;
160         param04.ParameterName = "GeneroId";
161         parametros.Add(param04);
162         DbParameter param05 = dpf.CreateParameter();
163         param05.Value = alumno.Telefono;
164         param05.ParameterName = "Telefono";
165         parametros.Add(param05);
166         DbParameter param06 = dpf.CreateParameter();
167         param06.Value = alumno.Correo;
168         param06.ParameterName = "Correo";
169         parametros.Add(param06);
170         DbParameter param07 = dpf.CreateParameter();
171         param07.Value = alumno.FechaNacimiento;
172         param07.ParameterName = "FechaNacimiento";
173         parametros.Add(param07);
174         DbParameter param08 = dpf.CreateParameter();
175         param08.Value = alumno.Curp;
176         param08.ParameterName = "Curp";
177         parametros.Add(param08);
178         try
179         {
180             return ejecutaNonQuery("Alumno_Agrega", parametros);
181         }
182         catch { return 0; }
183     }
184     1 referencia
185     public List<eAlumno> AlumnoConsulta()
186     {
187         List<DbParameter> parametros = new List<DbParameter>();
188         return ejecutaDataReader("Alumno_Consulta");
189     }
190     1 referencia
191     public eAlumno AlumnoConsultaXMat(int matricula)
192     {
193         // Lista que contendra los valores a agragr al SP
194         List<DbParameter> parametros = new List<DbParameter>();
195         //Parametro Ingresado a SP
196         DbParameter param01 = dpf.CreateParameter();
197         param01.Value = matricula;
198         param01.ParameterName = "Matricula";
199         parametros.Add(param01);
200         // Ejecuta y regresa el valor del DataReader
201         return ejecutaDataReader("Alumno_ConsultaXMat", parametros);
202     }

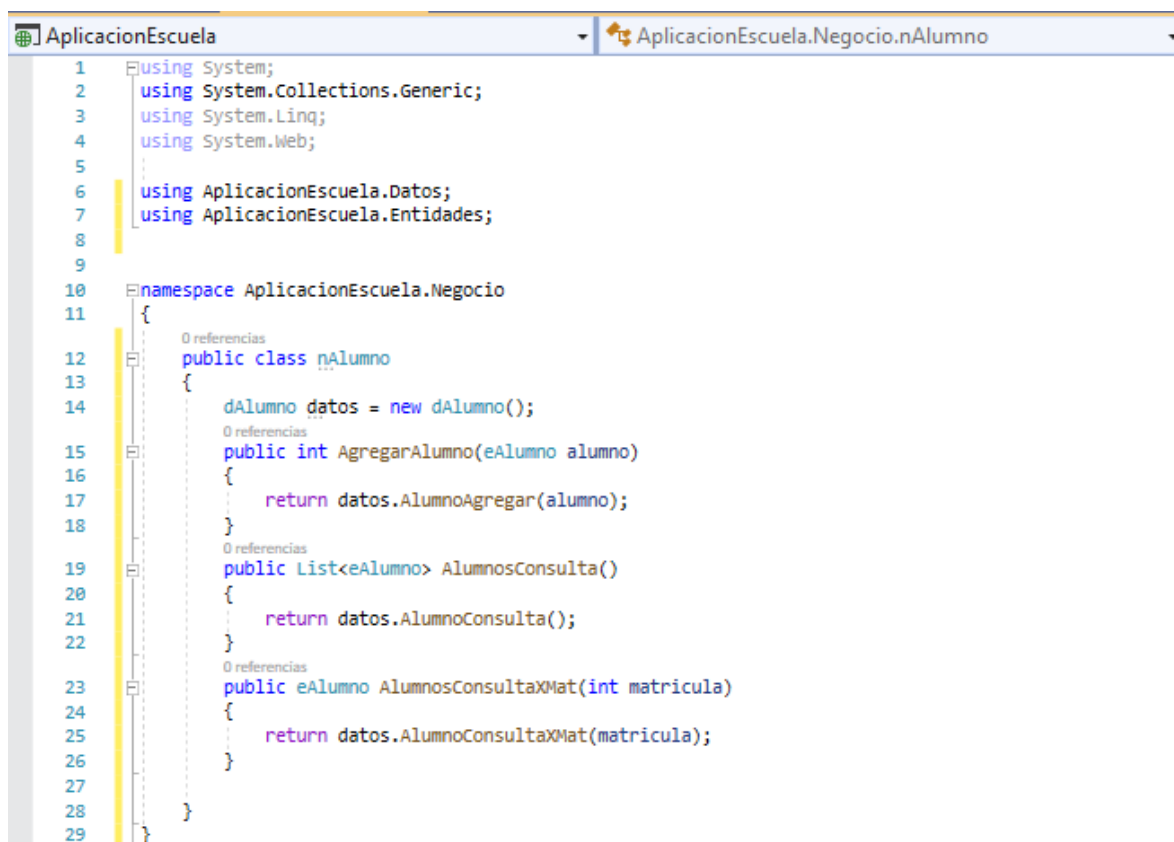
```


Capa de Negocio

clase y Alumno

Similar a la forma en que se agregó la clase eAlumno en la carpeta Entidades, se agregará una clase en la carpeta Negocio con el nombre nAlumno

1. Presionar botón derecho del mouse sobre la carpeta Negocio
2. Del menú desplegable seleccionar Agregar
3. Del menú desplegable seleccionar Clase
4. Indicar el nombre de la clase (nAlumno)



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  using AplicacionEscuela.Datos;
7  using AplicacionEscuela.Entidades;
8
9
10 namespace AplicacionEscuela.Negocio
11 {
12     0 referencias
13     public class nAlumno
14     {
15         dAlumno datos = new dAlumno();
16         0 referencias
17         public int AgregarAlumno(eAlumno alumno)
18         {
19             return datos.AlumnoAgregar(alumno);
20         }
21         0 referencias
22         public List<eAlumno> AlumnosConsulta()
23         {
24             return datos.AlumnoConsulta();
25         }
26         0 referencias
27         public eAlumno AlumnosConsultaxMat(int matricula)
28         {
29             return datos.AlumnoConsultaxMat(matricula);
30         }
31     }
32 }
```



8376-4717

1340-4020 ext. 5712



educont.fime@uanl.mx



Educación Continua (FIME)



811-002-6992



@EdContFime



@Edu_Cont



Educación Continua FIME

