

CURSO : DESARROLLO DE SERVICIOS WEB I  
PROFESOR : JORGE ENRIQUE VENTURA GONZALES  
SEMESTRE : 2023-II  
CICLO : QUINTO  
SECCION : T5YT  
GRUPO : 01  
FECHA DE INICIO: sábado 05 de agosto desde las 19:00 horas  
FECHA DE FIN : domingo 06 de agosto hasta las 10:00 horas

NOTA

ALUMNO (A) : MARIBEL MAZA AUCCATINCO

### SEGUNDO CASO PRÁCTICO – CL 02

Tenga presente que **el logro es que usted**, implemente algunas opciones de una aplicación web que utilice **ASP NET Core MVC, Recuperación de Datos con ADO NET e Inyección de Dependencias.**

#### Consideraciones generales:

- El desarrollo de la práctica es estrictamente individual.
- Está permitido el uso de apuntes, libros, materiales o recursos informáticos adicionales.
- Las preguntas deben responderse de acuerdo a los conceptos discutidos o analizados en clase y aplicando su propio criterio para dar una solución a los problemas planteados.

#### Consolidado

Pregunta		Puntaje		Llenar solo en caso de Recalificación justificada	
		Máximo	Obtenido	Sustento	Puntaje
1	A	4			
	B	4			
	C	4			
	D	4			
	E	4			
Nota Recalificada					

## **PREGUNTA 1**

La Universidad Nacional de Ingeniería le solicita implementar un nuevo sistema de matrícula de estudiantes para su proceso de admisión 2023-I. El sistema debe considerar los siguientes atributos para el postulante:

- Dni del postulante – Validar 8 dígitos (Obligatorio)
- Nombre del postulante (Obligatorio)
- Apellidos del postulante (Obligatorio)
- Nombre de colegio de educación secundaria que egresó. (Opcional)
- Año de egreso (Obligatorio)
- Carrera a la que postula (Ingeniería de Sistemas, Ingeniería Química, etc.) (Obligatorio)

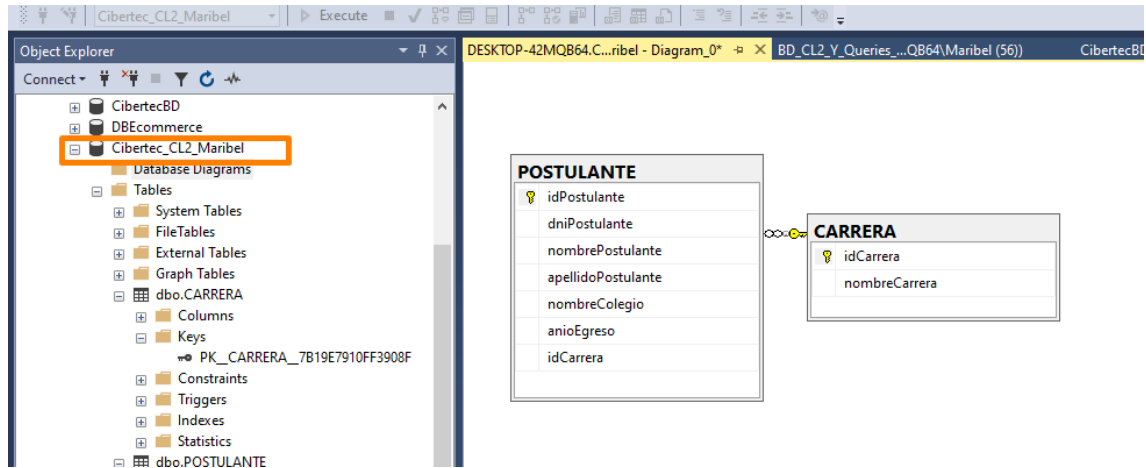
**NOTA: el alumno debe generar la base de datos y debe colocar el script de la base de datos en la solución**

Se pide:

- A.** Validación de Modelo Postulante:
  - a. Se define los atributos, propiedades de la clase. (2ptos)
  - b. Se establecen criterios de validación usando **DataAnnotations**. (2ptos)

# SOLUCION

## Creando la BD Cibertec\_CL2\_Maribel



```
1 CREATE DATABASE Cibertec_CL2_Maribel
2 GO
3
4 use Cibertec_CL2_Maribel
5 go
6
7 CREATE TABLE CARRERA(
8     idCarrera int identity(1,1) primary key,
9     nombreCarrera varchar(255) not null
10 );
11
12 CREATE TABLE POSTULANTE (
13     idPostulante int identity(1,1) primary key,
14     dniPostulante char(8) not null,
15     nombresPostulante varchar(255) not null,
16     apellidosPostulante varchar(255) not null,
17     nombreColegio varchar(255) null,
18     anioEgreso int not null,
19     idCarrera int not null,
20 );
21
22
23 ALTER TABLE POSTULANTE
24 ADD CONSTRAINT FK_PostulanteCarrera
25 FOREIGN KEY (idCarrera) REFERENCES CARRERA(idCarrera);
26
```

```

27
28 /**Insertamos valores**/
29 INSERT INTO CARRERA(nombreCarrera)
30 VALUES ('Arquitectura');
31 INSERT INTO CARRERA(nombreCarrera)
32 VALUES ('Ciencias Físicas');
33 INSERT INTO CARRERA(nombreCarrera)
34 VALUES ('Ciencias Matemáticas');
35 INSERT INTO CARRERA(nombreCarrera)
36 VALUES ('Ingeniería Ambiental');
37 INSERT INTO CARRERA(nombreCarrera)
38 VALUES ('Ingeniería Civil');
39 INSERT INTO CARRERA(nombreCarrera)
40 VALUES ('Ingeniería Económica');
41 INSERT INTO CARRERA(nombreCarrera)
42 VALUES ('Ingeniería Eléctrica');
43 INSERT INTO CARRERA(nombreCarrera)
44 VALUES ('Ingeniería Electromecánica');
45 INSERT INTO CARRERA(nombreCarrera)
46 VALUES ('Ingeniería Electrónica');
47 INSERT INTO CARRERA(nombreCarrera)
48 VALUES ('Ingeniería Estadística');
49 INSERT INTO CARRERA(nombreCarrera)
50 VALUES ('Ingeniería Física');
51 INSERT INTO CARRERA(nombreCarrera)
52 VALUES ('Ingeniería Geológica');
53 INSERT INTO CARRERA(nombreCarrera)
54 VALUES ('Ingeniería Industrial');
55 INSERT INTO CARRERA(nombreCarrera)
56 VALUES ('Ingeniería Mecatrónica');
57 INSERT INTO CARRERA(nombreCarrera)
58 VALUES ('Ingeniería Mecánica');
59 INSERT INTO CARRERA(nombreCarrera)
60 VALUES ('Ingeniería Metalúrgica');
61 INSERT INTO CARRERA(nombreCarrera)
62 VALUES ('Ingeniería Minas');
63 INSERT INTO CARRERA(nombreCarrera)
64 VALUES ('Ingeniería Naval');
65 INSERT INTO CARRERA(nombreCarrera)
66 VALUES ('Ingeniería Petroquímica');
67 INSERT INTO CARRERA(nombreCarrera)
68 VALUES ('Ingeniería Petróleo');
69 INSERT INTO CARRERA(nombreCarrera)
70 VALUES ('Ingeniería Química');
71 INSERT INTO CARRERA(nombreCarrera)
72 VALUES ('Ingeniería Sanitaria');
73 INSERT INTO CARRERA(nombreCarrera)
74 VALUES ('Ingeniería Sistemas');
75 INSERT INTO CARRERA(nombreCarrera)
76 VALUES ('Ingeniería Telecomunicaciones');
77 INSERT INTO CARRERA(nombreCarrera)
78 VALUES ('Ingeniería Textil');
79
80
81
82
83 INSERT INTO POSTULANTE(dniPostulante, nombresPostulante, apellidosPostulante,nombreColegio,
84 anioEgreso,idCarrera )
85 VALUES (77033456, 'Victor Alberto', 'Maza Auccatinco', 'IEP SEMILLITAS DEL SABER', 2020, 5);
86
87 --select * from POSTULANTE
88
89
90 ---- sp que lista Postulantes
91 CREATE OR ALTER PROCEDURE sp_GetPostulantes
92 AS
93 BEGIN
94     SELECT P.idPostulante, P.dniPostulante, P.nombresPostulante,
95            P.apellidosPostulante, P.nombreColegio, P.anioEgreso, C.nombreCarrera
96     FROM POSTULANTE P
97     INNER JOIN CARRERA C ON C.idCarrera = P.idCarrera
98 END
99 GO
100

```

```

100
101 --EXEC sp_GetPostulantes
102
103 --- sp que lista las carreras
104 CREATE OR ALTER PROCEDURE sp_GetCarreras
105 AS
106 BEGIN
107     SELECT C.idCarrera, C.nombreCarrera
108     FROM CARRERA C
109 END
110 GO
111 --EXEC sp_GetCarreras
112
113
114 -- INSERT
115 CREATE OR ALTER PROCEDURE sp_InsertPostulante
116     @prmDniPostulante char(8),
117     @prmNombresPostulante varchar(255),
118     @prmApellidosPostulante varchar(255),
119     @prmNombreColegio varchar(255),
120     @prmAnioEgreso int,
121     @prmIdCarrera int
122 AS
123 BEGIN
124     INSERT INTO POSTULANTE(dniPostulante, nombresPostulante, apellidosPostulante, nombreColegio,
125         anioEgreso, idCarrera)
126     VALUES (@prmDniPostulante, @prmNombresPostulante, @prmApellidosPostulante,
127         @prmNombreColegio, @prmAnioEgreso, @prmIdCarrera)
128 END
129 GO
130
131 -- UPDATE
132 CREATE OR ALTER PROCEDURE sp_UpdatePostulante
133     @prmIdPostulante int,
134     @prmDniPostulante char(8),
135     @prmNombresPostulante varchar(255),
136     @prmApellidosPostulante varchar(255),
137     @prmNombreColegio varchar(255),
138     @prmAnioEgreso int,
139     @prmIdCarrera int
140 AS
141 BEGIN
142
143     UPDATE POSTULANTE
144     SET
145         dniPostulante = @prmDniPostulante,
146         nombresPostulante = @prmNombresPostulante,
147         apellidosPostulante = @prmApellidosPostulante,
148         nombreColegio = @prmNombreColegio,
149         anioEgreso = @prmAnioEgreso,
150         idCarrera = @prmIdCarrera
151     WHERE idPostulante = @prmIdPostulante
152
153 END
154 GO
155
156 -- DELETE
157 CREATE OR ALTER PROCEDURE sp_DeletePostulante
158     @prmIdPostulante int
159 AS
160 BEGIN
161     DELETE FROM POSTULANTE WHERE idPostulante = @prmIdPostulante
162 END
163 GO

```

Se pide:

**A. Validación de Modelo Postulante:**

- a. Se define los atributos, propiedades de la clase. (2ptos)
- b. Se establecen criterios de validación usando **DataAnnotations**. (2ptos)

## Atributos y Validaciones

```

1 namespace Proyecto_C12_Maribel.Models
2 {
3     5 referencias
4     public class Carrera
5     {
6         1 referencia
7         public int idCarrera { get; set; }
8
9         1 referencia
10        public string? nombreCarrera { get; set; }
11    }
12 }

```

```

1 using System.ComponentModel.DataAnnotations;
2
3 namespace Proyecto_C12_Maribel.Models
4 {
5     29 referencias
6     public class Postulante
7     {
8         [Display(Name = "ID")]
9         16 referencias
10        public int idPostulante { get; set; }
11
12        [Display(Name = "DNI")]
13        [Required(ErrorMessage = "DNI requerido!")]
14        [RegularExpression(@"^\d{8}$", ErrorMessage = "Digite 8 numeros")]
15        public string? dniPostulante { get; set; }
16
17        [Display(Name = "Nombres")]
18        [Required(ErrorMessage = "Nombre(s) requerido(s)!")]
19        [RegularExpression(@"^[a-zA-Z ]*\s*([a-zA-Z ]*)$", ErrorMessage = "Digite solo letras")]
20        public string? nombresPostulante { get; set; }
21
22        [Display(Name = "Apellidos")]
23        [Required(ErrorMessage = "Apellido(s) requerido(s)!")]
24        [RegularExpression(@"^[a-zA-Z ]*\s*([a-zA-Z ]*)$", ErrorMessage = "Digite solo letras")]
25        public string? apellidosPostulante { get; set; }
26
27        [Display(Name = "Colegio de Procedencia")]
28        [RegularExpression(@"^[a-zA-Z ]*\s*([a-zA-Z ]*)$", ErrorMessage = "Digite solo letras y puntos")]
29        public string? nombreColegio { get; set; }
30
31        [Display(Name = "Año de Egreso")]
32        [Required(ErrorMessage = "Año requerido!")]
33        [RegularExpression(@"^\d{4}$", ErrorMessage = "Digite 4 numeros")]
34        public int anioEgreso { get; set; }
35
36        [Required]
37        [Display(Name = "ID Carrera")]
38        11 referencias
39        public int idCarrera { get; set; }
40        [Display(Name = "Carrera")]
41        5 referencias
42        public string? nombreCarrera { get; set; }
43    }
44 }

```

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 2 puntos	Regular: 1 punto	Deficiente: 0 puntos
04 Puntos	Cumple con lo solicitado en un 100%.	Cumple con lo solicitado en un 50%.	Cumple con lo solicitado en un 25%.	No cumple con lo solicitado.

### B. Se solicita caso de uso: **Registrar Postulante**

- Genere la vista de registro de postulante. (2ptos)
- Registrar postulante en la base de datos usando procedimientos almacenados e **Inyección de Dependencias**. (2ptos)



DNI

Nombres

Apellidos

Colegio de Procedencia

0

ID Carrera

Arquitectura

Guardar

[Regresar](#)

← → ↻  <https://localhost:7257/Postulante/Create>

Nombres

231

Apellidos

Colegio de Procedencia

Año de Egreso

22222222222

ID Carrera

Arquitectura

Guardar

[Regresar](#)

© 2023 - Proyecto\_Cl2\_Maribel - [Privacy](#)

# Registrando

Proyecto\_CI2\_Maribel [Home](#) [Privacy](#) [Postulantes](#)

## Registrar Postulante

DNI

57527956

Nombres

Ana Lucia

Apellidos

Valverde Jauregui

Colegio de Procedencia

Año de Egreso

2022

ID Carrera

Ingeniería Naval

Guardar

Regresar

Se ha agregado 1 postulante.

campo opcional

## Aparecerá .



Proyecto\_CI2\_Maribel [Home](#) [Privacy](#) [Postulantes](#)

## Lista de Postulantes

Nuevo Postulante

ID	DNI	Nombres	Apellidos	Colegio de Procedencia	Año de Egreso	Carrera	
1	77033456	Víctor Alberto	Maza Auccatínco	IEP SEMILLITAS DEL SABER	2020	Ingeniería Civil	<a href="#">Editar</a>   <a href="#">Eliminar</a>
2	57527956	Ana Lucia	Valverde Jauregui		2022	Ingeniería Naval	<a href="#">Editar</a>   <a href="#">Eliminar</a>



## Interfaces :

```
Proyecto_CL2_Maribel Proyecto_CL2_Maribel.Models.IPostulante
1 namespace Proyecto_CL2_Maribel.Models
2 {
3     3 referencias
4     public interface IPostulante
5     {
6         // Interface : donde definimos los metodos pero no los
7         // estamos implementando
8         4 referencias
9         IEnumerable<Postulante> GetPostulantes();
10
11         2 referencias
12         string Agregar(Postulante postulante);
13         2 referencias
14         string Editar(Postulante postulante);
15         2 referencias
16         string Delete(Postulante postulante);
17     }
18 }
```

```
Proyecto_CL2_Maribel Proyecto_CL2_Maribel.Models
1 namespace Proyecto_CL2_Maribel.Models
2 {
3     3 referencias
4     public interface ICarrera
5     {
6         5 referencias
7         IEnumerable<Carrera> GetCarreras();
8     }
9 }
```

## Program.cs

```
Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs Carrera.cs Create.cshtml
Proyecto_CL2_Maribel
1 using Proyecto_CL2_Maribel.Models;
2
3 var builder = WebApplication.CreateBuilder(args);
4
5 // Add services to the container.
6 builder.Services.AddControllersWithViews();
7
8 // Agregar repositorios
9 builder.Services.AddSingleton<IPostulante, PostulanteRepositorio>();
10 builder.Services.AddSingleton<ICarrera, CarreraRepositorio>();
11
12 var app = builder.Build();
13
14 // Configure the HTTP request pipeline.
15 if (!app.Environment.IsDevelopment())
16 {
17     app.UseExceptionHandler("/Home/Error");
18     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
19     app.UseHsts();
20 }
21
22 app.UseHttpsRedirection();
23 app.UseStaticFiles();
24
25 app.UseRouting();
26
27 app.UseAuthorization();
28
29 app.MapControllerRoute(
30     name: "default",
31     pattern: "{controller=Home}/{action=Index}/{id?}");
32
33 app.Run();
34
```

```

Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs Postular
Proyecto_CI2_Maribel Proyecto_CI2_Maribel.Models.CarreraRepositorio
1 using Microsoft.Data.SqlClient; //para los sp
2 using System.Data; // para los tipos de sp (command)
3
4 namespace Proyecto_CI2_Maribel.Models
5 {
6     3 referencias
7     public class CarreraRepositorio : ICarrera
8     {
9         string cadena; // 1. variable donde esta nuestra cadena de conexion
10
11         // 2. Constructor donde inicializamos la conexion
12         1 referencia
13         public CarreraRepositorio()
14         {
15             cadena = new ConfigurationBuilder()
16                 .AddJsonFile("appsettings.json")
17                 .Build()
18                 .GetConnectionString("connection");
19         }
20
21         5 referencias
22         public IEnumerable<Carrera> GetCarreras()
23         {
24             List<Carrera> carreras = new List<Carrera>();
25             using (SqlConnection connection = new SqlConnection(cadena))
26             {
27                 //sp para obtener carreras
28                 SqlCommand command = new SqlCommand("sp_GetCarreras", connection);
29                 command.CommandType = CommandType.StoredProcedure;
30
31                 connection.Open();
32                 SqlDataReader dr = command.ExecuteReader();
33
34                 while (dr.Read())
35                 {
36                     carreras.Add(new Carrera
37                     {
38                         idCarrera = dr.GetInt32(0),
39                         nombreCarrera = dr.GetString(1),
40                     });
41                 }
42             }
43             return carreras;
44         }
45     }
46 }

```

```

Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs
Proyecto_CI2_Maribel Proyecto_CI2_Maribel.Models.PostulanteRepositorio
1 using Microsoft.Data.SqlClient; //para los sp
2 using System.Data; // para los tipos de sp (command)
3
4 namespace Proyecto_CI2_Maribel.Models
5 {
6     3 referencias
7     public class PostulanteRepositorio : IPostulante
8     {
9         string cadena; // 1. variable donde esta nuestra cadena de conexion
10
11         // 2. Constructor donde inicializamos la conexion
12         1 referencia
13         public PostulanteRepositorio()
14         {
15             cadena = new ConfigurationBuilder()
16                 .AddJsonFile("appsettings.json")
17                 .Build()
18                 .GetConnectionString("connection");
19         }
20
21         4 referencias
22         public IEnumerable<Postulante> GetPostulantes()
23         {
24             //3. Agregamos nuestra lista de postulante
25             List<Postulante> postulantes = new List<Postulante>();
26
27             // 4. Agregamos la cadena de conexion
28             using (SqlConnection connection = new SqlConnection(cadena))
29             {
30                 //sp para pbtener categoria clientes
31                 SqlCommand command = new SqlCommand("sp_GetPostulantes", connection);
32                 command.CommandType = CommandType.StoredProcedure;
33
34                 connection.Open();
35                 SqlDataReader dr = command.ExecuteReader();
36
37                 while (dr.Read())
38                 {
39                     postulantes.Add(new Postulante
40                     {
41                         idPostulante = dr.GetInt32(0),
42                         dniPostulante = dr.GetString(1),
43                         nombresPostulante = dr.GetString(2),
44                         apellidosPostulante = dr.GetString(3),
45                         nombreColegio = dr.GetString(4),
46                         anioEgreso = dr.GetInt32(5),
47                         nombreCarrera = dr.GetString(6),
48                     });
49                 }
50             }
51         }
52     }
53 }

```

```

50     }
51
52     //5 . retornamos la lista de postulantes
53     return postulantes;
54 }
55
56 // 6. IMPLEMENTAMOS EL metodo agregar :
57 2 referencias
58 public string Agregar(Postulante postulante)
59 {
60     string mensaje = "";
61
62     using (SqlConnection connection = new SqlConnection(cadena))
63     {
64         try
65         {
66             SqlCommand cmd = new SqlCommand("sp_InsertPostulante", connection);
67             cmd.CommandType = CommandType.StoredProcedure;
68             cmd.Parameters.AddWithValue("@prmDniPostulante", postulante.dniPostulante);
69             cmd.Parameters.AddWithValue("@prmNombresPostulante", postulante.nombresPostulante);
70             cmd.Parameters.AddWithValue("@prmApellidosPostulante", postulante.apellidosPostulante);
71             if (postulante.nombreColegio != null)
72             {
73                 cmd.Parameters.AddWithValue("@prmNombreColegio", postulante.nombreColegio);
74             }
75             else
76             {
77                 cmd.Parameters.AddWithValue("@prmNombreColegio", " ");
78             }
79
80             cmd.Parameters.AddWithValue("@prmAnioEgreso", postulante.anioEgreso);
81             cmd.Parameters.AddWithValue("@prmIdCarrera", postulante.idCarrera);
82
83             connection.Open();
84             int filas = cmd.ExecuteNonQuery();
85             mensaje = $"Se ha agregado {filas} postulante.";
86         }
87         catch (Exception ex)
88         {
89             mensaje = ex.Message;
90         }
91     }
92
93     return mensaje;
94 }
95

```

Postulante.cs Delete.cshtml Edit.cshtml **PostulanteController.cs** Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs Carrera.cs Crea

Proyecto\_C12\_Manibel Proyecto\_C12\_Manibel.Controllers.PostulanteController

```

15 0 referencias
16 public PostulanteController()
17 {
18     iPostulante = new PostulanteRepositorio();
19     iCarrera = new CarreraRepositorio();
20 }
21
22 0 referencias
23 public async Task<IActionResult> Index()
24 {
25     return View(await Task.Run(() => iPostulante.GetPostulantes()));
26 }
27
28 // 5. Para CRUD : vista Create
29 0 referencias
30 public async Task<IActionResult> Create()
31 {
32     //viewbags para listar el combo para las carreras
33     ViewBag.carreras = new SelectList(
34         await Task.Run(() => iCarrera.GetCarreras()),
35         "idCarrera",
36         "nombreCarrera"
37     );
38     // retorna un postulante vacio
39     return View(new Postulante());
40 }
41
42 [HttpPost]
43 0 referencias
44 public async Task<IActionResult> Create(Postulante postulante)
45 {
46     //recibe un objeto Postulante
47     ViewBag.carreras = new SelectList(
48         await Task.Run(() => iCarrera.GetCarreras()),
49         "idCarrera",
50         "nombreCarrera",
51         postulante.idCarrera
52     );
53
54     if (!ModelState.IsValid)
55     {
56         return View(postulante);
57     }
58
59     //
60     ViewBag.mensaje = iPostulante.Agregar(postulante);
61     return View(postulante);
62 }

```

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 2 puntos	Regular: 1 punto	Deficiente: 0 puntos

<b>04 Puntos</b>	<b>Cumple con lo solicitado en un 100%.</b>	<b>Cumple con lo solicitado en un 50%.</b>	<b>Cumple con lo solicitado en un 25%.</b>	<b>No cumple con lo solicitado.</b>
----------------------	---	--	--	---

**C. Se solicita caso de uso: **Editar Postulante****

a. Genere la vista de edición de postulante. (2ptos)

Proyecto\_CI2\_Maribel Home Privacy Postulantes

## Editar Postulante

ID  
1

DNI  
77033456

Nombres  
Victor Manuel

Apellidos  
Maza Auccatinco

Colegio de Procedencia  
IEP SEMILLITAS DEL SABER

Año de Egreso  
2021

ID Carrera  
Ingeniería Minas

[Guardar](#) [Regresar](#)

Se ha editado 1 postulante.

© 2023 - Proyecto\_CI2\_Maribel - [Privacy](#)

b. Editar postulante en la base de datos usando procedimientos almacenados e **Inyección de Dependencias**. (2ptos)

```

Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs Carrera.cs
Proyecto_CI2_Maribel Proyecto_CI2_Maribel.Models.PostulanteRepositorio
91
92
93     return mensaje;
94 }
95
96
97 2 referencias
98 public string Editar(Postulante postulante)
99 {
100     string mensaje = "";
101     using (SqlConnection connection = new SqlConnection(cadena))
102     {
103         try
104         {
105             SqlCommand cmd = new SqlCommand("sp_UpdatePostulante", connection);
106             cmd.CommandType = CommandType.StoredProcedure;
107             cmd.Parameters.AddWithValue("@prmIdPostulante", postulante.idPostulante);
108             cmd.Parameters.AddWithValue("@prmDniPostulante", postulante.dniPostulante);
109             cmd.Parameters.AddWithValue("@prmNombresPostulante", postulante.nombresPostulante);
110             cmd.Parameters.AddWithValue("@prmApellidosPostulante", postulante.apellidosPostulante);
111             if (postulante.nombreColegio != null)
112             {
113                 cmd.Parameters.AddWithValue("@prmNombreColegio", postulante.nombreColegio);
114             }
115             else
116             {
117                 cmd.Parameters.AddWithValue("@prmNombreColegio", " ");
118             }
119             cmd.Parameters.AddWithValue("@prmAnioEgreso", postulante.anioEgreso);
120             cmd.Parameters.AddWithValue("@prmIdCarrera", postulante.idCarrera);
121
122             connection.Open();
123             int filas = cmd.ExecuteNonQuery();
124             mensaje = $"Se ha editado {filas} postulante.";
125         }
126         catch (Exception ex)
127         {
128             mensaje = ex.Message;
129         }
130     }
131     return mensaje;
132 }
133
134
135
136
137
138

```

```

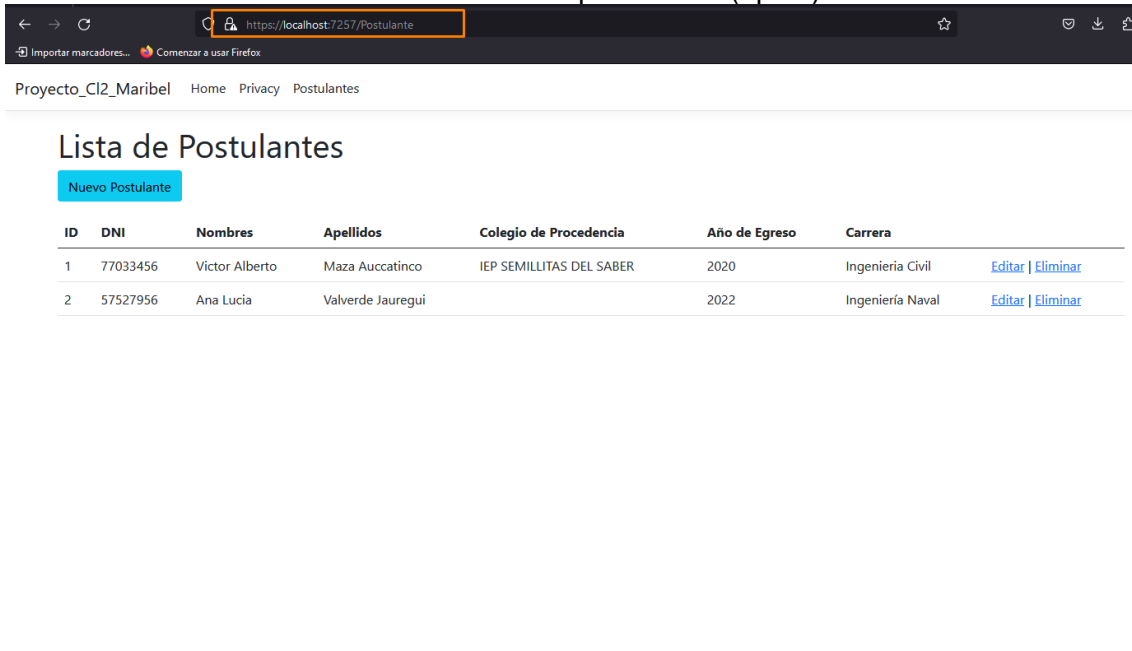
Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs
Proyecto_CI2_Maribel Proyecto_CI2_Maribel.Controllers.PostulanteController
57
58 ViewBag.mensaje = iPostulante.Agregar(postulante);
59 return View(postulante);
60 }
61
62
63 // 6. Para CRUD : vista Edit
64 0 referencias
65 public async Task<IActionResult> Edit(int idPostulante)
66 {
67     //
68     Postulante postulante = iPostulante.GetPostulantes()
69     .Where(item => item.idPostulante == idPostulante)
70     .FirstOrDefault();
71     if (postulante == null)
72     {
73         return RedirectToAction("Index");
74     }
75     //viewbags para listar el combo
76     ViewBag.carreras = new SelectList(
77         await Task.Run(() => iCarrera.GetCarreras()),
78         "idCarrera",
79         "nombreCarrera"
80     );
81     // retorna el postulante encontrado
82     return View(postulante);
83 }
84
85 [HttpPost]
86 0 referencias
87 public async Task<IActionResult> Edit(Postulante postulante)
88 {
89     //viewbags para listar el combo
90     ViewBag.carreras = new SelectList(
91         await Task.Run(() => iCarrera.GetCarreras()),
92         "idCarrera",
93         "nombreCarrera"
94     );
95     //
96     if (!ModelState.IsValid)
97     {
98         return View(postulante);
99     }
100     //
101     ViewBag.mensaje = iPostulante.Editar(postulante);
102     return View(postulante);
103 }
104
105

```

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 4 puntos	Regular: 4 punto	Deficiente: 0 puntos
04 Puntos	Cumple con lo solicitado en un 100%.	Cumple con lo solicitado en un 50%.	Cumple con lo solicitado en un 25%.	No cumple con lo solicitado.

## D. Se solicita caso de uso: **Listar Postulante**

### a. Genere la vista de listado de postulante. (2ptos)



Proyecto\_CI2\_Maribel Home Privacy Postulantes

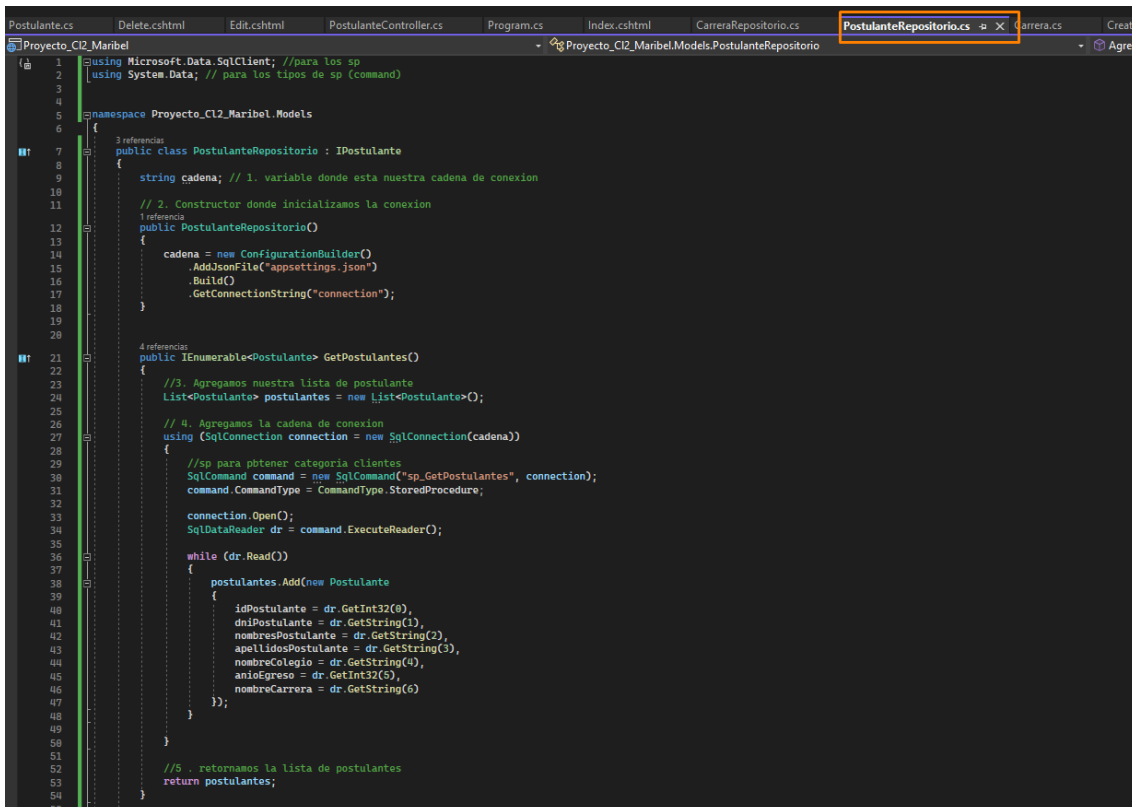
## Lista de Postulantes

[Nuevo Postulante](#)

ID	DNI	Nombres	Apellidos	Colegio de Procedencia	Año de Egreso	Carrera	
1	77033456	Victor Alberto	Maza Auccatinco	IEP SEMILLITAS DEL SABER	2020	Ingeniería Civil	<a href="#">Editar</a>   <a href="#">Eliminar</a>
2	57527956	Ana Lucia	Valverde Jauregui		2022	Ingeniería Naval	<a href="#">Editar</a>   <a href="#">Eliminar</a>

© 2023 - Proyecto\_CI2\_Maribel - [Privacy](#)

### b. Listar postulantes de la base de datos usando procedimientos almacenados e **Inyección de Dependencias**. (2ptos)



```
1 using Microsoft.Data.SqlClient; //para los sp
2 using System.Data; // para los tipos de sp (Command)
3
4
5 namespace Proyecto_CI2_Maribel.Models
6 {
7     3 referencias
8     public class PostulanteRepositorio : IPostulante
9     {
10         string cadena; // 1. variable donde esta nuestra cadena de conexion
11
12         // 2. Constructor donde inicializamos la conexion
13         1 referencia
14         public PostulanteRepositorio()
15         {
16             cadena = new ConfigurationBuilder()
17                 .AddJsonFile("appsettings.json")
18                 .Build()
19                 .GetConnectionString("connection");
20         }
21
22         4 referencias
23         public IEnumerable<Postulante> GetPostulantes()
24         {
25             //3. Agregamos nuestra lista de postulante
26             List<Postulante> postulantes = new List<Postulante>();
27
28             // 4. Agregamos la cadena de conexion
29             using (SqlConnection connection = new SqlConnection(cadena))
30             {
31                 //sp para obtener categoria clientes
32                 SqlCommand command = new SqlCommand("sp_GetPostulantes", connection);
33                 command.CommandType = CommandType.StoredProcedure;
34
35                 connection.Open();
36                 SqlDataReader dr = command.ExecuteReader();
37
38                 while (dr.Read())
39                 {
40                     postulantes.Add(new Postulante
41                     {
42                         idPostulante = dr.GetInt32(0),
43                         dniPostulante = dr.GetString(1),
44                         nombresPostulante = dr.GetString(2),
45                         apellidosPostulante = dr.GetString(3),
46                         nombreColegio = dr.GetString(4),
47                         anioEgreso = dr.GetInt32(5),
48                         nombreCarrera = dr.GetString(6)
49                     });
50                 }
51
52                 //5. retornamos la lista de postulantes
53                 return postulantes;
54             }
55         }
56     }
57 }
```

```

1 using Microsoft.AspNetCore.Mvc;
2 using Microsoft.AspNetCore.Mvc.Rendering;
3 // 1. agregamos referencias
4 using Proyecto_Cl2_Maribel.Models;
5
6 namespace Proyecto_Cl2_Maribel.Controllers
7 {
8     1 referencia
9     public class PostulanteController : Controller
10     {
11         // 2. Creamos los objetos interfaces
12         private IPostulante iPostulante;
13         private ICarrera iCarrera;
14
15         //3 . Las instanciamos en el constructor
16         0 referencias
17         public PostulanteController()
18         {
19             iPostulante = new PostulanteRepositorio();
20             iCarrera = new CarreraRepositorio();
21         }
22
23         0 referencias
24         public async Task<IActionResult> Index()
25         {
26             return View(await Task.Run(() => iPostulante.GetPostulantes()));
27         }
28     }
29 }

```

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 4 puntos	Regular: 4 punto	Deficiente: 0 puntos
04 Puntos	Cumple con lo solicitado en un 100%.	Cumple con lo solicitado en un 50%.	Cumple con lo solicitado en un 25%.	No cumple con lo solicitado.

E. Se solicita **eliminar postulante** de la base de datos usando procedimientos almacenados e **Inyección de Dependencias**.

Proyecto\_Cl2\_Maribel Home Privacy Postulantes

## Eliminar Postulante

¿Estas seguro que quieres eliminar este postulante?

ID	2
DNI	57527956
Nombres	Ana Lucía
Apellidos	Valverde Jauregui
Colegio de Procedencia	
Año de Egreso	2022
Carrera	Ingeniería Naval

[Eliminar](#) [Regresar](#)

© 2023 - Proyecto\_Cl2\_Maribel - [Privacy](#)

Proyecto\_CI2\_Maribel Home Privacy Postulantes

## Lista de Postulantes

[Nuevo Postulante](#)

ID	DNI	Nombres	Apellidos	Colegio de Procedencia	Año de Egreso	Carrera	
1	77033456	Victor Alberto	Maza Aucatincó	IEP SEMILLITAS DEL SABER	2020	Ingeniería Civil	<a href="#">Editar</a>   <a href="#">Eliminar</a>

2

ya no aparece

© 2023 - Proyecto\_CI2\_Maribel - [Privacy](#)

```
Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs Carrera.cs Create.cshtml Layout.cshtml
Proyecto_CI2_Maribel
126 int filas = cmd.ExecuteNonQuery();
127 mensaje = $"Se ha editado {filas} postulante.";
128 catch (Exception ex)
129 {
130     mensaje = ex.Message;
131 }
132 return mensaje;
133 }
134 }
135 }
136 }
137 }
138 }
139 2 referencias
140 public string Delete(Postulante postulante)
141 {
142     string mensaje = "";
143     using (SqlConnection connection = new SqlConnection(cadena))
144     {
145         try
146         {
147             SqlCommand cmd = new SqlCommand("sp.DeletePostulante", connection);
148             cmd.CommandType = CommandType.StoredProcedure;
149             cmd.Parameters.AddWithValue("@idPostulante", postulante.idPostulante);
150
151             connection.Open();
152             int filas = cmd.ExecuteNonQuery();
153             mensaje = $"Se ha editado {filas} postulante.";
154         }
155         catch (Exception ex)
156         {
157             mensaje = ex.Message;
158         }
159     }
160     return mensaje;
161 }
162 }
163 }
164 }
165 }
166 }
```

```
Postulante.cs Delete.cshtml Edit.cshtml PostulanteController.cs Program.cs Index.cshtml CarreraRepositorio.cs PostulanteRepositorio.cs Carrera.cs Create.cshtml Layout.cshtml
Proyecto_CI2_Maribel
107 // 7. Para CRUD : vista Delete
108 public async Task<IActionResult> Delete(int idPostulante)
109 {
110     //
111     Postulante postulante = iPostulante.GetPostulantes()
112     .Where(item => item.idPostulante == idPostulante)
113     .FirstOrDefault();
114     if (postulante == null)
115         return RedirectToAction("Index");
116     // retorna el postulante encontrado
117     return View(postulante);
118 }
119 }
120 }
121 }
122 }
123 }
124 // [HttpPost]
125 0 referencias
126 public async Task<IActionResult> Delete(Postulante postulante)
127 {
128     ViewBag.mensaje = iPostulante.Delete(postulante);
129     return RedirectToAction("Index");
130 }
131 }
132 }
133 }
134 }
135 }
```



<b>Rúbrica</b>				
<b>Puntos</b>	<b>Excelente: 4 puntos</b>	<b>Bueno: 4 puntos</b>	<b>Regular: 4 punto</b>	<b>Deficiente: 0 puntos</b>
<b>04 Puntos</b>	<b>Cumple con lo solicitado en un 100%.</b>	<b>Cumple con lo solicitado en un 50%.</b>	<b>Cumple con lo solicitado en un 25%.</b>	<b>No cumple con lo solicitado.</b>