

CURSO : Programación Orientada a Objetos II
PROFESOR : Edson Lhi Ugaz
SEMESTRE : 2025-23
CICLO : Cuarto
SECCIÓN : 2501
FECHA : Miércoles, 23 de Abril 2025
DURACIÓN : 90 minutos

NOTA

ALUMNO (A) : Gonzalo Pereyra

EVALUACIÓN LABORATORIO 2

LOGRO:

Tenga presente que el logro de esta evaluación es que usted aplique los conceptos aprendidos en el curso para construir una aplicación web en ASP.NET MVC y C# donde permita realizar operaciones de consultas y actualización de datos desde un origen de datos en SQL Server.

Consideraciones generales:

- Considerar el orden, la limpieza y la claridad de las respuestas.
- Inicie el proyecto con el nombre POOII_CL2_XXXX (donde XXXX serán sus apellidos y nombres)
- El desarrollo de la evaluación se desarrollará en un solo proyecto.

Consolidado

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	20			
Nota Recalificada				

Pregunta 1: CRUD Completo Tabla Products – Base de Datos NorthWind

Implementa un proceso donde permita Listar, Agregar, Actualizar y Eliminar los datos de la tabla **Products**.

Para este proceso use la base de datos **NorthWind** del SQL Server:

1. Un procedure donde permita listar los registros de la tabla **Products** y un procedure donde liste los registros de la tabla **Categories**.
Products: ProductID, ProductName, CategoryID, UnitPrice, UnitsInStock.
Categories: CategoryID, CategoryName.
2. Un procedure donde permita agregar un registro a la tabla **Products**, defina sus parámetros.
3. Un procedure donde permita actualizar un registro a la tabla **Products** por su campo **ProductID**, defina sus parámetros.

PROCEDIMIENTOS ALMACENADOS:

```
Pereyra_Gonzalo_T...8OE\USUARIO (53)) *  X
USE Northwind
GO

-----

CREATE OR ALTER PROCEDURE sp_Listar_Productos
AS
BEGIN
    SELECT p.ProductID, p.ProductName, p.CategoryID, p.UnitPrice, p.UnitsInStock, c.CategoryName, p.Discontinued
    FROM Products p
    INNER JOIN Categories c ON p.CategoryID = c.CategoryID;
END;

sp_Listar_Productos
GO

CREATE OR ALTER PROCEDURE sp_Listar_Categorias
AS
BEGIN
    SELECT CategoryID, CategoryName
    FROM Categories;
END;

sp_Listar_Categorias
GO

CREATE OR ALTER PROCEDURE sp_Agregar_Producto
    @ProductName NVARCHAR(40),
    @CategoryID INT,
    @UnitPrice MONEY,
    @UnitsInStock SMALLINT
AS
BEGIN
    INSERT INTO Products (ProductName, CategoryID, UnitPrice, UnitsInStock)
    VALUES (@ProductName, @CategoryID, @UnitPrice, @UnitsInStock);
END;

GO
```

```

CREATE OR ALTER PROCEDURE sp_Actualizar_Producto
    @ProductID INT,
    @ProductName NVARCHAR(40),
    @CategoryID INT,
    @UnitPrice MONEY,
    @UnitsInStock SMALLINT
AS
BEGIN
    UPDATE Products SET
        ProductName = @ProductName,
        CategoryID = @CategoryID,
        UnitPrice = @UnitPrice,
        UnitsInStock = @UnitsInStock
    WHERE ProductID = @ProductID;
END;
GO

```

```

CREATE OR ALTER PROCEDURE sp_Eliminar_Producto
    @ProductID INT
AS
BEGIN
    UPDATE Products
    SET Discontinued = 1
    WHERE ProductID = @ProductID;
END;
GO

```

En una aplicación ASP.NET MVC:

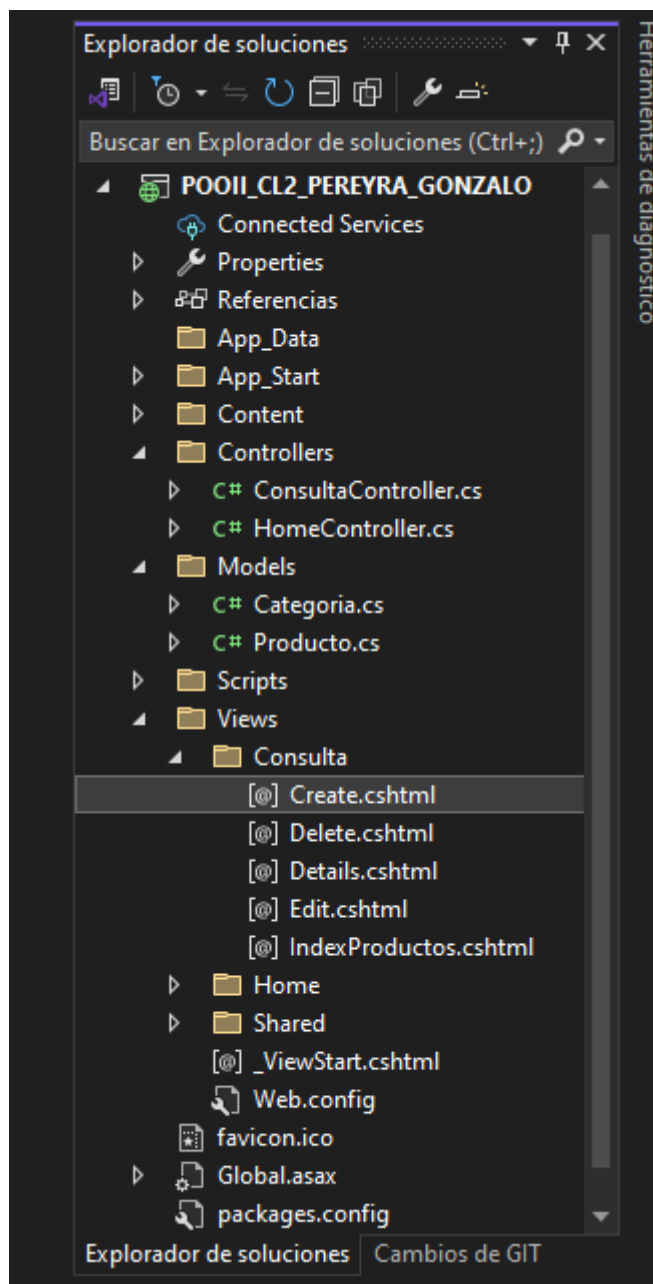
- En la carpeta Models, crea la clase **Products** y la clase **Categories**.
- En la carpeta Controllers, utilice el controlador ConsultaController.
- En el controlador:
 - Crea un IEnumerable para listar los registros de la tabla **Products** y otro IEnumerable para listar los registros de la Tabla **Categories**, ejecute cada uno su procedimiento almacenado para probar que funcionen.
 - Crea el ActionResult **IndexProductos** donde envía a la vista la lista de los registros de la tabla **Products** a través de su IEnumerable.
 - Crea el ActionResult **Create()** donde envía los registros de **Products** y la lista de la tabla **Categories** para poblar el **DropDownList** de la Vista. En el POST del ActionResult, ejecuta el procedimiento almacenado para insertar el registro visualizando un mensaje indicando si el proceso ha sido satisfactorio o el mensaje de error a través del ViewBag.
 - Create el ActionResult **Edit()**, donde al seleccionar desde la página **IndexProductos**, visualizamos los datos de los **Products**, incluyendo el DropDownList de la **Categories** que se ha registrado. En el POST del ActionResult, primero ejecute un mensaje de confirmación si desea Actualizar el registro, en caso de aceptar ejecutar el procedimiento almacenado de actualización por empleado, visualizando el mensaje de la ejecución.
 - Create el ActionResult **Details()** y **Delete()**, donde al seleccionar desde la página **IndexProductos**, visualizamos los datos de los **Products** o muestre una vista previa antes de eliminar.

En la Vista:

- Defina la vista para **IndexProductos**, donde se lista una vista de **Products**. En la Vista debe mostrar **ProductID**, **ProductName**, **CategoyID**, **UnitPrice**, **UnitsInStock**. Agregar un ActionLink “Nuevo” el cual nos redirecciona a la Vista para crear un nuevo producto.
- Defina la vista para **Create**, donde agrega un registro de **Products**. En la Vista debe agregar un DropDownList para visualizar la lista de **Categories** (ActionLink) Ingrese los datos en los controles, al presionar el Link “Agregar”, se visualiza el mensaje “Producto Agregado satisfactoriamente”.
- Defina la vista para **Edit**, donde actualiza los datos del producto seleccionado. En la Vista visualizamos los datos del producto seleccionado y agregue un DropDownList para visualizar la lista de categoría. Al presionar el botón Actualizar (input tipo submit), si presiona la opción ACEPTAR, se visualiza el mensaje “Producto Actualizado satisfactoriamente”.
- Defina la vista para **Delete**, donde elimina el producto seleccionado. Mostrar en una Vista los datos del producto seleccionado ante de eliminar.

Mejore las vistas utilizando un diseño moderno.

EXPLORADOR DE SOLUCIONES:



CONTROLADOR:

```
11 namespace POOTII_CL2_PEREYRA_GONZALO.Controllers
12 {
13     0 referencias
14     public class ConsultaController : Controller
15     {
16         // Listar tabla PRODUCTO
17         3 referencias
18         IEnumerable<Producto> listarProducto()
19         {
20             List<Producto> temporal = new List<Producto>();
21             using (SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["cadena"].ConnectionString))
22             {
23                 cn.Open();
24                 SqlCommand cmd = new SqlCommand("sp_Listar_Productos", cn);
25                 SqlDataReader dr = cmd.ExecuteReader();
26                 while (dr.Read())
27                 {
28                     bool discontinued = dr.GetBoolean(6);
29                     if (!discontinued)
30                     {
31                         temporal.Add(new Producto()
32                         {
33                             ProductID = dr.GetInt32(0),
34                             ProductName = dr.GetString(1),
35                             CategoryID = dr.GetInt32(2),
36                             UnitPrice = dr.GetDecimal(3),
37                             UnitsInStock = dr.GetInt16(4),
38                             CategoryName = dr.GetString(5)
39                         });
40                     }
41                 }
42                 dr.Close();
43                 cn.Close();
44             }
45             return temporal;
46         }
47     }
48 }
```

```
47 //Listar tabla CATEGORIA
48 4 referencias
49 IEnumerable<Categoria> listarCategoria()
50 {
51     List<Categoria> temporal = new List<Categoria>();
52     using (SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["cadena"].ConnectionString))
53     {
54         cn.Open();
55         SqlCommand cmd = new SqlCommand("sp_Listar_Categorias", cn);
56         SqlDataReader dr = cmd.ExecuteReader();
57         while (dr.Read())
58         {
59             temporal.Add(new Categoria()
60             {
61                 CategoryID = dr.GetInt32(0),
62                 CategoryName = dr.GetString(1)
63             });
64         }
65         dr.Close();
66         cn.Close();
67     }
68     return temporal;
69 }
```

```
71 //Crear PRODUCTO
72 1 referencia
73 string agregarProducto(Producto reg)
74 {
75     string mensaje = "";
76     using (SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["cadena"].ConnectionString))
77     {
78         cn.Open();
79         SqlCommand cmd = new SqlCommand("sp_Agregar_Producto @ProductName, @CategoryID, " +
80             "@UnitPrice, @UnitsInStock", cn);
81
82         cmd.Parameters.AddWithValue("@ProductName", reg.ProductName);
83         cmd.Parameters.AddWithValue("@CategoryID", reg.CategoryID);
84         cmd.Parameters.AddWithValue("@UnitPrice", reg.UnitPrice);
85         cmd.Parameters.AddWithValue("@UnitsInStock", reg.UnitsInStock);
86
87         int i = cmd.ExecuteNonQuery();
88
89         cn.Close();
90         mensaje = $"{i} Producto agregado satisfactoriamente";
91     }
92     return mensaje;
93 }
```

```

95 //Editar PRODUCTO
96 1 referencia
97 string actualizarProducto(Producto reg)
98 {
99     string mensaje = "";
100     using (SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["cadena"].ConnectionString))
101     {
102         cn.Open();
103         SqlCommand cmd = new SqlCommand("sp_Actualizar_Producto @ProductID, @ProductName, @CategoryID, " +
104             "@UnitPrice, @UnitsInStock", cn);
105
106         cmd.Parameters.AddWithValue("@ProductID", reg.ProductID);
107         cmd.Parameters.AddWithValue("@ProductName", reg.ProductName);
108         cmd.Parameters.AddWithValue("@CategoryID", reg.CategoryID);
109         cmd.Parameters.AddWithValue("@UnitPrice", reg.UnitPrice);
110         cmd.Parameters.AddWithValue("@UnitsInStock", reg.UnitsInStock);
111
112         int i = cmd.ExecuteNonQuery();
113
114         cn.Close();
115         mensaje = $"{i} Producto actualizado satisfactoriamente";
116     }
117     return mensaje;
118 }

```

```

120 //DETALLE - ELIMINAR PRODUCTO
121 1 referencia
122 string eliminarProducto(int id)
123 {
124     string mensaje = "";
125     using (SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["cadena"].ConnectionString))
126     {
127         cn.Open();
128         SqlCommand cmd = new SqlCommand("sp_Eliminar_Producto @ProductID", cn);
129
130         cmd.Parameters.AddWithValue("@ProductID", id);
131
132         int i = cmd.ExecuteNonQuery();
133
134         cn.Close();
135         mensaje = $"Se ha eliminado {i} insumo";
136     }
137     return mensaje;
138 }
139
140 2 referencias
141 Producto buscarProducto(int id)
142 {
143     return listarProducto().FirstOrDefault(x => x.ProductID == id);
144 }

```

VISTAS:

```

145 //=====SOLO LAS VISTAS=====
146 0 referencias
147 public ActionResult IndexProductos()
148 {
149     return View(listarProducto());
150 }
151
152 //Get CREATE
153 0 referencias
154 public ActionResult Create()
155 {
156     ViewBag.categoria = new SelectList(listarCategoria(), "CategoryID", "CategoryName");
157     return View(new Producto());
158 }
159
160 //POST CREATE
161 [HttpPost]
162 0 referencias
163 public ActionResult Create(Producto reg)
164 {
165     ViewBag.mensaje = agregarProducto(reg);
166     ViewBag.categoria = new SelectList(listarCategoria(), "CategoryID", "CategoryName", reg.CategoryID);
167     return View(reg);
168 }

```

```

170
171 //Get EDIT
172 0 referencias
173 public ActionResult Edit(int id)
174 {
175     Producto reg = buscarProducto(id);
176     ViewBag.categoria = new SelectList(listarCategoria(), "CategoryID", "CategoryName");
177     return View(reg);
178 }
179
180 //POST EDIT
181 [HttpPost]
182 0 referencias
183 public ActionResult Edit(Producto reg)
184 {
185     ViewBag.mensaje = actualizarProducto(reg);
186     ViewBag.categoria = new SelectList(listarCategoria(), "CategoryID", "CategoryName", reg.CategoryID);
187     return View(reg);
188 }
189
190 //-----
191 // esta es la vista de la opcion detalles
192 0 referencias
193 public ActionResult Details(int? id = null)
194 {
195     if (id == null)
196     {
197         return RedirectToAction("IndexProductos");
198     }
199     Producto reg = listarProducto().FirstOrDefault(x => x.ProductID == id);
200     return View(reg);
201 }
202
203 // GET Delete
204 0 referencias
205 public ActionResult Delete(int? id)
206 {
207     if (id == null)
208     {
209         return RedirectToAction("IndexProductos");
210     }
211     Producto reg = buscarProducto(id.Value);
212     if (reg == null)
213     {
214         return HttpNotFound();
215     }
216     return View(reg);
217 }
218
219 //POS DELETE
220 [HttpPost, ActionName("Delete")]
221 [ValidateAntiForgeryToken]
222 0 referencias
223 public ActionResult DeleteConfirmed(int id)
224 {
225     eliminarProducto(id);
226     return RedirectToAction("IndexProductos");
227 }
228 }

```


APP WEB FUNCIONAL:

Index de Productos - Mi aplicación

localhost:60880/Consulta/IndexProductos

Nombre de la aplicación Inicio Acerca de Contacto

Listado de Productos

+ Nuevo Producto

Id producto	Nombre	Id Categoría	Precio	Unidades disponibles	Acciones
4	Chef Anton's Cajun Seasoning	2	22,00	53	Editar Detalle Eliminar
6	Grandma's Boysenberry Spread	2	25,00	120	Editar Detalle Eliminar
7	Uncle Bob's Organic Dried Pears	7	30,00	15	Editar Detalle Eliminar
8	Northwoods Cranberry Sauce	2	40,00	6	Editar Detalle Eliminar
10	Ikura	8	31,00	31	Editar Detalle Eliminar
11	Queso Cabrales	4	21,00	22	Editar Detalle Eliminar
12	Queso Manchego La Pastora	4	38,00	86	Editar Detalle Eliminar

Crear Producto - Mi aplicación

localhost:60880/Consulta/Create

Nombre de la aplicación Inicio Acerca de Contacto

Crear Nuevo Producto

Nombre

Precio

Unidades disponibles

Nombre categoría

[Agregar Producto](#)

1 Producto agregado satisfactoriamente

[Regresar](#)

91	BONITO	8	29,00	100	Editar Detalle Eliminar
----	--------	---	-------	-----	-------------------------------------------------------------------------

Edit - Mi aplicación ASP.NET

localhost:60880/Consulta/Edit/91

Nombre de la aplicaciónInicioAcerca deContacto

Editar Productos

Nombre

CALAMAR

Precio

16

Unidades disponibles

200

Nombre categoria

Seafood

Actualizar

1 Producto actualizado satisfactoriamente

[Regresar](#)

© 2025 - Mi aplicación ASP.NET

91	CALAMAR	8	16,00	200	Editar	Detalle	Eliminar
----	---------	---	-------	-----	------------------------	-------------------------	--------------------------

Details - Mi aplicación ASP.NET

localhost:60880/Consulta/Details/91

Nombre de la aplicaciónInicioAcerca deContacto

Detalles

Nombre

CALAMAR

Precio

16,00

Unidades disponibles

200

Nombre categoria

Seafood

[Editar](#) | [Regresar](#)

© 2025 - Mi aplicación ASP.NET

Editar Productos

Nombre

PULPO

Precio

23

Unidades disponibles

500

Nombre categoria

Seafood



Actualizar

1 Producto actualizado satisfactoriamente

[Regresar](#)

© 2025 - Mi aplicación ASP.NET

91	PULPO	8	23,00	500	✎ Editar	🔍 Detalle	🗑 Eliminar
----	-------	---	-------	-----	--------------------------	---------------------------	----------------------------

¿Estás seguro que deseas eliminar este producto?

Producto

ID

91

Nombre

PULPO

Categoría

Seafood

Precio Unitario

23,0000

Unidades en Stock

500

Eliminar

Cancelar

© 2025 - Mi aplicación ASP.NET

84	PAPITAS	5	13,00	99	Editar	Detalle	Eliminar
85	CARNE	6	32,00	5	Editar	Detalle	Eliminar
86	CHANCHO	6	19,00	12	Editar	Detalle	Eliminar

© 2025 - Mi aplicación ASP.NET

Rúbrica				
Puntos	Excelente: 20 puntos	Bueno:15 puntos	Regular: 7 punto	Deficiente: 2 puntos
20 Puntos	Desarrollo de procedures, clases, desarrollo de Index y su Vista; desarrollo del Action NuevoProducto (GET y POST) y su vista. Desarrollo del Action Actualizar (GET y POST) y su vista donde visualiza los datos y confirma el proceso para ejecutarla, visualizando el mensaje en forma correcta.	Desarrollo de procedures, clases, desarrollo de Index y su Vista. Desarrollo del Action NuevoProducto (GET y POST) y su vista donde envia el mensaje de confirmación en forma correcta	Desarrollo de procedure y las clases, desarrollo correcto del Action Index , el cual se visualiza los registros en la Vista en forma correcta	Desarrollo correcto de los procedimientos almacenados y las clases