

# MEDIALUNAS SA

Medialunas a Domicilio



Roberto Piombi  
TP: Carrito de Compras

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

# 1 CONTENIDO

2	Descripción del negocio .....	4
	Beneficios para el cliente .....	4
	Beneficios para la panadería .....	4
2.1	Alcance .....	4
3	Estrategias de programación.....	5
3.1	Programación .....	5
3.2	Base de datos .....	5
3.2.1	Tipo de conexión: .....	6
3.2.2	Procedimientos almacenados .....	6
3.3	Seguridad.....	6
3.3.1	Protección de contraseñas: .....	6
3.3.2	Hashing.....	6
3.3.3	Algoritmo de encriptación.....	6
3.3.4	Algoritmo de dígitos verificadores: .....	7
4	Política de programación .....	8
4.1	Metodología de acceso y seguridad:.....	8
4.1.1	Usuarios de acceso:.....	8
4.1.2	Descripción de perfiles: .....	8
4.1.3	Autenticación de usuarios:.....	8
4.1.4	Olvido de contraseña: .....	9
4.1.5	Intentos fallidos y bloqueo de usuarios: .....	9
4.1.6	Plan de contingencia en caso de falla de integridad: .....	9
4.2	Política de copia de seguridad: .....	9
4.3	Bitácora .....	10
5	Mapa del sitio.....	11
6	Diagrama ciclo de vida autenticación .....	12
7	Diagrama de flujo .....	13
8	Caso de uso Login.....	14
9	Capa de presentación.....	15
9.1	Inicio.aspx.....	16

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

9.2	Login.aspx.....	19
9.3	crearUsuario.aspx.....	22
9.4	Webmaster.aspx .....	25
9.5	Bitacora.aspx .....	29
9.6	backup.aspx.....	31
9.7	administrador.aspx.....	33
9.8	alta_producto.aspx .....	35
9.9	error.aspx .....	38
10	Capa de negocio .....	41
10.1	BitacoraBLL.vb .....	41
10.2	CarritoBLL.vb .....	42
10.3	PermisosBLL.vb.....	43
10.4	ProductosBLL.vb .....	43
10.5	SeguridadBLL.vb .....	44
10.6	UsuariosBLL.vb .....	46
10.7	WebmasterBLL.vb .....	48
10.8	Base.vb .....	50
11	Capa de acceso a datos .....	51
11.1	Acceso.vb.....	51
11.2	backupDAL.vb.....	52
11.3	BitacoraDAL.vb .....	53
11.4	DigitoVerificadorDAL.vb .....	54
11.5	PanaderiasDAL.vb.....	55
11.6	PermisosDAL.vb.....	56
11.7	ProductosDAL.vb .....	56
11.8	UsuariosDAL.vb .....	57
12	Capa de entidades de negocio .....	61
12.1	Bitacora.vb .....	61
12.2	DigitoVerificador.vb .....	62
12.3	DigitoVerificadorVertical.vb .....	62
12.4	Panaderia.vb.....	63
12.5	Pedidos.vb .....	64

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

12.6	Permiso.vb.....	65
12.7	Persona.vb.....	65
12.8	Producto.vb .....	67
12.9	Usuario.vb .....	67
12.10	Usuario_Permiso.vb .....	69
13	Estructura de Base de Datos .....	70
13.1	DER .....	70
13.2	Procedimientos almacenados .....	71
13.3	BITACORA .....	71
13.4	DIGITOVERIFICADOR .....	71
13.5	ITEM .....	72
13.6	PANADERIA.....	72
13.7	PEDIDO .....	73
13.8	PERMISO.....	73
13.9	PRODUCTO .....	74
13.10	USUARIO.....	74
13.11	USUARIO_PERMISO.....	76

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 2 DESCRIPCIÓN DEL NEGOCIO

---

**MEDIALUNAS SA** se dedica a la gestión de venta de Medialunas y Facturas por internet, siendo únicamente un canal de comercialización online.

El sistema online permitirá adquirir productos previa registración de los clientes, puede agregar en el carrito de compras el producto deseado sin preocuparse de a quien se lo compra, el sistema mediante geolocalización se encarga ubicar la Panadería más cercana para que luego le sea enviado el pedido, el cual será pagado contra factura.

### BENEFICIOS PARA EL CLIENTE

- Conseguir rápidamente el producto buscado.
- No necesita conocer a quien, o donde se encuentra ubicado el vendedor.
- Paga contra entrega.

### BENEFICIOS PARA LA PANADERÍA

- Incremento de las ventas y la rentabilidad.
- Ventas más frecuentes, mayor repetición de negocios con los mismos clientes.
- Un más alto nivel de ventas individuales a cada cliente, (los clientes satisfechos compran más de los mismos productos).
- Menores gastos en actividades de Marketing (publicidad, promoción de ventas y similares).
- Optimización de producción al publicar productos seleccionados.
- Facilidad de actualización de precios, ya que este es informado al agregar al carrito luego de que le fue asignado el pedido.

### 2.1 ALCANCE

Los clientes a través de la interfaz web pueden adquirir el producto deseado.

Las panaderías pueden ofrecer sus productos, pudiendo cambiar sus precios y radio de acción.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

### 3 ESTRATEGIAS DE PROGRAMACIÓN

#### 3.1 PROGRAMACIÓN

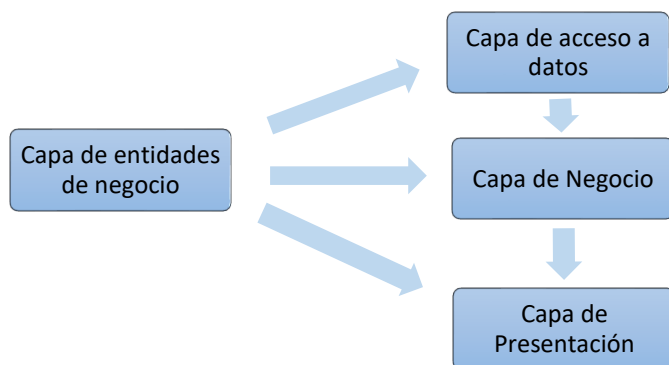
El sistema se encuentra dividido en 4 capas: capa de presentación, capa de negocio, capa de acceso a datos y una capa de entidades de negocio.

*Capa de presentación:* la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario. Se encuentra desarrollada con ASP.NET. Esta capa se comunica únicamente con la capa de negocio.

*Capa de negocio:* es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Se encuentra desarrollada en VB.NET

*Capa de acceso a datos:* es donde residen los datos y es la encargada de acceder a los mismos. Está formada por gestores que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Se utiliza ADO desconectado.

*Capa de entidades de negocio:* es donde entrarían todas las entidades o clases de la aplicación, las relaciones, atributos, colecciones de objetos del modelo conceptual.



#### 3.2 BASE DE DATOS

Se utiliza Microsoft SQL Server 2012

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

### 3.2.1 Tipo de conexión:

Se Utiliza ADO.NET Desconectado ya que cuenta con las siguientes ventajas:

- 1.- se puede trabajar con los datos de manera independiente de los otros usuarios
- 2.- las conexiones son pocas y consume menos recursos el servidor

### 3.2.2 Procedimientos almacenados

Se utilizan procedimientos almacenados, estos son programas almacenados en la base de datos, con la ventaja que al ser ejecutado en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos.

## 3.3 SEGURIDAD

### 3.3.1 Protección de contraseñas:

Para guardar las contraseñas en forma segura se utiliza el algoritmo de encriptación AES

### 3.3.2 Hashing

Se utiliza el algoritmo SHA-1 que ha sido examinado muy de cerca por la comunidad criptográfica pública, y no se ha encontrado ningún ataque efectivo, SHA-1 produce una salida resumen de 160 bits (20 bytes) de un mensaje que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados por el profesor Ronald L. Rivest del MIT en el diseño de los algoritmos de resumen de mensaje MD4 y MD5.

id	login	pass
1	administrador	9dbf7c1488382487931d10235fc84a74bff5d2f4
3	palemo	c84f35f9f4de4c55d6e68cdf5c1d4ae0f255cd65
2	webmaster	50f3f01caa053693ce619d596e14b0ff3901ab49

### 3.3.3 Algoritmo de encriptación

Se utilizará Advanced Encryption Standard (AES) es uno de los algoritmos de cifrado más utilizados y seguros actualmente disponibles. Es de acceso público, y es el cifrado que la NSA utiliza para asegurar documentos con la clasificación "top secret". Este utiliza una clave privada lo cual hace que esta encriptación sea única para la aplicación.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

### 3.3.4 Algoritmo de dígitos verificadores:

El sistema contendrá dos tipos de dígitos verificadores, un digito verificador *Horizontal* y *Vertical*. La función del digito verificador horizontal es verificar la integridad de los datos de cada registro en las tablas del sistema, el digito verificador vertical cumple la misma función, pero para controlar la integridad de la tabla. En resumen, con el horizontal se controlan que no se cambien datos y con el vertical que no se agreguen o borren registros.

El chequeo de la integridad de los datos se realizará cada vez que se inicia el sistema, en caso de encontrar un problema de consistencia, la misma se informará por pantalla al usuario mediante un mensaje de alerta.

El modo de funcionamiento consiste en calcular el valor hash del registro y que este sirva como evidencia para la verificación de si ha sido modificado.

*Horizontal:*

id	descripcion	imagen	DVH
1	Medialuna	1.png	4d696ac08ece69700303ccceabaa39b12cae8a63
2	Vigilante	2.png	834f94c6053a6c46d40df4c87b681bdca4f06a2d

*Vertical:*

id	tabla	DVV	DVH
1	Usuario	648d32f4333fb7d173c56d7cac7dfcc0704fa1b	74b7d1fd820bfc4ce561f0392f138742f05cdb0d
3	Producto	0cab4075ebec894352d6896a952237009e3614a9	a2f64c985d08c0c9a9982b81539036e734197ae0
6	Bitacora	73102fdddb320b8134f4044d2a0638e3da4c1848	247c78d7d443ebbc0b08c4e52d26b4e59ed22070



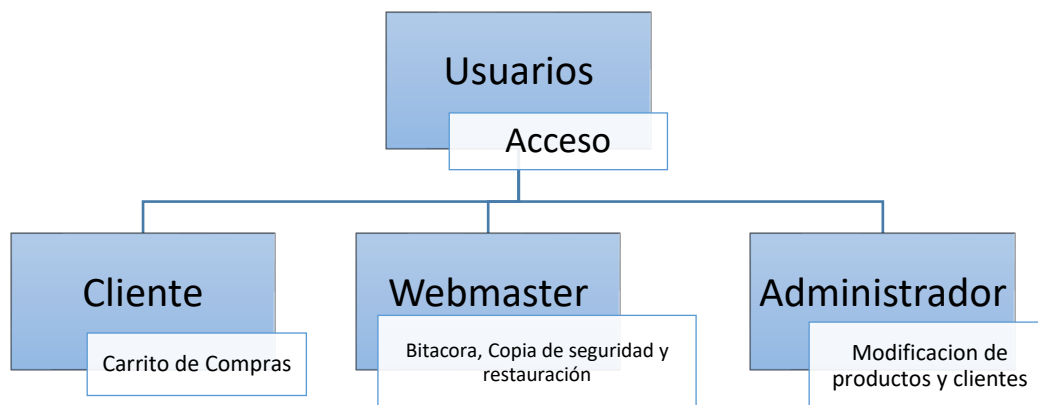
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 4 POLÍTICA DE PROGRAMACIÓN

### 4.1 METODOLOGÍA DE ACCESO Y SEGURIDAD:

#### 4.1.1 Usuarios de acceso:

Para acceder a la aplicación será necesario un nombre de usuario y una contraseña, esto permitirá identificar los siguientes Perfiles



#### 4.1.2 Descripción de perfiles:

*Cliente:* Tendrán los permisos necesarios para realizar pedidos a través de la página.

*Administrador:* Tendrá los permisos necesarios para acceder a las siguientes funcionalidades, de modificar información de productos y clientes.

*Webmaster:* Tendrá los permisos necesarios para acceder a las siguientes funcionalidades de Bitácora, Copia de seguridad y Restauración.

#### 4.1.3 Autenticación de usuarios:

Los clientes deberán registrarse en el sitio para poder realizar cualquier acción, la contraseña deberá cumplir con el requisito de tener mínimamente 8 caracteres.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

#### 4.1.4 Olvido de contraseña:

El usuario podrá acceder a la funcionalidad de recuperar contraseña desde la página de inicio de sesión. En esta página se le solicitara el usuario y el sistema generara una nueva contraseña de 8 caracteres en forma aleatoria.

Esta nueva contraseña será mostrada en pantalla y almacenada en la base de datos.

Usuario

Contraseña

Recuperar Contraseña

Su nueva pass es: hola1234

#### 4.1.5 Intentos fallidos y bloqueo de usuarios:

El usuario será bloqueado luego de 3 intentos fallidos, el *Webmaster* podrá acceder a una funcionalidad para desbloquear usuarios.

Usuario

Contraseña

3 intentos erroneos, usuario bloqueado

Recuperar Contraseña

#### 4.1.6 Plan de contingencia en caso de falla de integridad:

Si el usuario tiene el permiso *Webmaster* se le presenta una pantalla con la información pertinente, y opciones de recuperación.

Si el usuario es un cliente se le informará que el sitio se encuentra “En mantenimiento y regresará a la brevedad”

### 4.2 POLÍTICA DE COPIA DE SEGURIDAD:

La tarea de copia de seguridad es la de mantener una copia de resguardo de los datos que contiene el sistema.

El sistema tendrá una opción para realizar resguardo de la información y su posterior recupero.

Periodicidad de las Copias: se realiza bajo demanda desde la página destinada a tal fin.

Responsables: Usuarios con permisos de *Webmaster*.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## Perfil: Webmaster

### Copia de Seguridad y Restauración

Hacer Copia de seguridad

Ejecutar

Restaurar Copia de seguridad

Ejecutar

## 4.3 BITÁCORA

La bitácora debe registrar información sobre todos los eventos que estén relacionados con el sistema. El objetivo de la misma es servir de soporte y seguimiento para todas las operaciones que un usuario realiza en el sistema, brindar información para la detección y diagnóstico de problemas y garantizar el “no repudio”

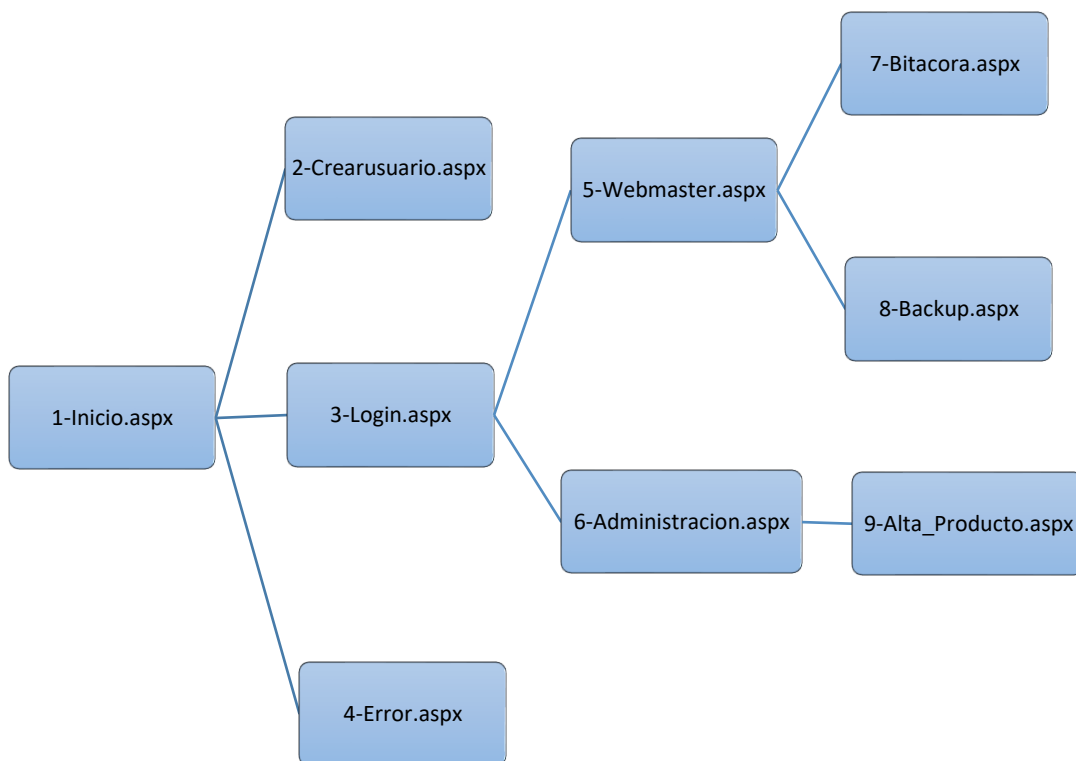
La bitácora proveerá información sobre los accesos realizados, usuario, acción, fecha y hora. También se incluirá en la bitácora, solicitudes de nuevas contraseñas y desbloqueo de usuarios.

90	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:29
91	webmaster	roberto piombi	Recalculo DV Vertical	23/11/2017 15:56:31
92	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:33
93	webmaster	roberto piombi	Verificación DV Horizontal	23/11/2017 15:56:35
94	webmaster	roberto piombi	Verificación DV Horizontal	23/11/2017 15:56:39
95	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:41
96	webmaster	roberto piombi	Se realizo Copia de seguridad	23/11/2017 15:56:46
97	webmaster	roberto piombi	Logueo de: Webmaster	23/11/2017 16:16:26
98	webmaster	roberto piombi	Logueo de: Webmaster	23/11/2017 16:35:53
99	webmaster	roberto piombi	Login erroneo intento: 1	23/11/2017 16:38:00
100	webmaster	roberto piombi	Logueo de: Webmaster	23/11/2017 16:38:04
101	webmaster	roberto piombi	Verificación DV Horizontal	23/11/2017 16:38:45
102	administrador	juan sanchez	Logueo de: Administrador	23/11/2017 16:42:53
103	webmaster	roberto piombi	Logueo de: Webmaster	23/11/2017 17:04:49

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

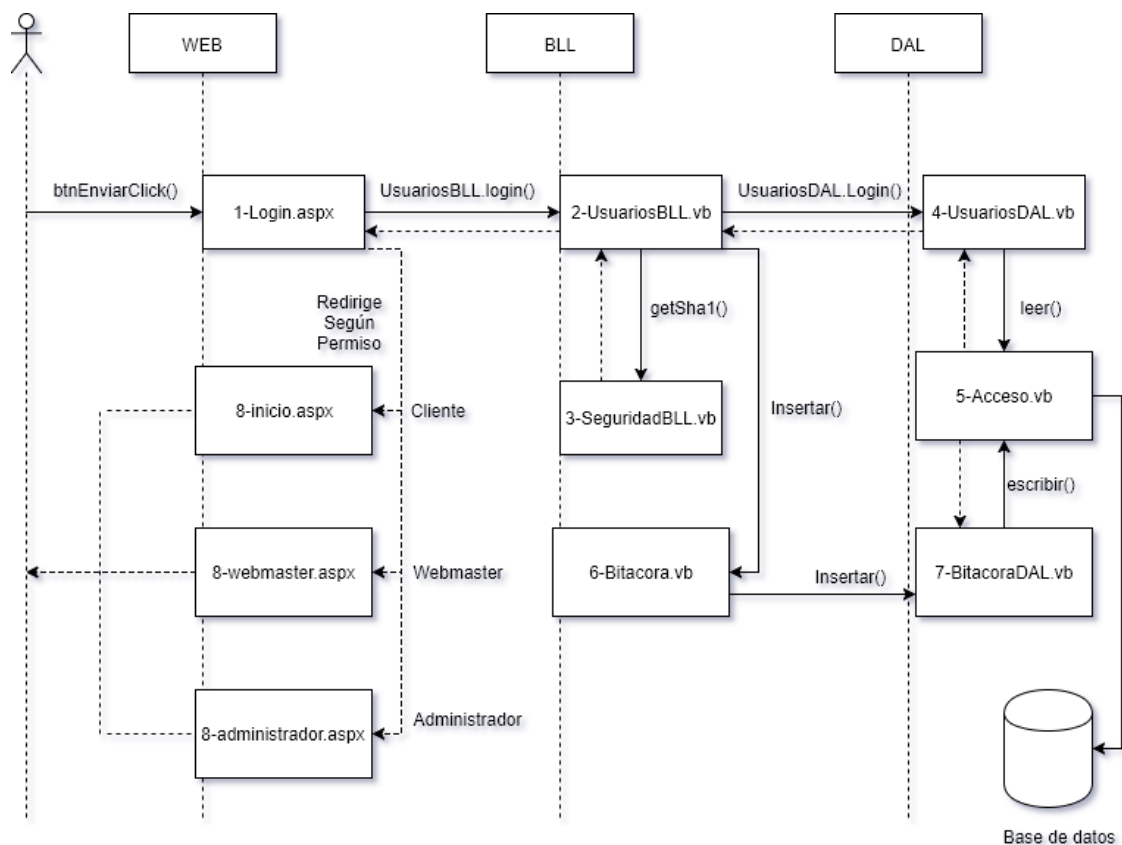
## 5 MAPA DEL SITIO

---



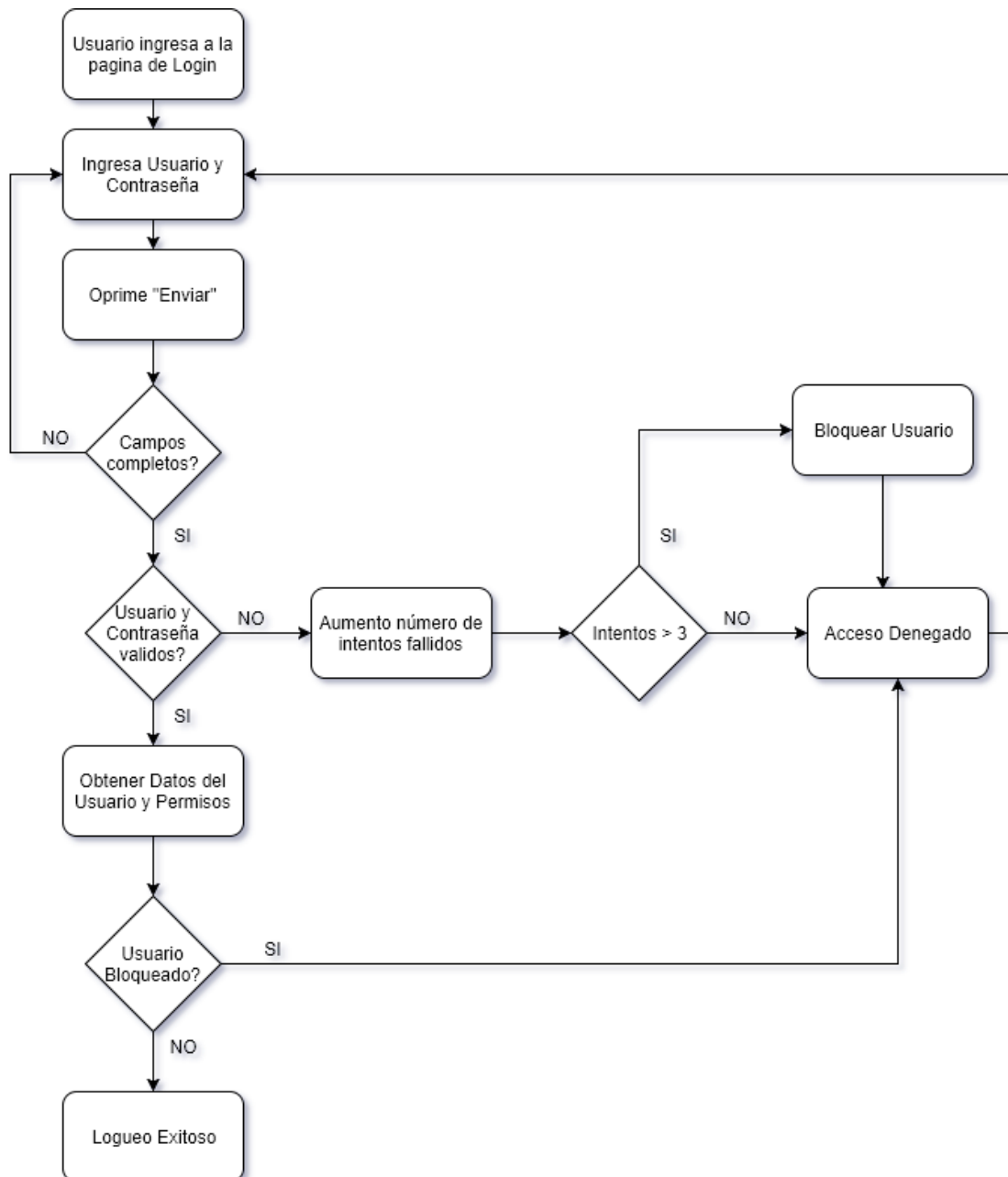
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017
Fecha 24/11/2017				

## 6 DIAGRAMA CICLO DE VIDA AUTENTICACIÓN



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017
Fecha 24/11/2017				

## 7 DIAGRAMA DE FLUJO























UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 8 CASO DE USO LOGIN

<b>NOMBRE:</b>	01 Login.
<b>DESCRIPCION:</b> Describe los pasos para la autenticación en el sistema.	
<b>ACTOR PRINCIPAL:</b> Usuario.	
<b>PRECONDICIONES:</b> El usuario debe estar creado.	
<b>CONDICION:</b>	
<b>ESCENARIO PRINCIPAL:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la funcionalidad de Login.</li> <li>2. El sistema muestra la pantalla de Login</li> <li>3. El usuario completa los campos "Usuario" y "Contraseña" y pulsa el botón "Enviar"</li> <li>4. El sistema valida que los campos no esten vacios.</li> <li>5. El sistema valida que el usuario sea un usuario valido.</li> <li>6. El sistema encripta la contraseña y valida que sea correcta.</li> <li>7. El sistema valida que el usuario tenga menos de 3 intentos fallidos.</li> <li>8. El sistema obtiene los permisos del usuario.</li> <li>9. El sistema redirecciona a la página correspondiente según su permiso.</li> </ol>	
<b>FLUJOS ALTERNATIVOS:</b> <p>A1 3. Cancelar el Logueo. El usuario es redireccionado a la pagina de inicio.</p> <p>A2 3. El usuario deja campos vacios. El sistema muestra un mensaje de error indicando los campos requeridos. El sistema vuelve al paso 2.</p> <p>A3 6. El sistema encuentra la contraseña es incorrecta El sistema muestra un mensaje de error indicándolo y aumenta en 1 los intentos fallidos. El sistema vuelve al paso 2.</p> <p>A4 7. El sistema encuentra que tiene más de 3 intentos fallidos. El sistema muestra un mensaje de error indicándolo. El sistema vuelve al paso 2.</p>	
<b>POSTCONDICIONES:</b>  El sistema redirecciona a la página correspondiente de acuerdo a su permiso.	

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9 CAPA DE PRESENTACIÓN

-  **WEB**
  -  My Project
  - ▷  Referencias
  - ▷  estilos
  - ▷  imagenes
  - ▷  imgProductos
  -  scripts
  -  administrador.aspx
  -  alta\_producto.aspx
  -  backup.aspx
  - ▷  Base.vb
  -  bitacora.aspx
  -  ClassDiagram1.cd
  -  crearUsuario.aspx
  -  error.aspx
  -  inicio.aspx
  -  login.aspx
  -  packages.config
  -  Web.config
  -  webmaster.aspx



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9.1 INICIO.ASPX



```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="inicio.aspx.vb"
Inherits="MedialunasTp.inicio" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
<tr>
<td class="aLeft">
<asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
<asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
</td>
<td class="aRight">
<asp:Label runat="server" Text="Usuario: "
ID="lblUsu"></asp:Label>
<asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
<asp:Button ID="btnLogin" runat="server" Text="Login"
Width="100px" />

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <asp:Button ID="btnCrearUsuario" runat="server" Text="Crear
Usuarios" />
        <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Visible="False" Width="100px" />
    </td>
</tr>
</table>

```

```

    <br />
    <asp:Label runat="server" ID="lblPermiso" text="Listado de Productos"
Font-Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
    <br />
    <asp:Table ID="tablaProductos" class="tablaProductos"
runat="server"></asp:Table>
</div>

```

```

    </form>
</body>
</html>

```

Imports BE, BLL

```

Public Class inicio
    Inherits System.Web.UI.Page

```

```

    Dim usuario As Usuario

```

```

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

```

```

        verificarIntegridad()

```

```

        Dim login As String = Session("Login")

```

```

        If Not String.IsNullOrEmpty(login) Then
            usuario = UsuariosBLL.buscar(login)
            lblUsuario.Text = usuario.ToString
            lblUsu.Visible = True
            lblUsuario.Visible = True
            btnLogin.Visible = False
            btnCrearUsuario.Visible = False
            btnCerrar.Visible = True

```

```

        Else
            usuarioNoLogueado()

```

```

        End If
        cargarTablaProductos()

```

```

    End Sub

```

```

    Protected Sub btnLogin_Click(sender As Object, e As EventArgs) Handles
btnLogin.Click
        Response.Redirect("login.aspx")
    End Sub

```

```

    Protected Sub btnCerrar_Click(sender As Object, e As EventArgs) Handles
btnCerrar.Click
        usuarioNoLogueado()

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Sub

```
Private Sub usuarioNoLogueado()
    Session("Login") = ""
    usuario = Nothing
    lblUsu.Visible = False
    lblUsuario.Visible = False
    btnLogin.Visible = True
    btnCrearUsuario.Visible = True
    btnCerrar.Visible = False
End Sub
```

```
Private Sub verificarIntegridad()
    If WebmasterBLL.verificarAInicio Then
        Response.Redirect("error.aspx")
    End If
End Sub
```

```
Private Sub cargarTablaProductos()

    lblPermiso.Text = "Listado de Productos"
    Dim lista As List(Of BE.Producto) = ProductosBLL.listar

    For Each item In lista
        Dim fila As New TableRow

        Dim id As New TableCell
        id.Text = item.id

        Dim descripcion As New TableCell
        descripcion.Text = item.descripcion

        Dim imagen As New TableCell
        Dim img As New Image
        img.ImageUrl = "~/imgProductos/" & item.imagen
        img.Height = 40
        imagen.Controls.Add(img)

        'fila.Cells.Add(id)
        fila.Cells.Add(imagen)
        fila.Cells.Add(descripcion)
        tablaProductos.Rows.Add(fila)
    Next
End Sub
```

```
Protected Sub btnCrearUsuario_Click(sender As Object, e As EventArgs) Handles
btnCrearUsuario.Click
    Response.Redirect("crearUsuario.aspx")
End Sub
End Class
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9.2 LOGIN.ASPX

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="login.aspx.vb"
Inherits="MedialunasTp.login" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
<tr>
<td class="aLeft">
<asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
<asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
</td>
<td>&nbsp;</td>
</tr>
</table>
<br />
<br />
<table style="width:620px;" class="center">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td>Usuario</td>
<td>
<asp:TextBox ID="txtUsuario" runat="server"
Width="300px"></asp:TextBox>
</td>
</tr>
</table>
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="txtUsuario" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>Contraseña</td>
    <td>
        <asp:TextBox ID="txtPass" runat="server" TextMode="Password"
Width="300px"></asp:TextBox>
    </td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ControlToValidate="txtPass" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td class="auto-style1"></td>
    <td class="auto-style1">
        <asp:Label ID="lblError" runat="server"
ForeColor="Red"></asp:Label>
    </td>
    <td class="auto-style1">
        <asp:Button ID="btnCancelar" runat="server" Text="Cancelar"
Width="100px" CausesValidation="False" />
        &nbsp;
        <asp:Button ID="btnEnviar" runat="server" Text="Enviar"
Width="100px" />
    </td>
</tr>
<tr>
    <td></td>
    <td>
        <asp:Button ID="btnRecuperar" runat="server" Text="Recuperar
Contraseña" BorderStyle="None" CausesValidation="False" />
    </td>
    <td>
        <asp:Label ID="lblNueva" runat="server" Text=""></asp:Label>
    </td>
</tr>
</table>

</div>
</form>
</body>
</html>

```

```

Imports BLL, BE
Public Class login
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles Me.Load
        Page.Form.DefaultButton = btnEnviar.ID
    End Sub
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Sub

```
Protected Sub btnCancelar_Click(sender As Object, e As EventArgs) Handles
btnCancelar.Click
    Session("Login") = ""
    Response.Redirect("inicio.aspx")
End Sub
```

```
Private Sub irA(param As Permiso)
    If param Is Nothing Then
        Response.Redirect("inicio.aspx")
    ElseIf BLL.PermisosBLL.Admin.Equals(param) Then
        Response.Redirect("administrador.aspx")
    ElseIf BLL.PermisosBLL.Web.Equals(param) Then
        Response.Redirect("webmaster.aspx")
    End If
End Sub
```

```
Protected Sub btnRecuperar_Click(sender As Object, e As EventArgs) Handles
btnRecuperar.Click
    Dim usuario As New Usuario
    Dim nueva As String = UsuariosBLL.generarPass(txtUsuario.Text)
    lblNueva.Text = "Su nueva pass es: " & nueva
End Sub
```

```
Protected Sub btnEnviar_Click(sender As Object, e As EventArgs) Handles
btnEnviar.Click
    Dim usuario As New Usuario
    usuario.login = txtUsuario.Text
    usuario.pass = txtPASS.Text
    Dim usuarioCompleto As Usuario = UsuariosBLL.login(usuario)
    If usuarioCompleto IsNot Nothing Then
        If usuarioCompleto.intentosLogin < 3 Then
            Session("Login") = usuarioCompleto.login
            FormsAuthentication.RedirectFromLoginPage(usuario.login, False)
            irA(usuarioCompleto.permiso)
        Else
            lblError.Text = "3 intentos erroneos, usuario bloqueado"
        End If
    Else
        lblError.Text = "No se encontro el usuario o la Contraseña es
erronea"
    End If
End Sub
End Class
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

### 9.3 CREARUSUARIO.ASPX



```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="crearUsuario.aspx.vb"
Inherits="MedialunasTp.crearUsuario" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

```
<title>>Medialunas SA</title>
```

```
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<table style="width: 100%;" class="center">
```

```
<tr>
```

```
<td class="aLeft">
```

```
<asp:Image ID="logo1" runat="server"
```

```
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
```

```
<asp:Image ID="logoTXT" runat="server"
```

```
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
```

```
</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
</table>
```

```
<br />
```

```
<br />
```

```
<table style="width:620px;" class="center">
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Usuario</td>
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <td>
            <asp:TextBox ID="txtUsuario" runat="server"
Width="300px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="txtUsuario" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td>Contraseña</td>
        <td>
            <asp:TextBox ID="txtPass" runat="server" TextMode="Password"
Width="300px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ControlToValidate="txtPass" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator ControlToValidate = "txtPass"
ID="RegularExpressionValidator2" ValidationExpression = "[\s\S]{8,}"
runat="server" ErrorMessage="Minimo 8 caracteres"
ForeColor="Red"></asp:RegularExpressionValidator>
        </td>
    </tr>
    <tr>
        <td>Nombre</td>
        <td>
            <asp:TextBox ID="txtNombre" runat="server"
Width="300px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ControlToValidate="txtNombre" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td>Apellido</td>
        <td>
            <asp:TextBox ID="txtApellido" runat="server"
Width="300px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ControlToValidate="txtApellido" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td>Correo</td>
        <td>
            <asp:TextBox ID="txtCorreo" runat="server"
Width="300px"></asp:TextBox>
        </td>

```



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server" ControlToValidate="txtCorreo" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server" ControlToValidate="txtCorreo"
ErrorMessage="Correo invalido" ForeColor="Red" ValidationExpression="\w+([-
+.' ]\w+)*@\w+([-.' ]\w+)*\. \w+([-.' ]\w+)*"></asp:RegularExpressionValidator>
        </td>
    </tr>
    <tr>
        <td class="auto-style1"></td>
        <td class="auto-style1">
            <asp:Label ID="lblError" runat="server"
ForeColor="Red"></asp:Label>
        </td>
        <td class="auto-style1">
            <asp:Button ID="btnCancelar" runat="server" Text="Cancelar"
Width="100px" CausesValidation="False" />
            &nbsp;
            <asp:Button ID="Button1" runat="server" Text="Crear"
Width="100px" />
        </td>
    </tr>
</table>

</div>
</form>
</body>
</html>

```

```

Imports BLL, BE
Public Class crearUsuario
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        Page.Form.DefaultButton = Button1.ID
    End Sub

    Protected Sub btnCancelar_Click(sender As Object, e As EventArgs) Handles
btnCancelar.Click
        Response.Redirect("inicio.aspx")
    End Sub

    Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim usuario As New Usuario
        usuario.login = txtUsuario.Text
        usuario.pass = txtPass.Text
        usuario.nombre = txtNombre.Text
        usuario.apellido = txtApellido.Text
        usuario.correo = txtCorreo.Text
        Dim resultado = UsuariosBLL.crearUsuario(usuario)
        If resultado = 1 Then

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        Session("Login") = usuario.login
    End If
    Response.Redirect("inicio.aspx")
End Sub
End Class

```

## 9.4 WEBMASTER.ASPX

```

<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="webmaster.aspx.vb"
Inherits="MedialunasTp.webmaster" %>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Medialunas SA</title>
    <link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

<form id="form1" runat="server">
  <div>
    <table style="width: 100%;" class="center">
      <tr>
        <td class="aLeft">
          <asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
          <asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
        </td>
        <td class="aRight">
          <asp:Label runat="server" Text="Usuario: "
ID="Label1"></asp:Label>
          <asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
          &nbsp;
          <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
        </td>
      </tr>
    </table>

    <br />
    <asp:Label runat="server" ID="lblPermiso" Text="Perfil: Webmaster"
Font-Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
    <br />
    <br />
    <br />
    <br />
    <br />
    <table style="width: 40%;" class="center">
      <tr>
        <td>
          <asp:Label ID="Label5" runat="server" Text="Ver
bitacora"></asp:Label>
        </td>
        <td>
          <asp:Button ID="btnBitacora" runat="server"
Text="Ejecutar" />
        </td>
      </tr>
      <tr>
        <td>
          <asp:Label ID="Label2" runat="server" Text="Copia de
seguridad y Restauración"></asp:Label>
        </td>
        <td>
          <asp:Button ID="btnBack" runat="server" Text="Ejecutar"
/>
        </td>
      </tr>
      <tr>
        <td>
          <asp:Label ID="Label3" runat="server" Text="Verificar
integridad Horizontal"></asp:Label>
        </td>

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <td>
            <asp:Button ID="btnVerificarH" runat="server"
Text="Ejecutar" />
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label4" runat="server" Text="Recalcular
Digitos Horizontales"></asp:Label>
        </td>
        <td>
            <asp:Button ID="btnRecalcularH" runat="server"
Text="Ejecutar" />
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label6" runat="server" Text="Verificar
integridad Vertical"></asp:Label>
        </td>
        <td>
            <asp:Button ID="btnVerificarV" runat="server"
Text="Ejecutar" />
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label7" runat="server" Text="Recalcular
Digitos Verticales"></asp:Label>
        </td>
        <td>
            <asp:Button ID="btnRecalcularV" runat="server"
Text="Ejecutar" />
        </td>
    </tr>
</table>

<br />
<br />

<table style="width: 600px;" class="center">
    <tr>
        <td>
            <asp:TextBox ID="txtResultado" runat="server"
ReadOnly="True" TextMode="Multiline" Width="600px" Height="200px"></asp:TextBox>
        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

```

Imports BLL, BE
Public Class webmaster

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

Inherits Base

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    UsuarioLogueado(BLL.PermisosBLL.Web)
    lblUsuario.Text = usuario.ToString
End Sub

Protected Sub btnCerrar_Click(sender As Object, e As EventArgs) Handles
btnCerrar.Click
    usuarioNoLogueado()
End Sub

Protected Sub btnBitacora_Click(sender As Object, e As EventArgs) Handles
btnBitacora.Click
    Response.Redirect("bitacora.aspx")
End Sub

Protected Sub btnBack_Click(sender As Object, e As EventArgs) Handles
btnBack.Click
    Response.Redirect("backup.aspx")
End Sub

Protected Sub btnVerificarH_Click(sender As Object, e As EventArgs) Handles
btnVerificarH.Click
    txtResultado.Text = WebmasterBLL.verificarDVHBase(usuario)
End Sub

Protected Sub btnRecalcularH_Click(sender As Object, e As EventArgs) Handles
btnRecalcularH.Click
    txtResultado.Text = WebmasterBLL.recalcularDVHBase(usuario)
End Sub

Protected Sub btnVerificarV_Click(sender As Object, e As EventArgs) Handles
btnVerificarV.Click
    txtResultado.Text = WebmasterBLL.verificarDVVerticalBase(usuario)
End Sub

Protected Sub btnRecalcularV_Click(sender As Object, e As EventArgs) Handles
btnRecalcularV.Click
    txtResultado.Text = WebmasterBLL.recalcularDVVerticalBase(usuario)
End Sub
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9.5 BITACORA.ASPX


**Medialunas SA**

Usuario: roberto piombi

**Perfil: Webmaster**

**Bitacora**

90	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:29
91	webmaster	roberto piombi	Recalculo DV Vertical	23/11/2017 15:56:31
92	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:33
93	webmaster	roberto piombi	Verificación DV Horizontal	23/11/2017 15:56:35
94	webmaster	roberto piombi	Verificación DV Horizontal	23/11/2017 15:56:39
95	webmaster	roberto piombi	Verificación DV Vertical	23/11/2017 15:56:41
		roberto	Se realizo	23/11/2017

```

<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="bitacora.aspx.vb"
Inherits="MedialunasTp.bitacora" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
<tr>
<td class="aLeft">
<asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
<asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
</td>
<td class="aRight">
<asp:Label runat="server" Text="Usuario: "
ID="Label1"></asp:Label>
<asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
&nbsp;

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
    </td>
</tr>
</table>

<br />
<asp:Label runat="server" ID="lblPermiso" text="Perfil: Webmaster" Font-
Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
<br />
<br />
<asp:Label runat="server" ID="Label2" text="Bitacora" Font-Bold="True"
Font-Size="Medium" CssClass="aCenter"></asp:Label>
<br />
<br />
<asp:Table ID="tablaBitacora" class="tablaBitacora"
runat="server"></asp:Table>
</div>
</form>
</body>
</html>

```

Imports BLL, BE

```

Public Class bitacora
    Inherits Base

```

```

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        UsuarioLogueado(BLL.PermisosBLL.Web)
        lblUsuario.Text = usuario.ToString
        cargarBitacora()
    End Sub

```

```

    Protected Sub btnCerrar_Click(sender As Object, e As EventArgs) Handles
btnCerrar.Click
        usuarioNoLogueado()
    End Sub

```

```

    Private Sub cargarBitacora()
        Dim lista As List(Of BE.Bitacora) = BLL.BitacoraBLL.listar

```

```

    For Each item In lista
        Dim fila As New TableRow

```

```

        Dim id As New TableCell
        Dim usuario As New TableCell
        Dim usuario2 As New TableCell
        Dim accion As New TableCell
        Dim fecha As New TableCell

```

```

        id.Text = item.id
        usuario.Text = item.usuario.login
        usuario2.Text = item.usuario.ToString
        accion.Text = item.accion

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```
fecha.Text = item.fecha.ToString("dd/MM/yyyy H:mm:ss")
```

```
fila.Cells.Add(id)
fila.Cells.Add(usuario)
fila.Cells.Add(usuario2)
fila.Cells.Add(accion)
fila.Cells.Add(fecha)
tablaBitacora.Rows.Add(fila)
```

```
Next
```

```
End Sub
```

```
End Class
```

## 9.6 BACKUP.ASPX



```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="backup.aspx.vb"
Inherits="MedialunasTp.backup" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Medialunas SA</title>
  <link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <table style="width: 100%;" class="center">
        <tr>
          <td class="aLeft">
            <asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
            <asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
          </td>
          <td class="aRight">
```



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <asp:Label runat="server" Text="Usuario: "
ID="Label1"></asp:Label>
        <asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
        &nbsp;
        <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
    </td>
</tr>
</table>

<br />
<asp:Label runat="server" ID="lblPermiso" Text="Perfil: Webmaster"
Font-Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
<br />
<br />
<asp:Label runat="server" ID="Label2" Text="Copia de Seguridad y
Restauración" Font-Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
<br />
<br />
<br />
<br />
<br />
<table style="width: 40%;" class="center">
    <tr>
        <td>
            <asp:Label ID="Label5" runat="server" Text="Hacer Copia
de seguridad"></asp:Label>
        </td>
        <td>
            <asp:Button ID="btnBack" runat="server" Text="Ejecutar"
/>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label3" runat="server" Text="Restaurar
Copia de seguridad"></asp:Label>
        </td>
        <td>
            <asp:Button ID="btnRestaurar" runat="server"
Text="Ejecutar" />
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <asp:Label ID="lblmensaje" runat="server"
ForeColor="Red"></asp:Label>
        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

Imports BE, BLL

Public Class backup

Inherits Base

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

UsuarioLogueado(BLL.PermisosBLL.Web)

lblUsuario.Text = usuario.ToString

End Sub

Protected Sub btnCerrar\_Click(sender As Object, e As EventArgs) Handles btnCerrar.Click

usuarioNoLogueado()

End Sub

Protected Sub btnBack\_Click(sender As Object, e As EventArgs) Handles btnBack.Click

lblmensaje.Text = WebmasterBLL.hacerBackup(usuario)

End Sub

Protected Sub btnRestaurar\_Click(sender As Object, e As EventArgs) Handles btnRestaurar.Click

lblmensaje.Text = WebmasterBLL.restaurarBackup(usuario)

End Sub

End Class

## 9.7 ADMINISTRADOR.ASPX

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="administrador.aspx.vb"
Inherits="MedialunasTp.administrador" %>
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <tr>
            <td class="aLeft">
                <asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
                <asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
            </td>
            <td class="aRight">
                <asp:Label runat="server" Text="Usuario: "
ID="Label1"></asp:Label>
                <asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
                &nbsp;
                <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
            </td>
        </tr>
    </table>

    <br />
    <asp:Label runat="server" ID="lblPermiso" text="Perfil: Administrador"
Font-Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>

    <br />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Alta de Productos" />

</div>
</form>
</body>
</html>

```

```

Imports BLL, BE
Public Class administrador
    Inherits Base

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        UsuarioLogueado(BLL.PermisosBLL.Admin)
        lblUsuario.Text = usuario.ToString
    End Sub

    Protected Sub btnCerrar_Click(sender As Object, e As EventArgs) Handles
btnCerrar.Click
        usuarioNoLogueado()
    End Sub

    Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Response.Redirect("alta_producto.aspx")
    End Sub
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9.8 ALTA\_PRODUCTO.ASPX

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="alta_producto.aspx.vb"
Inherits="MedialunasTp.producto" %>
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
<tr>
<td class="aLeft">
<asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
<asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
</td>
<td class="aRight">
<asp:Label runat="server" Text="Usuario: "
ID="Label1"></asp:Label>
<asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
<br />
<asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
</td>
</tr>
</table>
<br />
<asp:Label runat="server" ID="Label2" text="Perfil: Administrador" Font-
Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

<br />
<br />
<asp:Label runat="server" ID="lblPermiso" text="Carga de Productos" Font-
Bold="True" Font-Size="Medium" CssClass="aCenter"></asp:Label>

<br />
<br />
<table style="width:620px;" class="center">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td>Descripcion</td>
<td>
<asp:TextBox ID="txtDesc" runat="server"
Width="300px"></asp:TextBox>
</td>
<td>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="txtDesc" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td>Imagen</td>
<td>
<asp:TextBox ID="txtImg" runat="server"
Width="300px"></asp:TextBox>
</td>
<td>
<asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ControlToValidate="txtImg" ErrorMessage="Campo Requerido"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td class="auto-style1"></td>
<td class="auto-style1">
<asp:Label ID="lblError" runat="server"
ForeColor="Red"></asp:Label>
</td>
<td class="auto-style1">
<asp:Button ID="btnCancelar" runat="server" Text="Cancelar"
Width="100px" CausesValidation="False" />
&nbsp;<br>
<asp:Button ID="btnEnviar" runat="server" Text="Aceptar"
Width="100px" />
</td>
</tr>
</table>
</div>
</form>
</body>
</html>

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

Public Class producto

Inherits Base

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

UsuarioLogueado(BLL.PermisosBLL.Admin)

lblUsuario.Text = usuario.ToString

End Sub

Protected Sub btnCerrar\_Click(sender As Object, e As EventArgs) Handles btnCerrar.Click

usuarioNoLogueado()

End Sub

Protected Sub btnEnviar\_Click(sender As Object, e As EventArgs) Handles btnEnviar.Click

Dim obj = New BE.Producto

obj.descripcion = txtDesc.Text

obj.imagen = txtImg.Text

BLL.ProductosBLL.insertarProducto(usuario, obj)

Response.Redirect("administrador.aspx")

End Sub

Protected Sub btnCancelar\_Click(sender As Object, e As EventArgs) Handles btnCancelar.Click

Response.Redirect("administrador.aspx")

End Sub

End Class

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 9.9 ERROR.ASPX



```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="error.aspx.vb"
Inherits="MedialunasTp._error" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Medialunas SA</title>
<link href="/estilos/estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="width: 100%;" class="center">
<tr>
<td class="aLeft">
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        <asp:Image ID="logo1" runat="server"
ImageUrl="~/imagenes/logo.png" Height="30" ImageAlign="Middle" />
        <asp:Image ID="logoTXT" runat="server"
ImageUrl="~/imagenes/logoTXT.png" ImageAlign="Middle" />
    </td>
    <td class="aRight">
        <asp:Label runat="server" Text="Usuario: "
ID="lblUsu"></asp:Label>
        <asp:Label ID="lblUsuario" runat="server"
Text="usuario"></asp:Label>
        &nbsp;
        <asp:Button ID="btnLogin" runat="server" Text="Login"
Width="100px" />
        <asp:Button ID="btnCerrar" runat="server" Text="Cerrar"
Width="100px" />
    </td>
</tr>
</table>

<br />

<br />
<table style="width: 650px;" class="center">
    <tr>
        <td>
            <asp:Image ID="Image1" runat="server"
ImageUrl="~/imagenes/mantenimiento.png" ImageAlign="Middle" />
        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

Imports BE, BLL

```

Public Class _error
    Inherits System.Web.UI.Page

```

```

    Dim usuario As Usuario

```

```

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

```

```

        Dim login As String = Session("Login")

```

```

        If Not String.IsNullOrEmpty(login) Then
            usuario = UsuariosBLL.buscar(login)
            lblUsuario.Text = usuario.ToString
            lblUsu.Visible = True
            btnLogin.Visible = False
            lblUsuario.Visible = True
            btnCerrar.Visible = True

```

```

        Else

```



UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

```

        usuarioNoLogueado()
    End If
End Sub

Private Sub usuarioNoLogueado()
    Session("Login") = ""
    usuario = Nothing
    lblUsu.Visible = False
    lblUsuario.Visible = False
    btnCerrar.Visible = False
    btnLogin.Visible = True
End Sub

Protected Sub btnLogin_Click(sender As Object, e As EventArgs) Handles
btnLogin.Click
    Response.Redirect("login.aspx")
End Sub

Protected Sub btnCerrar_Click(sender As Object, e As EventArgs) Handles
btnCerrar.Click
    usuarioNoLogueado()
End Sub
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 10 CAPA DE NEGOCIO

- VB BLL
  - My Project
  - Referencias
  - VB BitacoraBLL.vb
  - VB CarritoBLL.vb
  - ClassDiagram1.cd
  - VB PermisosBLL.vb
  - VB ProductosBLL.vb
  - VB SeguridadBLL.vb
  - VB UsuariosBLL.vb
  - VB WebmasterBLL.vb

### 10.1 BITACORABLL.VB

Imports BE, DAL

Public Class BitacoraBLL

```
Public Shared Function listar() As List(Of BE.Bitacora)
    Return BitacoraDAL.Listar
End Function
```

```
Public Shared Function Insertar(param As BE.Bitacora) As Integer
    param.fecha = DateTime.Now
    param.DVH = SeguridadBLL.calcularDVH(param)
    Dim resultado As Integer = BitacoraDAL.Insertar(param)
    recalcularDVVertical()
    Return resultado
End Function
```

```
Public Shared Function Insertar(usuario As Usuario, accion As String) As Integer
    Dim bi As New Bitacora
    bi.usuario = usuario
    bi.accion = accion
    Return Insertar(bi)
End Function
```

```
Public Shared Function verificarDVH() As String
    Dim resultado As String = ""
    For Each item In listar()
        If Not SeguridadBLL.verificarDVH(item) Then
            resultado += "Error en el registro: " & item.id &
Environment.NewLine
        End If
    Next
    If String.IsNullOrEmpty(resultado) Then
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        Return "Ok"
    End If
    Return resultado
End Function

Public Shared Function verificarDVVertical() As String
    Dim digito As String = ""
    For Each item In listar()
        digito += item.DVH
    Next
    Return SeguridadBLL.getSHA1(digito)
End Function

Public Shared Function recalcularDVVertical() As String
    Return SeguridadBLL.insertarDVVertical("Bitacora", verificarDVVertical())
End Function

```

End Class

## 10.2 CARRITO BLL.VB

```

Imports BE, DAL
Public Class CarritoBLL

    Private _listaProductos As New List(Of Item)

    Public Shared Function listarPanaderias() As List(Of Panaderia)
        Return PanaderiasDAL.Listar
    End Function

    Public Function confirmarCompra(param As Pedidos) As Integer
        Return 0
    End Function

    Public Function armarPedido(usuario As Usuario) As Pedidos
        Dim pedido As New Pedidos
        pedido.usuario = usuario
        pedido.items = _listaProductos
        pedido.panaderia = ubicarPanaderiaCercana()
        Return pedido
    End Function

    Public Sub agregarItemCarrito(producto As Producto, precio As Single,
cantidad As Integer)
        Dim item = New Item
        item.producto = producto
        item.cantidad = cantidad
        item.precio = precio
        _listaProductos.Add(item)
    End Sub

    Private Function ubicarPanaderiaCercana() As Panaderia
        'Temporal

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Dim pan As New Panaderia
pan.nombre = "Panaderia 1"
pan.ubicacion = 1

Return PanaderiasDAL.Buscar(pan)
End Function

End Class

```

### 10.3 PERMISOSBLL.VB

```

Imports BE, DAL
Public Class PermisosBLL

    Private Shared _permisoAdmin As Permiso
    Private Shared _permisoWeb As Permiso

    Public Shared Function Admin() As Permiso
        If _permisoAdmin Is Nothing Then
            _permisoAdmin = PermisosDAL.permisoAdmin()
        End If
        Return _permisoAdmin
    End Function

    Public Shared Function Web() As Permiso
        If _permisoWeb Is Nothing Then
            _permisoWeb = PermisosDAL.permisoWeb()
        End If
        Return _permisoWeb
    End Function
End Class

```

### 10.4 PRODUCTOSBLL.VB

```

Imports BE, DAL
Public Class ProductosBLL

    Public Shared Function listar() As List(Of Producto)
        Return ProductosDAL.Listar
    End Function

    Public Shared Function insertarProducto(usuario As Usuario, param As Producto)
        param.DVH = SeguridadBLL.calcularDVH(param)
        Dim resultado As Integer = ProductosDAL.Insertar(param)
        If resultado > 0 Then
            BitacoraBLL.Insertar(usuario, "Producto Ins: " & param.descripcion)
        End If
        recalcularDVVertical()
        Return resultado
    End Function

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Public Shared Function verificarDVH() As String
    Dim resultado As String = ""
    For Each item In listar()
        If Not SeguridadBLL.verificarDVH(item) Then
            resultado += "Error en el registro: " & item.id &
Environment.NewLine
        End If
    Next
    If String.IsNullOrEmpty(resultado) Then
        Return "Ok"
    End If
    Return resultado
End Function

Public Shared Function recalcularDV() As String
    Dim cantidadModificados As Integer = 0
    For Each item In listar()
        item.DVH = SeguridadBLL.calcularDVH(item)
        cantidadModificados += ProductosDAL.Modificar(item)
    Next
    Return cantidadModificados
End Function

Public Shared Function verificarDVVertical() As String
    Dim digito As String = ""
    For Each item In listar()
        digito += item.DVH
    Next
    Return SeguridadBLL.getSHA1(digito)
End Function

Public Shared Function recalcularDVVertical() As String
    Return SeguridadBLL.insertarDVVertical("Producto", verificarDVVertical())
End Function

End Class

```

## 10.5 SEGURIDADBLL.VB

```

Imports BE, DAL, System.Security.Cryptography
Public Class SeguridadBLL

    Private Shared pass As String = "lamedialuna"

    Public Shared Function getSHA1(ByVal param As String) As String

        Dim sha1obj As New SHA1CryptoServiceProvider
        Dim hacerHash() As Byte = System.Text.Encoding.ASCII.GetBytes(param)
        hacerHash = sha1obj.ComputeHash(hacerHash)
        Dim resultado As String = ""
        For Each b As Byte In hacerHash
            resultado += b.ToString("x2")
        Next
    End Function
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Next
Return resultado
End Function

Private Shared Function getSHA1Hash(ByVal param As String) As Byte()
    Dim sha1obj As New SHA1CryptoServiceProvider
    Dim hash() As Byte = System.Text.Encoding.ASCII.GetBytes(param)
    hash = sha1obj.ComputeHash(hash)
    Return hash
End Function

Public Shared Function AES_Encrypt(ByVal input As String) As String
    Dim AES As New System.Security.Cryptography.RijndaelManaged
    Dim Hash_AES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim encrypted As String = ""
    Try
        AES.Key = getSHA1Hash(pass)
        AES.Mode = Security.Cryptography.CipherMode.CBC
        Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateEncryptor
        Dim Buffer As Byte() =
System.Text.Encoding.ASCII.GetBytes(input)
        encrypted =
Convert.ToBase64String(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
        Return encrypted
    Catch ex As Exception
        Return Nothing
    End Try
End Function

Public Shared Function AES_Decrypt(ByVal input As String) As String
    Dim AES As New System.Security.Cryptography.RijndaelManaged
    Dim Hash_AES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim decrypted As String = ""
    Try
        AES.Key = getSHA1Hash(pass)
        AES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateDecryptor
        Dim Buffer As Byte() = Convert.FromBase64String(input)
        decrypted =
System.Text.Encoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer
, 0, Buffer.Length))
        Return decrypted
    Catch ex As Exception
        Return Nothing
    End Try
End Function

Public Shared Function calcularDVH(param As BE.DigitoVerificador)
    Return getSHA1(param.verificar)
End Function

Public Shared Function verificarDVH(param As BE.DigitoVerificador) As Boolean

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Dim verificar As String = calcularDVH(param)
Return param.DVH.Equals(verificar)
End Function

```

```

Public Shared Function insertarDVVertical(tabla As String, digito As String)
As String
    Dim dv As New DigitoVerificadorVertical
    dv.tabla = tabla
    dv.DVV = digito
    dv.DVH = SeguridadBLL.calcularDVH(dv)
    DigitoVerificadorDAL.Modificar(dv)
    Return digito
End Function
End Class

```

## 10.6 USUARIOSBLL.VB

```
Imports BE, DAL
```

```
Public Class UsuariosBLL
```

```

Public Shared Function listar() As List(Of Usuario)
    Return UsuariosDAL.Listar
End Function

```

```

Public Shared Function login(param As Usuario) As Usuario
    param.pass = SeguridadBLL.getSHA1(param.pass)
    Dim usuario As Usuario = UsuariosDAL.Login(param)
    Dim existe As Usuario
    If usuario Is Nothing Then
        existe = UsuariosDAL.Buscar(param)
        If existe IsNot Nothing Then
            existe.intentosLogin += 1
            UsuariosDAL.Modificar(existe)
            BitacoraBLL.Insertar(existe, "Login erroneo intento: " &
existe.intentosLogin)
        End If
        Return Nothing
    End If
    If usuario.permiso IsNot Nothing Then
        BitacoraBLL.Insertar(usuario, "Logueo de: " &
usuario.permiso.descripcion)
    End If
    Return usuario
End Function

```

```

Public Shared Function crearUsuario(param As Usuario) As Integer
    param.pass = SeguridadBLL.getSHA1(param.pass)
    param.DVH = SeguridadBLL.calcularDVH(param)
    Dim resultado As Integer = UsuariosDAL.Insertar(param)
    If resultado > 0 Then
        BitacoraBLL.Insertar(param, "Usuario creado")
    End If
End Function

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

End If
recalcularDVVertical()
Return resultado
End Function

Public Shared Function buscar(param As String) As Usuario
Return UsuariosDAL.Buscar(param)
End Function

Public Shared Function verificarPermiso(param As Usuario) As Permiso
Return UsuariosDAL.VerificarPermiso(param)
End Function

Public Shared Function cambiarPass(param As Usuario) As Integer
param.pass = SeguridadBLL.getSHA1(param.pass)
Dim resultado As Integer = UsuariosDAL.cambiasPass(param)
If resultado > 0 Then
    BitacoraBLL.Insertar(param, "Cambio su contraseña")
End If
recalcularDV()
Return resultado
End Function

Public Shared Function generarPass(param As String) As String
Dim usuario As Usuario = buscar(param)
Dim nueva As String = "No existe el usuario"
If usuario IsNot Nothing Then
    nueva = "hola1234"
    usuario.pass = nueva
    cambiarPass(usuario)
End If
Return nueva
End Function

Public Shared Function verificarDVH() As String
Dim resultado As String = ""
For Each item In listar()
    If Not SeguridadBLL.verificarDVH(item) Then
        resultado += "Error en el registro: " & item.id &
Environment.NewLine
    End If
Next
If String.IsNullOrEmpty(resultado) Then
    Return "Ok"
End If
Return resultado
End Function

Public Shared Function recalcularDV() As String
Dim cantidadModificados As Integer = 0
For Each item In listar()
    item.DVH = SeguridadBLL.calcularDVH(item)
    cantidadModificados += UsuariosDAL.Modificar(item)
Next
Return cantidadModificados
End Function

```



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Public Shared Function verificarDVVertical() As String
    Dim digito As String = ""
    For Each item In listar()
        digito += item.DVH
    Next
    Return SeguridadBLL.getSHA1(digito)
End Function

Public Shared Function recalcularDVVertical() As String
    Return SeguridadBLL.insertarDVVertical("Usuario", verificarDVVertical())
End Function

```

End Class

## 10.7 WEBMASTERBLL.VB

```

Imports BE, DAL
Public Class WebmasterBLL
    Public Shared Function verificarDVHBase(param As Usuario) As String
        Dim resultado As String = verificarDVH()
        BitacoraBLL.Insertar(param, "Verificación DV Horizontal")
        Return resultado
    End Function

    Public Shared Function verificarDVH() As String
        Dim resultado As String = "Inicio verificacion de la Base de datos" &
Environment.NewLine
        resultado += "Tabla Producto: "
        resultado += ProductosBLL.verificarDVH & Environment.NewLine
        resultado += "Tabla Usuario: "
        resultado += UsuariosBLL.verificarDVH & Environment.NewLine
        resultado += "Tabla Bitacora: "
        resultado += BitacoraBLL.verificarDVH & Environment.NewLine
        resultado += "Fin."
        Return resultado
    End Function

    Public Shared Function recalcularDVHBase(param As Usuario) As String
        Dim resultado As String = "Inicio de recalcule de DVH de la Base de
datos" & Environment.NewLine
        resultado += "Tabla Producto:" & Environment.NewLine
        resultado += ProductosBLL.recalcularDV & Environment.NewLine
        resultado += "Tabla Usuario:" & Environment.NewLine
        resultado += UsuariosBLL.recalcularDV & Environment.NewLine
        resultado += "Fin."
        BitacoraBLL.Insertar(param, "Recalcule DV Horizontal ")
        Return resultado
    End Function

    Public Shared Function verificarDVVerticalBase(param As Usuario) As String
        Dim resultado As String = verificarDVVertical()
        BitacoraBLL.Insertar(param, "Verificación DV Vertical ")
        Return resultado
    End Function
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Function

```

Public Shared Function verificarDVVertical() As String
    Dim resultado As String = "Inicio de verificacion de DV Vertical de la
Base de datos" & Environment.NewLine

    Dim lista As List(Of DigitoVerificadorVertical) =
DigitoVerificadorDAL.Listar
    For Each item In lista
        If item.tabla = "Usuario" Then
            If item.DVV.Equals(UsuariosBLL.verificarDVVertical) Then
                resultado += "Tabla Usuario: Ok" & Environment.NewLine
            Else
                resultado += "Tabla Usuario: Error" & Environment.NewLine
            End If
        ElseIf item.tabla = "Producto" Then
            If item.DVV.Equals(ProductosBLL.verificarDVVertical) Then
                resultado += "Tabla Producto: Ok" & Environment.NewLine
            Else
                resultado += "Tabla Producto: Error" & Environment.NewLine
            End If
        ElseIf item.tabla = "Bitacora" Then
            If item.DVV.Equals(BitacoraBLL.verificarDVVertical) Then
                resultado += "Tabla Bitacora: Ok" & Environment.NewLine
            Else
                resultado += "Tabla Bitacora: Error" & Environment.NewLine
            End If
        End If
    Next
    resultado += "Fin."
    Return resultado
End Function

Public Shared Function recalcularDVVerticalBase(param As Usuario) As String
    Dim resultado As String = "Inicio de recalcu de DV Vertical de la Base
de datos" & Environment.NewLine
    resultado += "Tabla Usuario:" & Environment.NewLine
    resultado += UsuariosBLL.recalcularDVVertical & Environment.NewLine
    resultado += "Tabla Producto:" & Environment.NewLine
    resultado += ProductosBLL.recalcularDVVertical & Environment.NewLine
    resultado += "Tabla Bitacora:" & Environment.NewLine
    resultado += BitacoraBLL.recalcularDVVertical & Environment.NewLine
    BitacoraBLL.Insertar(param, "Recalcu de DV Vertical ")
    resultado += "Fin."
    Return resultado
End Function

Public Shared Function verificarAInicio() As Boolean
    Dim resultado As String = verificarDVH() + verificarDVVertical()
    Return resultado.Contains("Error")
End Function

Public Shared Function hacerBackup(param As Usuario) As String
    BitacoraBLL.Insertar(param, "Se realizo Copia de seguridad")
    Return backupDAL.hacerBackup()
End Function

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```
Public Shared Function restaurarBackup(param As Usuario) As String
    BitacoraBLL.Insertar(param, "Se Restauro Copia de seguridad")
    Return backupDAL.restaurarBackup()
End Function
```

```
End Class
```

## 10.8 BASE.VB

```
Imports BLL, BE
Public Class Base
    Inherits System.Web.UI.Page

    Protected usuario As Usuario

    Protected Sub UsuarioLogueado()

        Dim login As String = Session("Login")
        usuario = UsuariosBLL.buscar(login)
        If usuario Is Nothing Then
            usuarioNoLogueado()
        End If
    End Sub

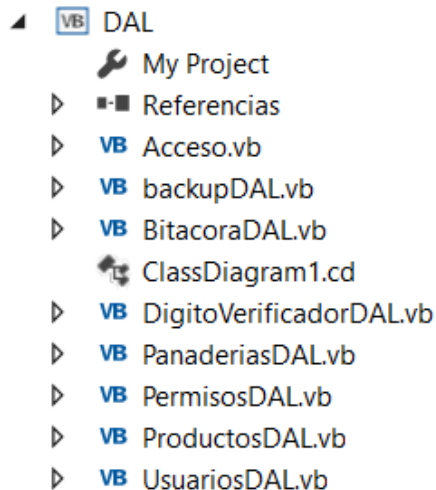
    Protected Sub UsuarioLogueado(permisoRequerido As BE.Permiso)

        Dim login As String = Session("Login")
        usuario = UsuariosBLL.buscar(login)
        If usuario Is Nothing OrElse Not permisoRequerido.Equals(usuario.permiso)
Then
            usuarioNoLogueado()
        End If
    End Sub

    Protected Sub usuarioNoLogueado()
        Session("Login") = ""
        usuario = Nothing
        Response.Redirect("inicio.aspx")
    End Sub
End Class
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 11 CAPA DE ACCESO A DATOS



### 11.1 ACCESO.VB

```
Imports System.Data.SqlClient

Public Class Acceso
    Dim conexion As New SqlConnection("INITIAL CATALOG=Medialunas; DATA
SOURCE=.\SQLEXPRESS; INTEGRATED SECURITY= SSPI")

    Public Sub abrir()
        conexion.Open()
    End Sub

    Public Sub cerrar()
        conexion.Close()
    End Sub

    Public Function leer(nombre As String, parametros() As SqlParameter) As
DataTable
        abrir()

        Dim adapter As New SqlDataAdapter
        Dim tabla As New DataTable
        adapter.SelectCommand = New SqlCommand(nombre, conexion)
        adapter.SelectCommand.CommandType = CommandType.StoredProcedure
        If parametros IsNot Nothing Then
            adapter.SelectCommand.Parameters.AddRange(parametros)
        End If
        adapter.Fill(tabla)

        cerrar()
        Return tabla
    End Function
End Class
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Function

```
Public Function escribir(nombre As String, parametros() As SqlParameter) As Integer
```

```
    abrir()
```

```
    Dim cmd As New SqlCommand(nombre, conexion)
    cmd.CommandType = CommandType.StoredProcedure
    If parametros IsNot Nothing Then
        cmd.Parameters.AddRange(parametros)
    End If
    Dim afectadas As Integer = cmd.ExecuteNonQuery
```

```
    cerrar()
    Return afectadas
```

End Function

```
Public Function ejecutarSQL(nombre As String, parametros() As SqlParameter) As Integer
```

```
    Dim conn As New SqlConnection("INITIAL CATALOG=master; DATA SOURCE=.\SQLEXPRESS; INTEGRATED SECURITY= SSPI")
    conn.Open()
```

```
    Dim cmd As New SqlCommand(nombre, conn)
    cmd.CommandType = CommandType.Text
    If parametros IsNot Nothing Then
        cmd.Parameters.AddRange(parametros)
    End If
    Dim afectadas As Integer = cmd.ExecuteNonQuery
```

```
    conn.Close()
    Return afectadas
```

End Function

End Class

## 11.2 BACKUPDAL.VB

```
Imports BE, System.Data.SqlClient
Public Class backupDAL
```

```
    Public Shared Function hacerBackup() As String
```

```
        Dim acceso As New Acceso
        Dim sql As String = "backup database Medialunas to disk='c:\back\backup.sql'"
        Dim afectadas As Integer = acceso.ejecutarSQL(sql, Nothing)
        Return afectadas
    End Function
```

```
    Public Shared Function restaurarBackup() As String
```

```
        Dim acceso As New Acceso
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Dim antes As String = "alter database Medialunas set offline with
rollback immediate"
Dim despues As String = "alter database Medialunas set online"
Dim sql As String = "restore database Medialunas from
disk='c:\back\backup.sql'"
Dim antesI As Integer = acceso.ejecutarSQL(antes, Nothing)
Dim afectadas As Integer = acceso.ejecutarSQL(sql, Nothing)
Dim despuesI As Integer = acceso.ejecutarSQL(despues, Nothing)
Return "Antes: " & antesI & "Restore: " & afectadas & "Despues: " &
despuesI
End Function
End Class

```

### 11.3 BITACORADAL.VB

```
Imports BE, System.Data.SqlClient
```

```
Public Class BitacoraDAL
```

```
Public Shared Function Listar() As List(Of Bitacora)
```

```
Dim lista = New List(Of Bitacora)
```

```
Dim acceso As New Acceso
```

```
Dim tabla As DataTable = acceso.leer("bitacora_leer", Nothing)
```

```
For Each item In tabla.Rows
```

```
Dim obj As New Bitacora
```

```
obj.id = item("id")
```

```
obj.usuario = UsuariosDAL.Buscar(item("usuario"))
```

```
obj.accion = item("accion")
```

```
obj.fecha = item("fecha")
```

```
obj.DVH = item("DVH")
```

```
lista.Add(obj)
```

```
Next
```

```
Return lista
```

```
End Function
```

```
Public Shared Function Insertar(param As Bitacora) As Integer
```

```
Dim acceso As New Acceso
```

```
Dim parametros(3) As SqlParameter
```

```
parametros(0) = New SqlParameter("@usuario", param.usuario.login)
```

```
parametros(1) = New SqlParameter("@accion", param.accion)
```

```
Dim paramFecha As New SqlParameter("@fecha", SqlDbType.DateTime)
```

```
paramFecha.Value = param.fecha
```

```
parametros(2) = paramFecha
```

```
parametros(3) = New SqlParameter("@dvh", param.DVH)
```

```
Return acceso.escribir("bitacora_alta", parametros)
```

```
End Function
```

```
Public Shared Function Buscar(param As Bitacora) As Bitacora
```

```
Dim lista = Listar()
```

```
Dim encontro As Integer = lista.IndexOf(param)
```

```
If encontro <> -1 Then
```

```
Return lista.Item(encontro)
```

```
Else
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        Return Nothing
    End If
End Function
End Class

```

## 11.4 DIGITOVERIFICADORDAL.VB

```

Imports BE, System.Data.SqlClient

Public Class DigitoVerificadorDAL
    Public Shared Function Listar() As List(Of DigitoVerificadorVertical)
        Dim lista = New List(Of DigitoVerificadorVertical)
        Dim acceso As New Acceso
        Dim tabla As DataTable = acceso.leer("digitoverificador_leer", Nothing)
        For Each item In tabla.Rows
            Dim obj As New DigitoVerificadorVertical
            obj.id = item("id")
            obj.tabla = item("tabla")
            obj.DVV = item("DVV")
            obj.DVH = item("DVH")
            lista.Add(obj)
        Next
        Return lista
    End Function

    Public Shared Function baja(param As DigitoVerificadorVertical) As Integer

        Return 0
    End Function

    Public Shared Function Insertar(param As DigitoVerificadorVertical) As Integer
        Dim acceso As New Acceso
        Dim parametros(2) As SqlParameter
        parametros(0) = New SqlParameter("@tabla", param.tabla)
        parametros(1) = New SqlParameter("@dvv", param.DVV)
        parametros(2) = New SqlParameter("@dvh", param.DVH)
        Return acceso.escribir("digitoverificador_alta", parametros)
    End Function

    Public Shared Function Modificar(param As DigitoVerificadorVertical) As Integer
        Dim acceso As New Acceso
        Dim parametros(2) As SqlParameter
        parametros(0) = New SqlParameter("@tabla", param.tabla)
        parametros(1) = New SqlParameter("@dvv", param.dvv)
        parametros(2) = New SqlParameter("@dvh", param.DVH)
        Return acceso.escribir("digitoverificador_modificar", parametros)
    End Function

    Public Shared Function Buscar(param As DigitoVerificadorVertical) As DigitoVerificadorVertical

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Dim lista = Listar()
Dim encontro As Integer = lista.IndexOf(param)
If encontro <> -1 Then
    Return lista.Item(encontro)
Else
    Return Nothing
End If
End Function
End Class

```

## 11.5 PANADERIASDAL.VB

```
Imports BE, System.Data.SqlClient
```

```

Public Class PanaderiasDAL
    Public Shared Function Listar() As List(Of Panaderia)
        Dim lista = New List(Of Panaderia)
        Dim acceso As New Acceso
        Dim tabla As DataTable = acceso.leer("panaderia_leer", Nothing)
        For Each item In tabla.Rows
            Dim obj As New Panaderia
            obj.id = item("id")
            obj.nombre = item("nombre")
            obj.ubicacion = item("ubicacion")
            lista.Add(obj)
        Next
        Return lista
    End Function

    Public Shared Function baja(param As Panaderia) As Integer

        Return 0
    End Function

    Public Shared Function Insertar(param As Panaderia) As Integer

        Return 0
    End Function

    Public Shared Function Modificar(param As Panaderia) As Integer

        Return 0
    End Function

    Public Shared Function Buscar(param As Panaderia) As Panaderia
        Dim lista = Listar()
        Dim encontro As Integer = lista.IndexOf(param)
        If encontro <> -1 Then
            Return lista.Item(encontro)
        Else
            Return Nothing
        End If
    End Function
End Class

```



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Function  
End Class

## 11.6 PERMISOSDAL.VB

Imports BE, System.Data.SqlClient

Public Class PermisosDAL

```
Public Shared Function buscarPermiso(id As Integer) As Permiso
    Dim obj As Permiso = Nothing
    Dim acceso As New Acceso
    Dim parametros(0) As SqlParameter
    parametros(0) = New SqlParameter("@id", id)
    Dim tabla As DataTable = acceso.leer("permiso_buscar", parametros)
    For Each item In tabla.Rows
        obj = New Permiso
        obj.id = item("id")
        obj.descripcion = item("descripcion")
    Next
    Return obj
End Function
```

```
Public Shared Function permisoAdmin() As Permiso
    Return buscarPermiso(1)
End Function
```

```
Public Shared Function permisoWeb() As Permiso
    Return buscarPermiso(2)
End Function
```

End Class

## 11.7 PRODUCTOSDAL.VB

Imports BE, System.Data.SqlClient

Public Class ProductosDAL

```
Public Shared Function Listar() As List(Of Producto)
    Dim lista = New List(Of Producto)
    Dim acceso As New Acceso
    Dim tabla As DataTable = acceso.leer("producto_leer", Nothing)
    For Each item In tabla.Rows
        Dim obj As New Producto
        obj.id = item("id")
        obj.descripcion = item("descripcion")
        obj.imagen = item("imagen")
        obj.DVH = item("DVH")
        lista.Add(obj)
    Next
End Function
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Next
Return lista
End Function

Public Shared Function baja(param As Producto) As Integer

    Return 0
End Function

Public Shared Function Insertar(param As Producto) As Integer
    Dim acceso As New Acceso
    Dim parametros(2) As SqlParameter
    parametros(0) = New SqlParameter("@descripcion", param.descripcion)
    parametros(1) = New SqlParameter("@imagen", param.imagen)
    parametros(2) = New SqlParameter("@dvh", param.DVH)
    Return acceso.escribir("producto_alta", parametros)
End Function

Public Shared Function Modificar(param As Producto) As Integer
    Dim acceso As New Acceso
    Dim parametros(3) As SqlParameter
    parametros(0) = New SqlParameter("@id", param.id)
    parametros(1) = New SqlParameter("@descripcion", param.descripcion)
    parametros(2) = New SqlParameter("@imagen", param.imagen)
    parametros(3) = New SqlParameter("@dvh", param.DVH)
    Return acceso.escribir("producto_modificar", parametros)
End Function

Public Shared Function Buscar(param As Producto) As Producto
    Dim lista = Listar()
    Dim encontro As Integer = lista.IndexOf(param)
    If encontro <> -1 Then
        Return lista.Item(encontro)
    Else
        Return Nothing
    End If
End Function
End Class

```

## 11.8 USUARIOSDAL.VB

```

Imports BE, System.Data.SqlClient

Public Class UsuariosDAL

    Public Shared Function Listar() As List(Of Usuario)
        Dim lista = New List(Of Usuario)

        Dim acceso As New Acceso
        Dim tabla As DataTable = acceso.leer("usuario_leer", Nothing)
        For Each item In tabla.Rows
            Dim obj As New Usuario
            obj.id = item("id")
            obj.login = item("login")

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        obj.pass = item("pass")
        obj.nombre = item("nombre")
        obj.apellido = item("apellido")
        obj.correo = item("correo")
        obj.intentosLogin = item("intentos")
        obj.habilitado = item("habilitado")
        obj.DVH = item("DVH")
        lista.Add(obj)
    Next
    Return lista
End Function

Public Shared Function baja(param As Usuario) As Integer

    Return 0
End Function

Public Shared Function Insertar(param As Usuario) As Integer
    Dim acceso As New Acceso
    Dim parametros(5) As SqlParameter
    parametros(0) = New SqlParameter("@login", param.login)
    parametros(1) = New SqlParameter("@pass", param.pass)
    parametros(2) = New SqlParameter("@nombre", param.nombre)
    parametros(3) = New SqlParameter("@apellido", param.apellido)
    parametros(4) = New SqlParameter("@correo", param.correo)
    parametros(5) = New SqlParameter("@dvh", param.DVH)
    Return acceso.escribir("usuario_alta", parametros)
End Function

Public Shared Function cambiasPass(param As Usuario) As Integer
    Dim acceso As New Acceso
    Dim parametros(1) As SqlParameter
    parametros(0) = New SqlParameter("@login", param.login)
    parametros(1) = New SqlParameter("@pass", param.pass)
    Return acceso.escribir("usuario_modificarPass", parametros)
End Function

Public Shared Function Modificar(param As Usuario) As Integer
    Dim acceso As New Acceso
    Dim parametros(7) As SqlParameter
    parametros(0) = New SqlParameter("@login", param.login)
    parametros(1) = New SqlParameter("@pass", param.pass)
    parametros(2) = New SqlParameter("@nombre", param.nombre)
    parametros(3) = New SqlParameter("@apellido", param.apellido)
    parametros(4) = New SqlParameter("@correo", param.correo)
    parametros(5) = New SqlParameter("@intentos", param.intentosLogin)
    parametros(6) = New SqlParameter("@habilitado", param.habilitado)
    parametros(7) = New SqlParameter("@dvh", param.DVH)
    Return acceso.escribir("usuario_modificar", parametros)
End Function

Public Shared Function Login(param As Usuario) As Usuario
    Dim obj As Usuario = Nothing
    Dim acceso As New Acceso
    Dim parametros(1) As SqlParameter
    parametros(0) = New SqlParameter("@login", param.login)

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

parametros(1) = New SqlParameter("@pass", param.pass)
Dim tabla As DataTable = acceso.leer("usuario_login", parametros)
For Each item In tabla.Rows
    obj = New Usuario
    obj.id = item("id")
    obj.login = item("login")
    obj.pass = item("pass")
    obj.nombre = item("nombre")
    obj.apellido = item("apellido")
    obj.correo = item("correo")
    obj.intentosLogin = item("intentos")
    obj.habilitado = item("habilitado")
    obj.DVH = item("DVH")
Next
If obj IsNot Nothing Then
    Dim permiso = VerificarPermiso(obj)
    If permiso IsNot Nothing Then
        obj.permiso = permiso
    End If
    Return obj
Else
    Return Nothing
End If
End Function

Public Shared Function Buscar(param As Usuario) As Usuario
    Dim obj As Usuario = Nothing
    Dim acceso As New Acceso
    Dim parametros(0) As SqlParameter
    parametros(0) = New SqlParameter("@login", param.login)
    Dim tabla As DataTable = acceso.leer("usuario_buscar", parametros)
    For Each item In tabla.Rows
        obj = New Usuario
        obj.id = item("id")
        obj.login = item("login")
        obj.pass = item("pass")
        obj.nombre = item("nombre")
        obj.apellido = item("apellido")
        obj.correo = item("correo")
        obj.intentosLogin = item("intentos")
        obj.habilitado = item("habilitado")
        obj.DVH = item("DVH")
    Next
    If obj IsNot Nothing Then
        obj.permiso = VerificarPermiso(obj)
        Return obj
    Else
        Return Nothing
    End If
End Function

Public Shared Function Buscar(param As String) As Usuario
    Dim obj = New Usuario
    obj.login = param
    Return Buscar(obj)
End Function

```

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

```

Public Shared Function VerificarPermiso(param As Usuario) As Permiso
    Dim acceso As New Acceso
    Dim parametros(0) As SqlParameter
    parametros(0) = New SqlParameter("@usuario", param.login)
    Dim tabla As DataTable = acceso.leer("usuario_permisos", parametros)
    For Each item In tabla.Rows
        Return PermisosDAL.buscarPermiso(item("permiso"))
    Next
    Return Nothing
End Function

```

End Class

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 12 CAPA DE ENTIDADES DE NEGOCIO

- VB BE
  - My Project
  - Referencias
  - VB Bitacora.vb
  - DiagramaClasesBE.cd
  - VB DigitoVerificador.vb
  - VB DigitoVerificadorVertical.vb
  - VB Item.vb
  - VB Panaderia.vb
  - VB Pedidos.vb
  - VB Permiso.vb
  - VB Persona.vb
  - VB Producto.vb
  - VB Usuario.vb
  - VB Usuario\_Permiso.vb

### 12.1 BITACORA.VB

```
Public Class Bitacora
    Inherits DigitoVerificador

    Private _usuario As Usuario
    Public Property usuario() As Usuario
        Get
            Return _usuario
        End Get
        Set(ByVal value As Usuario)
            _usuario = value
        End Set
    End Property

    Private _accion As String
    Public Property accion() As String
        Get
            Return _accion
        End Get
        Set(ByVal value As String)
            _accion = value
        End Set
    End Property

    Private _fecha As DateTime
    Public Property fecha() As DateTime
        Get
            Return _fecha
        End Get
        Set(ByVal value As DateTime)
            _fecha = value
        End Set
    End Property
End Class
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        End Get
        Set(ByVal value As DateTime)
            _fecha = value
        End Set
    End Property

    Public Overrides Function verificar() As String
        Return usuario.ToString & accion & fecha.ToString("dd/MM/yyyy H:mm:ss")
    End Function
End Class

```

## 12.2 DIGITOVERIFICADOR.VB

```

Public MustInherit Class DigitoVerificador

    Private _id As Integer
    Public Property id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Private _DVH As String
    Public Property DVH() As String
        Get
            Return _DVH
        End Get
        Set(ByVal value As String)
            _DVH = value
        End Set
    End Property

    Public MustOverride Function verificar() As String

End Class

```

## 12.3 DIGITOVERIFICADORVERTICAL.VB

```

Public Class DigitoVerificadorVertical
    Inherits DigitoVerificador

    Private _tabla As String
    Public Property tabla() As String
        Get
            Return _tabla
        End Get

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        Set(ByVal value As String)
            _tabla = value
        End Set
    End Property

    Private _dvv As String
    Public Property DVV() As String
        Get
            Return _dvv
        End Get
        Set(ByVal value As String)
            _dvv = value
        End Set
    End Property

    Public Overrides Function verificar() As String
        Return tabla & dvv
    End Function

End Class

```

## 12.4 PANADERIA.VB

```

Public Class Panaderia
    Private _id As Integer
    Public Property id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Private _nombre As String
    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property

    Private _ubicacion As String
    Public Property ubicacion() As String
        Get
            Return _ubicacion
        End Get
        Set(ByVal value As String)
            _ubicacion = value
        End Set
    End Property

```



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Public Overrides Function ToString() As String
    Return _nombre
End Function

Public Overrides Function Equals(obj As Object) As Boolean
    Try
        If Me.nombre.Equals(obj.nombre) AndAlso
Me.ubicacion.Equals(obj.ubicacion) Then
            Return True
        Else
            Return False
        End If
    Catch ex As Exception
        Return False
    End Try
End Function

End Class

```

## 12.5 PEDIDOS.VB

```

Public Class Pedidos

    Private _usuario As Usuario
    Public Property usuario() As Usuario
    Get
        Return _usuario
    End Get
    Set(ByVal value As Usuario)
        _usuario = value
    End Set
End Property

    Private _items As List(Of Item)
    Public Property items() As List(Of Item)
    Get
        Return _items
    End Get
    Set(ByVal value As List(Of Item))
        _items = value
    End Set
End Property

    Private _panaderia As Panaderia
    Public Property panaderia() As Panaderia
    Get
        Return _panaderia
    End Get
    Set(ByVal value As Panaderia)
        _panaderia = value
    End Set
End Property

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

End Class

## 12.6 PERMISO.VB

```
Public Class Permiso
    Private _id As Integer
    Public Property id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Private _descripcion As String
    Public Property descripcion() As String
        Get
            Return _descripcion
        End Get
        Set(ByVal value As String)
            _descripcion = value
        End Set
    End Property

    Public Overrides Function ToString() As String
        Return _descripcion
    End Function

    Public Overrides Function Equals(obj As Object) As Boolean
        Try
            If Me.id.Equals(obj.id) Then
                Return True
            Else
                Return False
            End If
        Catch ex As Exception
            Return False
        End Try
    End Function
End Class
```

## 12.7 PERSONA.VB

```
Public Class Persona
    Inherits DigoVerificador

    Private _nombre As String
```

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

```

Public Property nombre() As String
    Get
        Return _nombre
    End Get
    Set(ByVal value As String)
        _nombre = value
    End Set
End Property

Private _apellido As String
Public Property apellido() As String
    Get
        Return _apellido
    End Get
    Set(ByVal value As String)
        _apellido = value
    End Set
End Property

Private _dni As String
Public Property dni() As String
    Get
        Return _dni
    End Get
    Set(ByVal value As String)
        _dni = value
    End Set
End Property

Private _direccion As String
Public Property direccion() As String
    Get
        Return _direccion
    End Get
    Set(ByVal value As String)
        _direccion = value
    End Set
End Property

Private _correo As String
Public Property correo() As String
    Get
        Return _correo
    End Get
    Set(ByVal value As String)
        _correo = value
    End Set
End Property

Public Overrides Function verificar() As String
    Throw New NotImplementedException()
End Function
End Class

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 12.8 PRODUCTO.VB

```

Public Class Producto
    Inherits DigitoVerificador

    Private _descripcion As String
    Public Property descripcion() As String
        Get
            Return _descripcion
        End Get
        Set(ByVal value As String)
            _descripcion = value
        End Set
    End Property

    Private _imagen As String
    Public Property imagen() As String
        Get
            Return _imagen
        End Get
        Set(ByVal value As String)
            _imagen = value
        End Set
    End Property

    Public Overrides Function ToString() As String
        Return _descripcion
    End Function

    Public Overrides Function Equals(obj As Object) As Boolean
        Try
            If Me.id = obj.id Then
                Return True
            Else
                Return False
            End If
        Catch ex As Exception
            Return False
        End Try
    End Function

    Public Overrides Function verificar() As String
        Return descripcion & imagen
    End Function
End Class

```

## 12.9 USUARIO.VB

```

Public Class Usuario
    Inherits Persona

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

Private _login As String
Public Property login() As String
    Get
        Return _login
    End Get
    Set(ByVal value As String)
        _login = value
    End Set
End Property

Private _pass As String
Public Property pass() As String
    Get
        Return _pass
    End Get
    Set(ByVal value As String)
        _pass = value
    End Set
End Property

Private _habilitado As Boolean
Public Property habilitado() As Boolean
    Get
        Return _habilitado
    End Get
    Set(ByVal value As Boolean)
        _habilitado = value
    End Set
End Property

Private _intentosLogin As Integer
Public Property intentosLogin() As Integer
    Get
        Return _intentosLogin
    End Get
    Set(ByVal value As Integer)
        _intentosLogin = value
    End Set
End Property

Private _permiso As Permiso
Public Property permiso() As Permiso
    Get
        Return _permiso
    End Get
    Set(ByVal value As Permiso)
        _permiso = value
    End Set
End Property

Public Overrides Function ToString() As String
    Return Me.nombre & " " & Me.apellido
End Function

Public Overrides Function Equals(obj As Object) As Boolean
    Try

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```

        If Me.login.Equals(obj.login) AndAlso Me.pass.Equals(obj.pass) Then
            Return True
        Else
            Return False
        End If
    Catch ex As Exception
        Return False
    End Try
End Function

```

```

Public Overrides Function verificar() As String
    Return login & pass & nombre & apellido & correo
End Function

```

End Class

## 12.10 USUARIO\_PERMISO.VB

```
Public Class Usuario_Permiso
```

```

    Private _usuario As Usuario
    Public Property usuario() As Usuario
        Get
            Return _usuario
        End Get
        Set(ByVal value As Usuario)
            _usuario = value
        End Set
    End Property

```

```

    Private _permiso As Permiso
    Public Property permiso() As Permiso
        Get
            Return _permiso
        End Get
        Set(ByVal value As Permiso)
            _permiso = value
        End Set
    End Property

```

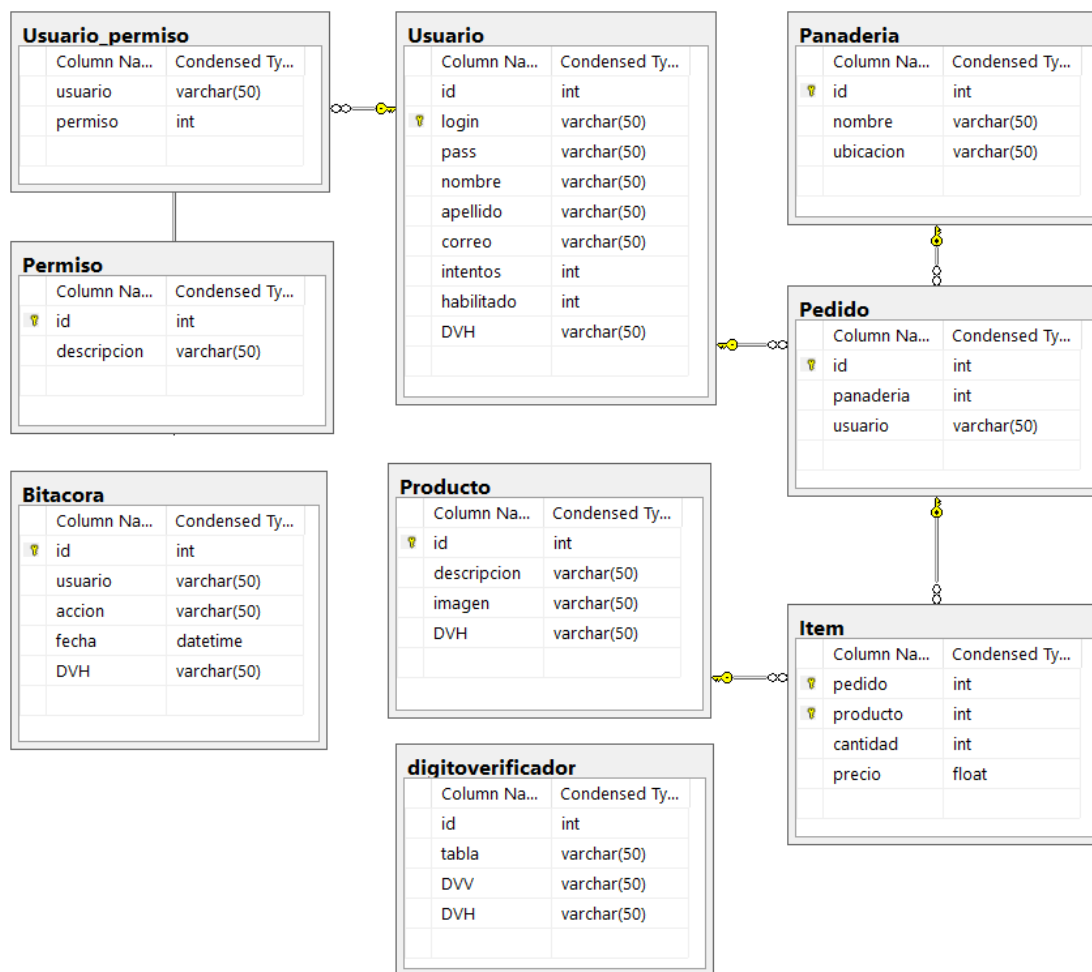
End Class

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi		Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 13 ESTRUCTURA DE BASE DE DATOS

La base de datos que contiene toda la información es una base de datos relacional Microsoft SQL server 2012.

### 13.1 DER



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

## 13.2 PROCEDIMIENTOS ALMACENADOS

## 13.3 BITACORA

CREATE procedure [dbo].[**BITACORA\_ALTA**]

@usuario varchar(50), @accion varchar(50),@fecha Datetime, @dvh varchar(50) as

begin

insert into bitacora values (@usuario,@accion, @fecha, @dvh);

end

create procedure [dbo].[**BITACORA\_LEER**]

as

begin

select \* from bitacora

end

## 13.4 DIGITOVERIFICADOR

CREATE procedure [dbo].[**DIGITOVERIFICADOR\_ALTA**]

@tabla varchar(50), @dvv varchar(50), @dvh varchar(50) as

begin

insert into digitoverificador values (@tabla, @dvv, @dvh);

end;

CREATE procedure [dbo].[**DIGITOVERIFICADOR\_LEER**]

as

begin



UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

```
select * from digitoverificador
```

```
end
```

```
CREATE procedure [dbo].[DIGITOVERIFICADOR_MODIFICAR]
```

```
@tabla varchar(50), @dvv varchar(50), @dvh varchar(50) as
```

```
begin
```

```
update digitoverificador set dvv = @dvv, DVH = @dvh where tabla = @tabla ;
```

```
end;
```

## 13.5 ITEM

```
create procedure [dbo].[ITEM_ALTA]
```

```
@pedido int, @producto int, @cantidad int, @precio float as
```

```
begin
```

```
insert into item values (@pedido,@producto, @cantidad, @precio);
```

```
end
```

```
create procedure [dbo].[ITEM_LEER]
```

```
as
```

```
begin
```

```
select * from item
```

```
end
```

## 13.6 PANADERIA

```
create procedure [dbo].[PANADERIA_LEER]
```

```
as
```

```
begin
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```
select * from panaderia
```

```
end
```

## 13.7 PEDIDO

```
create procedure [dbo].[PEDIDO_ALTA]
```

```
@panaderia int, @usuario varchar(50)as
```

```
begin
```

```
insert into pedido values (@panaderia,@usuario);
```

```
end
```

```
create procedure [dbo].[PEDIDO_LEER]
```

```
as
```

```
begin
```

```
select * from pedido
```

```
end
```

## 13.8 PERMISO

```
create procedure [dbo].[PERMISO_BUSCAR]
```

```
@id int as
```

```
begin
```

```
select * from Permiso where permiso.id = @id;
```

```
end
```

```
create procedure [dbo].[PERMISO_LEER]
```

```
as
```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

begin

select \* from permiso

end

## 13.9 PRODUCTO

CREATE procedure [dbo].[**PRODUCTO\_ALTA**]

@descripcion varchar(50), @imagen varchar(50), @dvh varchar(50) as

begin

insert into producto values (@descripcion, @imagen, @dvh);

end;

create procedure [dbo].[**PRODUCTO\_LEER**]

as

begin

select \* from producto

end

CREATE procedure [dbo].[**PRODUCTO\_MODIFICAR**]

@id int, @descripcion varchar(50), @imagen varchar(50), @dvh varchar(50) as

begin

update producto set descripcion = @descripcion, imagen = @imagen, DVH = @dvh where id = @id ;

end;

## 13.10 USUARIO

CREATE procedure [dbo].[**USUARIO\_ALTA**]

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de tecnología informática				
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato	
	Alumno: Roberto Piombi			Legajo: 49093
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017

```
@login varchar(50),@pass varchar(50),@nombre varchar(50),
@apellido varchar(50),@correo varchar(50), @dvh varchar(50) as
begin
insert into Usuario values (@login,@pass, @nombre,@apellido,@correo, 0, 0,@dvh);
end
```

```
CREATE procedure [dbo].[USUARIO_BUSCAR]
@login varchar(50) as
begin
select * from usuario where usuario.login = @login;
end
```

```
create procedure [dbo].[usuario_leer]
as
begin
select * from usuario
end
```

```
CREATE procedure [dbo].[USUARIO_LOGIN]
@login varchar(50), @pass varchar(50) as
begin
select * from usuario where usuario.login = @login and usuario.pass = @pass;
end
```

```
CREATE procedure [dbo].[USUARIO_MODIFICAR]
@login varchar(50),@pass varchar(50),@nombre varchar(50),
@apellido varchar(50),@correo varchar(50), @intentos int , @habilitado int, @dvh
varchar(50) as
```

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de tecnología informática					
	Materia: Lenguajes de Programacion para la administración		Docente: Santiago Sabato		Fecha 24/11/2017
	Alumno: Roberto Piombi			Legajo: 49093	
	Sede: Centro	Comisión: 4ºJ	Turno: Noche	Año: 2017	

begin

update Usuario set nombre = @nombre, apellido = @apellido, correo = @correo, intentos = @intentos,

habilitado = @habilitado, DVH = @dvh where login = @login;

end

create procedure [dbo].[**USUARIO\_MODIFICARPASS**]

@login varchar(50),@pass varchar(50) as

begin

update Usuario set pass = @pass where login = @login;

end

### 13.11 USUARIO\_PERMISO

create procedure [dbo].[**USUARIO\_PERMISO\_LEER**]

as

begin

select \* from usuario\_permiso

end

create procedure [dbo].[**USUARIO\_PERMISOS**]

@usuario varchar(50) as

begin

select \* from usuario\_permiso where Usuario\_permiso.usuario = @usuario;

end