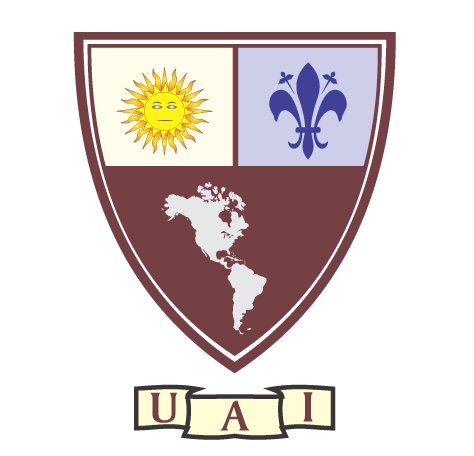
**Facultad de Tecnología Informática**

**Ingeniería en Sistemas Informático**

**Modelos Computacionales de Gestión Administrativa** 

**Trabajo Práctico: El Clavo oxidado**

**Profesor: Sábato, Santiago Juan**

**Alumnos: Castillo, Eduardo**

**Tripelhorn, Facundo**

**Suárez, Gastón**

**2017**

# Índice

[Índice 2](#_Toc497858969)

[1. Descripción del negocio 4](#_Toc497858970)

[2. Explorador de Soluciones – IDE 5](#_Toc497858971)

[3. Capa de presentación 6](#_Toc497858972)

[3.1. Default.aspx 7](#_Toc497858976)

[3.2. Site.Master 9](#_Toc497858977)

[3.3. LogIn.aspx 13](#_Toc497858978)

[3.4. Registrarse.aspx 16](#_Toc497858979)

[3.5. MiCuenta.aspx 20](#_Toc497858980)

[3.6. UserManager.aspx 23](#_Toc497858981)

[3.7. Bitacora.aspx 25](#_Toc497858982)

[3.8. Productos.aspx 28](#_Toc497858983)

[3.9. Carrito.aspx 30](#_Toc497858984)

[3.10. Contact.aspx 33](#_Toc497858985)

[3.11. Backup.aspx 35](#_Toc497858986)

[3.12. Restore.aspx 37](#_Toc497858987)

[3.13. About.aspx 39](#_Toc497858988)

[3.14. EditarProductos.aspx 40](#_Toc497858989)

[3.15. ErrorPage.aspx 41](#_Toc497858990)

[3.16. AdRotator.xml 43](#_Toc497858991)

[4. Capa de Entidades 44](#_Toc497858992)

[4.1. Usuario.vb 44](#_Toc497858994)

[4.2. Bitacora.vb 45](#_Toc497858995)

[4.3. Categoria.vb 45](#_Toc497858996)

[4.4. Carrito.vb 46](#_Toc497858997)

[4.5. ErrorBD.vb 47](#_Toc497858998)

[4.6. Excepcion\_Login.vb 47](#_Toc497858999)

[4.7. Excepcion\_Personalizada.vb 47](#_Toc497859000)

[4.8. Producto.vb 48](#_Toc497859001)

[4.9. Proveedor.vb 48](#_Toc497859002)

[5. Capa de Negocio 49](#_Toc497859003)

[5.1. Usuario 50](#_Toc497859009)

[5.2. Bitacora 52](#_Toc497859016)

[5.3. Categoria 52](#_Toc497859019)

[5.4. Proveedor 53](#_Toc497859023)

[5.5. BackupRestore 53](#_Toc497859027)

[5.6. Producto 54](#_Toc497859030)

[6. Capa de Datos 55](#_Toc497859035)

[6.1. DAL\_Usuario 56](#_Toc497859037)

[6.2. DAL\_Bitacora 60](#_Toc497859045)

[6.3. DAL\_Producto 61](#_Toc497859048)

[6.4. DAL\_BackupRestore 63](#_Toc497859053)

[6.5. DAL\_Categoria 64](#_Toc497859056)

[6.6. DAL\_Proveedor 66](#_Toc497859060)

[6.7. SQL\_Desconectado 68](#_Toc497859064)

[7. Capa Web Services 71](#_Toc497859072)

[7.1. ClavoService.asmx 72](#_Toc497859074)

[8. XML 73](#_Toc497859075)

[1. ANEXOS 74](#_Toc497859076)

[a. Caso de Uso Log in 74](#_Toc497859077)

[b. Diagrama de secuencia Login 76](#_Toc497859078)

[c. DER 76](#_Toc497859079)

[d. Mapa de Navegación 77](#_Toc497859080)

[e. Relación BD/Tabla/Pagina 77](#_Toc497859081)

[f. Estrategias de programación 78](#_Toc497859082)

[g. Políticas de programación 90](#_Toc497859083)

# Descripción del negocio

El Clavo Oxidado es una empresa que tiene origen en el año 2016 dedicada a la comercialización de artículos de ferretería con sede en el barrio de San Telmo, Capital Federal. La empresa nace con el objetivo de brindar productos de la más alta calidad junto con una atención centrada en las necesidades particulares de cada cliente.

Para lograr estos objetivos, trabajamos con las marcas líderes del mercado como Karcher, DeWalt, Bosch y Black&Decker.

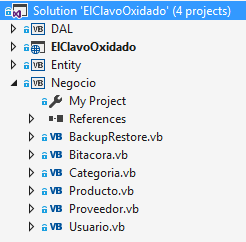
Actualmente la empresa cuenta con 10 empleados que poseen una larga experiencia en el rubro. Lo cual, junto a una constante actualización realizada a través de jornadas de capacitaciones efectuadas en las empresas proveedoras, nos permite asesorar profesionalmente y de manera personalizada a cada uno de nuestros clientes.

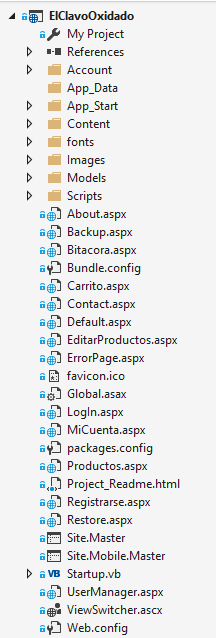
Al focalizar la atención en las necesidades del cliente, la empresa se desempeña tanto en la comercialización de productos tanto B2C como B2B.

Para ventas minoristas, se aceptan pagos únicamente en el local de forma presencial con efectivo o tarjetas de débito y crédito que son procesadas a través de terminales financiera POSNET. Para ventas mayoristas se permiten otros medios de pago bancarizados como transferencias o cheques.

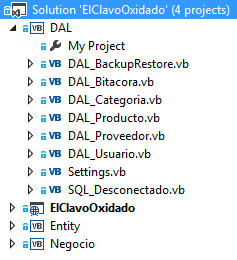
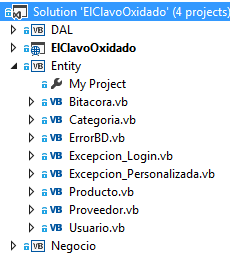
Actualmente, con el objetivo de ofrecer mayores servicios a nuestros clientes, El Clavo Oxidado se encuentra en el proceso de desarrollar el cuarto canal de venta, el comercio electrónico, a través de la apertura de su propio sitio web.

# Explorador de Soluciones – IDE

 Secuencia para Login



**1**



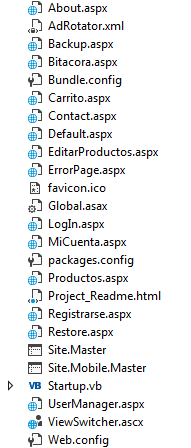
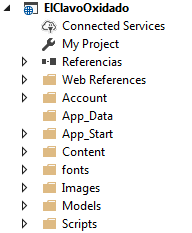
**4**

**3**

**21**

# Capa de presentación

En la solución, el proyecto denominado “ElClavoOxidado” contiene los formularios y lógica de front-end.





# Default.aspx



<%@ Page Title="Inicio" Language="VB" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Default.aspx.vb" Inherits="ElClavoOxidado.\_Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<link rel="stylesheet" href="Content/slideshow.css" />

<script type="text/javascript" src="Scripts/imageslide.js"></script>

<div class="container buscar">

<asp:TextBox ID="BuscarTxt" runat="server" BorderStyle="Solid" Width="219px" CssClass="BuscarTxt"></asp:TextBox>

<asp:Button ID="BuscarBtn" runat="server" Text="Buscar" CssClass="btn btn-default" />

</div>

<section id="BannerSection">

<ul id="galeria">

<li><img src="Images/nuestramarcas.jpg" border="0" alt="Nuestras marcas" /></li>

<li><img src="Images/atencionycalidad.jpg" border="0" alt="Atención y calidad" /></li>

<li><img src="Images/bannerferreteria.jpeg" border="0" alt="Ferreteria" /></li>

</ul>

</section>

</asp:Content>

A ingresar a la página con un usuario con perfil webmaster, se corrobora que no se haya corrompido la integridad de la base de datos a través de las funciones “check” correspondientes ubicadas dentro del Page\_Load.

Para identificar los registros corrompidos, se efectúa el cálculo de una función hash tomando los campos del registro correspondiente. Se utilizan respectivos Store Procedure para tal finalidad.

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

If Session.Item("CheckErrorValidate") = 0 Then

Dim \_listaProveedor As List(Of Entity.Proveedor) = \_dal\_proveedor.Get\_Proveedores

Dim \_listaCategoria As List(Of Entity.Categoria) = \_dal\_categoria.Get\_Categorias

Dim \_listaProducto As List(Of Entity.Producto) = \_dal\_producto.Obtener\_Productos

Dim \_listaError As New List(Of Entity.ErrorBD)

Dim \_id As Integer = 1

For Each \_producto As Entity.Producto In \_listaProducto

If Not \_dal\_producto.Check\_Producto(\_producto) Then

\_listaError.Add(New Entity.ErrorBD(\_id, "Producto", \_listaProducto.IndexOf(\_producto) + 1))

\_id += 1

End If

Next

For Each \_proveedor As Entity.Proveedor In \_listaProveedor

If Not \_dal\_proveedor.Check\_Proveedor(\_proveedor) Then

\_listaError.Add(New Entity.ErrorBD(\_id, "Proveedor", \_listaProveedor.IndexOf(\_proveedor) + 1))

\_id += 1

End If

Next

For Each \_categoria As Entity.Categoria In \_listaCategoria

If Not \_dal\_categoria.Check\_Categoria(\_categoria) Then

\_listaError.Add(New Entity.ErrorBD(\_id, "Categoria", \_listaCategoria.IndexOf(\_categoria) + 1))

\_id += 1

End If

Next

If \_listaError.Count > 0 And Context.User.IsInRole("1") Then

Session("ListaError") = \_listaError

Session("CheckError") = 1

Response.Redirect("ErrorPage", True)

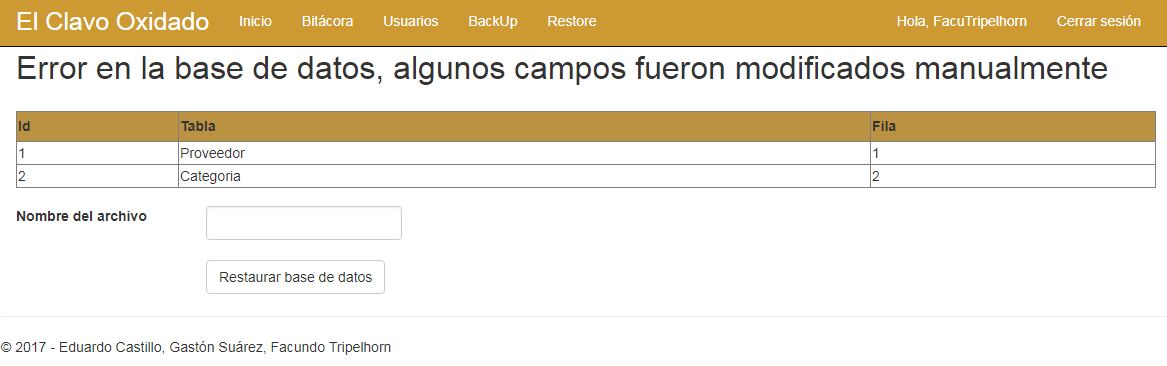
End If

Session("CheckErrorValidate") = 1

End If

End Sub

De suceder tal situación, al ingresar con un usuario webmaster se visualizara un mensaje de error indicando detalladamente la causa del mismo y habita al usuario a efectuar el restore de la base de datos correspondiente.



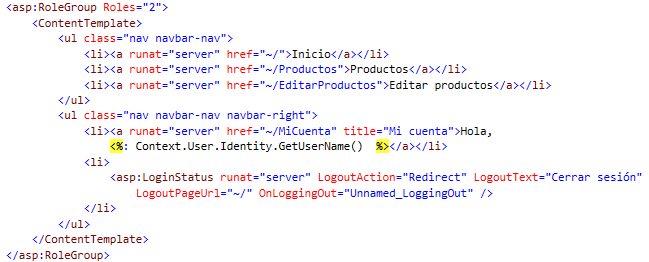
# Site.Master

Definimos el template de las solapas a visualizar



Definimos los 3 roles







Page\_Init

Protected Sub Page\_Init(sender As Object, e As EventArgs)

' El código siguiente ayuda a proteger frente a ataques XSRF

Dim requestCookie = Request.Cookies(AntiXsrfTokenKey)

Dim requestCookieGuidValue As Guid

If requestCookie IsNot Nothing AndAlso   
Guid.TryParse(requestCookie.Value, requestCookieGuidValue) Then

' Utilizar el token Anti-XSRF de la cookie

\_antiXsrfTokenValue = requestCookie.Value

Page.ViewStateUserKey = \_antiXsrfTokenValue

Else

' Generar un nuevo token Anti-XSRF y guardarlo en la cookie

\_antiXsrfTokenValue = Guid.NewGuid().ToString("N")

Page.ViewStateUserKey = \_antiXsrfTokenValue

Dim responseCookie = New HttpCookie(AntiXsrfTokenKey) With { \_

.HttpOnly = True, \_

.Value = \_antiXsrfTokenValue \_

}

If FormsAuthentication.RequireSSL AndAlso Request.IsSecureConnection Then

responseCookie.Secure = True

End If

Response.Cookies.[Set](responseCookie)

End If

AddHandler Page.PreLoad, AddressOf master\_Page\_PreLoad  
 End Sub

master\_Page\_PreLoad

Protected Sub master\_Page\_PreLoad(sender As Object, e As EventArgs)

If Not IsPostBack Then

' Establecer token Anti-XSRF

ViewState(AntiXsrfTokenKey) = Page.ViewStateUserKey

ViewState(AntiXsrfUserNameKey) = If(Context.User.Identity.Name, [String].Empty)

Else

' Validar el token Anti-XSRF

If DirectCast(ViewState(AntiXsrfTokenKey), String) <> \_antiXsrfTokenValue OrElse DirectCast(ViewState(AntiXsrfUserNameKey), String) <> (If(Context.User.Identity.Name, [String].Empty)) Then

Throw New InvalidOperationException("Error de validación del token Anti-XSRF.")

End If

End If  
 End Sub

Unnamed\_LoggingOut

Protected Sub Unnamed\_LoggingOut(sender As Object, e As LoginCancelEventArgs)

Context.GetOwinContext().Authentication.SignOut()

CargarBitacora(HttpContext.Current.User.Identity.Name)  
 End Sub

CargarBitacora

Private Sub CargarBitacora(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = usuario

\_bitacoraEntity.Tipo = "Cierre de sesión"

\_bitacoraEntity.Actividad = usuario & " cerró sesión"

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

# LogIn.aspx



<%@ Page Title="Iniciar sesión" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="LogIn.aspx.vb" Inherits="ElClavoOxidado.LogIn" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<h2></h2>

<div class="row">

<div class="col-md-8">

<section id="loginForm">

<div class="form-horizontal">

<h4>Si tiene una cuenta inicie sesión con su nombre de usuario</h4>

<hr />

<asp:PlaceHolder runat="server" ID="ErrorMessage" Visible="false">

<p class="text-danger">

<asp:Literal runat="server" ID="FailureText" />

</p>

</asp:PlaceHolder>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="UserName" CssClass="col-md-2 control-label">Nombre de usuario</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="UserName" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="UserName"

CssClass="text-danger" ErrorMessage="Ingrese un nombre de usuario." />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2 control-label">Contraseña</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Password" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Password" CssClass="text-danger" ErrorMessage="Ingrese una contraseña." />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<div class="checkbox">

<asp:CheckBox runat="server" ID="RememberMe" />

<asp:Label runat="server" AssociatedControlID="RememberMe">Mantener sesión iniciada?</asp:Label>

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Iniciar sesión" CssClass="btn btn-default" ID="LogInBtn" OnClientClick="javascript:getHash()" OnClick="LogInBtn\_Click" />

</div>

</div>

</div>

<p>

En caso de que no posea una cuenta registrese

<asp:HyperLink runat="server" ID="RegisterHyperLink" ViewStateMode="Disabled" NavigateUrl="Registrarse.aspx" >aquí</asp:HyperLink>.

</p>

</section>

</div>

</div>

<script type="text/javascript" src="Scripts/sha.js"></script>

<script type="text/javascript">

function getHash() {

var hashInput = document.getElementById("<%=Password.ClientID%>");

if (hashInput.value.length > 5) {

var hash = new jsSHA(hashInput.value, "TEXT");

var hashOutput = document.getElementById("<%=Password.ClientID%>");

hashOutput.value = hash.getHash("SHA-256", "HEX");

}

}</script>

</asp:Content>

LogInBtn\_Click

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

\_usuarioEntity = New Entity.Usuario With {.Usuario = UserName.Text, .Password = Password.Text, .Cant\_Bloqueos = Session("cant\_Intentos")}

Try

If Password.Text.Length > 5 Then

If \_usuarioNeg.Validar\_Login(\_usuarioEntity, \_usuarioEntity.Password) Then

If RememberMe.Checked Then

FormsAuthentication.SetAuthCookie(UserName.Text, True)

Else

FormsAuthentication.SetAuthCookie(UserName.Text, False)

End If

If Roles.IsUserInRole(\_usuarioEntity.Usuario, \_usuarioEntity.Familia) = False Then

Roles.AddUserToRole(\_usuarioEntity.Usuario, \_usuarioEntity.Familia)

End If

CargarBitacora(\_usuarioEntity.Usuario)

Response.Redirect("/Default", True)

Else

End If

End If

Catch ex As Exception

If Not TryCast(ex, Entity.Excepcion\_Login) Is Nothing Then

Dim miError As Entity.Excepcion\_Login = DirectCast(ex, Entity.Excepcion\_Login)

If miError.Descripcion = "Usuario invalido" Then

Session("cant\_Intentos") += 1

ScriptManager.RegisterClientScriptBlock(Me, Me.GetType, "Usuario Invalido", "alert('Usuario inválido. Si aún no se registro presione la opción Registrarse')", True)

ElseIf miError.Descripcion = "Usuario Bloqueado" Then

ScriptManager.RegisterClientScriptBlock(Me, Me.GetType, "Usuario bloqueado", "alert('Usuario bloqueado, por favor contactese con el web master')", True)

ElseIf miError.Descripcion = "Password invalida" Then

Session("cant\_Intentos") += 1

ScriptManager.RegisterClientScriptBlock(Me, Me.GetType, "Contraseña invalida", "alert('Contraseña incorrecta')", True)

End If

End If

End Try

End Sub

LogIn\_PreLoad

Private Sub LogIn\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

Dim isSomeoneLoggedIn As Boolean = HttpContext.Current.User.Identity.IsAuthenticated

If isSomeoneLoggedIn Then

Response.Redirect("~/Default", True)

End If  
 End Sub

CargarBitacora

Private Sub CargarBitacora(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

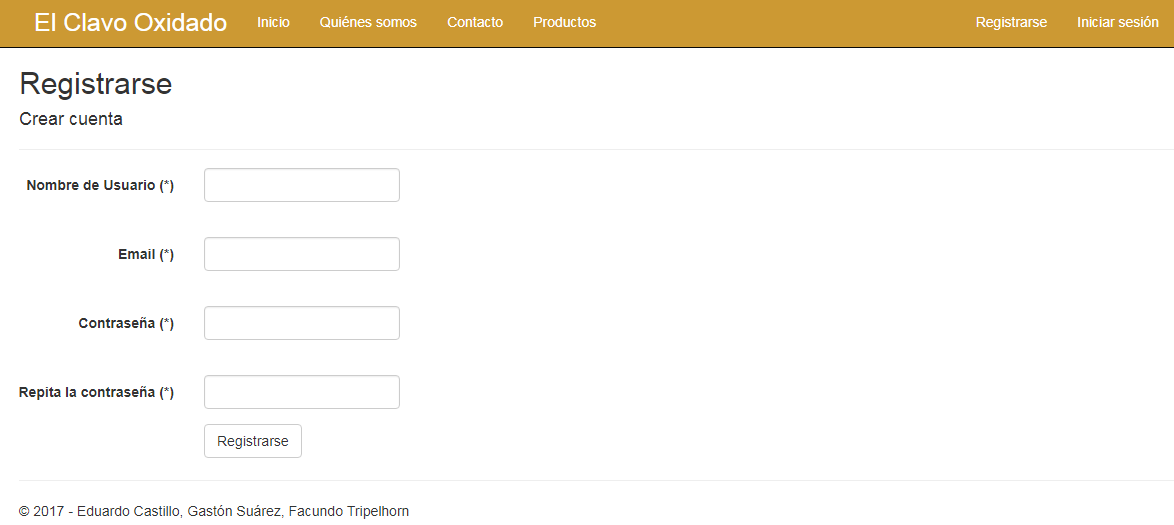
\_bitacoraEntity.Usuario = usuario

\_bitacoraEntity.Tipo = "Inicio de sesión"

\_bitacoraEntity.Actividad = usuario & " inició sesión"

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)  
 End Sub

# Registrarse.aspx



<%@ Page Title="Registrarse" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Registrarse.aspx.vb" Inherits="ElClavoOxidado.Registrarse" %>

<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">

<h2><%: Title %></h2>

<p class="text-danger">

<asp:Literal runat="server" ID="ErrorMessage" />

</p>

<div class="form-horizontal">

<h4>Crear cuenta</h4>

<hr />

<asp:ValidationSummary runat="server" CssClass="text-danger" />

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="UserName" CssClass="col-md-2 control-label">Nombre de Usuario (\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="UserName" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="UserName"

CssClass="text-danger" ErrorMessage="Ingrese un nombre de usuario." />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2 control-label">Email (\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Email" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Email"

CssClass="text-danger" ErrorMessage="Ingrese un email correcto." />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2 control-label">Contraseña (\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Password" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Password"

CssClass="text-danger" ErrorMessage="Ingrese una contraseña." />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="ConfirmPassword" CssClass="col-md-2 control-label">Repita la contraseña (\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="ConfirmPassword"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Ingrese nuevamente la contraseña." />

<asp:CompareValidator runat="server" ControlToCompare="Password" ControlToValidate="ConfirmPassword"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Las contraseñas no coinciden." />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Registrarse" CssClass="btn btn-default" ID="RegistrarseBtn" OnClientClick="javascript:getHash()" OnClick="Registrarse" />

</div>

</div>

</div>

<script type="text/javascript" src="Scripts/sha.js"></script>

<script type="text/javascript">

function getHash() {

var hashInput = document.getElementById("<%=Password.ClientID%>");

if (hashInput.value.length > 5) {

var hash = new jsSHA(hashInput.value, "TEXT");

var hashOutput = document.getElementById("<%=Password.ClientID%>");

var hash2Input = document.getElementById("<%=ConfirmPassword.ClientId%>");

var hash2 = new jsSHA(hash2Input.value, "TEXT");

var hash2Output = document.getElementById("<%=ConfirmPassword.ClientID%>");

hashOutput.value = hash.getHash("SHA-256", "HEX");

hash2Output.value = hash2.getHash("SHA-256", "HEX");

} else {

alert("La contraseña debe contener al menos 6 caracteres")

}

}</script>

</asp:Content>

Registrarse\_PreLoad

Private Sub Registrarse\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

Dim isSomeoneLoggedIn As Boolean = HttpContext.Current.User.Identity.IsAuthenticated

If isSomeoneLoggedIn Then

Response.Redirect("~/Default", True)

End If

End Sub

Registrarse

Protected Sub Registrarse(sender As Object, e As EventArgs)

If Password.Text.Length > 5 Then

If Password.Text = ConfirmPassword.Text Then

\_usuarioEntity = New Entity.Usuario

\_usuarioEntity.Usuario = UserName.Text

\_usuarioEntity.Email = Email.Text

\_usuarioEntity.Password = Password.Text

\_usuarioEntity.Familia = 3

If \_usuarioNegocio.Chequear\_Usuario(\_usuarioEntity.Usuario) = False Then

ErrorMessage.Text = "El nombre de usuario ingresado ya existe, pruebe otro nombre de usuario"

ErrorMessage.Visible = True

Else

\_usuarioNegocio.Alta\_Usuario(\_usuarioEntity)

CargarBitacoraRegistrar(\_usuarioEntity.Usuario)

LogIn()

End If

End If

End If  
 End Sub

LogIn

Private Sub LogIn()

Try

FormsAuthentication.SetAuthCookie(UserName.Text, False)

If Roles.IsUserInRole(\_usuarioEntity.Usuario, \_usuarioEntity.Familia) = False Then

Roles.AddUserToRole(\_usuarioEntity.Usuario, \_usuarioEntity.Familia)

End If

CargarBitacoraLogIn(\_usuarioEntity.Usuario)

Response.Redirect("/Default", True)

Catch ex As Exception

End Try

End Sub

CargarBitacoraLogin

Private Sub CargarBitacoraLogIn(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = usuario

\_bitacoraEntity.Tipo = "Inicio de sesión"

\_bitacoraEntity.Actividad = usuario & " inició sesión"

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

CargarBitacoraRegistrar

Private Sub CargarBitacoraRegistrar(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

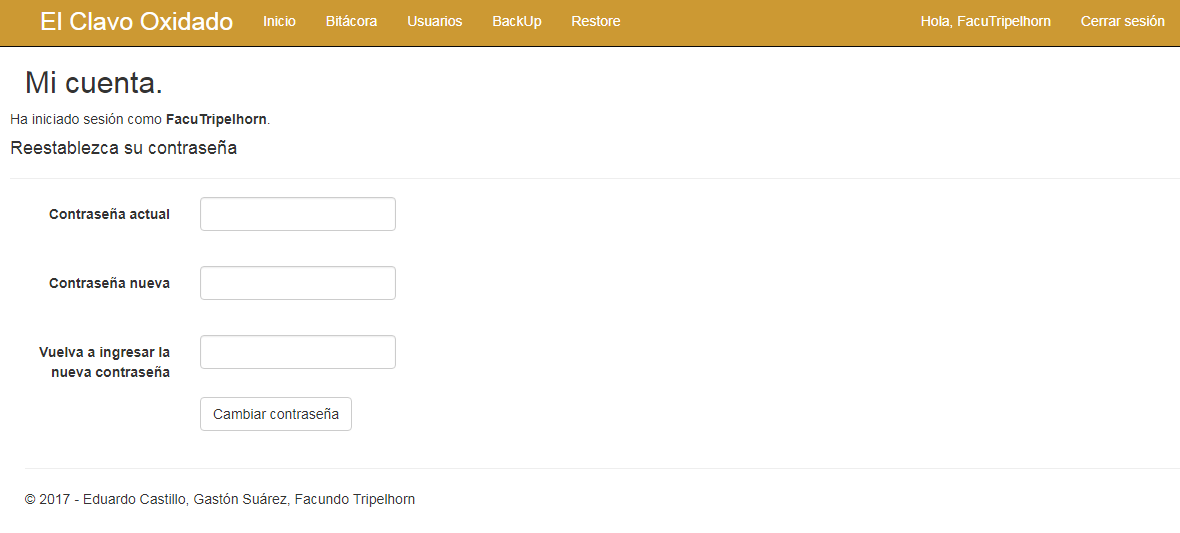
\_bitacoraEntity.Usuario = usuario

\_bitacoraEntity.Tipo = "Registro"

\_bitacoraEntity.Actividad = "Se ha registrado el usuario " & usuario

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)  
 End Sub

# MiCuenta.aspx



<%@ Page Title="Mi cuenta" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="MiCuenta.aspx.vb" Inherits="ElClavoOxidado.MiCuenta" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<h2><%:Title %>.</h2>

<div>

<asp:PlaceHolder runat="server" ID="successMessage" Visible="false" ViewStateMode="Disabled">

<p class="text-success"><%:SuccessMessageText %></p>

</asp:PlaceHolder>

</div>

<div class="row">

<div class="span7">

<section id="passwordForm">

<asp:PlaceHolder runat="server" ID="changePasswordHolder">

<p>Ha iniciado sesión como <strong><%: User.Identity.Name%></strong>.</p>

<div class="form-horizontal">

<h4>Reestablezca su contraseña</h4>

<hr />

<asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="text-danger" />

<div class="form-group">

<asp:Label runat="server" ID="ContraseñaLbl" AssociatedControlID="ContraseñaActual" CssClass="col-md-2 control-label">Contraseña actual</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="ContraseñaActual" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="ContraseñaActual"

CssClass="text-danger" ErrorMessage="Ingrese su contraseña actual."

ValidationGroup="ChangePassword" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" ID="ContraseñaNuevaLbl" AssociatedControlID="ContraseñaNueva" CssClass="col-md-2 control-label">Contraseña nueva</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="ContraseñaNueva" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="ContraseñaNueva"

CssClass="text-danger" ErrorMessage="Ingrese la nueva contraseña."

ValidationGroup="ChangePassword" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" ID="ConfirmarContraseñaLbl" AssociatedControlID="ConfirmarContraseña" CssClass="col-md-2 control-label">Vuelva a ingresar la nueva contraseña</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="ConfirmarContraseña" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="ConfirmarContraseña"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Ingrese nuevamente la nueva contraseña."

ValidationGroup="ChangePassword" />

<asp:CompareValidator runat="server" ControlToCompare="ContraseñaNueva" ControlToValidate="ConfirmarContraseña"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Las contraseñas no coinciden."

ValidationGroup="ChangePassword" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Cambiar contraseña" CssClass="btn btn-default" OnClientClick="javascript:getHash()" ValidationGroup="ChangePassword" ID="CambiarContraseñaBtn" />

</div>

</div>

</div>

</asp:PlaceHolder>

</section>

</div>

</div>

<script type="text/javascript" src="Scripts/sha.js"></script>

<script type="text/javascript">

function getHash() {

var hashInput = document.getElementById("<%=ContraseñaActual.ClientID%>");

var hash = new jsSHA(hashInput.value, "TEXT");

var hashOutput = document.getElementById("<%=ContraseñaActual.ClientID%>");

var hash2Input = document.getElementById("<%=ContraseñaNueva.ClientID%>");

var hash2 = new jsSHA(hash2Input.value, "TEXT");

var hash2Output = document.getElementById("<%=ContraseñaNueva.ClientID%>");

var hash3Input = document.getElementById("<%=ConfirmarContraseña.ClientID%>");

var hash3 = new jsSHA(hash2Input.value, "TEXT");

var hash3Output = document.getElementById("<%=ConfirmarContraseña.ClientID%>");

hashOutput.value = hash.getHash("SHA-256", "HEX");

hash2Output.value = hash2.getHash("SHA-256", "HEX");

hash3Output.value = hash3.getHash("SHA-256", "HEX");

}</script>

</asp:Content>

CambiarContraseñaBtn\_Click

Private Sub CambiarContraseñaBtn\_Click(sender As Object, e As EventArgs) Handles CambiarContraseñaBtn.Click

\_usuarioEntity = New Entity.Usuario

If ContraseñaNueva.Text = ConfirmarContraseña.Text Then

\_usuarioEntity.Usuario = HttpContext.Current.User.Identity.Name

\_usuarioEntity.Password = ContraseñaActual.Text

If \_usuarioNeg.Validar\_Login(\_usuarioEntity, \_usuarioEntity.Password) Then

\_usuarioNeg.Cambiar\_Contraseña(\_usuarioEntity.Usuario, ContraseñaNueva.Text)

CargarBitacora(\_usuarioEntity.Usuario)

SuccessMessageText = "Cambio de contraseña exitoso"

successMessage.Visible = True

changePasswordHolder.Visible = False

ContraseñaActual.Text = ""

ContraseñaNueva.Text = ""

ConfirmarContraseña.Text = ""

End If

End If  
 End Sub

CargarBitacora

Private Sub CargarBitacora(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = usuario

\_bitacoraEntity.Tipo = "Cambio de contraseña"

\_bitacoraEntity.Actividad = usuario & " cambió su contraseña"

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)  
 End Sub

# UserManager.aspx

<%@ Page Title="Gestión de usuarios" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="UserManager.aspx.vb" Inherits="ElClavoOxidado.UserManager" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<link rel="stylesheet" href="Content/estilo.css" />

<h2>Lista de usuarios</h2><br />

<asp:GridView ID="GrillaUsuarios" runat="server">

<HeaderStyle BackColor="#BB9241" />

</asp:GridView>

<br />

<br />

<div class="asignar\_rol\_container">

<div class="rol\_container">

<div class="rol\_container\_left">

<asp:Label runat="server" ID="UsuarioLbl" AssociatedControlID="Username" CssClass="control-label">Usuario</asp:Label>

<asp:TextBox runat="server" ID="Username" CssClass="form-control" />

</div>

<div class="rol\_container\_right">

<asp:Label runat="server" ID="FamiliaLbl" AssociatedControlID="Familia" CssClass="control-label FamiliaLbl">Rol</asp:Label>

<asp:DropDownList ID="Familia" runat="server" CssClass="form-control"></asp:DropDownList>

</div>

</div>

<div class="asignar\_rol">

<asp:Button ID="UpdateUsuarioBtn" runat="server" Text="Asignar rol" CssClass="btn btn-default" ValidationGroup="UserRole" />

</div>

</div>

<br />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Username"

CssClass="text-danger" ErrorMessage="Ingrese el nombre de usuario." ValidationGroup="UserRole"/>

<br />

</asp:Content>

UserManager\_PreLoad

Private Sub UserManager\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("1") Then

Response.Redirect("Default", True)

End If  
 End Sub

Page\_Load

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

Llenar\_Grilla()

Llenar\_Combo()

End Sub

UpdateUsuarioBtn\_Click

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

\_usuarioNeg.Asignar\_Rol(Username.Text, Familia.SelectedIndex + 1)

Llenar\_Grilla()

End Sub

Llenar\_Grilla

Private Sub Llenar\_Grilla()

Dim \_listaUsuario As New List(Of VistaUsuario)

GrillaUsuarios.DataSource = Nothing

For Each \_usuario As Entity.Usuario In \_usuarioNeg.Get\_Usuarios

\_listaUsuario.Add(New VistaUsuario(\_usuario.Usuario, \_usuario.Email, \_usuario.Familia, \_usuario.Bloqueado))

Next

GrillaUsuarios.DataSource = \_listaUsuario

GrillaUsuarios.DataBind()

End Sub

Llenar\_Combo

Private Sub Llenar\_Combo()

Familia.Items.Add("Webmaster")

Familia.Items.Add("Administrador")

Familia.Items.Add("Cliente")

End Sub

CargarBitacora

Private Sub CargarBitacora(usuario As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = usuario

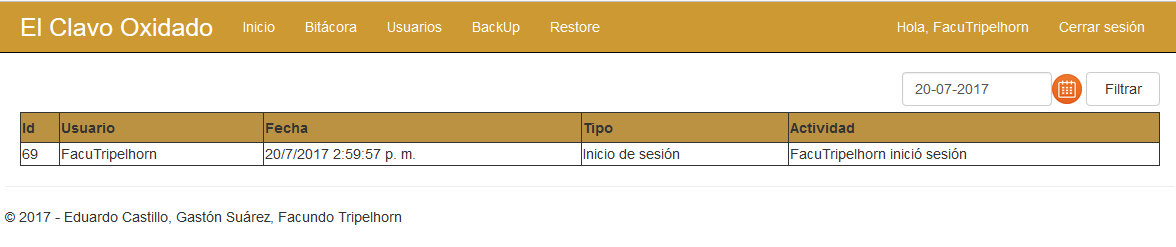
\_bitacoraEntity.Tipo = "Asignación de rol"

\_bitacoraEntity.Actividad = "El usuario " & usuario & "le asigno al usuario " & Username.Text & " el rol de " & Familia.Text

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

# Bitacora.aspx



<%@ Page Title="Bitácora" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Bitacora.aspx.vb" Inherits="ElClavoOxidado.Bitacora" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<link rel="stylesheet" href="Content/estilo.css" />

<br />

<div class="container">

<div class="Filtro">

<asp:TextBox ID="txtDate" runat="server" ReadOnly = "true" CssClass="FiltroTxt"></asp:TextBox>

<asp:Button ID="Button1" runat="server" Text="Filtrar" CssClass="btn btn-default" Width="74px" />

</div>

<br />

<br />

<asp:GridView ID="GrillaBitacora" runat="server" >

<HeaderStyle BackColor="#BB9241" Height="30px" />

</asp:GridView>

<br />

<br />

<br />

<asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="AdRotator.xml" Width="100%" />

</div>

<style type="text/css">

.ui-datepicker-trigger {

height:30px;

}

.FiltroTxt {

display: inline-block;

width:150px;

padding: 6px 12px;

font-size: 14px;

line-height: 1.428571429;

color: #555555;

vertical-align: middle;

background-color: #ffffff;

border: 1px solid #cccccc;

border-radius: 4px;

-webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);

box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);

-webkit-transition: border-color ease-in-out 0.15s, box-shadow ease-in-out 0.15s;

transition: border-color ease-in-out 0.15s, box-shadow ease-in-out 0.15s;

}

.Filtro {

float: right;

}

</style>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6/jquery.min.js" type="text/javascript"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js" type="text/javascript"></script>

<link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="Stylesheet" type="text/css" />

<script type="text/javascript">

$(document).ready(function () {

$("#<% = txtDate.ClientID %>").datepicker({

showOn: 'button',

buttonImageOnly: true,

buttonImage: 'https://s-media-cache-ak0.pinimg.com/736x/c0/74/6d/c0746d1a59f56b5147e1d71c385c2cef--save-the-date-calendar-icon-png.jpg',

dateFormat: "dd-mm-yy",

});

});

</script>

</asp:Content>

Page\_Load

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

mListaOriginal = \_Bitacora\_Negocio.Obtener\_Bitacora

GrillaBitacora.DataSource = Nothing

GrillaBitacora.DataSource = mListaOriginal

GrillaBitacora.DataBind()

End Sub

Bitacora\_PreLoad

Private Sub Bitacora\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("1") Then

Response.Redirect("Default", True)

End If

End Sub

FiltrarBitacoraPorFecha

Private Sub FiltrarBitacoraPorFecha(fecha As Date)

Try

Dim mLista As New List(Of Entity.Bitacora)

For Each elemento As Entity.Bitacora In mListaOriginal

If elemento.Fecha.Date = fecha.Date Then

mLista.Add(elemento)

End If

Next

GrillaBitacora.DataSource = mLista

GrillaBitacora.DataBind()

Catch ex As Exception

End Try

End Sub

Button1\_Click

Protected Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Try

Dim dt As String = Request.Form(txtDate.UniqueID)

txtDate.Text = dt

FiltrarBitacoraPorFecha(Date.Parse(txtDate.Text))

Catch ex As Exception

End Try

End Sub

# Productos.aspx

<%@ Page Title="Productos" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Productos.aspx.vb" Inherits="ElClavoOxidado.Productos" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" Runat="Server">

<h1>Página en Construcción</h1>

<h2>Estamos trabajando para brindarle la mejor experiencia</h2>

<!--<div class="container buscar">

<asp:TextBox ID="BuscarTxt" runat="server" BorderStyle="Solid" Width="219px" CssClass="BuscarTxt"></asp:TextBox>

<asp:Button ID="BuscarBtn" runat="server" Text="Buscar" CssClass="btn btn-default" />

</div> -->

<br />

<br />

<div class="ProdGallery">

<a target="\_blank" href="fjords.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Fjords" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="forest.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Forest" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="lights.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Northern Lights" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="ProdGallery">

<a target="\_blank" href="mountains.jpg">

<img src="http://static.mercadoshops.com/tijera-corta-canos-pvc-pinza-hasta-42mm-latyn\_iZ61475331XvZcXpZ1XfZ92418743-597355278-1.jpgXsZ92418743xIM.jpg" alt="Mountains" width="300" height="200">

</a>

<div class="desc">Add a description of the image here</div>

</div>

</asp:Content>

# Carrito.aspx

<%@ Page Title="Carrito" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Carrito.aspx.vb" Inherits="ElClavoOxidado.Carrito" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" Runat="Server">

<h1>Mi Carrito</h1>

<asp:GridView runat="server" ID="gvShoppingCart" AutoGenerateColumns="false" EmptyDataText="Su carrito se encuentra vacío." GridLines="None" Width="100%" CellPadding="5" ShowFooter="true" DataKeyNames="Codigo" >

<HeaderStyle HorizontalAlign="Left" BackColor= "#cc9933" ForeColor="#FFFFFF" />

<FooterStyle HorizontalAlign="Right" BackColor= "#cc9933" ForeColor="#FFFFFF" />

<AlternatingRowStyle BackColor="#F8F8F8" />

<Columns>

<asp:BoundField DataField="Descripcion" HeaderText="Producto" />

<asp:BoundField DataField="cantidad" HeaderText="Cantidad" />

<asp:TemplateField HeaderText="Precio" ItemStyle-HorizontalAlign="Right" HeaderStyle-HorizontalAlign="Right" >

<ItemTemplate>

$<%# FormatNumber(Eval("Precio"), 2)%>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Total" ItemStyle-HorizontalAlign="Right" HeaderStyle-HorizontalAlign="Right" >

<ItemTemplate>

$<%# FormatNumber(Eval("Total"), 2)%>

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

<div style="float:left;">

<h4>Calcular Costo de envío</h4>

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem Text="CABA" />

<asp:ListItem Text="Buenos Aires" />

<asp:ListItem Text="Córdoba" />

</asp:DropDownList>

<asp:Button CssClass="btn btn-default" Text="Calcular" runat="server" onclick="CalcularEnvio\_Click"/>

<asp:TextBox cssclass="BuscarTxt" ID="txtCosto" runat="server" readonly="true"/>

</div>

<div style="float:right">

<br /><br />

<asp:Button Id="GuardarBtn" CssClass="btn btn-default" Text="Guardar compra" runat="server" />

</div>

</asp:Content>

Page\_Load

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

If (Page.IsPostBack = False) Then

mCarrito = DirectCast(Session("Carrito"), Entity.Carrito)

gvShoppingCart.DataSource = mCarrito.getLista

gvShoppingCart.DataBind()

End If

End Sub

Carrito\_PreLoad

Private Sub Carrito\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

'TODO: Activar verificacion de roles

' If Not Context.User.IsInRole("2") Then

'Response.Redirect("Default", True)

'End If

DirectCast(Session("Carrito"), Entity.Carrito).agregarProd(New Entity.Producto With {.Codigo = 1, .Descripcion = "Producto1", .cantidad = 1, .Precio = 2.5})

DirectCast(Session("Carrito"), Entity.Carrito).agregarProd(New Entity.Producto With {.Codigo = 1, .Descripcion = "Producto1", .cantidad = 1, .Precio = 2.5})

DirectCast(Session("Carrito"), Entity.Carrito).agregarProd(New Entity.Producto With {.Codigo = 2, .Descripcion = "Producto2", .cantidad = 1, .Precio = 6.4000000000000004})

End Sub

gvShoppingCart\_RowDataBound

Private Sub gvShoppingCart\_RowDataBound(sender As Object, e As GridViewRowEventArgs) Handles gvShoppingCart.RowDataBound

Debug.Write("Prueba")

Try

If e.Row.RowType = DataControlRowType.Footer Then

e.Row.Cells(3).Text = "$" + FormatNumber(mCarrito.Total, 2).ToString

End If

Catch

End Try

End Sub

gvShoppingCart\_RowCommand

Private Sub gvShoppingCart\_RowCommand(sender As Object, e As GridViewCommandEventArgs) Handles gvShoppingCart.RowCommand

Debug.Write("Prueba2")

End Sub

CalcularEnvio\_Click

Protected Sub CalcularEnvio\_Click(sender As Object, e As EventArgs)

Try

Dim mWebService As New CostoEnvioWebService.ClavoService

txtCosto.Text = "$" + FormatNumber(mWebService.CostoEnvio(DropDownList1.Text), 2).ToString

Catch

End Try

End Sub

GuardarBtn\_Click

Private Sub GuardarBtn\_Click(sender As Object, e As EventArgs) Handles GuardarBtn.Click

mCarrito = DirectCast(Session("Carrito"), Entity.Carrito)

Dim vEscritor As New XmlTextWriter(Server.MapPath("Carrito de " & Context.User.Identity.Name & ".xml"), Nothing)

vEscritor.Formatting = Formatting.Indented

vEscritor.WriteStartDocument()

vEscritor.WriteComment("Carrito de compras de " & Context.User.Identity.Name)

vEscritor.WriteStartElement("Carrito")

For Each Producto As Entity.Producto In mCarrito.getLista

vEscritor.WriteStartElement("Producto")

vEscritor.WriteStartElement("Codigo", "")

vEscritor.WriteString(Producto.Codigo)

vEscritor.WriteEndElement()

vEscritor.WriteStartElement("Descripcion", "")

vEscritor.WriteString(Producto.Descripcion)

vEscritor.WriteEndElement()

vEscritor.WriteStartElement("Precio", "")

vEscritor.WriteString(Producto.Precio)

vEscritor.WriteEndElement()

vEscritor.WriteStartElement("Cantidad", "")

vEscritor.WriteString(Producto.cantidad)

vEscritor.WriteEndElement()

vEscritor.WriteStartElement("Total", "")

vEscritor.WriteString(Producto.total)

vEscritor.WriteEndElement()

vEscritor.WriteEndElement()

Next

vEscritor.WriteStartElement("Total del carrito", "")

vEscritor.WriteString(mCarrito.Total)

vEscritor.WriteEndDocument()

vEscritor.Flush()

vEscritor.Close()

MostrarMensaje()

End Sub

MostrarMensaje

Private Sub MostrarMensaje()

Dim message As String = "El carrito se guardó con éxito"

Dim sb As New System.Text.StringBuilder()

sb.Append("<script type = 'text/javascript'>")

sb.Append("window.onload=function(){")

sb.Append("alert('")

sb.Append(message)

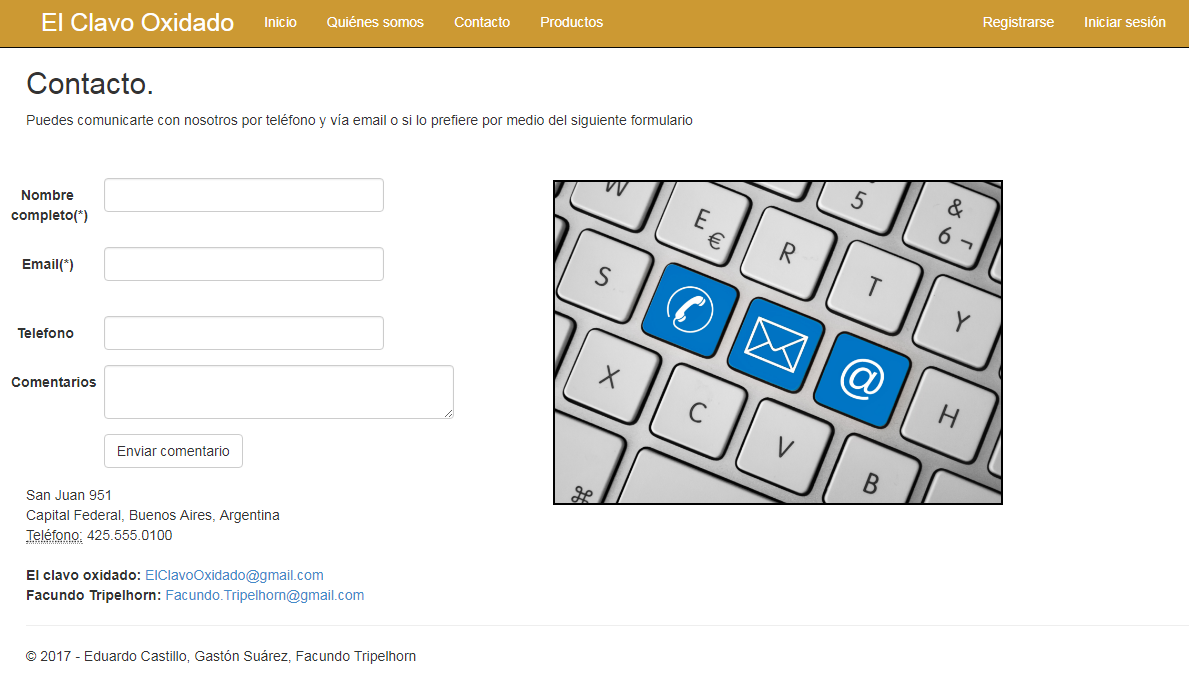
sb.Append("')};")

sb.Append("</script>")

ClientScript.RegisterClientScriptBlock(Me.GetType(), "alert", sb.ToString())

End Sub

# Contact.aspx



<%@ Page Title="Contacto" Language="VB" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Contact.aspx.vb" Inherits="ElClavoOxidado.Contact" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<h2><%: Title %></h2>

<p>Puedes comunicarte con nosotros por teléfono y vía email o si lo prefiere por medio del siguiente formulario</p>

<br />

<br />

<div class="form-left">

<div class="form-horizontal">

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="NombreYApellido" CssClass="col-md-2 control-label">Nombre completo(\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="NombreYApellido" CssClass="form-control" Width="350px" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="NombreYApellido"

CssClass="text-danger" ErrorMessage="Ingrese su nombre" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2 control-label">Email(\*)</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Email" CssClass="form-control" Width="350px" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Email"

CssClass="text-danger" ErrorMessage="Ingrese su dirección de email" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Telefono" CssClass="col-md-2 control-label">Telefono</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Telefono" CssClass="form-control" Width="350px" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Comentarios" CssClass="col-md-2 control-label">Comentarios</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Comentarios" CssClass="form-control" TextMode="MultiLine" Width="350px" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" OnClick="EnviarComentario" Text="Enviar comentario" CssClass="btn btn-default" />

</div>

</div>

</div>

</div>

<div class="form-right">

<asp:Image ID="ContactoImg" runat="server" ImageUrl="~/Images/contacto.jpg" Width="450px" BorderColor="Black" BorderStyle="Solid" BorderWidth="2px" />

</div>

<address>

San Juan 951<br />

Capital Federal, Buenos Aires, Argentina<br />

<abbr title="Phone">Teléfono:</abbr> 425.555.0100

</address>

<address>

<strong>El clavo oxidado:</strong>

<a href="mailto:ElClavoOxidado@gmail.com">ElClavoOxidado@gmail.com</a><br />

<strong>Facundo Tripelhorn:</strong>

<a href="mailto:Facundo.Tripelhorn@gmail.com">Facundo.Tripelhorn@gmail.com</a>

</address>

</asp:Content>

Contact\_PreLoad

Private Sub Contact\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

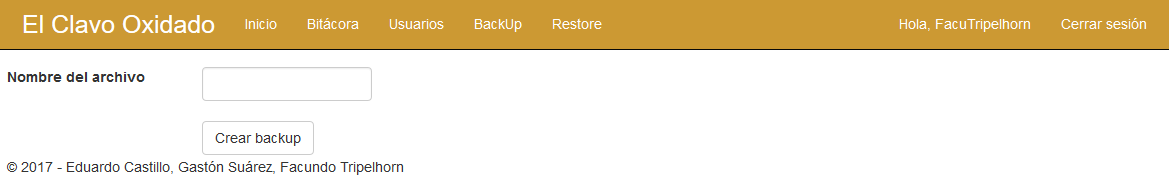
If Context.User.IsInRole("1") Or Context.User.IsInRole("3") Then

Response.Redirect("Default", True)

End If

End Sub

# Backup.aspx



<%@ Page Title="" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Backup.aspx.vb" Inherits="ElClavoOxidado.Backup" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<br />

<div>

<asp:PlaceHolder runat="server" ID="successMessage" Visible="false" ViewStateMode="Disabled">

<p class="text-success"><%:SuccessMessageText %></p>

</asp:PlaceHolder>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Nombre" CssClass="col-md-2 control-label">Nombre del archivo</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Nombre" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Nombre" CssClass="text-danger" ErrorMessage="Ingrese el nombre del archivo" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Crear backup" CssClass="btn btn-default" ID="BackupBtn" OnClick="BackupBtn\_Click" />

</div>

</div>

</asp:Content>

Backup\_PreLoad

Private Sub Backup\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("1") Then

Response.Redirect("Default", True)

End If

End Sub

BackupBtn\_Click

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

Dim \_usuarioEntity As New Entity.Usuario

\_usuarioEntity.Usuario = HttpContext.Current.User.Identity.Name

If Not Directory.Exists("C:\BK\") Then

Directory.CreateDirectory("C:\BK\")

End If

Dim \_path As String = "C:\BK\"

Dim \_nombre As String = Nombre.Text

\_backup.Crear\_Backup(\_path, \_nombre)

CargarBitacora(\_usuarioEntity.Usuario, \_nombre)

SuccessMessageText = "Se ha realizado el backup de la base exitosamente en el archivo " + \_path + \_nombre + ".bak"

successMessage.Visible = True

Nombre.Text = ""

End Sub

CargarBitacora

Private Sub CargarBitacora(\_usuario As String, \_nombre As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = \_usuario

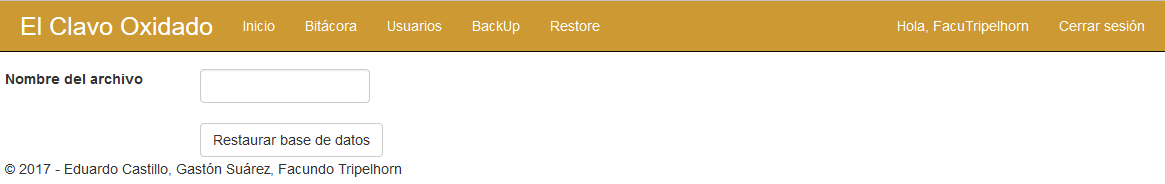
\_bitacoraEntity.Tipo = "Creación de Backup"

\_bitacoraEntity.Actividad = \_usuario & " creó el backup: " & \_nombre

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

# Restore.aspx



<%@ Page Title="" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="Restore.aspx.vb" Inherits="ElClavoOxidado.Restore" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<br />

<div>

<asp:PlaceHolder runat="server" ID="successMessage" Visible="false" ViewStateMode="Disabled">

<p class="text-success"><%:SuccessMessageText %></p>

</asp:PlaceHolder>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Nombre" CssClass="col-md-2 control-label">Nombre del archivo</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Nombre" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Nombre" CssClass="text-

danger" ErrorMessage="Ingrese el nombre del archivo" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Restaurar base de datos" CssClass="btn btn-default" ID="RestoreBtn" />

</div>

</div>

</asp:Content>

Restore\_PreLoad

Private Sub Restore\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("1") Then

Response.Redirect("Default", True)

End If

End Sub

RestoreBtn\_Click

Private Sub RestoreBtn\_Click(sender As Object, e As EventArgs) Handles RestoreBtn.Click

Dim \_usuarioEntity As New Entity.Usuario

\_usuarioEntity.Usuario = HttpContext.Current.User.Identity.Name

Dim \_path As String = "C:\BK\"

Dim \_nombre As String = Nombre.Text

\_backup.Restore\_BD(\_path, \_nombre)

CargarBitacora(\_usuarioEntity.Usuario, \_nombre)

SuccessMessageText = "La base de datos se restauró correctamente"

successMessage.Visible = True

Nombre.Text = ""

End Sub

CargarBitacora

Private Sub CargarBitacora(\_usuario As String, \_nombre As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = \_usuario

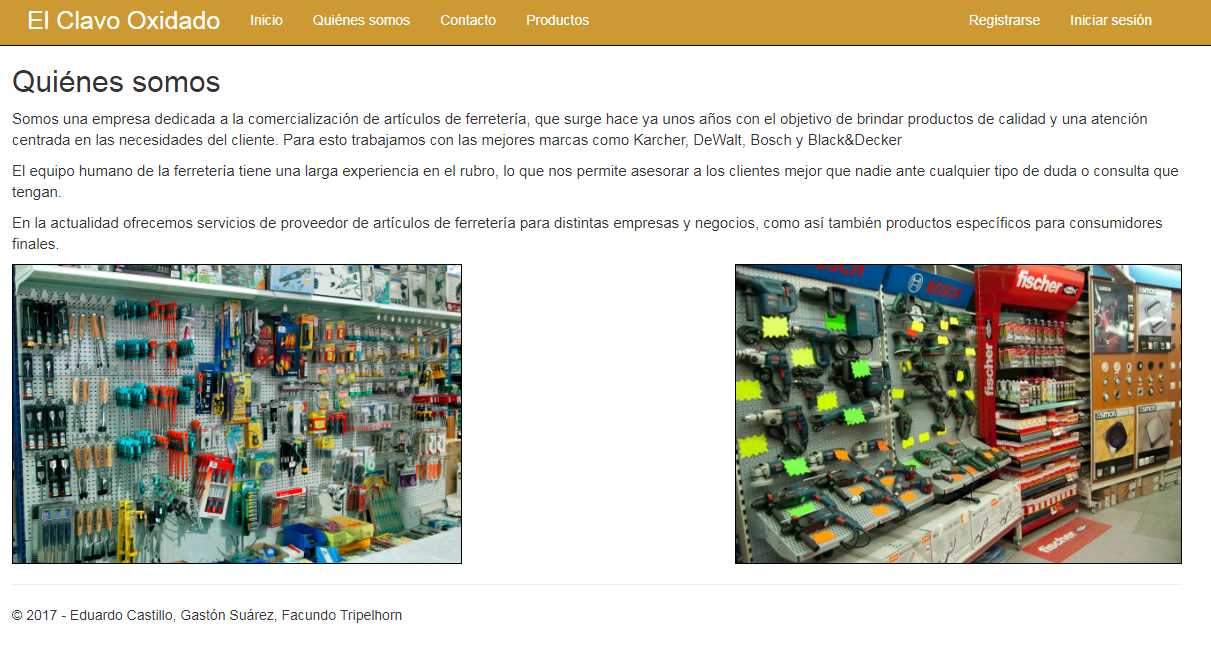
\_bitacoraEntity.Tipo = "Restauración de backup"

\_bitacoraEntity.Actividad = \_usuario & " restauró el backup: " & \_nombre

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

# About.aspx



<%@ Page Title="Quiénes somos" Language="VB" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="About.aspx.vb" Inherits="ElClavoOxidado.About" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<h2><%: Title %></h2>

<p class="About">Somos una empresa dedicada a la comercialización de artículos de ferretería, que surge hace ya unos años con el objetivo de brindar

productos de calidad y una atención centrada en las necesidades del cliente. Para esto trabajamos con las mejores marcas como Karcher,

DeWalt, Bosch y Black&Decker</p>

<p class="About">El equipo humano de la ferretería tiene una larga experiencia en el rubro, lo que nos permite asesorar a los clientes mejor que nadie

ante cualquier tipo de duda o consulta que tengan.</p>

<p class="About">En la actualidad ofrecemos servicios de proveedor de artículos de ferretería para distintas empresas y negocios, como así también

productos específicos para consumidores finales.</p>

<asp:Image ID="Herramientas1" runat="server" BorderColor="Black" BorderStyle="Solid" BorderWidth="1px" CssClass="Herramientas" ImageUrl="~/Images/herramientas.jpg" Height="300px" />

<asp:Image ID="Herramientas2" runat="server" BorderColor="Black" BorderStyle="Solid" BorderWidth="1px" CssClass="Herramientas" ImageUrl="~/Images/herramientas2.jpg" ImageAlign="Right" Height="300px" />

<br />

</asp:Content>

About\_PreLoad

Private Sub About\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Context.User.IsInRole("1") Or Context.User.IsInRole("3") Then

Response.Redirect("Default", True)

End If

End Sub

# EditarProductos.aspx

<%@ Page Title="" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="EditarProductos.aspx.vb" Inherits="ElClavoOxidado.EditarProductos" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<h1>Página en Construcción</h1>

<h2>Estamos trabajando para brindarle la mejor experiencia</h2>

</asp:Content>

EditarProductos\_PreLoad

Private Sub EditarProductos\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("2") Then

Response.Redirect("Default", True)

End If

End Sub

# ErrorPage.aspx

<%@ Page Title="" Language="vb" AutoEventWireup="false" MasterPageFile="~/Site.Master" CodeBehind="ErrorPage.aspx.vb" Inherits="ElClavoOxidado.ErrorPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<link rel="stylesheet" href="Content/estilo.css" />

<div class="container">

<asp:Label ID="Label1" runat="server" Text="Error en la base de datos, algunos campos fueron modificados manualmente" Font-Size="XX-Large"></asp:Label>

<br />

<br />

<asp:GridView ID="GrillaError" runat="server" >

<HeaderStyle BackColor="#BB9241" Height="30px" />

</asp:GridView>

<br />

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Nombre" CssClass="col-md-2 control-label">Nombre del archivo</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Nombre" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Nombre" CssClass="text-danger" ErrorMessage="Ingrese el nombre del archivo" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" Text="Restaurar base de datos" CssClass="btn btn-default" ID="RestoreBtn" />

</div>

</div>

</div>

</asp:Content>

Page\_Load

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

GrillaError.DataSource = Session("ListaError")

GrillaError.DataBind()

End Sub

ErrorPage\_PreLoad

Private Sub ErrorPage\_PreLoad(sender As Object, e As EventArgs) Handles Me.PreLoad

If Not Context.User.IsInRole("1") Then

Response.Redirect("Default", True)

End If

If Session("CheckError") = 0 Then

Response.Redirect("Default", True)

End If

End Sub

RestoreBtn\_Click

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

Dim \_usuarioEntity As New Entity.Usuario

\_usuarioEntity.Usuario = HttpContext.Current.User.Identity.Name

Dim \_path As String = "C:\BK\"

Dim \_nombre As String = Nombre.Text

\_backup.Restore\_BD(\_path, \_nombre)

Session("CheckError") = 0

CargarBitacora(\_usuarioEntity.Usuario, \_nombre)

Response.Redirect("Default", True)

End Sub

CargarBitacora

Private Sub CargarBitacora(\_usuario As String, \_nombre As String)

Dim \_bitacoraNeg As New Negocio.Bitacora

Dim \_bitacoraEntity As New Entity.Bitacora

Dim \_listaBitacora = \_bitacoraNeg.Obtener\_Bitacora()

If \_listaBitacora.Count > 0 Then

\_bitacoraEntity.Id = \_listaBitacora.Last.Id + 1

Else

\_bitacoraEntity.Id = 1

End If

\_bitacoraEntity.Usuario = \_usuario

\_bitacoraEntity.Tipo = "Restauración de backup"

\_bitacoraEntity.Actividad = \_usuario & " restauró el backup: " & \_nombre

\_bitacoraNeg.Alta\_Bitacora(\_bitacoraEntity)

End Sub

# AdRotator.xml

<?xml version="1.0" encoding="utf-8" ?>

<Advertisements>

<Ad>

<ImageUrl>Images\atencionycalidad.jpg</ImageUrl>

<NavigateUrl>http://www.google.com</NavigateUrl>

<AlternateText>

atencion y calidad

</AlternateText>

<Impressions>20</Impressions>

<Keyword>atencion</Keyword>

</Ad>

<Ad>

<ImageUrl>Images\nuestramarcas.jpg</ImageUrl>

<NavigateUrl>http://www.google.com</NavigateUrl>

<AlternateText>marcas</AlternateText>

<Impressions>20</Impressions>

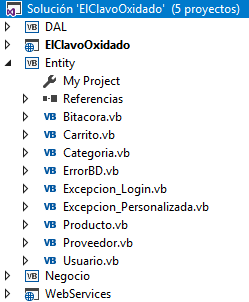
<Keyword>banner</Keyword>

</Ad>

</Advertisements>

# Capa de Entidades

Esta capa representa los objetos utilizados por la aplicación. Solo define esta capa las propiedades de los objetos.





# Usuario.vb

Public Class Usuario

Public Property Usuario As String

Public Property Email As String

Public Property Password As String

Public Property IsValid As Boolean

Public Property Bloqueado As Boolean

Public Property Cant\_Bloqueos As Integer

Public Property Ultimo\_intento As Date

Public Property Familia As Integer

Sub New()

End Sub

Sub New(\_usuario As String, \_email As String, \_password As String, \_familia As Integer, \_bloqueado As Boolean)

Usuario = \_usuario

Email = \_email

Password = \_password

Familia = \_familia

Bloqueado = \_bloqueado

End Sub

End Class

# Bitacora.vb

Public Class Bitacora

Property Id As Integer

Property Usuario As String

Property Fecha As Date

Property Tipo As String

Property Actividad As String

Sub New()

End Sub

Sub New(pUsuario As String, pFecha As Date, pTipo As String, pActividad As String)

Usuario = pUsuario

Fecha = pFecha

Tipo = pTipo

Actividad = pActividad

End Sub

Sub New(pId As Integer, pUsuario As String, pFecha As Date, pTipo As String, pActividad As String)

Id = pId

Usuario = pUsuario

Fecha = pFecha

Tipo = pTipo

Actividad = pActividad

End Sub

End Class

# Categoria.vb

Public Class Categoria

Property Id As Integer

Property Nombre As String

Sub New()

End Sub

Sub New(\_Id As Integer, \_Nombre As String)

Id = \_Id

Nombre = \_Nombre

End Sub

End Class

# Carrito.vb

Imports System.Xml

Public Class Carrito

Dim mListaProd As New List(Of Producto)

Public Sub agregarProd(pProd As Producto)

If validarExistencia(pProd.Codigo) = False Then

mListaProd.Add(pProd)

End If

End Sub

Public Sub quitarProd(pProd As Producto)

Try

mListaProd.Remove(pProd)

Catch ex As Exception

End Try

End Sub

Public Function getLista() As List(Of Producto)

Return mListaProd

End Function

Private Function validarExistencia(pProdId As Integer) As Boolean

Dim answer As Boolean = False

For Each pProd As Producto In mListaProd

If pProd.Codigo = pProdId Then

answer = True

pProd.cantidad += 1

Exit For

End If

Next

Return answer

End Function

ReadOnly Property Total As Decimal

Get

Dim mTotal As Decimal = 0

For Each mProd As Producto In mListaProd

mTotal += mProd.total

Next

Return mTotal

End Get

End Property

End Class

# ErrorBD.vb

Public Class ErrorBD

Property Id As Integer

Property Tabla As String

Property Fila As Integer

Sub New()

End Sub

Sub New(\_id As Integer, \_tabla As String, \_fila As Integer)

Id = \_id

Tabla = \_tabla

Fila = \_fila

End Sub

End Class

# Excepcion\_Login.vb

Public Class Excepcion\_Login

Inherits Excepcion\_Personalizada

Public Overrides Property Codigo As String

Public Overrides Property Descripcion As String

Sub New(\_mensaje As String)

Codigo = "C001"

Descripcion = \_mensaje

End Sub

End Class

# Excepcion\_Personalizada.vb

Public MustInherit Class Excepcion\_Personalizada

Inherits System.Exception

MustOverride Property Codigo As String

MustOverride Property Descripcion As String

End Class

# Producto.vb

Public Class Producto

Property Codigo As Integer

Property Categoria As Categoria

Property Proveedor As Proveedor

Property Descripcion As String

Property Precio As Decimal

ReadOnly Property total As Decimal

Get

Return Precio \* cantidad

End Get

End Property

Property cantidad As Integer

Sub New()

End Sub

Sub New(\_codigo As Integer, \_categoria As Categoria, \_proveedor As Proveedor, \_descripcion As String, \_precio As Decimal)

Codigo = \_codigo

Categoria = \_categoria

Proveedor = \_proveedor

Descripcion = \_descripcion

Precio = \_precio

End Sub

End Class

# Proveedor.vb

Public Class Proveedor

Property Id As Integer

Property Nombre As String

Sub New()

End Sub

Sub New(\_Id As Integer, \_Nombre As String)

Id = \_Id

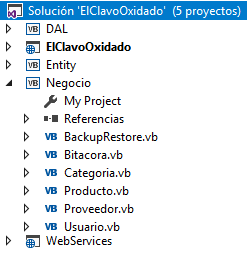
Nombre = \_Nombre

End Sub

End Class

# Capa de Negocio

En la solución, el proyecto denominado “Negocio” contiene las clases con la lógica de negocio.





# Usuario

# Alta\_Usuario

Esta rutina registra un nuevo usuario persistiéndolo en la base de datos.

Public Sub Alta\_Usuario(\_usuario As Entity.Usuario)

\_dal.Alta\_Usuario(\_usuario)

End Sub

# Chequear\_Usuario

Esta función permite verificar contra la base de datos del sistema, si el nombre de usuario recibido por parámetro existe. En caso afirmativo se obtienen todos los datos asociados al mismo.

Public Function Chequear\_Usuario(\_username As String) As Boolean

Return \_dal.Chequear\_Usuario(\_username)

End Function

# Get\_Usuarios

Esta rutina permite recuperar todos los usuarios registrados en la página web.

Public Function Get\_Usuarios() As List(Of Entity.Usuario)

Return \_dal.Get\_Usuarios

End Function

# Cambiar\_Contraseña

Esta rutina permite registrar la actualización de la contraseña del usuario. Se recibe como parámetro tanto el nombre del usuario a actualizar como el hash correspondiente a la contraseña.

Public Sub Cambiar\_Contraseña(\_username As String, \_password As String)

\_dal.Cambiar\_Contraseña(\_username, \_password)

End Sub

# Asignar\_Rol

Esta rutina permite registrar la asignación de roles para un usuario determinado. Se recibe como parámetro tanto el nombre del usuario a actualizar como el la familia a la cual se desea asociar el mismo.

Public Sub Asignar\_Rol(\_username As String, \_familia As Integer)

\_dal.Asignar\_Rol(\_username, \_familia)

End Sub

# Validar\_Login

Esta rutina realiza las validaciones necesarias para conceder autorización a la página web. A partir del usuario ingresado se recuperan sus perfiles asociados lo que permite acceder a las funcionalidades permitidas para el mismo.

Public Function Validar\_Login(\_user As Entity.Usuario, \_pass As String) As Boolean

***' Validamos el usuario***

\_dal.Obtener\_Usuario(\_user, \_pass)

***' Verificamos que sea un usuario registrado***

If \_user.Usuario Is Nothing And \_user.IsValid = False Then

Dim \_ex As New Entity.Excepcion\_Login("Usuario invalido")

Throw \_ex

End If

***' Verificamos que no supere la cantidad de intentos - Caso contrario usuario bloqueado***

If \_user.Bloqueado = True Then

Dim \_ex As New Entity.Excepcion\_Login("Usuario Bloqueado")

Throw \_ex

End If

***' Si usuario esta informado pero no es valido es error en clave***

If \_user.IsValid = False Then

If \_user.Cant\_Bloqueos >= 3 Then

\_user.Bloqueado = True

***' Actualizo el usuario y retorno el error***

\_dal.Bloquear\_Usuarios(\_user)

Dim \_ex As New Entity.Excepcion\_Login("Usuario Bloqueado")

Throw \_ex

End If

Dim \_ex2 As New Entity.Excepcion\_Login("Password invalida")

Throw \_ex2

End If

***'Obtenemos los perfiles habilitados***

***' Si la validaciones son correcta retornamos el usuario y true indicando que es un usuario valido.***

Return True

End Function

# Bitacora

# Obtener\_Bitacora

Esta rutina permite recuperar todos los eventos registrados en la bitácora para listar en un reporte.

Public Function Obtener\_Bitacora() As List(Of Entity.Bitacora)

Return \_BitacoraDAL.Obtener\_Bitacora()

End Function

# Alta\_Bitacora

Esta rutina permite la registración de la bitácora en la base de datos de la aplicación.

Public Sub Alta\_Bitacora(ByRef bitacora As Entity.Bitacora)

\_BitacoraDAL.Alta\_Bitacora(bitacora)

End Sub

# Categoria

# Check\_Categoria

Esta rutina permite verificar contra la base de datos la existencia de la categoría recibida por parámetro.

Public Function Check\_Categoria(\_categoria As Entity.Categoria) As Boolean

Return \_dal.Check\_Categoria(\_categoria)

End Function

# Get\_Categoria

Esta rutina permite recuperar a partir del ID recibido por parámetro la categoría correspondiente a dicho ID.

Public Function Get\_Categoria(\_id As Integer) As Entity.Categoria

Return \_dal.Get\_Categoria(\_id)

End Function

# Get\_Categorias

Esta rutina permite recuperar un listado con las categorías vigentes en el sistema.

Public Function Get\_Categorias() As List(Of Entity.Categoria)

Return \_dal.Get\_Categorias

End Function

# Proveedor

# Check\_Proveedor

Esta rutina permite verificar contra la base de datos la existencia del proveedor recibido por parámetro.

Public Function Check\_Proveedor(\_proveedor As Entity.Proveedor) As Boolean

Return \_dal.Check\_Proveedor(\_proveedor)

End Function

# Get\_Proveedor

Esta rutina permite recuperar a partir del ID recibido por parámetro el proveedor correspondiente al mismo.

Public Function Get\_Proveedor(\_id As Integer) As Entity.Proveedor

Return \_dal.Get\_Proveedor(\_id)

End Function

# Get\_Proveedores

Esta rutina permite recuperar un listado con los proveedores vigentes en el sistema.

Public Function Get\_Proveedores() As List(Of Entity.Proveedor)

Return \_dal.Get\_Proveedores

End Function

# BackupRestore

# Crear\_Backup

Esta rutina permite generar un back up completo de la base de datos del sistema. Para ello se debe enviar por parámetro la ruta y nombre destino con el cual se desea guardar el backup.

Public Sub Crear\_Backup(\_path As String, \_nombre As String)

\_dal.Crear\_Backup(\_path, \_nombre)

End Sub

# Restore\_BD

Esta rutina permite restaurar de forma completa la base de datos del sistema a partir de un backup generado previamente. Para ello se debe enviar por parámetro la ruta y nombre del backup a restaurar.

Public Sub Restore\_BD(\_path As String, \_nombre As String)

\_dal.Restore\_BD(\_path, \_nombre)

End Sub

# Producto

# Check\_Producto

Esta rutina permite verificar contra la base de datos la existencia del producto recibido por parámetro.

Public Function Check\_Producto(\_producto As Entity.Producto) As Boolean

Return \_dal.Check\_Producto(\_producto)

End Function

# Obtener\_Productos

Esta rutina permite recuperar un listado con los productos vigentes en el sistema.

Public Function Obtener\_Productos() As List(Of Entity.Producto)

Return \_dal.Obtener\_Productos

End Function

# Alta\_Producto

Esta rutina permite dar de alta un nuevo producto en el sistema. Es función de la rutina que persiste el producto verificar que el mismo no exista ya dado de alta previamente.

Public Sub Alta\_Producto(\_producto As Entity.Producto)

\_dal.Alta\_Producto(\_producto)

End Sub

# Get\_Producto

Esta rutina permite recuperar a partir del código recibido por parámetro el producto asociado al mismo.

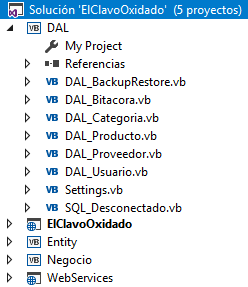
Public Function Get\_Producto(\_codigo As Integer) As Entity.Producto

Return \_dal.Get\_Producto(\_codigo)

End Function

# Capa de Datos

En la solución, el proyecto denominado “DAL” contiene las clases que permiten el acceso a los datos.





# DAL\_Usuario

# Función Alta\_Usuario

Define el acceso a la base para registrar un nuevo usuario. Utiliza el store procedure AltaUsuario el cual requiere como parámetros de entradas los datos del usuario a persistir. Es requisito que la contraseña recibida para el usuario se encuentre encriptada.

Public Sub Alta\_Usuario(ByRef usuario As Entity.Usuario)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "AltaUsuario"

\_parametros.Clear()

\_parametros.Add("@Usuario", usuario.Usuario)

\_parametros.Add("@Email", usuario.Email)

\_parametros.Add("@Password", usuario.Password)

\_parametros.Add("@Familia", usuario.Familia)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# Bloquear\_Usuarios

Define la forma de bloquear a un usuario en particular como consecuencia de haber alcanzado el límite de intentos erróneos de accesos definidos por el negocio. Para ello se hace efectivo el bloqueo a través del store procedure “BloquearUser” enviando como único parámetro el nombre de usuario.

Public Sub Bloquear\_Usuarios(ByRef usuario As Entity.Usuario)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet ***' Obtengo de la BD los datos***

Dim \_storeprocedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeprocedure = "BloquearUser" ***' Nombre del store procedure de persistencia***

\_parametros.Clear()

\_parametros.Add("@Usuario", usuario.Usuario)

***' Obtengo el tipo***

\_ds = \_dal.EjecutarEscalar\_StoreProcedure(\_storeprocedure, \_parametros)

End Sub

# Obtener\_Usuarios

Define el acceso a la base para validar un usuario a partir del usuario y contraseña ingresada para efectuar un login. La función define el store procedure ObtenerUser que será el encargado de ejecutar la acción en la base de datos. Para ello al store es necesario pasarle por parámetro el nombre de usuario y el hash de la contraseña ingresada.

Public Sub Obtener\_Usuario(ByRef usuario As Entity.Usuario, \_password As String)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet ***' Obtengo de la BD los datos***

Dim \_storeprocedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeprocedure = "ObtenerUser" ***' Nombre del store procedure de persistencia***

\_parametros.Clear()

\_parametros.Add("@Usuario", usuario.Usuario)

\_parametros.Add("@Pass", \_password)

***' Obtengo el tipo***

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeprocedure, \_parametros)

usuario.Usuario = Nothing

If \_ds.Tables(0).Rows.Count > 0 Then

For Each Item As DataRow In \_ds.Tables(0).Rows

usuario.Usuario = Item("Usuario")

usuario.IsValid = Item("IsValid")

usuario.Bloqueado = Item("Bloqueado")

usuario.Familia = Item("Familia")

Next

Else

End If

End Sub

# Cambiar\_Contraseña

Define el acceso a la base para actualizar la contraseña del usuario recibido por parámetro. Es requisito que la contraseña recibida para el usuario se encuentre encriptada.

Public Sub Cambiar\_Contraseña(\_username As String, \_password As String)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "CambiarContraseña"

\_parametros.Clear()

\_parametros.Add("@Usuario", \_username)

\_parametros.Add("@Password", \_password)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# Chequear\_Usuario

Define el acceso a la base para verificar si existe el nombre de usuario ingresado durante el proceso de login.

Public Function Chequear\_Usuario(username As String) As Boolean

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "ChequearUsuario"

\_parametros.Clear()

\_parametros.Add("@Usuario", username)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Return False

Else

Return true

End If

End Function

# Asignar\_Rol

Define el acceso a la base para registrar la asignación de un rol al usuario recibido por parámetro.

Public Sub Asignar\_Rol(\_username As String, \_familia As Integer)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "AsignarRol"

\_parametros.Clear()

\_parametros.Add("@Usuario", \_username)

\_parametros.Add("@Familia", \_familia)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# Get\_Usuarios

Define el acceso a la base para obtener un listado con todos los usuarios del sistema.

Public Function Get\_Usuarios() As List(Of Entity.Usuario)

Dim \_listaUsuarios As New List(Of Entity.Usuario)

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

\_storeProcedure = "GetUsuarios"

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure)

For Each Item As DataRow In \_ds.Tables(0).Rows

\_listaUsuarios.Add(New Entity.Usuario(Item(0), Item(1), Item(2), Item(3), Item(4)))

Next

Catch ex As Exception

End Try

Return \_listaUsuarios

End Function

# DAL\_Bitacora

# Función Obtener\_Bitacora

Define el acceso a la base para obtener los registros de la bitácora y guardarlos en una lista.

Public Function Obtener\_Bitacora() As List(Of Bitacora)

Dim \_listaBitacora As New List(Of Bitacora)

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

\_storeProcedure = "GetBitacora"

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure)

For Each Item As DataRow In \_ds.Tables(0).Rows

\_listaBitacora.Add(New Bitacora(Item(0), Item(1), Item(2), Item(3), Item(4)))

Next

Catch ex As Exception

End Try

Return \_listaBitacora

End Function

# Función AltaBitacora

Define el acceso a la base para dar de alta un nuevo registro en la bitácora, recibiendo como parámetros de entrada el id de la actividad, el usuario que la haya realizado, el tipo de actividad y la actividad.

Public Sub Alta\_Bitacora(ByRef bitacora As Entity.Bitacora)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "AltaBitacora"

\_parametros.Clear()

\_parametros.Add("@Id", bitacora.Id)

\_parametros.Add("@Usuario", bitacora.Usuario)

\_parametros.Add("@Tipo", bitacora.Tipo)

\_parametros.Add("@Actividad", bitacora.Actividad)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# DAL\_Producto

# Función AltaProducto

Define el acceso a la base para dar de alta un nuevo registro de producto.  
Se debe recibir por parámetros todos los datos asociados al producto. Será el store procedure invocado el encargado de validar que el producto a dar de alta no exista previamente en la base de datos.

Public Sub Alta\_Producto(\_producto As Producto)

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "AltaProducto"

\_parametros.Clear()

\_parametros.Add("@Codigo", \_producto.Codigo)

\_parametros.Add("@Categoria", \_producto.Categoria.Id)

\_parametros.Add("@Proveedor", \_producto.Proveedor.Id)

\_parametros.Add("@Descripcion", \_producto.Descripcion)

\_parametros.Add("@Precio", \_producto.Precio)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# Función Check\_Producto

Define el método para consultar los productos existentes en la base de datos y recuperar los datos asociados a los mismos.

Public Function Check\_Producto(\_producto As Producto) As Boolean

Try

Dim \_ds As DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "CheckProducto"

\_parametros.Clear()

\_parametros.Add("@Codigo", \_producto.Codigo)

\_parametros.Add("@Categoria", \_producto.Categoria.Id)

\_parametros.Add("@Proveedor", \_producto.Proveedor.Id)

\_parametros.Add("@Descripcion", \_producto.Descripcion)

\_parametros.Add("@Precio", \_producto.Precio)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Return True

Else

Return False

End If

Catch ex As Exception

Return False

End Try

End Function

# Get\_Producto

Permite obtener un producto en particular a partir de su codigo.

Public Function Get\_Producto(\_codigo As Integer) As Producto

Try

Dim \_producto As New Producto

Dim \_ds As New DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "GetProducto"

\_parametros.Clear()

\_parametros.Add("@Codigo", \_codigo)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Dim \_drow As DataRow = \_ds.Tables(0).Rows(0)

\_producto.Codigo = \_drow(0)

\_producto.Categoria = \_dalcategoria.Get\_Categoria(\_drow(1))

\_producto.Proveedor = \_dalproveedor.Get\_Proveedor(\_drow(2))

\_producto.Descripcion = \_drow(3)

\_producto.Precio = \_drow(4)

End If

Return \_producto

Catch ex As Exception

Return Nothing

End Try

End Function

# Obtener\_Productos

Permite obtener el listado de productos vigentes.

Public Function Obtener\_Productos() As List(Of Producto)

Dim \_listaProductos As New List(Of Producto)

Try

Dim \_ds As DataSet

Dim \_storeProcedure As String

\_storeProcedure = "GetProductos"

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure)

For Each Item As DataRow In \_ds.Tables(0).Rows

Dim \_categoria As Categoria = \_dalcategoria.Get\_Categoria(Item(1))

Dim \_proveedor As Proveedor = \_dalproveedor.Get\_Proveedor(Item(2))

\_listaProductos.Add(New Producto(Item(0), \_categoria, \_proveedor, Item(3), Item(4)))

Next

Catch ex As Exception

End Try

Return \_listaProductos

End Function

# DAL\_BackupRestore

# Función Crear\_Backup

Define el método para acceder a la base de datos a realizar un backup completo. La copia se genera en la ruta indicada por parámetro bajo el nombre recibido.

Public Sub Crear\_Backup(\_path As String, \_nombre As String)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "CrearBackup"

\_parametros.Clear()

\_parametros.Add("@Path", \_path)

\_parametros.Add("@Nombre", \_nombre)

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

End Sub

# Función Restore\_BD

Define el método para acceder a la base de datos con el fin de restaurar un backup en forma completa. Se accede al directorio de la ruta recibida por parámetro junto con el nombre para recuperar el backup de la base.

Public Sub Restore\_BD(\_path As String, \_nombre As String)

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

Dim \_archivo As String = \_path + \_nombre + ".bak"

\_storeProcedure = "KillProcess"

\_parametros.Clear()

\_parametros.Add("@dbName", "ElClavoOxidado")

\_dal.EjecutarNonQuery\_StoreProcedure(\_storeProcedure, \_parametros)

\_dal.ConexionIniciar()

\_dal.EjecutarConsulta("USE master; RESTORE DATABASE ElClavoOxidado FROM DISK = '" & \_archivo & "' WITH REPLACE")

End Sub

# DAL\_Categoria

# Función Check\_Categoria

Esta función permite comprobar para la categoría recibida por parámetro si la misma existe o no en el sistema.

Public Function Check\_Categoria(\_categoria As Categoria) As Boolean

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "CheckCategoria"

\_parametros.Clear()

\_parametros.Add("@Id", \_categoria.Id)

\_parametros.Add("@Nombre", \_categoria.Nombre)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Return True

Else

Return False

End If

Catch ex As Exception

Return False

End Try

End Function

# Función Get\_Categorias

Define el método para recuperar todas las categorías registradas en el sistema.

Public Function Get\_Categorias() As List(Of Categoria)

Dim \_listaCategoria As New List(Of Categoria)

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

\_storeProcedure = "GetCategorias"

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure)

For Each Item As DataRow In \_ds.Tables(0).Rows

\_listaCategoria.Add(New Categoria(Item(0), Item(1)))

Next

Catch ex As Exception

End Try

Return \_listaCategoria

End Function

# Función Get\_Categorias

Esta función define la forma de recuperar una categoría existente a partir del ID recibido por parámetro.

Public Function Get\_Categoria(\_id As Integer) As Categoria

Try

Dim \_categoria As New Categoria

Dim \_dal As New SQL\_Desconectado

Dim \_ds As New DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "GetCategoria"

\_parametros.Clear()

\_parametros.Add("@Id", \_id)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Dim \_drow As DataRow = \_ds.Tables(0).Rows(0)

\_categoria.Id = \_drow(0)

\_categoria.Nombre = \_drow(1)

End If

Return \_categoria

Catch ex As Exception

Return Nothing

End Try

End Function

# DAL\_Proveedor

# Función Check\_Proveedor

Esta función permite comprobar para el proveedor recibido por parámetro si el mismo existe o no en el sistema.

Public Function Check\_Proveedor(\_proveedor As Proveedor) As Boolean

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "CheckProveedor"

\_parametros.Clear()

\_parametros.Add("@Id", \_proveedor.Id)

\_parametros.Add("@Nombre", \_proveedor.Nombre)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Return True

Else

Return False

End If

Catch ex As Exception

Return False

End Try

End Function

# Función Get\_Proveedores

Define el método para recuperar todos los proveedores registrados en el sistema.

Public Function Get\_Proveedores() As List(Of Proveedor)

Dim \_listaProveedores As New List(Of Proveedor)

Try

Dim \_dal As New DAL.SQL\_Desconectado

Dim \_ds As DataSet

Dim \_storeProcedure As String

\_storeProcedure = "GetProveedores"

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure)

For Each Item As DataRow In \_ds.Tables(0).Rows

\_listaProveedores.Add(New Proveedor(Item(0), Item(1)))

Next

Catch ex As Exception

End Try

Return \_listaProveedores

End Function

# Función Get\_Proveedor

Esta función define la forma de recuperar un proveedor existente a partir del ID recibido por parámetro.

Public Function Get\_Proveedor(\_id As Integer) As Proveedor

Try

Dim \_proveedor As New Proveedor

Dim \_dal As New SQL\_Desconectado

Dim \_ds As New DataSet

Dim \_storeProcedure As String

Dim \_parametros As New Dictionary(Of String, Object)

\_storeProcedure = "GetProveedor"

\_parametros.Clear()

\_parametros.Add("@Id", \_id)

\_ds = \_dal.Obtener\_DatasetStoreProcedure(\_storeProcedure, \_parametros)

If \_ds.Tables(0).Rows.Count > 0 Then

Dim \_drow As DataRow = \_ds.Tables(0).Rows(0)

\_proveedor.Id = \_drow(0)

\_proveedor.Nombre = \_drow(1)

End If

Return \_proveedor

Catch ex As Exception

Return Nothing

End Try

End Function

# SQL\_Desconectado

Esta clase es la encargada de efectuar el acceso a la base de datos.

La clase cuenta con las siguientes definiciones para iniciar la conexión con la base de datos “ElClavoOxidado”:

La base de datos se aloja en el mismo servidor que la aplicación.

La seguridad es integrada a Windows

Sub New()

Dim server As String = "."

Try

ConexionIniciar()

Catch \_ex As Exception'

End Try

End Sub

# ConexionIniciar

Public Sub ConexionIniciar()

Try

Dim connectionString As String = "Data Source=.;Initial Catalog=ElClavoOxidado;Integrated Security=True"

\_sqlConnection = New SqlConnection(connectionString)

\_sqlConnection.Open()

Catch ex As Exception

End Try

End Sub

# ConexionFinalizar

Public Sub ConexionFinalizar()

Try

\_sqlConnection.Close()

SqlConnection.ClearAllPools()

Catch ex As Exception

End Try

End Sub

# CambiarBase

Public Sub CambiarBase(\_string As String)

\_sqlConnection.ChangeDatabase(\_string)

End Sub

# Obtener\_DatasetStoreProcedure

Public Function Obtener\_DatasetStoreProcedure(sp As String, Optional parametros As Dictionary(Of String, Object) = Nothing)

Try

Dim unComando As New SqlCommand()

***'1) La conexion abierta.***

unComando.Connection = \_sqlConnection

***'2) Texto de la consulta.***

unComando.CommandText = sp

If Not parametros Is Nothing Then

For Each p In parametros

unComando.Parameters.AddWithValue(p.Key, p.Value)

Next

End If

***'3) Tipo de consulta.***

unComando.CommandType = CommandType.StoredProcedure

***'SE CREA EL OBJETO DATAADAPTER PARA LLENAR EL DATASET***

mDa = New SqlDataAdapter(unComando)

mDs = New DataSet

mDa.Fill(mDs)

ConexionFinalizar()

Catch ex As Exception

ConexionFinalizar()

End Try

Return mDs

End Function

# EjecutarConsulta

Public Sub EjecutarConsulta(\_consulta As String)

Dim \_comando As New SqlCommand

Try

\_comando.Connection = \_sqlConnection

\_comando.CommandText = \_consulta

\_comando.ExecuteNonQuery()

\_comando.Connection.Close()

Catch ex As Exception

\_comando.connection.close

End Try

End Sub

# Función EjecutarEscalar\_StoreProcedure

Public Function EjecutarEscalar\_StoreProcedure(consulta As String, parametros As Dictionary(Of String, Object)) As Object

Try

Dim unComando As New SqlCommand()

***'1) La conexion abierta.***

unComando.Connection = \_sqlConnection

'unComando.Transaction = \_unaTransaccion

***'2) Texto de la consulta.***

unComando.CommandText = consulta

For Each p In parametros

unComando.Parameters.AddWithValue(p.Key, p.Value)

Next

***'3) Tipo de consulta.***

unComando.CommandType = CommandType.StoredProcedure

***'4) Ejecutar y esperar el resultado.***

Return unComando.ExecuteScalar()

ConexionFinalizar()

Catch ex As Exception

ConexionFinalizar()

Return -1

End Try

End Function

# Función EjecutarNonQuery\_StoreProcedure

Public Sub EjecutarNonQuery\_StoreProcedure(sp As String, Optional parametros As Dictionary(Of String, Object) = Nothing)

Try

Dim unComando As New SqlCommand()

***'1) La conexion abierta.***

unComando.Connection = \_sqlConnection

***'2) Texto de la consulta.***

unComando.CommandText = sp

unComando.Parameters.Clear()

If Not parametros Is Nothing Then

For Each p In parametros

unComando.Parameters.AddWithValue(p.Key, p.Value)

Next

End If

***'3) Tipo de consulta.***

unComando.CommandType = CommandType.StoredProcedure

***'4) Ejecutar y esperar el resultado.***

unComando.ExecuteNonQuery()

ConexionFinalizar()

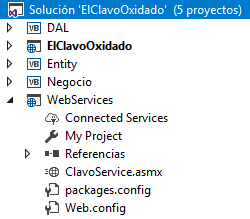
Catch ex As Exception

ConexionFinalizar()

End Try

End Sub

# Capa Web Services



|  |  |  |
| --- | --- | --- |
| 7.1. ClavoService.asmx | Descripción | Quien lo consume? |
| Método CostoEnvio | Servicio que a partir de la provincia recibida por parámetro permite determinar el costo de envío que deberá adicionarse a un pedido. | 3.9 - Carrito.aspx a través del método CalcularEnvio\_Click |



# ClavoService.asmx

Imports System.Web.Services

Imports System.Web.Services.Protocols

Imports System.ComponentModel

<System.Web.Services.WebService(Namespace:="http://tempuri.org/")> \_

<System.Web.Services.WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1\_1)> \_

<ToolboxItem(False)> \_

Public Class ClavoService

Inherits System.Web.Services.WebService

<WebMethod()>

Public Function CostoEnvio(pProvincia As String) As Decimal

Return buscarCosto(pProvincia)

End Function

Private Function buscarCosto(pProvincia As String) As Decimal

Dim answer As Decimal = 0

Select Case pProvincia

Case "CABA"

answer = 100

Case "Buenos Aires"

answer = 150

Case Else

answer = 250

End Select

Return answer

End Function

End Class

# XML

|  |  |  |
| --- | --- | --- |
|  | Que utiliza del modelo de objetos? | Quien lo consume? |
| Carrito de User.xml | Utiliza el objeto XmlTextWriter del modelo DOM para generar un documento XML con el pedido efectuado por un cliente. | 3.9 - Carrito.aspx |

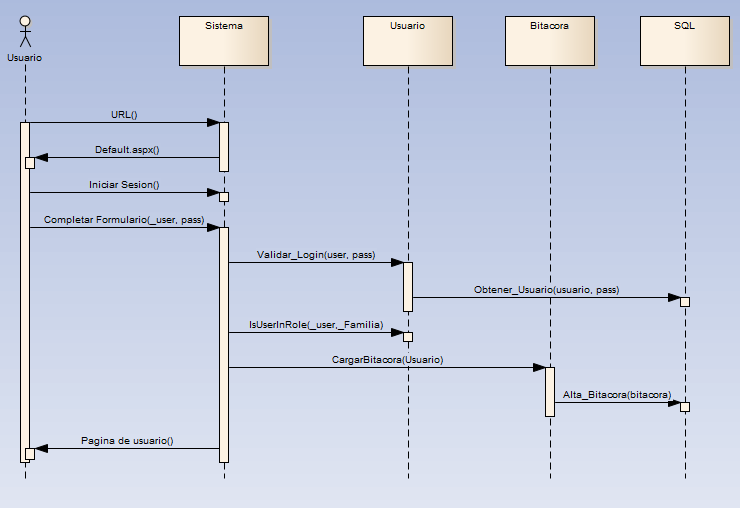
# ANEXOS

# Caso de Uso Log in

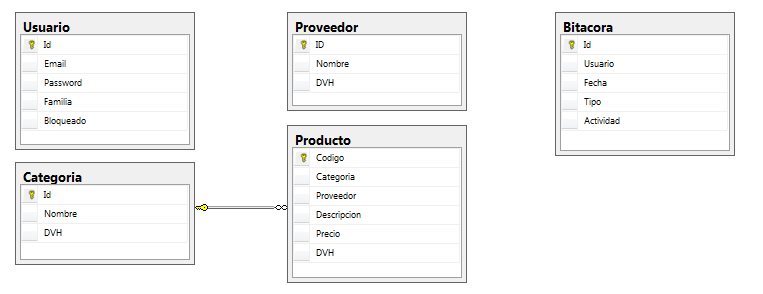
|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del CU: | Login en la pagina | | |
| Estado: | Approved | Versión: 1.0 | Fase: 1.0 |
| Creado el: | 05/06/2017 | Modificado el: |  |
| Notas: | Permite el acceso a las funcionalidades del sitio web asociadas al usuario a través del ingreso de un nombre de usuario y contraseña. | | |
| **Escenarios:** | | | |
| |  |  | | --- | --- | | 1.- | El Usuario accede a la URL de la empresa. | | 2.- | El sistema accede a la página web principal (Default.aspx) | | 3.- | El Usuario hace click sobre el botón Iniciar Sesión. | | 4.- | El sistema habilita un formulario que posee:   * Nombre de Usuario * Contraseña. Se enmascara con \* * Botones de acción. Ingresar para confirmar el login, cancelar para salir del formulario y retornar a la página de inicio. * Opcionalmente se puede tildar mantener sesión iniciada para que se recuerde el usuario. | | 5.- | El Usuario completa el formulario y selecciona Ingresar. | | 6.- | El sistema valida:   * Usuario existente * Usuario no bloqueado * Contraseña ingresada se corresponde con la almacenada por la aplicación. | | 7.- | El sistema obtiene los permisos del usuario que ingresó. | | 8.- | El sistema registra el ingreso del usuario en la bitácora. | | 9.- | El sistema presenta al usuario la página correspondiente a su rol. | | | | |
|

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Flujo alternativo | | Usuario Invalido |  | | 1.- | El sistema validó que el usuario no se encuentra registrado en la aplicación. | | | | | 2.- | El sistema muestra el mensaje de error “Usuario inválido” | | | | | 3.- | Vuelve al punto 4. | | | | | Flujo alternativo | | Usuario bloqueado |  | | 1.- | El sistema valida que el usuario ingresado supera la cantidad de intentos de acceso a la aplicación, la cual está definida en 3 intentos. | | | | | 2.- | El sistema muestra el mensaje de error “Usuario bloqueado” | | | | | 3.- | Vuelve al punto 4. | | | | | Flujo alternativo | | Contraseña errónea |  | | 1.- | El sistema valida que la contraseña ingresada no se corresponde con la registrada en la aplicación. | | | | | 2.- | El sistema acumula un intento más de ingreso al sistema erróneo. | | | | | 3.- | El sistema verifica si el usuario se bloquea con el intento erróneo.  Si con el intento erróneo el usuario se bloquea, se visualiza el mensaje “Usuario bloqueado”.  Si con el intento erróneo el usuario no se bloquea, se visualiza el mensaje “Contraseña inválida”. | | | | | 4.- | Vuelve al punto 4. | | | | | Flujo alternativo | | Usuario sin rol asociado |  | | 1.- | El sistema valida que el usuario no posee permisos asociados en la aplicación. | | | | | 2.- | El sistema muestra el mensaje de error “Usuario sin permisos asociados”. | | | | | 3.- | Vuelve al punto 2. | | | | |
|
| **Restricciones:** |
| Precondición: Usuarios con perfiles registrados en la base de datos de la aplicación.  Postcondición: Acceso a las funcionalidades de la aplicación habilitadas para el usuario ingresado. |
|  |

# Diagrama de secuencia Login



# DER



# Mapa de Navegación

3.1 Default.aspx

3.3 LogIn.aspx

3.4 Registrarse.aspx

3.12 Restore.aspx

3.5. MiCuenta.aspx

3.13 About.aspx

3.10 Contact.aspx

Cliente

3.8 Productos.aspx

3.9 Carrito.aspx

WebMaster

3.7 Bitacora.aspx

3.11 Backup.aspx

# Relación BD/Tabla/Pagina

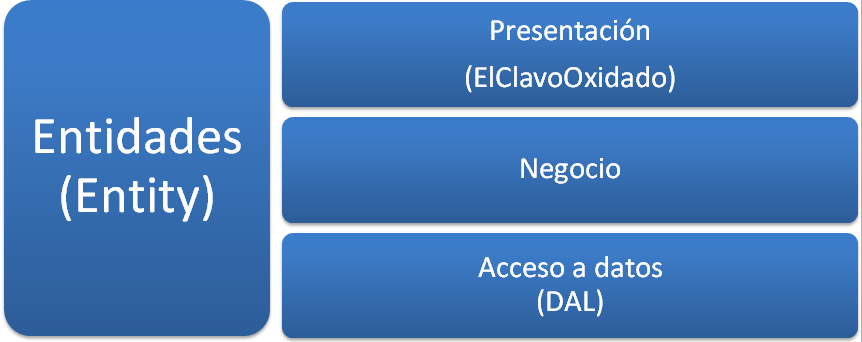
La siguiente tabla presenta las referencias a la relación Base de Datos / Tabla / Pagina según la vinculación que posee en las páginas de la aplicación

|  |  |  |
| --- | --- | --- |
| Base de datos | Tabla | Pagina |
| ElClavoOxidado | Usuario | 3.3 Login.aspx |
| ElClavoOxidado | Usuario | 3.5 MiCuenta.aspx |
| ElClavoOxidado | Usuario | 3.4 Registrarse.aspx |
| ElClavoOxidado | Bitacora | 3.7 Bitacora.aspx |
| ElClavoOxidado | Usuario | 3.11 UserManager.aspx |

# Estrategias de programación

* **¿Programa en Capas? ¿Cuántas?**

Programamos en 4 capas. Las mismas son:



En la capa de presentación “El Clavo Oxidado”, se encuentra el front-end de la aplicación con sus formularios y funciones correspondientes.

En la capa “Entity”, están las clases solo con sus propiedades.

La capa “Negocio”, implementa los métodos asociados a las entidades.

La capa “DAL”, implementa los métodos que permiten el acceso a la base de datos SQL para recuperar los datos utilizados por el sitio.

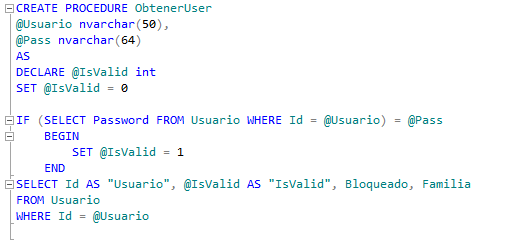
* **¿Usa Store Procedure?**

El sistema utiliza store procedure para efectuar las consultas y transacciones sobre la base de datos.

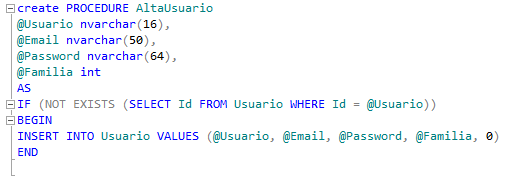
A continuación, se detallan los store procedures utilizados por el sistema:

ObtenerUser

Se encarga de validar que el usuario ingresado se encuentre registrado y que la contraseña ingresada sea la correcta. La contraseña ingresada se compara directamente con la almacenada en la base de datos, con lo cual se prevé que debe ingresar el hash de la contraseña como parámetro del store.

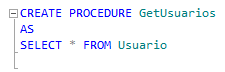


AltaUsuario

Se encarga de registrar un nuevo usuario en el sistema en caso de que el mismo no exista.

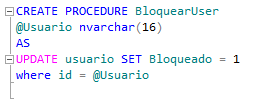
GetUsuarios

Se encarga de obtener todos los usuarios registrados en el sistema.



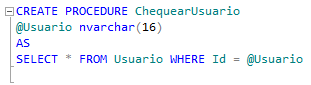
BloquearUser

Se encarga de bloquear un usuario ante reiterados intentos erróneos de ingresar al sistema.



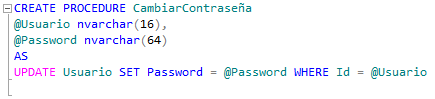
ChequearUsuario

Se encarga de obtener los datos del usuario recibido por parámetros.



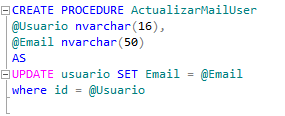
CambiarContraseña

Se encarga de actualizar la contraseña del usuario ingresado. La misma debe estar encriptada.



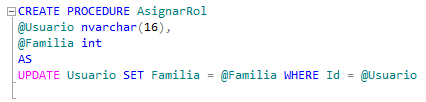
ActualizarMailUser

Se encarga de actualizar el mail del usuario ingresado. Se asume responsabilidad de la capa de negocio proporcionar un usuario registrado en el sistema.



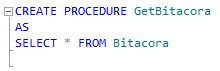
AsignarRol

Se encarga de asociarle un rol a un usuario.



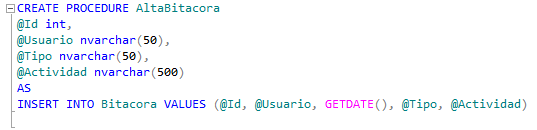
GetBitacora

Se encarga de proporcionar los datos de la bitácora del sistema.

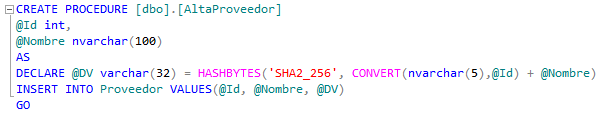


AltaBitacora

Se encarga de registrar en la bitácora un evento en particular. Se registra la fecha y hora en la cual se produce junto con los datos recibidos por parámetro (actividad a registrar, tipo y usuario que la generó).

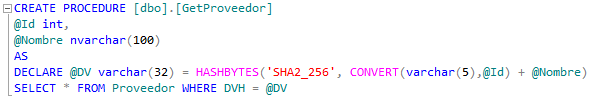


AltaProveedor

Se encarga de registrar un nuevo proveedor en el sistema. 

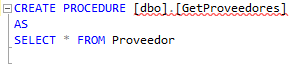
GetProveedor

Permite obtener todos los proveedores registrados en el sistema.



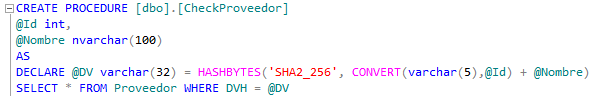
GetProveedores

Permite obtener todos los proveedores registrados en el sistema.



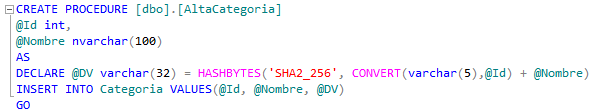
Check Proveedor

Permite verificar que un proveedor no haya sido alterado. Para ello se realiza el cálculo de la función hash SHA2-256 sobre los campos del registro utilizando la funcionalidad disponible del motor de base de datos.



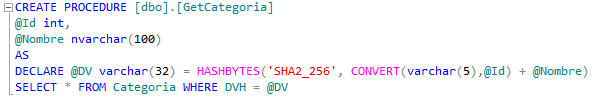
AltaCategoria

Se encarga de registrar una nueva categoría que se permitirá a asociar a un producto en el sistema.



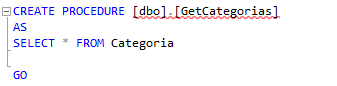
Get Categoria

Permite obtener las categorías dadas de alta para los productos registrados.



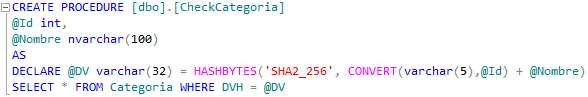
Get Categorias

Permite obtener todas las categorías dadas de alta.



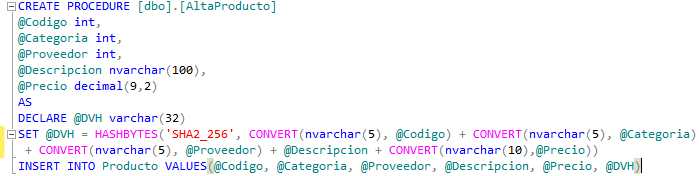
Check Categoria

Permite verificar que una categoría no haya sido alterada. Para ello se realiza el cálculo de la función hash SHA2-256 sobre los campos del registro utilizando la funcionalidad disponible del motor de base de datos.



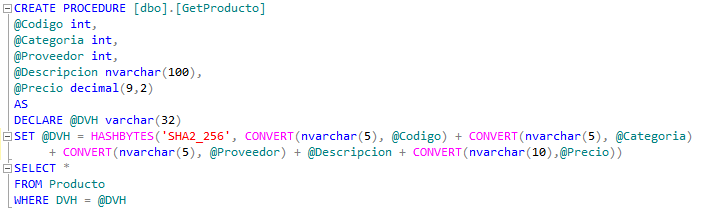
AltaProducto

Se encarga de registrar un nuevo producto en el sistema. Se han de enviar todos los datos asociados al producto a dar de alta.



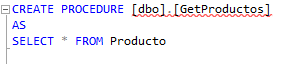
Get Producto

Permite obtener todos los productos dados de alta en el sistema.



Get Producto

Permite obtener todos los productos dados de alta en el sistema.



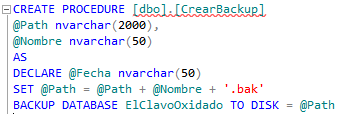
Check Producto

Permite verificar que un producto no haya sido alterado. Para ello se realiza el cálculo de la función hash SHA2-256 sobre los campos del registro utilizando la funcionalidad disponible del motor de base de datos.



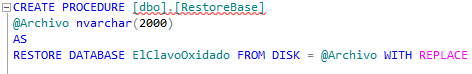
Crear Backup

Permite generar un backup de la base de datos dejando la copia en el path especificado.



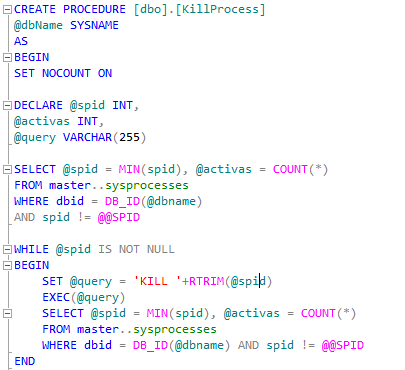
RestoreBase

Permite restaurar la base de datos a partir del backup especificado por parámetro.



KillProcess

Para poder efectuar el restore de la base, este store procedure se encarga de cerrar todas las conexiones existentes en la base de datos.



* **El GLOBAL.ASAX ha sido modificado contemplando la conexión a la base de datos. Si hay modificación indicar cuál.**

Fue necesario modificar la GLOBAL.ASAX para al iniciar una sesión generar un flag de check error.

Public Class Global\_asax

Inherits HttpApplication

Sub Application\_Start(sender As Object, e As EventArgs)

' Se desencadena al iniciar la aplicación

RouteConfig.RegisterRoutes(RouteTable.Routes)

BundleConfig.RegisterBundles(BundleTable.Bundles)

Try

Roles.CreateRole("1")

Roles.CreateRole("2")

Roles.CreateRole("3")

Catch

End Try

End Sub

Sub Session\_Start(sender As Object, e As EventArgs)

Session("CheckError") = 0

Session("CheckErrorValidate") = 0 ''Use this to optimize default.aspx'

Session("Carrito") = New Entity.Carrito

End Sub

End Class

* **Usamos el modo conectado en ADO.NET en estas circunstancias….**

**Usamos el modo desconectado en ADO.NET en estas otras circunstancias….**

Para los accesos a la base de datos siempre utilizamos el modo desconectado de ADO NET. No tenemos ningún acceso realizado a través del modo conectado.

La manera en la cual se efectúan los accesos a la base de datos es desde la capa DAL. La clase correspondiente se encarga de especificar el nombre del store procedure a ejecutar, e invoca la clase SQL\_Desconectado. Esta última clase es quien a través de la función Obtener\_DatasetStoreProcedure recupera un DATASET con los datos requeridos.

Finalmente la DAL que invoco el acceso a los datos, es la encargada de trasnsformar los registros recuperados en los objetos de negocio correspondiente.

* **La encriptación la hacemos mediante….**

La función definida en el archivo “sha.js”.

Se decide utilizar una función hash para almacenar la contraseña de los usuarios registrados en el sistema. En una relación de compromiso-beneficios se decide utilizar SHA-256, dado que es más robusta que sus antecesoras MD5 y SHA1, mientras que posee menos procesamiento que otras funciones de la misma familia SHA2.

Cálculo:

El modelo de los algoritmos SHA-2 sigue la misma estructura adoptada por otros algoritmos como el MD5 y SHA-1 implementando funciones de resumen iteradas. El SHA-256 trabaja con palabras de 32 bits.

Los pasos se pueden enumerar de la siguiente manera:

a) Un mensaje M se convierte en un bloque múltiplo de 512 bits, añadiendo bits si es necesario al final del mismo. El relleno del mensaje en el algoritmo SHA-2 añade un bit 1 al final del mensaje. A continuación, se añade 0 hasta que la palabra tenga el tamaño proporcional a la cantidad de bloques, y por fin concatenarse con bits para señalar el tamaño del mensaje al último Bloque.

b) Se inicializan 8 vectores de 32 bits cada uno dando un total de 256 bits.

c) Se inicializan un array con 64 constantes a utilizar en las rondas

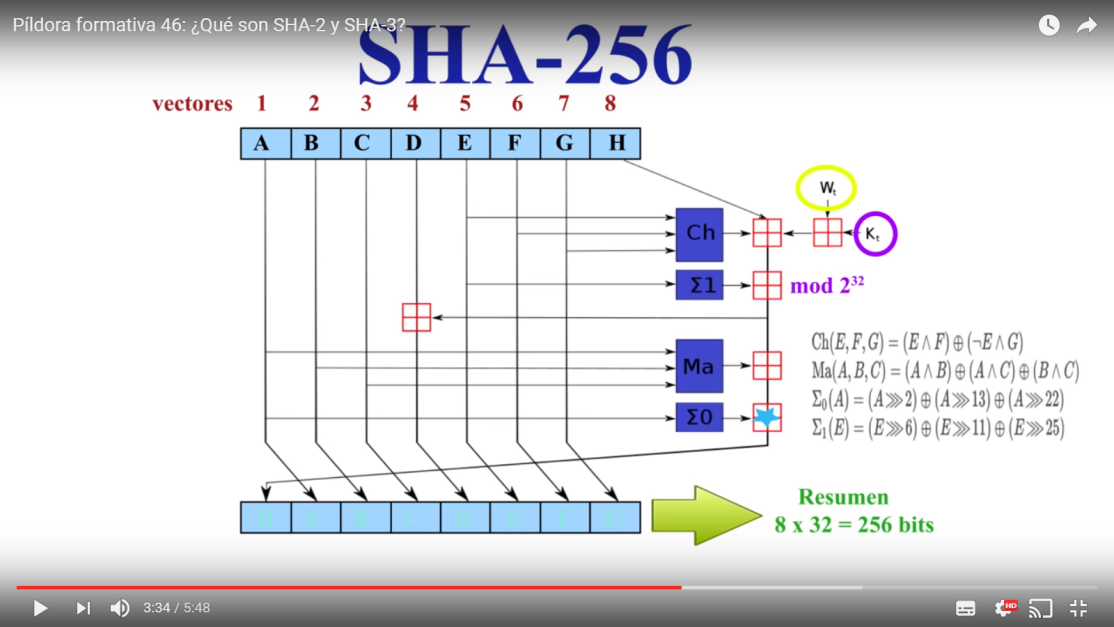
d) El mensaje de 512 bits se divide en 16 palabras, cada una con 32 bits.

e) Las palabras se extienden a 64 palabras de 32 bits, para ello se realiza una expansión de las primeras 16 palabras para generar otras 48 palabras a través de operaciones lógicas descritas por funciones para obtener el valor hash del mensaje. Estas operaciones son desplazamientos, XOR (o exclusivo), AND y NOT.

f) Posteriormente, para cada palabra se realiza una ronda del algoritmo, pero una particularidad del SHA-2 es que cada ronda está compuesta por una constante diferente, la cual se obtuvo en el punto b.

g) Por último, las variables de estados utilizados inicialmente se suman al módulo 2 (XOR) con los valores resultantes de las rondas.

h) La salida de esta operación (256 bits) se convierte en el nuevo conjunto de 8 vectores y se realiza la misma función con el segundo bloque de 512 bits del mensaje, y así hasta el último bloque del mensaje. Al terminar, el algoritmo entrega un resumen que corresponde a los últimos 256 bits de estas operaciones.



En todos los algoritmos de la familia SHA-2, estos procedimientos son los mismos, varían tan sólo los tamaños de 32 bits para 64 bits para los bloques de mensajes.

* **De que manera evitan que un intruso acceda a la bitácora a través de la URL.**

Cada formulario de la capa de presentación cuenta con un método asociado al evento PreLoad donde verificamos los permisos que tiene el usuario asignado. Si tiene permisos para ver el formulario se permite el acceso. En caso contrario, se visualiza el mensaje de error correspondiente.

# Políticas de programación

* **¿Con qué frecuencia hace un backup?**

No es política del sistema realizar backup de manera automática. Sin embargo, el sistema dispone de las herramientas necesarias para que el web master del sistema pueda efectuarlo cuando desee.

Por lo tanto, la frecuencia del backup de la base de datos depende de la decisión del web master. Por procedimiento, se establece que el web master debe efectuar como mínimo un backup semanal.

* **¿Es Total o Incremental?**

La herramienta puesta a disposición del web master únicamente permite efectuar backup de la base de datos de manera total.

* **Si el usuario olvida la contraseña. ¿Qué hace?**

Ante el olvida de la contraseña por parte del usuario, el mismo al ingresar en la opción de Inicio de Sesión, debe seleccionar Olvido de Contraseña. El sistema solicitará el nombre del usuario a resetear y la clave. Al aceptar automáticamente se enviará un mail al correo electrónico registrado para el usuario. El correo contendrá un código de seguridad que se solicitará ingresar en la página de recuperación de contraseña. El código enviado por el mail tendrá validez por 5 minutos, pasado dicho tiempo será necesario volver a generar el procedimiento.

Tras ingresar el código y ser validado por el sistema, se habilitará al usuario a indicar la nueva contraseña que desea tener.

* **Si hay 3 intentos por parte de un intruso ¿qué hace?**

Ante el intento reiterado de un usuario de ingresar al sistema con una contraseña inválida, se procede automáticamente a bloquear dicho usuario. El bloqueo sucederá al tercer intento erróneo y mantendrá dicha situación de bloqueo hasta que el usuario proceda al reseteo de la clave con la funcionalidad olvido de contraseña.