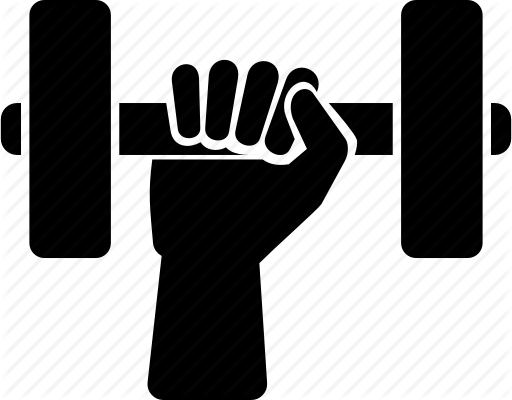
**TRABAJO PRÁCTICO**

**LPPA**

**4 AÑO**

**Megatlin S.A.**



**CASTELLI JUAN, MARRACO AGUSTIN**

[Quienes Somos 3](#_Toc10506815)

[Diagrama de secuencia con clases 4](#_Toc10506816)

[Log-in 5](#_Toc10506817)

[Login.aspx 5](#_Toc10506818)

[Login.aspx.cs 6](#_Toc10506819)

[LoginService.cs 7](#_Toc10506820)

[LoginDAO.cs 7](#_Toc10506821)

[SeguridadUtilities.cs 8](#_Toc10506822)

# Quienes Somos

La empresa “Megatlin.sa” es un gimnasio pequeño, de carácter familiar, el cual se encuentra ubicado en el partido de Lomas de Zamora, y recientemente a tenido un crecimiento en la clientela el cual puede producir una expansión, el modelo de negocio que maneja esta empresa es mediante la venta de servicios los cuales son:

· Las membresías para en ingreso al Gimnasio

· La venta de las rutinas a realizar (las cuales son supervisadas y autorizadas por entrenadores especializados)

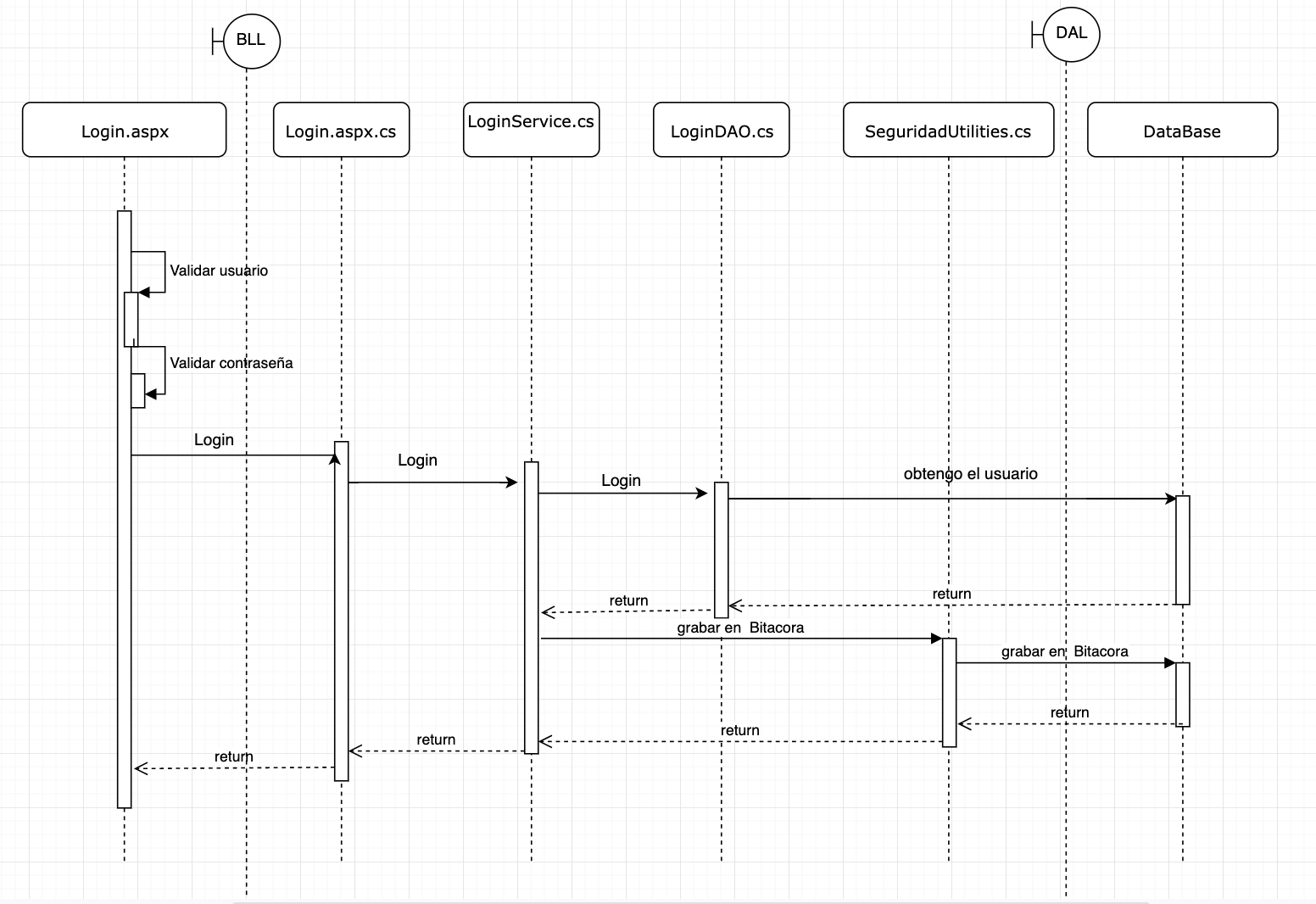
El software a realizar tiene 3 pilares donde aporta valor:

* Para la empresa:
  + Permitirá automatizar el registro y seguimiento de los usuarios de usuarios.
  + Poseerá un carrito de compras donde los usuarios podrán acumular la membresía y las rutinas que deseen comprar, lo cual facilitara la venta y el seguimiento de dicha información.
  + Permitirá mantener el histórico de entrada y salida de los usuarios

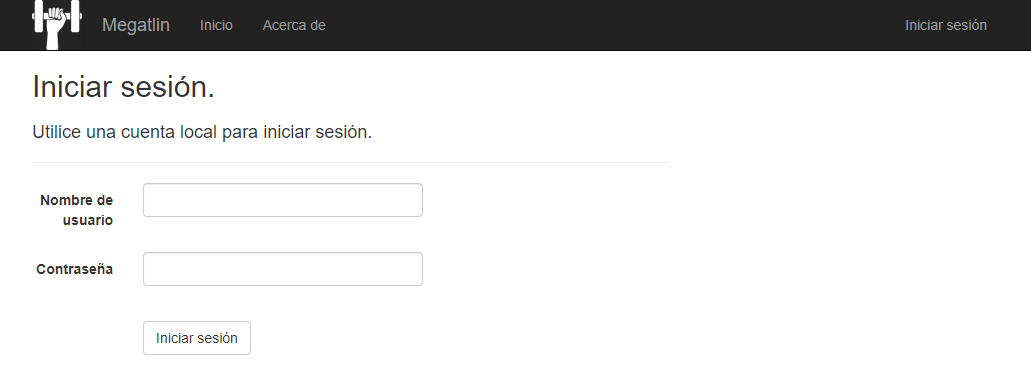
* Para el Usuario:
  + Permitirá al usuario registrar que días asistirá y revisar la rutina que le corresponderá para dicho día.

* Para los Entrenadores:
  + Permitirá a los entrenadores tanto ver quienes asistirán cada día como visar las rutinas y las máquinas que se deberán utilizar.

# Diagrama de secuencia con clases



## Log-in



## Login.aspx (1)

1. Se verifica que los campos no estén vacíos desde frontend

<div class="row">

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

<section id="loginForm">

<div class="form-horizontal">

<h4>Utilice una cuenta local para iniciar sesión.</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="El campo de nombre de usuario es obligatorio." />

</div>

A

</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="El campo de contraseña es obligatorio." />

</div>

</div>

<div class="form-group">

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

<asp:Button runat="server" OnClick="LogIn" Text="Iniciar sesión" CssClass="btn btn-default" />

</div>

</div>

</div>

</section>

</div>

## Login.aspx.cs (1)

1. Se verifica que los validadores de frontend estén correctos.
2. Se loguea al usuario con los datos provistos en los textBox.
3. Se verifica que estén cargadas las cookies
   1. de ser correcto se redirige a la url correspondiente
   2. de ser erróneo se le informa al usuario

using Microsoft.AspNet.Identity;

using Microsoft.Owin.Security;

using System;

using System.Web;

using System.Web.UI;

using ProyectoLPPA;

using System.Data.SqlClient;

using System.Configuration;

public partial class Account\_Login : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

var returnUrl = HttpUtility.UrlEncode(Request.QueryString["ReturnUrl"]);

}

protected void LogIn(object sender, EventArgs e)

{

A

if (IsValid)

{

B

Response.Cookies.Clear();

LoginService.login(UserName.Text, Password.Text,Response);

C.a

if (Response.Cookies.Count > 0)

{

IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);

}

else

{

FailureText.Text = "Datos invalidos.";

C.b

ErrorMessage.Visible = true;

}

}

}

}

## LoginService.cs

1. Se obtiene el usuario desde la base de datos
2. Se registra en la Bitácora el logueo del usuario con el perfil que este posee
3. Se agregan las Cookies que validan el correcto login del usuario

using ProyectoLPPA;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

public class LoginService

{

public LoginService()

{ }

public static void login(String user, String password,HttpResponse response)

{

A

User usuario = LoginDAO.login(user, password);

if (usuario != null)

{

SeguridadUtiles.grabarBitacora(usuario.id, "Se logueo " + usuario.username + " que tiene el tipo " + usuario.tipo);

B

response.Cookies.Add(new HttpCookie("user", usuario.username));

response.Cookies.Add(new HttpCookie("tipo", usuario.tipo));

response.Cookies.Add(new HttpCookie("userId", usuario.id.ToString()));

C

}

}

}

## LoginDAO.cs

1. Se obtiene la conexión con la base de datos
2. Se solicita el registro correspondiente
3. se crea un nuevo objeto de la clase User con los datos obtenidos
4. se cierra la conexión a base de datos se devuelve el objeto user

using ProyectoLPPA;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

public class LoginDAO

{

public LoginDAO()

{}

public static User login(String username, String password)

{

SqlConnection cn = ConexionSingleton.obtenerConexion();

A

if (cn.State == System.Data.ConnectionState.Open)

cn.Close();

cn.Open();

SqlTransaction tx = cn.BeginTransaction();

SqlCommand cmd = new SqlCommand("SELECT \* from Usuario where Alias = @id and Password = @password");

B

cmd.Parameters.Add(new SqlParameter("id", username));

cmd.Parameters.Add(new SqlParameter("password", password));

cmd.Connection = cn;

cmd.Transaction = tx;

SqlDataReader reader = cmd.ExecuteReader();

C

User user = null;

if (reader.Read())

{

user = new User(reader.GetString(5),reader.GetString(4).Trim(), reader.GetInt32(0));

}

reader.Close();

D

tx.Commit();

cn.Close();

return user;

}

}

## SeguridadUtilities.cs

1. Se conecta con la base de datos
2. Se crea el insert para la Bitácora
3. Se ejecuta el insert en base de datos
   1. Se cierra conexión a base de datos
   2. Se recalcula el dvv
   3. Si falla se realiza un rollback
   4. Se cierra la conexión a base de datos

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Security.Cryptography;

using System.IO;

using System.Data.SqlClient;

using System.Web.Script.Serialization;

namespace ProyectoLPPA

{

public static class SeguridadUtiles

{

private static String dbName = "[" + AppDomain.CurrentDomain.BaseDirectory + "APP\_DATA\\BASEDEDATOS.MDF]";

public static string recalcularDigitoHorizontal(String[] campos)

{

StringBuilder builder = new StringBuilder();

for (int x = 0; x < campos.Length; x++)

{

builder.Append(campos[x]);

}

return SeguridadUtiles.encriptarMD5(builder.ToString());

}

public static void recalcularDigitoVertical(String tabla)

{

SqlConnection connection = ConexionSingleton.obtenerConexion();

connection.Open();

SqlTransaction tx = connection.BeginTransaction();

String updateCommand = " UPDATE DIGITO\_VERTICAL SET DV\_DIGITO\_CALCULADO = @HASH WHERE DV\_NOMBRE\_TABLA = @NOMBRE\_TABLA ";

SqlCommand query = new SqlCommand("", connection, tx);

StringBuilder sb = new StringBuilder();

sb.Append(" SELECT dvh FROM " + tabla);

query.CommandText = sb.ToString();

SqlDataReader reader;

String sumaDeDVH = "";

try

{

reader = query.ExecuteReader();

while (reader.Read())

{

if (!reader.IsDBNull(0))

{

sumaDeDVH += reader.GetString(0);

}

}

reader.Close();

query.Parameters.Clear();

query.CommandText = updateCommand;

query.Parameters.Add(new SqlParameter("@HASH", System.Data.SqlDbType.VarChar)).Value = SeguridadUtiles.encriptarMD5(sumaDeDVH);

query.Parameters.Add(new SqlParameter("@NOMBRE\_TABLA", System.Data.SqlDbType.VarChar)).Value = tabla;

query.ExecuteNonQuery();

tx.Commit();

connection.Close();

}

catch (Exception ex)

{

try

{

tx.Rollback();

}

catch (Exception ex2)

{

connection.Close();

throw ex2;

}

connection.Close();

throw ex;

}

finally

{

connection.Close();

}

}

public static String encriptarMD5(String stringAEncriptar)

{

String hash = "";

if (!String.IsNullOrEmpty(stringAEncriptar))

{

MD5 md5 = MD5.Create();

byte[] bytes = Encoding.UTF8.GetBytes(stringAEncriptar);

byte[] datoEncodeado = md5.ComputeHash(bytes);

hash = BitConverter.ToString(datoEncodeado).Replace("-", "");

}

return hash;

}

public static void grabarBitacora(int usuId, String mensaje)

{

A

SqlConnection connection = ConexionSingleton.obtenerConexion();

connection.Open();

SqlTransaction tx = connection.BeginTransaction();

B

StringBuilder builder = new StringBuilder(" INSERT INTO BITACORA (");

builder.Append("idUsuario,");

builder.Append("mensaje,");

builder.Append("dvh )");

builder.Append(" VALUES (");

builder.Append(" @USUARIO,");

builder.Append(" @MENSAJE,");

builder.Append(" @DVH");

builder.Append(" ) ");

SqlCommand cmd = new SqlCommand(builder.ToString(), connection, tx);

DateTime fecha = DateTime.Now;

cmd.Parameters.Add(new SqlParameter("@MENSAJE", System.Data.SqlDbType.Text)).Value = mensaje;

cmd.Parameters.Add(new SqlParameter("@DVH", System.Data.SqlDbType.VarChar)).Value = recalcularDigitoHorizontal(new string[] { fecha.ToString(), mensaje });

cmd.Parameters.Add(new SqlParameter("@USUARIO", System.Data.SqlDbType.BigInt)).Value = usuId;

try

{

cmd.ExecuteNonQuery();

C

tx.Commit();

connection.Close();

C.a

recalcularDigitoVertical("BITACORA");

}

catch (Exception ex)

C.b

{

try

{

tx.Rollback();

}

catch (Exception)

C.c

{ }

connection.Close();

throw ex;

}

C.d

}

}

# User.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

public class User

{

private string \_username;

public string username

{

get { return \_username; }

set { \_username = value; }

}

private string \_tipo;

public string tipo

{

get { return \_tipo; }

set { \_tipo = value; }

}

private int \_id;

public int id

{

get { return \_id; }

set { \_id = value; }

}

public User(String username,String tipo,int id)

{

this.username = username;

this.tipo = tipo;

this.id = id;

}

}

# Bitacora

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

public class Bitacora

{

private DateTime fecha;

private String mensaje;

private String nombre;

public Bitacora(DateTime fecha, String mensaje,String nombre)

{

this.fecha = fecha;

this.mensaje = mensaje;

this.nombre = nombre;

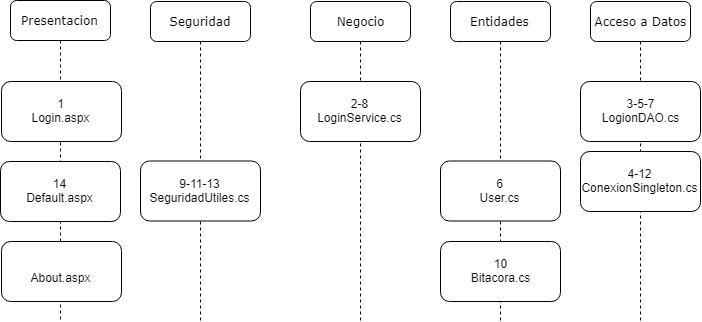
}

}

# Anexo

## Mapas de Navegación

# Diagrama General



|  |  |  |
| --- | --- | --- |
| Número | Página | Tablas Involucradas |
|  |  |  |