

Septiembre de 2024

Prácticas Profesionales

Inmobiliaria Herdim

Instituto Superior de Formación Docente y Técnica N° 46

Grupo II:

Aiello Hernán, Astargo Diego, Duchi Mara y Lacolla Micaela.

Índice	2
Manual de uso	3
Manual de sistema	6

Manual de uso:

Caso de uso 1: "Iniciar sesión".

Breve descripción: En este caso de uso se ingresa al sistema con un usuario y contraseña autorizado.

Descripción paso a paso: Completar los campos de usuario y contraseña con los datos previamente autorizados. Al hacer clic en el botón ingresar si el usuario está autorizado se loguea en el sistema de lo contrario valida que los datos no están en la base de datos y/o que el campo está vacío o incompleto.

Además, estando en los formularios de Cliente y Alquiler-Ventas pasan más de 2 segundos se desloguea y te redirige al formulario de inicio de sesión.

Caso de uso 2: "Ingresar cliente".

Breve descripción: En este caso de uso mediante un formulario se ingresan los datos del cliente y se guardan en la base de datos.

Descripción paso a paso:

Completar los campos del formulario del cliente: Nombre, Apellido, edad y contraseña.

Hacer clic en el botón guardar.

Validación: De no tener los campos completos se visualizará una leyenda en rojo indicando que el campo es obligatorio.

Caso de uso 3: "Realizar una compra o alquiler de un inmueble".

Breve descripción: En este caso de uso se podrá realizar una compra o alquiler de un inmueble.

Descripción paso a paso: Mediante un formulario de compras se podrá realizar una compra o alquiler de un inmueble. Además, la solapa permite realizar un abm, agregar, borrar, modificar y consultar las transacciones realizadas, discriminando por tipo de inmueble (Casa/Departamento/ Lote) y tipo de transacción (Alquiler/Venta).

Manual de sistemas:

Formularios principal:

Formulario Web.Config:

- Es el formulario principal donde se sacan los estilos y se habilitan los formularios dependientes.
- Este archivo contiene, en lenguaje xml, todos los parámetros de configuración necesarios para el funcionamiento de nuestra aplicación.
- Conexión con la base de datos de los formularios de login y cliente:

```
<connectionStrings>
```

```
<add name="Cadenacontacto" connectionString = "Data Source = EMINENCIA-EN-PR\SQLLE; Integrated Security = true; Initial Catalog = Gonzalo; " providerName="System.Data.SqlClient" />
```

```
<add name="CadenaT_Login" connectionString = "Data Source = EMINENCIA-EN-PR\SQLLE; Integrated Security = true; Initial Catalog = Gonzalo; " providerName="System.Data.SqlClient" />
```

```
</connectionStrings>
```

- Se determina el tiempo, configurado en 2 segundos para considerar que en la página no se realizaron movimientos por ende se debe redirigir al inicio de sesión y loguearse nuevamente:

```
<sessionState mode="InProc" cookieless="false" timeout="2"></sessionState>
```

paginamaestra.Master:

Muestra el nav, enlaces de navegación interna de:

Clientes> Formulario cliente.

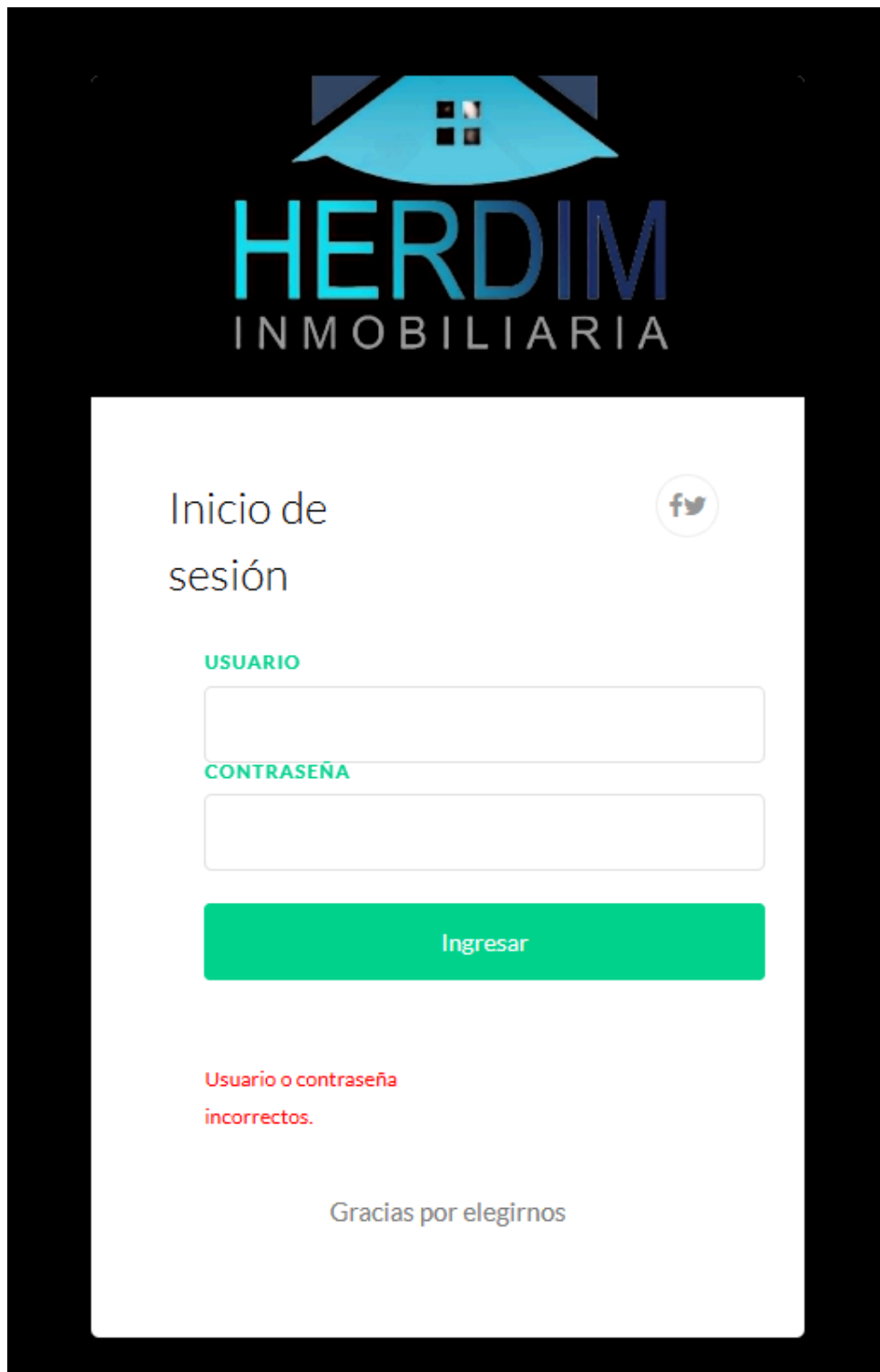
Inmuebles> Casas, Departamentos, Lotes.

VentasAlquiler.

Nosotros.

Formularios dependientes con sus Content Place Holder:

Formulario Login:



HERDIM
INMOBILIARIA

Inicio de sesión

USUARIO

CONTRASEÑA

Ingresar

Usuario o contraseña incorrectos.

Gracias por elegirnos

- Se establece la conexión con la tabla en la base de datos:

```
private static string Cadena = ConfigurationManager.ConnectionStrings["CadenaT_Login"].ConnectionString;
```

- Al hacer clic en el botón iniciar se verifica si las variables de usuario y contraseña están vacías o son nulas y si no coinciden con la información de la base de datos válida en un label "Usuario o contraseña incorrectos.", de lo contrario se loguea en nuestro sistema y te redirige al formulario de "Nosotros". Utiliza `Response.Redirect` para redirigir al formulario "Nosotros".

```
using (SqlConnection connection = new SqlConnection(Cadena))
{
    try
    {

        string script = "SELECT NOMBRE FROM T_LOGIN WHERE NOMBRE = '" + TextUsuario.Text + "'
AND" + " PASS = '" + TxtPassword.Text + "'";

        connection.Open();
        SqlCommand command = new SqlCommand(script, connection);
        SqlDataReader reader = command.ExecuteReader();
        string user = String.Empty;
        while (reader.Read())
        {
            user = reader.GetString(0);
        }
        reader.Close();

        if (user != String.Empty)
        {
            Session["Nombre"] = TextUsuario.Text;
            //Agregar "false" en el redirect. Le indicamos a la aplicacion
            //que debe terminar el codigo en esta pagina y redigir a la siguiente.
            Response.Redirect("Nosotros.aspx", false);
        }
        else
        lblError.Text = "Usuario o contraseña incorrectos.";
    }
    catch (Exception exception)
    {
        Console.WriteLine(exception.Message);
    }
}
```

Formulario Nosotros:

Formulario de visualización de información.

Se observa información de la empresa inmobiliaria Hermin. Su locación principal, principios organizacionales y servicio a brindar.

Formulario paginados:

- Formulario de ingreso de información del cliente. Los datos se pueden consultar y modificar.
- Se genera un POST desde el cliente hacia el servidor y una respuesta. Este ida y vuelta dentro de un mismo formulario web se llama Postback. (La propiedad IsPostBack indica si la página se está mostrando por primera vez o se está cargando como respuesta a un Postback).
- Se utiliza IsPostBack para determinar si la variable Nombre está vacía o es nula durante el tiempo estimado en el formulario de web.config como timeout="2" y se redirige al inicio de sesión para loguearse nuevamente y continuar trabajando en la página web.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        if (Session["Nombre"] == null)
        {
            Response.Redirect("Login.aspx");
        }
        else
        {
            string ses = Session["Nombre"].ToString();
            if (Session["Nombre"].ToString() == String.Empty)
                Response.Redirect("Login.aspx");
        }
    }
}
```

Clientes

Dejanos tu contacto:

Nombre *

Apellido *

Edad *

Contraseña *

Repita Contraseña *

- Complete nombre
- Complete apellido
- Ponga entre 18 y 100 años
- Las contraseñas deben ser iguales

- Dentro del formulario ASPX determinamos los textBox a completar>

Se determina el label y el textBox a utilizarse con el atributo **runat="server"** que especifica su uso en el servidor:

```
<asp:Label Text="Nombre" Id="lblNombre" runat="server" />  
<asp:TextBox ID="txtNombre" runat="server" />
```

- Validaciones utilizadas:

- ❖ **RequiredFieldValidator:**

Valida que se completen valores en un control (campo obligatorio).
Si todos los controles son válidos(tienen valores) la página es válida.

- ❖ **CompareValidator:**

- Compara los valores de dos controles
- Utiliza tres propiedades clave para realizar su validación
 - ☐ **ControlToValidate** (contiene el valor a comparar)
 - ☐ **ControlToCompare** (contiene el valor a comparar)

- ❖ **RangeValidator:**

Valida que una entrada se encuentre dentro de un determinado rango.
Puede ser numérica o por cantidad de caracteres.
Utiliza tres propiedades:

ControlToValidate (contiene el valor a comparar).

MinimumValue (define el valor mínimo a comparar).

MaximumValue (define el valor máximo a comparar).

Type (define el tipo de dato a comparar, String, Integer, Double, Date, Currency).

- ❖ **ValidationSummary:**

- Se muestra cuando la propiedad IsValid de la página está establecida a "false"
- Verifica cada control de validación de la página y agrega el texto de error (ErrorMessage) que cada uno muestra

- Dentro del control del servidor asp RequiredFieldValidator se coloca la validación de ese campo con ControlToValidate y el nombre del textBox. Además, se indica el texto que apareciera cuando el campo está vacío y al poner en text "*" Permite indicar el error debajo de los textbox.

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" ControlToValidate="txtNombre" runat="server" ErrorMessage="Complete nombre" Text="*"></asp:RequiredFieldValidator>
```

- Dentro del control del servidor asp:RangeValidator con el campo edad podemos validar que el rango numerico que se puede poner en ese campo.

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator3" ControlToValidate="txtEdad" runat="server" ErrorMessage="Complete edad" Text="*">
```

```
</asp:RequiredFieldValidator>
```

```
<asp:RangeValidator ID="RangeValidator1" ControlToValidate="txtEdad" MinimumValue="18" MaximumValue="100" Type="Integer" runat="server"
```

```
ErrorMessage="Ponga entre 18 y 100 años" Text="*"></asp:RangeValidator>
```

- Dentro de los labels de contraseña al indicar el TextMode="Password" nos da la visibilidad del campo como contraseña ofuscada pero pudiendo visualizarla al hacer clic en el icono del ojo.



```
<asp:Label ID="lbdContraseña" CssClass="form-control" runat="server" Text="Contraseña"></asp:Label>
```

```
<asp:Label ID="lbdContraseña2" runat="server" Text="Repita Contraseña"></asp:Label>
```

```
<asp:TextBox TextMode="Password" ID="txtPass" runat="server"></asp:TextBox>
```

```
<asp:TextBox TextMode="Password" ID="txtRepass" runat="server"></asp:TextBox>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ErrorMessage="" ControlToValidate="txtPass" Text="*" ForeColor="Red"></asp:RequiredFieldValidator>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server" ErrorMessage=" Las contraseñas deben ser iguales."
```

```
ControlToValidate="txtRepass" Text="*" ForeColor="Red" ></asp:RequiredFieldValidator>
```

```
<asp:CompareValidator ID="CompareValidator1" runat="server" ErrorMessage="Las contras deben ser iguales" ControlToValidate="txtPass" ControlToCompare="txtRepass" Text="*"></asp:CompareValidator>
```

Formulario Casas:

Formulario de visualización de información de inmuebles de tipo casa.

Formulario Anuncio:

Al hacer clic en ver propiedad se abre este formulario mostrando un detalle de la opción seleccionada.

Formulario Departamentos:

Formulario de visualización de información de inmuebles de tipo Departamentos.

Formulario VerDpt:

Al hacer clic en ver propiedad se abre este formulario mostrando un detalle de la opción seleccionada.

Formulario Lotes:

Formulario de visualización de información de inmuebles de tipo Lotes.

Formulario VerLote:

Al hacer clic en ver propiedad se abre este formulario mostrando un detalle de la opción seleccionada.

Formulario VentasOalquiler:

Formulario de transacciones de Ventas o alquileres de inmuebles.

Las transacciones se pueden consultar, modificar y eliminar.

Concateno nombre y apellido en el dropdownlist

Create View vw_Clientes As

Select id, RTRIM(apellido) + ' ' + RTRIM(nombre) As Contacto from dbo.contacto

Traigo los datos de clientes e inmuebles de la base de datos, no se selecciona manual.

Formularios con amb conectados a la base de datos:

Formulario ABM:

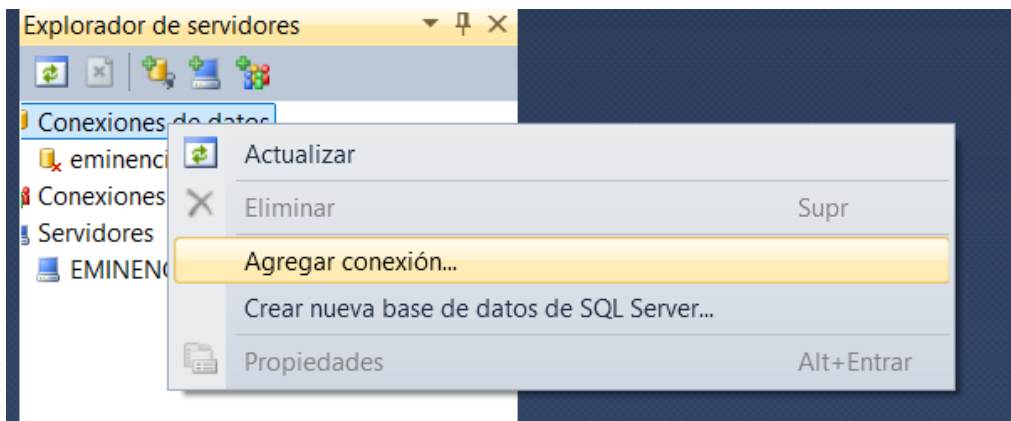
Formulario de para consultar , modificar y/o borrar datos de clientes.

Formulario ABM Ventas-Alquiler:

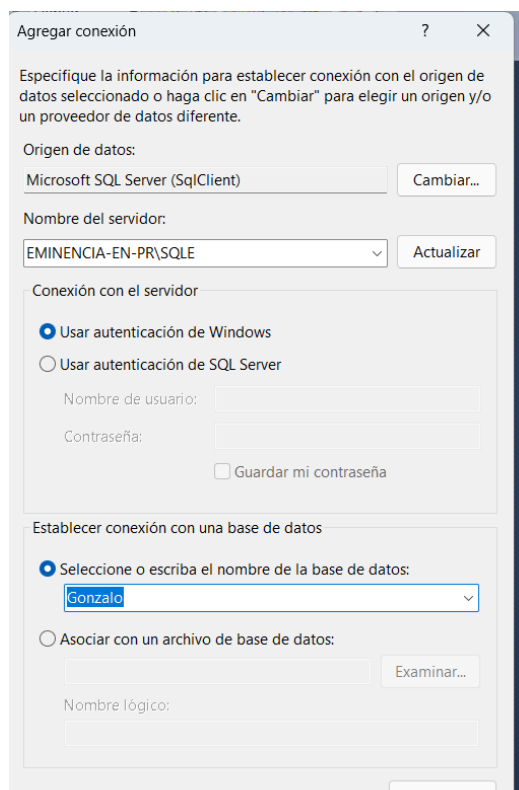
Formulario de para consultar , modificar y/o borrar datos de transacciones de inmuebles.

Base de datos dentro de visual studio:

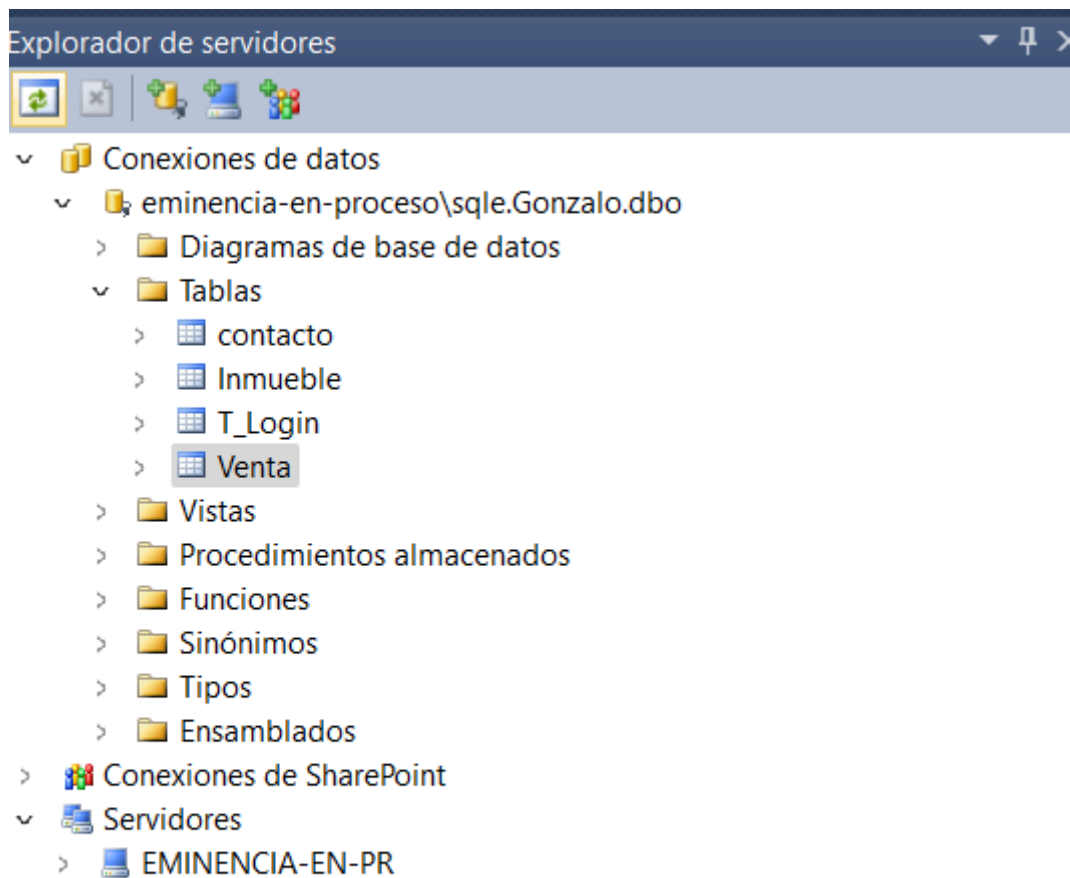
1)



2)



- 3) Una vez realiza la conexión a la base de datos, se arrastra la tabla para realizar el abm al Content Place Holder del formulario de abm:



4) Se habilita las funcionalidades realizar como editar y eliminar.

The screenshot shows a Visual Studio IDE with a file named 'abm.aspx' open. The main window displays a GridView control with the following columns: 'id', 'nombre', 'apellido0', 'edad', and 'contraseña'. Each row in the GridView has 'DataBound' text in the first five columns and a pair of 'Editar' and 'Eliminar' links in the sixth column. A context menu titled 'Tareas de GridView' is open over the GridView, showing various configuration options. The 'Habilitar edición' and 'Habilitar eliminación' checkboxes are checked, indicating that the edit and delete functionalities are enabled. The bottom status bar shows the current view is 'Diseño' (Design) and the selected element is '<asp:GridView#GridView1>'. The Explorer pane on the right shows the project structure, including 'pagmaestra.Master' and 'Gonzalo'.

	id	nombre	apellido0	edad	contraseña
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound
Editar Eliminar	DataBound	DataBound	DataBound	DataBound	DataBound

12

Diseño | Dividir | Código | <asp:Content#Content2> | <asp:GridView#GridView1>

Tareas de GridView

- Formato automático...
- Elegir origen de datos: SqlDataSource1
- Configurar origen de datos...
- Actualizar esquema
- Editar columnas...
- Agregar nueva columna...
- ☒ Habilitar paginación
- ☒ Habilitar ordenación
- ☒ Habilitar edición
- ☒ Habilitar eliminación
- ☐ Habilitar selección
- Editar plantillas

SQL server:

/***** Creación de base de datos *****/

```
USE [master]
GO
```

/***** Object: Database [Gonzalo] Script Date: 24/9/2024 19:13:02 *****/

```
CREATE DATABASE [Gonzalo]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'Gonzalo', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS03\MSSQL\DATA\Gonzalo.mdf' , SIZE = 8192KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'Gonzalo_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS03\MSSQL\DATA\Gonzalo_log.ldf' , SIZE = 8192KB , MAXSIZE =
2048GB , FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO
```

```
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Gonzalo].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
```

```
ALTER DATABASE [Gonzalo] SET ANSI_NULL_DEFAULT OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET ANSI_NULLS OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET ANSI_PADDING OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET ANSI_WARNINGS OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET ARITHABORT OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET AUTO_CLOSE OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET AUTO_SHRINK OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET AUTO_UPDATE_STATISTICS ON  
GO
```

```
ALTER DATABASE [Gonzalo] SET CURSOR_CLOSE_ON_COMMIT OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET CURSOR_DEFAULT GLOBAL  
GO
```

```
ALTER DATABASE [Gonzalo] SET CONCAT_NULL_YIELDS_NULL OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET NUMERIC_ROUNDABORT OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET QUOTED_IDENTIFIER OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET RECURSIVE_TRIGGERS OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET DISABLE_BROKER  
GO
```

```
ALTER DATABASE [Gonzalo] SET AUTO_UPDATE_STATISTICS_ASYNC OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET DATE_CORRELATION_OPTIMIZATION OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET TRUSTWORTHY OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET ALLOW_SNAPSHOT_ISOLATION OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET PARAMETERIZATION SIMPLE  
GO
```

```
ALTER DATABASE [Gonzalo] SET READ_COMMITTED_SNAPSHOT OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET HONOR_BROKER_PRIORITY OFF  
GO
```

```
ALTER DATABASE [Gonzalo] SET RECOVERY SIMPLE
```


GO

```
ALTER DATABASE [Gonzalo] SET MULTI_USER
GO
```

```
ALTER DATABASE [Gonzalo] SET PAGE_VERIFY CHECKSUM
GO
```

```
ALTER DATABASE [Gonzalo] SET DB_CHAINING OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
```

```
ALTER DATABASE [Gonzalo] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
```

```
ALTER DATABASE [Gonzalo] SET DELAYED_DURABILITY = DISABLED
GO
```

```
ALTER DATABASE [Gonzalo] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET QUERY_STORE = OFF
GO
```

```
ALTER DATABASE [Gonzalo] SET READ_WRITE
GO
```

/***** TABLA CONTACTOS *****/

```
CREATE TABLE [dbo].[contacto] (
    [id] [int] IDENTITY(1,1) primary key NOT NULL,
    [nombre] [nchar](10) NOT NULL,
    [apellido] [nchar](10) NOT NULL,
    [edad] [int] NOT NULL,
    [contraseña] [nchar](10) NOT NULL,
);
```

/***** TABLA LOGIN *****/

```
CREATE TABLE [dbo].[T_Login] (
    [Id] [int] IDENTITY(1,1) Primary key NOT NULL,
    [Nombre] [varchar](50) NOT NULL,
    [Pass] [varchar](10) NOT NULL,
);
```

```

/***** TABLA INMUEBLE *****/
CREATE TABLE [dbo].[Inmueble](
    [Id_Inmuebles] [int] IDENTITY(1,1) Primary Key NOT NULL,
    [Cod_Prod] [int] NOT NULL,
    [Direccion] [varchar](100) NULL,
    [P_Venta] [int] NOT NULL,
);

/**** Se agregan datos a la tabla inmueble****/
INSERT INTO dbo.Inmueble Values (0001,'Puerrydon 7859', 1500900);
INSERT INTO dbo.Inmueble Values (0002,'Soler 7859', 650900);
INSERT INTO dbo.Inmueble Values (0003,'Rivadavia 7859', 32500800);
INSERT INTO dbo.Inmueble Values (0004,'Av.San Martin 7859', 957000630);

/***** TABLA VENTA *****/

CREATE TABLE [dbo].[Venta](
    [Id_Venta] [int] IDENTITY(1,1) Primary Key NOT NULL,
    [Id_Inmueble] [int] NOT NULL,
    [Id_Contacto] [int] NOT NULL,
    [V_Venta] [int] NOT NULL,
);

/***** Crear usuario autorizado logeo *****/

INSERT INTO T_Login Values ('Diego', 1234);
INSERT INTO T_Login Values ('Hernan', 1234);
INSERT INTO T_Login Values ('Mara', 5555);
INSERT INTO T_Login Values ('Micaela', 1234);
```

