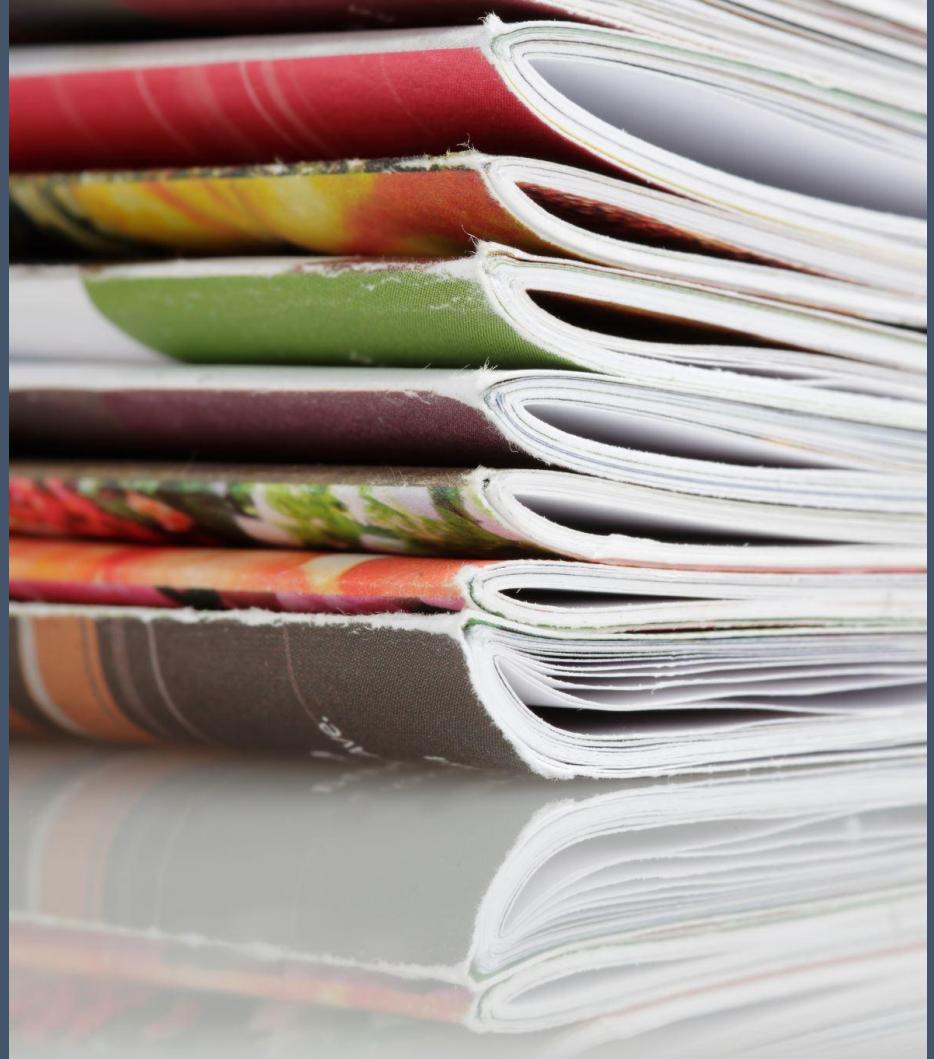


WebEscuelaFinal

Proyecto final de webEscuela





•En base a lo aprendido den los trabajos anteriores desarrollaremos el proyecto final de WebWEscuela

Iniciamos el proyecto con el aprestamiento

Elegimos un directorio Base del proyecto WebEscuela2021

En él creamos los subdirectorios

Datos

Desarrollo

Documentación

Testing

- 1) Agregamos en el directorio de **DATOS** el Script de creación de la base de datos DBESCUELA.SQL
- 2) En el directorio de **DESARROLLO** Creamos 3 Directorios
 - a) **CAPA DE DATOS**
Dentro de esta carpeta copiamos la Carpeta que contiene la librería de clases **BASICA**
 - b) **CAPA DE NEGOCIOS**
Dentro de esta carpeta Agregamos un nuevo proyecto de tipo ClassLibrary (.Net), denominado **BUSSINESCHOOL**
 - c) **CAPA DE PRESENTACION**
Dentro de esta carpeta creamos un proyecto de tipo Web vacío que denominaremos **WEBESCUELA2021**
- 3) En el directorio **DOCUMENTACIÓN** guardamos una copia de este documento
- 4) En el directorio **TESTING** se guardará una copia de WebEscuela que nos sirvió para probar las distintas actividades que hemos desarrollado





Base de Datos y tablas Roles Usuarios y Usuariosroles

- Vamos a Crear la base de datos
- BBWebEscuela
- En ella crearemos inicialmente tres tablas:
 - Roles
 - Usuarios
 - UsuariosRoles

Script de base de datos

DBESCUELA.sql (1)

```
use master
go
if exists (select * from sys.sysdatabases where name='DBWebEscuela')
drop database DBWebEscuela
go
create database DBWebEscuela
go
use DBWebEscuela
go
--CREACIÓN DE TABLA ROLES
create table Roles
(
Rol varchar(20) primary key
)
go
--HARDCODING ROLES
insert Roles values('ADMINISTRADOR')
insert Roles values('DIRECTOR DE ESTUDIOS')
insert Roles values('PROFESOR')
insert Roles values('PRECEPTOR')
insert Roles values('ALUMNO')
insert Roles values('INSCRIPTO')
insert Roles values('EXCLUIDO')
go
```

Script de base de datos

DBESCUELA.sql (2)

```
--CREACION DE TABLA USUARIO
create table Usuarios
(
ID int identity(1,1) primary key,
Nombre varchar(30)not null,
Dni int unique not null,
Direccion varchar (50)not null,
FechaNac smalldatetime not null,
Mail varchar(40)unique not null,
Telefono varchar(15)not null,
Password varchar(40)not null
)
go
```

```
create procedure Usuarios_Insert
(
@Nombre varchar(30),
@Dni int,
@Direccion varchar(50),
@FechaNac smallDateTime,
@Mail varchar(30),
@Telefono varchar(15),
@Password Varchar(40)
)
as begin
insert Usuarios
values(@Nombre,@Dni,@Direccion,@FechaNac,@Mail,@Telefono,@Password)
select @@IDENTITY
end
go
```

```
create procedure Usuarios_Update
(
@ID int,
@Nombre varchar(30),
@Dni int,
@Direccion varchar(50),
@FechaNac smallDateTime,
@Mail varchar(30),
@Telefono varchar(15),
@Password varchar(40)
)
as
```

Script de base de datos DBESCUELA.sql (3)

```
-- if @Password<>'' begin
--     Update Usuarios
--         set nombre=@Nombre,
--             Dni=@Dni, Mail=@Mail,
--             Direccion=@Direccion,
--             FechaNac=@FechaNac,
--             Telefono=@Telefono,
--             Password=@Password
--     where ID=@ID
-- end
else begin
    Update Usuarios
        set nombre=@Nombre,
            Dni=@Dni,
            Direccion=@Direccion,
            mail=@mail,
            FechaNac=@FechaNac,
            Telefono=@Telefono
    where ID=@ID
end
```

go

Script de base de datos

DBESCUELA.sql (4)

```
create procedure Usuarios_Delete(@ID int)
as
delete Usuarios where ID=@ID
go
create procedure Usuarios_Find(@ID int)
as
select * from Usuarios where ID=@ID
go
--Procedures de Usuarios Especiales de existencia
create procedure Usuarios_MailExists(@ID int,@Mail varchar(40))
as
select count(*) from Usuarios where
ID<>@ID and Mail=@Mail
go
create procedure Usuarios_DniExists(@ID int,@Dni int)
as
select count(*) from Usuarios where
ID<>@ID and Dni=@Dni
Go
```

Script de base de datos

DBESCUELA.sql (5)

```
--Procedures de Usuarios Especiales de búsqueda

create procedure Usuarios_FindByMail(@Mail
varchar(40))
as
select * from Usuarios where Mail=@Mail
go
create procedure Usuarios_FindByDni(@Dni int)
as
select * from Usuarios where Dni=@Dni
go
```

```
create procedure
Usuarios_FindByMailAndDni(@Mail
varchar(40),@Dni int)
as
select * from Usuarios where Mail=@Mail and
Dni=@Dni
go
create procedure Usuarios_Login(@Mail
varchar(40),@Password varchar(40))
as
select * from Usuarios where Mail=@Mail and
Password=@Password
go
```

```
--Procedures de listado
create procedure Usuarios_List
as select * from Usuarios
go
```

```
create Table UsuariosRoles
(
    ID int identity(1,1)Primary key,
    IDUsuario int foreign key References Usuarios(ID) on delete cascade not null,
    Rol varchar(20) foreign Key References Roles(Rol) not null,
    constraint UKUR unique(IDUsuario,Rol)
)
go
```

Script de base de datos DBESCUELA.sql (6)

```
--STORE PROCEDURES DE USUARIOSROLES
create procedure UsuariosRoles_Insert(@IDUsuario
int,@Rol varchar(20))
as begin
insert UsuariosRoles values(@IDUsuario,@Rol)
select @@IDENTITY
end
go
```

```
--create procedure UsuariosRoles_Update( @ID int,@IDUsuario int,@Rol varchar(20))
--as
--update UsuariosRoles set IDUsuario=@IDUsuario,Rol=@Rol where ID=@ID
--go
--create procedure UsuariosRoles_Delete( @IDUsuario int,@Rol varchar(20))
--as
--    Delete UsuariosRoles where IDUsuario=@IDUsuario and Rol=@Rol
--go
--create procedure UsuariosRoles_Find( @ID int)
--as
--    select * from UsuariosRoles where ID=@ID
--go
--procedures de listados
--create procedure UsuariosRoles_ListByUsuario(@IDUsuario int)
--as select Rol From UsuariosRoles where IDUsuario=@IDUsuario
--go
```

Script de base de datos

DBESCUELA.sql (7)

```
--HardCoding Usuario Administrador
--Nombre='admin',
--Dni=1,
--Direccion='dir admin',
--FechaNac='01-01-1990'
--Mail='admin@mail.com'
--Telefono='000000000000'
--El password se obtiene una dispersión de
'admin'
--Password=''
```

```
insert usuarios values(
'admin',
1,
'dir admin',
'01-01-1990',
'admin@mail.com',
'000000000000',
'D033E22AE348AEB5660FC2140AEC35850C4DA997')
--password ='admin' (con dispersion Sha1)
--hardcoding usuarios roles
insert UsuariosRoles
values(1, 'ADMINISTRADOR')
insert UsuariosRoles values(1, 'DIRECTOR DE
ESTUDIOS')
```

Agregado de código en BASICA

- Se modificará a la clase **Json** en el método ***string AddObject()***
 - Donde se tendrá en cuenta también si **value** es un objeto o un arreglo
- Se agregará en la Interface **IABMFIMG** ***string MakeData{get;}***
 - Devuelve un string que contiene **ID , BaseDirectory , DirectorY , Prefix y la extensión**
- Se agrega en IMakelmg la property ***string Extensión{get;set;}***
- Modifico en Image
 - El método **void SplitData(*string Data*);**
 - ***string Extension{get;set;}***
 - ***string Path***
 - ***String URL(strig Data)***

Agregado de Código en Básica

```
public interface IABMFIMG : IABMF
{
    string DirBase { get; }
    string DirectorY { get; }
    string Prefix { get; }
    string MakeData { get; }
    System.Web.HttpPostedFile Fu { get; set; }
```

```
string Q_ = "\"";
public string addObject(string parent, string name, string value)
{
    string Sufix="";
    if (parent.StartsWith("{})){ Sufix = "}" ); }
    if(!value.StartsWith("[")&&!value.StartsWith("{"))value=Q_ + value + Q_;
    if (parent.StartsWith("[")){ Sufix = "]"; }
    name = Q_ + name + Q_ + ":" + value+ Sufix ;
    if(parent!="{}"&&parent!="[]") { name = "," + name; }
    return parent.Remove(parent.Length - 1) + name;
}
```

```
public class Image : IMakeImg
{
    public string DirBase { get; set; }
    public string Directory { get; set; }
    public string Path => AppDomain.CurrentDomain.BaseDirectory+
        DirBase+
        @"\"+Directory+"\"+
        Prefix+ID+"."+Extension;
    public string Prefix { get; set; }
    public int ID { get; set; }
    public string Extension { get; set; }
    public void ResetPrefix()...
    public void ChangPrefix()...
    public void Add(HttpPostedFile FU, string Data)...
    public void Erase(string Data)...
    public void SplitData(string Data)
    {
        string[] Datos = Data.Split(new char[] { ';' });
        ID = int.Parse(Datos[0]);
        DirBase = Datos[1];
        Directory = Datos[2];
        Prefix = Datos[3];
        Extension = Datos[4];
    }
    public string URL(string Data)
    {
        SplitData(Data);
        ResetPrefix();
        if (File.Exists(Path))
        {
            return DirBase + "/" + Directory + "/" + Prefix + ID + "."+Extension;
        }
        ChangPrefix();
        if (File.Exists(Path))
        {
            return DirBase + "/" + Directory + "/" + Prefix + ID + "."+Extension;
        }
        ResetPrefix();
        return DirBase + "/" + Directory + "/" + Prefix+"default."+Extension;
    }
}
```

```
namespace BASICA
{
    public interface IMakeImg : IIImage
    {
        string DirBase { get; set; }
        string Directory { get; set; }
        string Path { get; }
        string Extension { get; set; }
        string Prefix { get; set; }
        void ResetPrefix();
        void ChangPrefix();
        void SplitData(string Data);
    }
}
```

Agregado de Código en Básica

**CLICK EN LA
IMAGEN PARA VER
EL VIDEO
EXPLICATIVO**

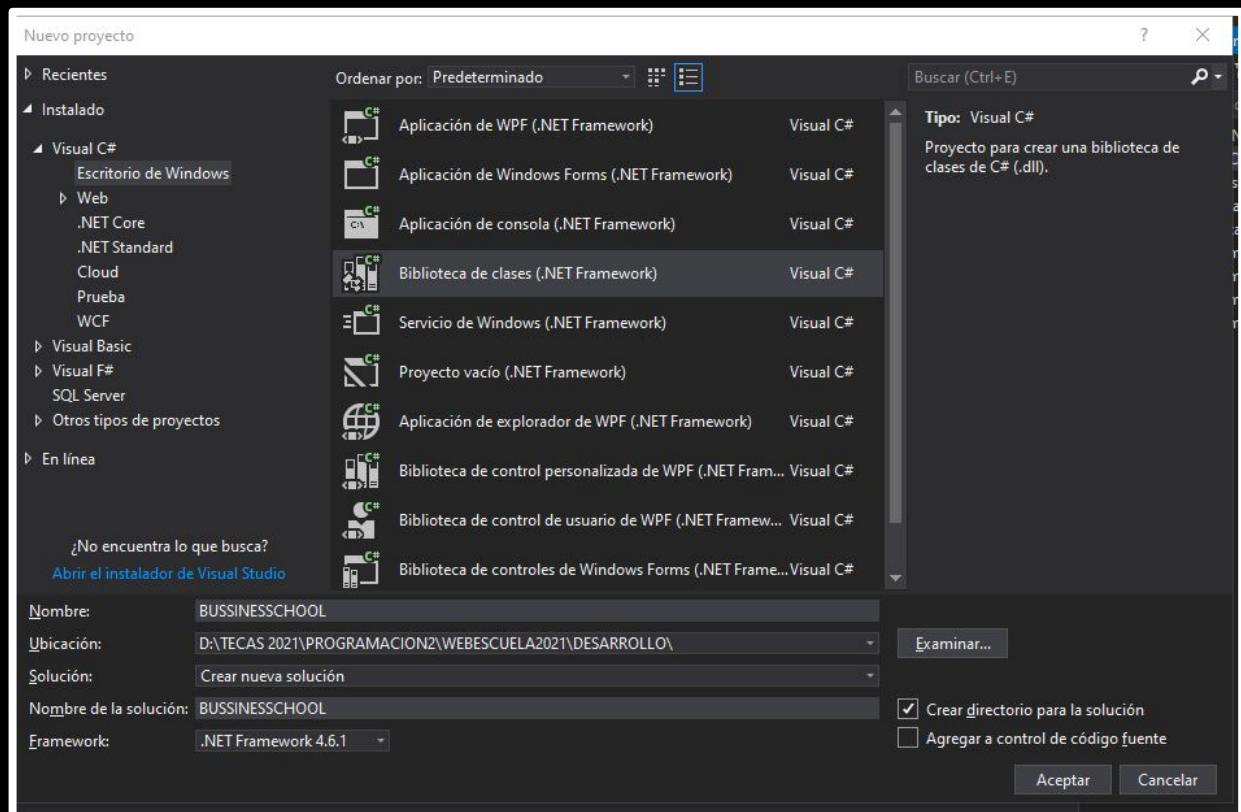


- Creación de la capa de negocios
- BUSSINESSCHOOL
- Pasos a seguir
 - Agregar proyecto existente BASIC
 - Referenciar
 - System.Web
 - BASICA
 - Creación de clases
 - Usuario
 - RolUsuario
 - Singleton

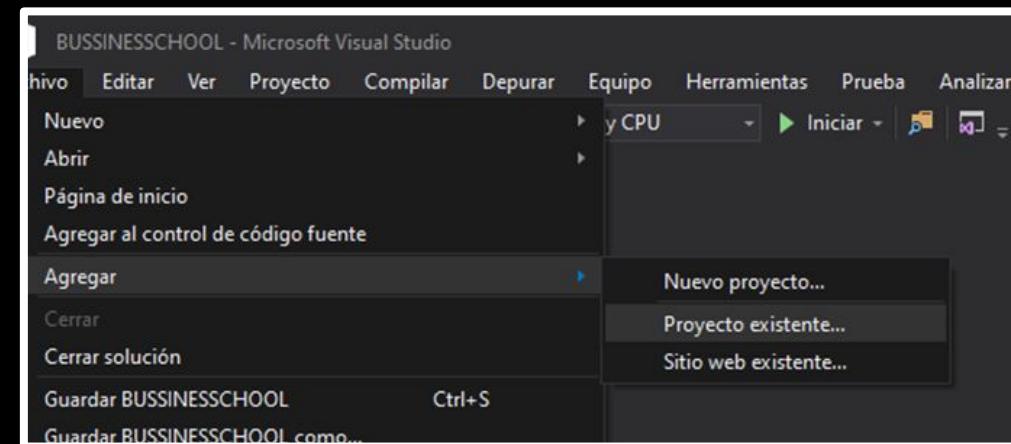


BUSSINESCHOOL

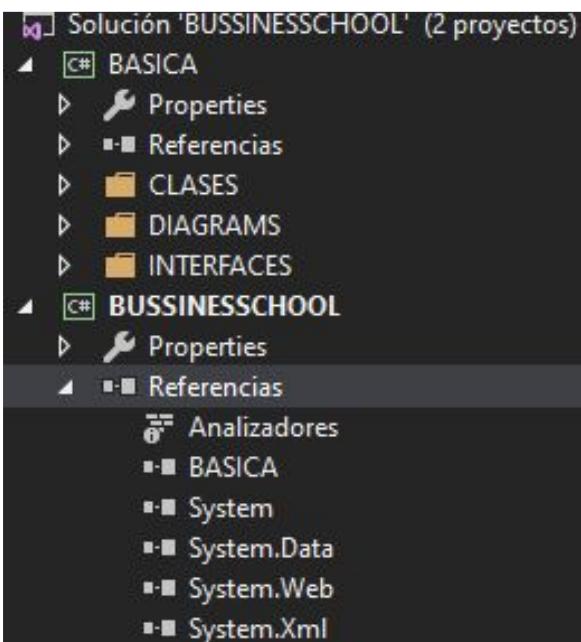
1 Creamos una biblioteca de Clases BUSSINESCHOOL



2 Agregamos Proyecto BASICA



3 Agregamos referencias BASICA System.Web



Clases de la capa Bussines

Usuario

- Contiene imagen
- Va a implementar una interface Heredera IABMFIImg

RolUsuario

- No contiene imagen
- Implementa una interface heredera de IABMF

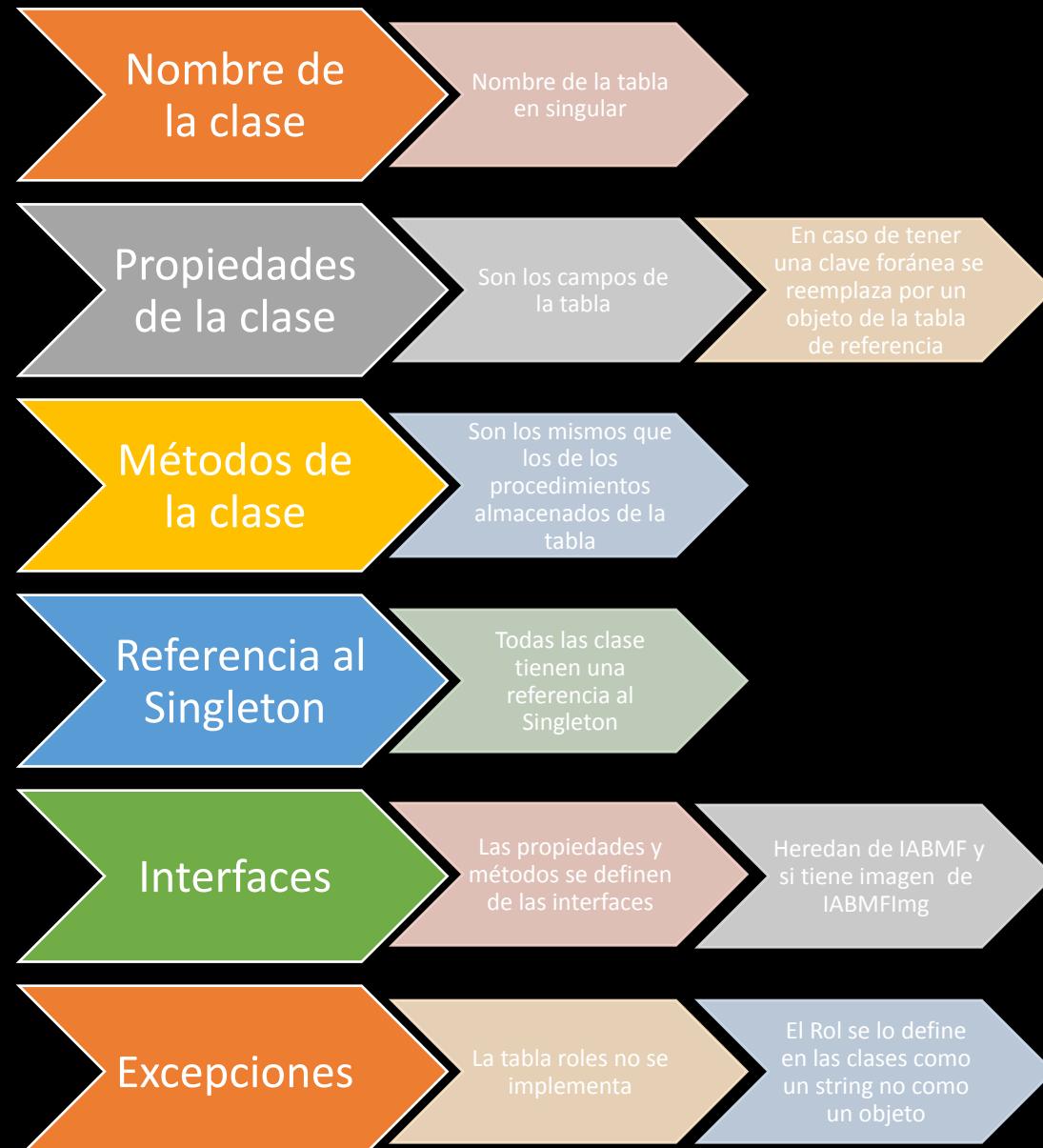
Singleton

- Hereda de Parent Singleton
- Clase parcial e interna

En qué se basa el criterio de elección de las clases en la capa de negocios



Ejemplo de construcción de Datos



Lo primero que se construye en la capa de negocios es el Singleton

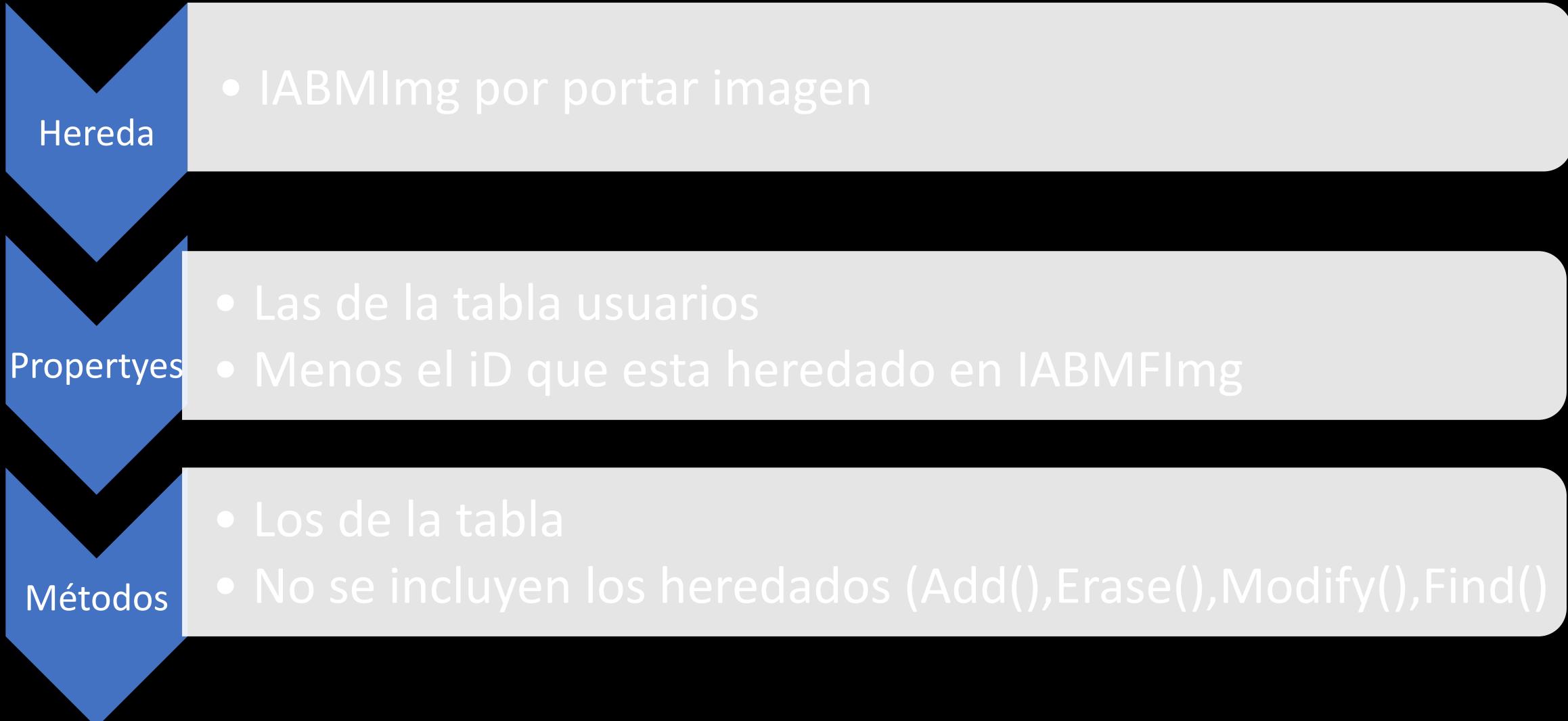
Singleton

- Es el eje sobre el que va a girar la capa de negocios
- Todas las clases lo van a referenciar
- Es clase parcial y de acceso internal
 - (No se puede utilizar desde una capa externa a la capa de negocios)

Código de Singleton

```
using BASICA;
namespace BUSSINESCHOOL
{
    internal partial class Singleton:ParentSingleton
    {
        static Singleton instance => new Singleton();
        private Singleton() { }
        public static Singleton GetInstance => instance;
    }
}
```

Interface del usuario IUsuario



Contrato de datos respecto a la tabla

Clase Usuario	Tabla Usuarios
<ul style="list-style-type: none">• int ID{get;set;}• string Nombre{get;set;}• int Dni{get;set;}• string Direccion{get;set;}• DateTime FechaNac{get;set;}• string Mail{get;set;}• string Telefono{get;set;}• string Password	<ul style="list-style-type: none">• ID int• Nombre VC(30)• Dni int• Direccion vc(50)• FechaNac smalldatetime• Mail vc(40)• Telefono vc(15)• Password vc(40)

Contrato de métodos respecto a la tabla

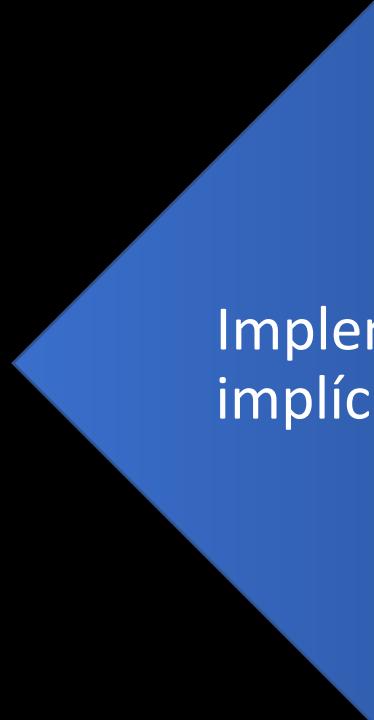
Métodos Usuario	Store Procedures Usuarios
<ul style="list-style-type: none">• void Add()• void Modify()• void Erase()• string Find()• bool MailExists()• bool DniExists()• string FindByMail()• string FindByDni()• string FinByMailAndDni• string Login()• string List()	<ul style="list-style-type: none">• Usuarios_Insert• Usuarios_Update• Usuarios_Delete• Usuarios_Find• Usuarios_MailExists• Usuarios_DniExists• Usuarios_FindByMail• Usuarios_FindByDni• Usuarios_FindByMailAndDni• Usuarios_Login• Usuarios_List

```
namespace BUSSINESCHOOL
{
    public interface IUsuario : BASICA.IABMFIMG
    {
        //data contract
        string Nombre { get; set; }
        int Dni { get; set; }
        string Direccion { get; set; }
        System.DateTime FechaNac { get; set; }
        string Mail { get; set; }
        string Telefono { get; set; }
        string Password { get; set; }
        //Method contract
        bool MailExists();
        bool DniExists();
        string FindByMail();
        string FindByDni();
        string FindByMailAndDni();
        string Login();
        string List();
    }
}
```

Código de interface Iusuario

```
using System;
using System.Web;
namespace BUSSINESCHOOL
{
    public class Usuario : IUsuario
    {
        Properties
        Methods
    }
}
```

Class Usuario



Implementamos en forma
implícita IUsuario

Class Usuario dataContract

- Se quitan las excepciones de las properties y se le asignan valores

```
internal Singleton S => Singleton.GetInstance;
#region Properties
public string Nombre { get; set; }
public int Dni { get; set; }
public string Direccion { get; set; }
public DateTime FechaNac { get; set; }
public string Mail { get; set; }
public string Telefono { get; set; }
public string Password {get; set; }
public string DirBase => "IMAGENES";
public string DirectorY =>"USUARIOS";
public string Prefix => "USUARIO";
public HttpPostedFile Fu { get ; set; }
public int ID { get ; set; }
public string MakeData => ID + ";" + DirBase + ";" + DirectorY + ";" + Prefix + ";" +"jpg";
#endregion
Methods
```

Class Usuario Method Contract

```
using System;
using System.Web;
namespace BUSSINESCHOOL
{
    public class Usuario : IUsuario
    {
        #region Properties
        #region Methods
        public void Add()...
        public bool DniExists()...
        public void Erase()...
        public string Find()...
        public string FindByDni()...
        public string FindByMail()...
        public string FindByMailAndDni()...
        public string List()...
        public string Login()...
        public bool MailExists()...
        public void Modify()...
        #endregion
    }
}
```



Los métodos quedan para implementarse después

Interface ISingletonUsuario

ISingletonUsuario

- Hereda de IGenericSingleton<Usuario>
- Tiene el desarrollo del Method Contract de usuarios se pasa Como Dato un objeto de la clase Usuario

Métodos a Agregar en SingletonUsuario

ISingletonUsuario

- bool MailExists(Usuario Data);
- bool DniExists(Usuario Data);
- string FindByDni(Usuario Data);
- string FindByMail(Usuario Data);
- string FindByDniAndMail(Usuario Data);
- string Login(Usuario Data);
- string List(Usuario Data);

IUsuario

- bool MailExists();
- bool DniExists();
- string FindByDni();
- string FindByMail();
- string FindByDniAndMail();
- string Login();
- string List();



ISingletonUsuario

```
namespace BUSSINESCHOOL
{
    interface ISingletonUsuario:BASICA.IGenericSingleton<Usuario>
    {
        bool MailExists(Usuario Data);
        bool DniExists(Usuario Data);
        string FindByDni(Usuario Data);
        string FindByMail(Usuario Data);
        string FindByMailAndDni(Usuario Data);
        string Login(Usuario Data);
        string List(Usuario Data);
    }
}
```

Implementación parcial de Singleton.Usuario.cs (1)

```
using System;
using System.Data;
using BASICA;

namespace BUSSINESCHOOL
{
    internal partial class Singleton : ISingletonUsuario
    {
        public ISingletonUsuario ISU => this;
        void IGenericSingleton<Usuario>.Add(Usuario Data)...
        bool ISingletonUsuario.DniExists(Usuario Data)...
        void IGenericSingleton<Usuario>.Erase(Usuario Data)...
        string IGenericSingleton<Usuario>.Find(Usuario Data)...
        string ISingletonUsuario.FindByDni(Usuario Data)...
        string ISingletonUsuario.FindByMail(Usuario Data)...
        string ISingletonUsuario.FindByMailAndDni(Usuario Data)...
        string ISingletonUsuario.List(Usuario Data)...
        string ISingletonUsuario.Login(Usuario Data)...
        bool ISingletonUsuario.MailExists(Usuario Data)...
        string IGenericSingleton<Usuario>.MakeJson(DataRow Dr, Usuario Data)...
        void IGenericSingleton<Usuario>.Modify(Usuario Data)...
    }
}
```

La implementación es explícita

Puntero que apunta a Singleton de modificador de acceso Público

Implementación parcial de Singleton.Usuario.cs (2)

```
void IGenericSingleton<Usuario>.Add(Usuario Data)
{
    if (Data.DniExists()) { throw new Exception("Existe otro usuario con el mismo Dni"); }
    if (Data.MailExists()) { throw new Exception("Existe otro usuario con el mismo Mail"); }
    Data.Password = IHashing.Hash(Data.Password);
    IConnection.CreateCommand("Usuarios_Insert","Usuario");
    IConnection.AddVarchar("Nombre", 30, Data.Nombre);
    IConnection.AddInt("Dni", Data.Dni);
    IConnection.AddVarchar("Direccion", 50, Data.Direccion);
    IConnection.AddDatetime("FechaNac", Data.FechaNac);
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    IConnection.AddVarchar("Telefono", 15, Data.Telefono);
    IConnection.AddVarchar("Password", 40, Data.Password);
    Data.ID = IConnection.Insert();
    IImage.Add(Data.Fu, Data.MakeData);
}
```

Implementación parcial de SingletonUsuario.cs (3)

```
bool ISingletonUsuario.DniExists(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_DniExists", "Usuario");
    IConnection.AddInt("ID", Data.ID);
    IConnection.AddInt("Dni", Data.Dni);
    return IConnection.Exists();
}

void IGenericSingleton<Usuario>.Erase(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_Delete", "Usuario");
    IConnection.AddInt("ID", Data.ID);
    IConnection.Delete();
    IIImage.Erase(Data.MakeData());
}
```

Implementación parcial de SingletonUsuario.cs (4)

```
string IGenericSingleton<Usuario>.Find(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_Find", "usuario");
    IConnection.AddInt("ID", Data.ID);
    DataRow Dr = IConnection.Find();
    return ISU.MakeJson(Dr, Data);
}
string ISingletonUsuario.FindByDni(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_FindByDni", "usuario");
    IConnection.AddInt("Dni", Data.Dni);
    DataRow Dr = IConnection.Find();
    return ISU.MakeJson(Dr, Data);
}
```

Implementación parcial de SingletonUsuario.cs (5)

```
string ISingletonUsuario.FindByMail(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_FindByMail", "usuario");
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    DataRow Dr = IConnection.Find();
    return ISU.MakeJson(Dr, Data);
}
string ISingletonUsuario.FindByMailAndDni(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_FindByMailAndDni", "usuario");
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    IConnection.AddInt("Dni", Data.Dni);
    DataRow Dr = IConnection.Find();
    return ISU.MakeJson(Dr, Data);
}
```

Implementación parcial de SingletonUsuario.cs (6)

```
string ISingletonUsuario.List(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_List", "Usuarios");
    DataTable Dt = IConnection.List();
    return this.IListToTable.TableToJson<Usuario>(Dt, ISU);
}
string ISingletonUsuario.Login(Usuario Data)
{
    Data.Password = IHashing.Hash(Data.Password);
    IConnection.CreateCommand("Usuarios.login", "Usuario");
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    IConnection.AddVarchar("Password", 40, Data.Password);
    DataRow Dr = IConnection.Find();
    return ISU.MakeJson(Dr, Data);
}
```

Implementación parcial de SingletonUsuario.cs (7)

```
        bool ISingletonUsuario.MailExists(Usuario Data)
{
    IConnection.CreateCommand("Usuarios_MailExists", "Usuario");
    IConnection.AddInt("ID", Data.ID);
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    return IConnection.Exists();
}
void IGenericSingleton<Usuario>.Modify(Usuario Data)
{
    if (Data.DniExists()) { throw new Exception("Existe otro usuario con el mismo Dni"); }
    if (Data.MailExists()) { throw new Exception("Existe otro usuario con el mismo Mail"); }
    if (Data.Password != "") { Data.Password = IHashing.Hash(Data.Password); }
    IConnection.CreateCommand("Usuarios_Update", "Usuario");
    IConnection.AddInt("ID", Data.ID);
    IConnection.AddVarchar("Nombre", 30, Data.Nombre);
    IConnection.AddInt("Dni", Data.Dni);
    IConnection.AddVarchar("Direccion", 50, Data.Direccion);
    IConnection.AddDatetime("FechaNac", Data.FechaNac);
    IConnection.AddVarchar("Mail", 40, Data.Mail);
    IConnection.AddVarchar("Telefono", 15, Data.Telefono);
    IConnection.AddVarchar("Password", 40, Data.Password);
    IConnection.Update();
    IIImage.Add(Data.Fu, Data.MakeData);
}
```

Implementación parcial de SingletonUsuario.cs (8)

```
string IGenericSingleton<Usuario>.MakeJson(DataRow Dr, Usuario Data)
{
    string json = "{}";
    string s = "ID";
    Data.ID = int.Parse(Dr[s].ToString());
    json = IJson.addObject(json, s, Dr[s].ToString());
    s = "Nombre";
    Data.Nombre = Dr[s].ToString();
    json = IJson.addObject(json, s, Dr[s].ToString());
    s = "Dni";
    Data.Dni = int.Parse(Dr[s].ToString());
    json = IJson.addObject(json, s, Dr[s].ToString());
    s = "Direccion";
    Data.Direccion = Dr[s].ToString();
    json = IJson.addObject(json, s, Dr[s].ToString());
    s = "FechaNac";
    Data.FechaNac = DateTime.Parse(Dr[s].ToString());
    json = IJson.addObject(json, s, Data.FechaNac.Year+"-"+Data.FechaNac.Month+"-"+Data.FechaNac.Day);
    s = "Mail";
    Data.Mail = Dr[s].ToString();
    json = IJson.addObject(json, s, Dr[s].ToString());
    s = "Telefono";
    Data.Telefono = Dr[s].ToString();
    json = IJson.addObject(json, s, Dr[s].ToString());
    string Url = IIImage.URL(Data.ID + ";" + Data.DirBase + ";" + Data.DirectorY + ";" + Data.Prefix);
    json = IJson.addObject(json, "Url", Url);
    return json;
}
```

Devuelve el formato de fecha que utilizamos (aaaa-mm-dd)

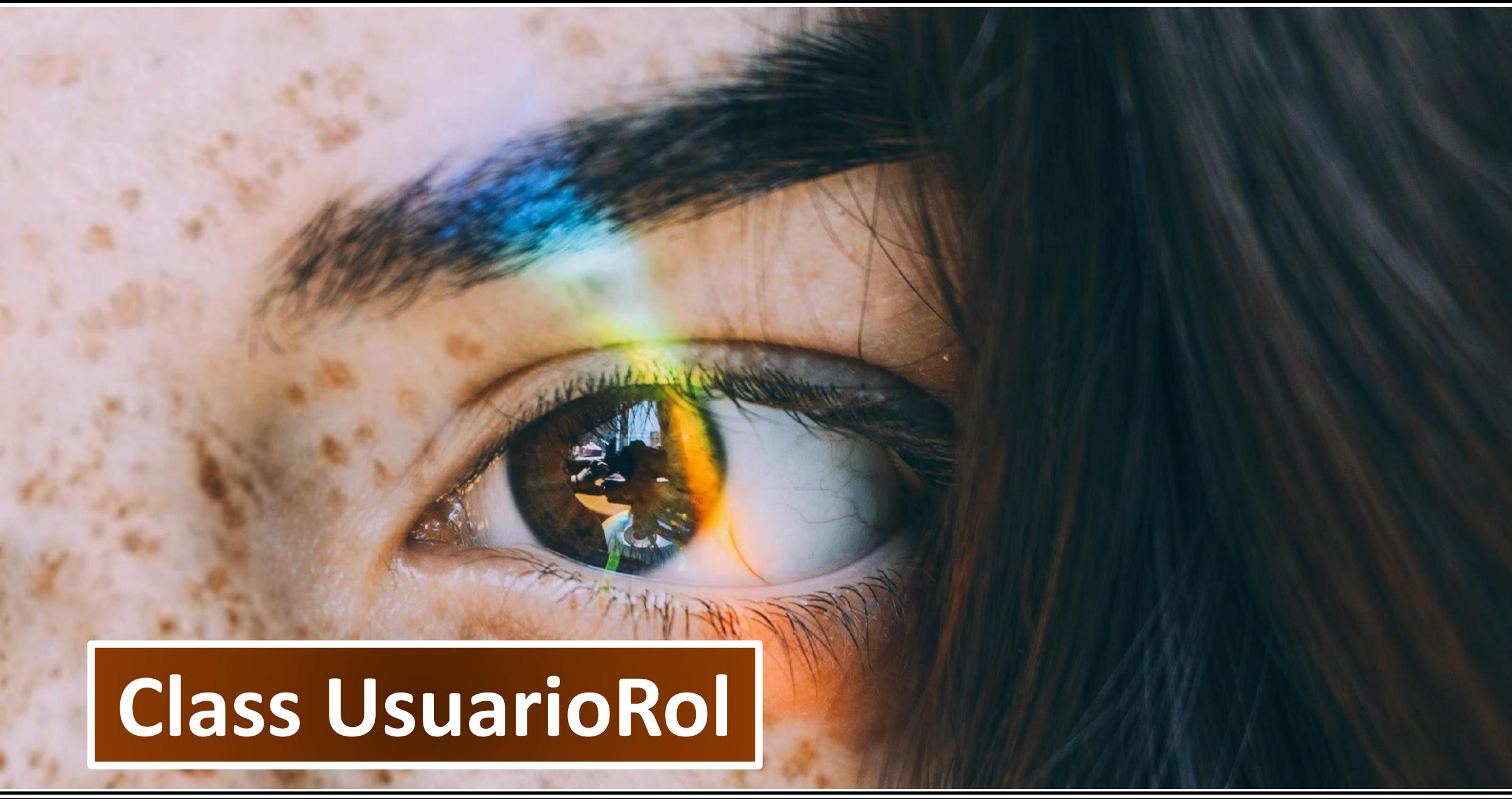
Ver video
click aquí



Class Usuario

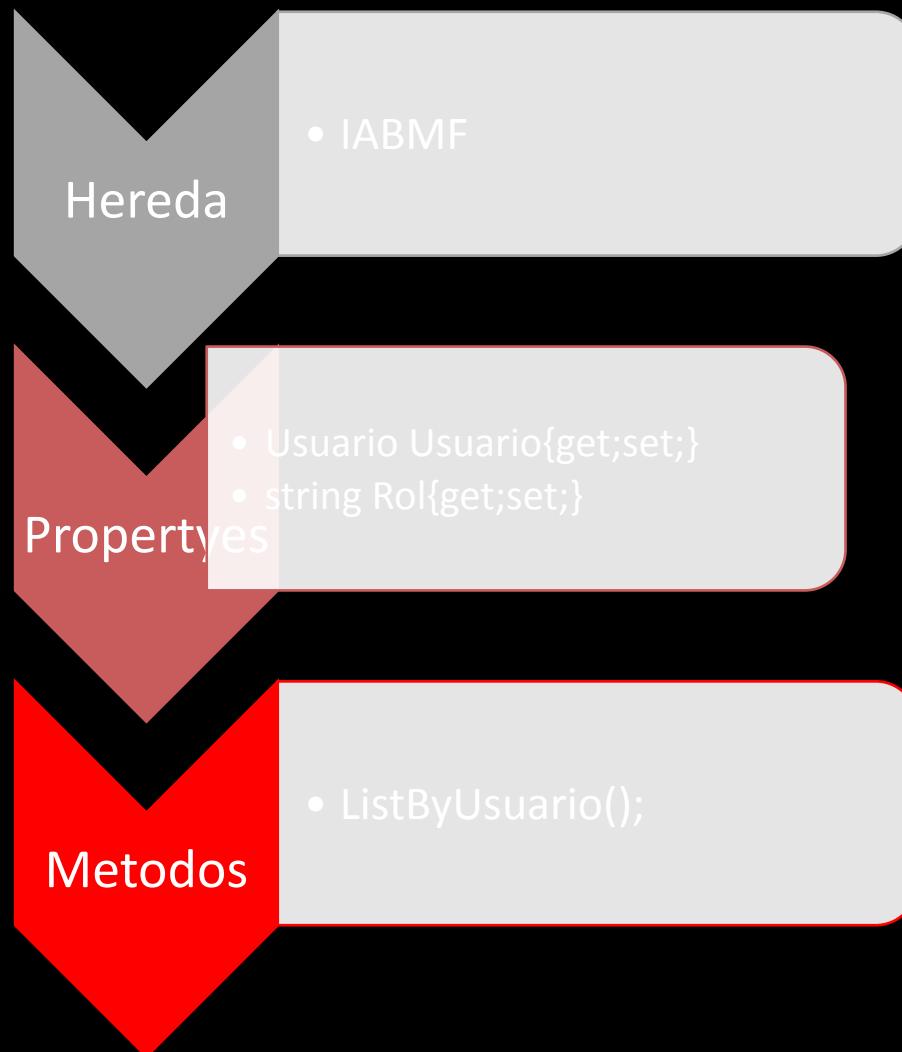
Class Usuario method Contract

```
using System;
using System.Web;
namespace BUSSINESCHOOL
{
    public class Usuario : IUsuario
    {
        internal Singleton S => Singleton.GetInstance;
        #region Properties
        #region Methods
        public void Add(){ S.ISU.Add(this); }
        public bool DniExists(){return S.ISU.DniExists(this);}
        public void Erase(){S.ISU.Erase(this);}
        public string Find(){return S.ISU.Find(this);}
        public string FindByDni(){return S.ISU.FindByDni(this);}
        public string FindByMail(){return S.ISU.FindByMail(this);}
        public string FindByMailAndDni(){return S.ISU.FindByMailAndDni(this);}
        public string List(){return S.ISU.List(this);}
        public string Login(){return S.ISU.Login(this);}
        public bool MailExists(){return S.ISU.MailExists(this);}
        public void Modify(){S.ISU.Modify(this);}
        #endregion
    }
}
```

A close-up photograph of a person's eye, likely brown or hazel, with a vibrant rainbow reflection in the iris. The eye is framed by dark, wavy hair. The background is a soft, out-of-focus beige color.

Class UsuarioRol

Interface IUsuarioRol



Interface IUsuarioRol

```
namespace BUSSINESCHOOL
{
    public interface IUsuarioRol:BASICA.IABMF
    {
        Usuario Usuario { get; set; }
        string Rol { get; set; }
        string ListByUsuario();
    }
}
```

Class UsuarioRol

```
namespace BUSSINESCHOOL
{
    public class UsuarioRol : IUsuarioRol
    {
        internal Singleton S => Singleton.GetInstance;

        public Usuario Usuario { get; set; }
        public string Rol { get; set; }
        public int ID { get; set; }
        public void Add()...
        public void Erase()
        public string Find()...
        public string ListByUsuario()...
        public void Modify()...
    }
}
```

Referencia a Singleton

Implementación
Implicita



Interface ISingletonUsuarioRol

```
namespace BUSSINESCHOOL
{
    public interface ISingletonUsuarioRol:BASICA.IGenericSingleton<UsuarioRol>
    {
        string ListByUsusario(UsuarioRol Data);
    }
}
```

Singleton.UsuarioRol.js (1)

```
using System.Data;
using BASICA;

namespace BUSSINESCHOOL
{
    internal partial class Singleton : ISingletonUsuarioRol
    {
        public ISingletonUsuarioRol ISUR => this;

        void IGenericSingleton<UsuarioRol>.Add(UsuarioRol Data)...
        void IGenericSingleton<UsuarioRol>.Erase(UsuarioRol Data)...
        string IGenericSingleton<UsuarioRol>.Find(UsuarioRol Data)...
        string ISingletonUsuarioRol.ListByUsusario(UsuarioRol Data)...
        string IGenericSingleton<UsuarioRol>.MakeJson(DataRow Dr, UsuarioRol Data)...
        void IGenericSingleton<UsuarioRol>.Modify(UsuarioRol Data)...
    }
}
```

Referencia Pública a interface

Implementación
Explicita de interface

Singleton.UsuarioRol.js (2)

```
void IGenericSingleton<UsuarioRol>.Add(UsuarioRol Data)
{
    IConnection.CreateCommand("UsuariosRoles_Insert", "Rol de Usuario");
    IConnection.AddInt("IDUsuario", Data.Usuario.ID);
    IConnection.AddVarchar("Rol", 20, Data.Rol);
    Data.ID = IConnection.Insert();
}
void IGenericSingleton<UsuarioRol>.Erase(UsuarioRol Data)
{
    IConnection.CreateCommand("UsuariosRoles_Delete", "Rol de Usuario");
    IConnection.AddInt("IDUsuario", Data.Usuario.ID);
    IConnection.AddVarchar("Rol", 20, Data.Rol);
    IConnection.Delete();
}
```

Singleton.UsuarioRol.js (3)

```
string ISingletonUsuarioRol.ListByUsuario(UsuarioRol Data)
{
    IConnection.CreateCommand("UsuariosRoles_ListByUsuario", "Roles de Usuario");
    IConnection.AddInt("IDUsuario", Data.Usuario.ID);
    DataTable Dt = IConnection.List();
    string JsonUser = Data.Usuario.Find();
    string Roles = IListToTable.TableToJson<UsuarioRol>(Dt, ISUR);
    return IJson.addObject(Data.Usuario.Find(), "Roles", Roles);
}
string IGenericSingleton<UsuarioRol>.MakeJson(DataRow Dr, UsuarioRol Data)
{
    return "\"" + Dr["ROL"] + "\"";
}
void IGenericSingleton<UsuarioRol>.Modify(UsuarioRol Data)
{
    IConnection.CreateCommand("UsuariosRoles_Update", "Rol de Usuario");
    IConnection.AddInt("ID", Data.ID);
    IConnection.AddInt("IDUsuario", Data.Usuario.ID);
    IConnection.AddVarchar("Rol", 20, Data.Rol);
    IConnection.Update();
}
```

Class UsuarioRol



Class UsuarioRol

```
namespace BUSSINESCHOOL
{
    public class UsuarioRol : IUsuarioRol
    {
        internal Singleton S => Singleton.GetInstance;

        public Usuario Usuario { get; set; }
        public string Rol { get; set; }
        public int ID { get; set; }
        public void Add() { S.ISUR.Add(this); }
        public void Erase() { S.ISUR.Erase(this); }
        public string Find() { return S.ISUR.Find(this); }
        public string ListByUsuario() { return S.ISUR.ListByUsuario(this); }
        public void Modify() { S.ISUR.Modify(this); }

    }
}
```

[Ver video Tema
anterior](#)
[click aquí](#)



Capa de presentacion

WebEscuelaFinal

Proyecto

- Sitio Web vacío
- En el web.config.agregar la connectionString
- Agregar BUSSINESCHOOL
- Agregar BASICA
- Referenciar a BUSSINESCHOOL

Carpetas

- Se agregan
 - IMÁGENES dentro de ella
 - GENERAL
 - CARRERAS con carreraDefault.jpg
 - USUARIOS con usuarioDefault.jpg
 - CSS
 - JS

**se agregan
Default.aspx y
default.html y
los archivos de
imagen**

Default.aspx borrado el Html

El Default Html el mismo de Web escuela

Los CSS se cargan los de WebEscuela

En JS

Cargamos de WebEscuela

Main.js

Datos.js

Dom.js

Form.js

Usuarios.js

Nav.js

General.js

Carreras.js

Crear sitio Web

The screenshot shows the Microsoft Visual Studio 2022 interface. On the left, there's a sidebar with sections like 'Introducción', 'Cree su primera aplicación', 'Maximice su producto', 'Aproveche las últimas mejoras de costo y confiabilidad', 'Desarrolle modernamente', 'Recientemente abierto' (WEBSCUELAFINAL.DATECAS 2022, WEBGESTAL.DATECAS 2022), 'Ayer' (BUSSINESS.DATECAS 2022, BASICA.csproj), and 'Hoy' (BASICAS.csproj). The main area shows the 'Nuevo proyecto' (New Project) dialog for Visual C#. The 'Tipos' dropdown is set to 'Visual C#'. The list includes: Sitio web vacío de ASP.NET, Sitio de ASP.NET Web Forms, Sitio web ASP.NET (Razor v3), Sitio web de entidades de datos dinámicos de ASP.NET, Sitio web de Linq to SQL de datos dinámicos de ASP.NET, and Servicio WCF. The 'Nombre:' field is filled with 'WebEscuelaFinal', and the 'Framework:' dropdown is set to '.NET Framework 4.6.1'. The 'Ubicación:' field is empty. Below the dialog, the 'Examinar...' (Browse...) button is highlighted. To the right, the 'Explorador de soluciones' (Solution Explorer) shows the solution 'WebEscuelaFinal' containing three projects: 'BASICA', 'BUSSINESSCHOOL', and 'WebEscuelaFinal'. The 'WebEscuelaFinal' project is expanded, showing its files: 'Bin' (containing 'BASICA.dll' and 'BASICA.pdb'), 'CSS' (containing 'FORMULARIOS.css', 'GENERAL.css', 'NAVEGACION.css', 'TABLAS.css'), 'IMAGENES' (containing 'CARRERAS' with 'carreradefault.jpg', 'GENERAL' with 'AULA UNIVERSITARIA.jpg', 'CLEAR.png', 'LOGO1.jpg', and 'USUARIOS' with 'usuariodefault.jpg'), 'JS' (containing 'Default.aspx', 'Default.html', 'packages.config', and 'Web.config'). The status bar at the bottom shows '15:56 viernes 13/8/2021'.

Default.aspx Default.aspx.cs

DeFault.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

DeFault.aspx.cs

```
using System;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request["accion"] != null)
        {
            switch (Request["accion"])
            {
                ...
            }
        }
    }
}
```

Ejecutando el sitio

INICIO CARRERAS NOTICIAS TESTING INGRESAR AL SISTEMA



DERECHOS HUMANOS
HELENITA ESPAÑA, DEFENSORA HABLA

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541
v.BALLESTER BS.AS.
TEL 25433234 MAIL
cballester@mail.com.ar



En el nav suprimimos el Testing



The image shows a classroom setting. Two men are standing at the front of the room, facing a group of students who are seated at their desks. The man on the left is wearing a white t-shirt and jeans, while the man on the right is wearing a grey t-shirt and jeans. They appear to be giving a presentation or lecture. In the background, there is a projector screen displaying a slide with the text 'DERECHOS HUMANOS' and some small icons. The room has a yellow wall with various educational posters and a blue chalkboard. There are also windows with blinds and a clock on the wall.

INICIO CARRERAS NOTICIAS INGRESAR AL SISTEMA

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541
v.BALLESTER BS.AS.
TEL 25433234 MAIL
cballester@mail.com.ar

```
const makeLinkAnonimous = function () {
    Nav.innerHTML = "";
    ul = $dc.ul(Nav);
    makeButton("INICIO", Home);
    makeButton("CARRERAS", $ec.listCarreras);
    makeButton("NOTICIAS", News);
    //TESTING//
    //dropdownButton("TESTING", ["ALUMNO DE TABLA", "HASHING"], [MakeAlumnos, MakePassword]);
};
```

En Datos.js
corregimos
el script de
esta manera

```
const Post = function (data)
{
    //funciona con browsers da ultima generacion
    const request = new XMLHttpRequest();
    var resp = null;
    request.onreadystatechange = function ()
    {
        if (request.readyState === 4 && request.status === 200) {
            resp = request.responseText;
        }
    };
    request.open("POST", "default.aspx", false);
    request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    request.send(data);
    return resp;
};
```

Asigno la
connectionString
en el Web.config

Pongo el nombre de El
Motor del servidor de
mi máquina



```
<connectionStrings>
  <add name="My Connection" connectionString="Data Source=DESKTOP-CAADJB2\SQLEXPRESS;Initial Catalog=DBWebEscuela;Integrated Security=true"/>
</connectionStrings>
<system.web>
```

Modifico en
formLogin el
submit

```
this.formLogin = function () {
    const Submit = function () {
    {
        let Data = "accion=LOGIN&Mail=" + mail.value + "&Password=" + pass.value;
        let res = Post(Data);
        try {
            MainUser = JSON.parse(res);
            $uf.formLogout();
        } catch (e) {
            alert(res);
        }
        return false;
    };
    let f = $df.form("ingresar", "form w40 l13 top7", "ingreso al sistema");
    let mail = $dlc.labelAndMail("mail");
    mail.required = true;
    let pass = $dlc.labelAndPassword();
    pass.placeholder = "ingrese contraseña";
    $dc.a(f, "", "OLVIDE MI CONTRASEÑA", "alogin").onclick = $uf.formRecoveryPassword;
    pass.required = true;
    f.onsubmit = function () {return Submit();};
    return f;
};
```

Modifico el formLogout

```
this.formLogout = function () {
    const Submit = function () {
        MainUser = undefined;
        Rol = undefined;
        Home();
        $n.init();
        return false;
    };

    const RolChanged = function () {
        Rol = $db.id("Roles").value;
        $n.init();
    };
    let f = $df.form("salir del sistema", "form w40 l15 top7", "usuario " + MainUser.Nombre);
    let select = $dlc.labelAndSelect("Roles", "Roles", MainUser.Roles, MainUser.Roles);
    $db.rc(Pie, $db.id("Reset"));
    $dc.img(Pie, "", MainUser.Url, "imglogout");
    $db.class("submit").style.margin = "2rem 0 0 1rem";

    //selecciono la opcion dados el rol
    let options = $db.cn(select);
    for (var i = 0; i < options.length; i++) {
        options[i].selected = options[i].value === Rol;
    }
    RolChanged();
    select.onchange = function () { RolChanged(); };
    f.onsubmit = function () { return Submit(); };
    return f;
};
```

Default.aspx.cs

AGREGO ESTE
USING

```
using System;
using BUSSINESCHOOL;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request["accion"] != null)
        {
            switch (Request["accion"])
            {
                case "LOGIN":Login();break;
            }
        }
    }

    private void Login()
    {
        Usuario Usuario = new Usuario();
        Usuario.Mail = Request["Mail"];
        Usuario.Password = Request["Password"];
        try
        {
            Usuario.Login();
            UsuarioRol UsuarioRol = new UsuarioRol();
            UsuarioRol.Usuario = Usuario;
            Response.Write(UsuarioRol.ListByUsuario());
        }
        catch(Exception er) { Response.Write(er.Message); }
    }
}
```

The image displays two side-by-side screenshots of a web application interface for 'INSTITUTO EDUCATIVO BALLESTER'.

Left Screenshot (Login Page):

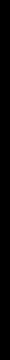
- Header:** INICIO, CARRERAS, NOTICIAS, INGRESAR AL SISTEMA.
- Form:** INGRESO AL SISTEMA with fields for MAIL (admin@mail.com) and CONTRASEÑA (*****), and buttons for INGRESAR and CANCELAR.
- Link:** OLVIDE MI CONTRASEÑA.
- Right Panel:** Features the school's logo (two lions flanking a shield), the text 'INSTITUTO EDUCATIVO BALLESTER', and 'EXCELENCIA EN LA EDUCACIÓN'. It also lists the address: AVDA MITRE 541 v.BALLESTER BS.AS. TEL 25433234 MAIL cballester@mail.com.ar.

Right Screenshot (User Profile Page):

- Header:** INICIO, PERFIL, FUNCIONES (highlighted), ALTA USUARIO, MOD. USUARIOS POR DNI, MOD. USUARIOS POR MAIL, ABM. CARRERAS, ADMIN.
- Form:** USUARIO ADMIN with a ROLES dropdown set to ADMINISTRADOR and a user icon.
- Link:** SALIR DEL SISTEMA.
- Right Panel:** Features the school's logo, the text 'INSTITUTO EDUCATIVO BALLESTER', and 'EXCELENCIA EN LA EDUCACIÓN'. It also lists the address: AVDA MITRE 541 v.BALLESTER BS.AS. TEL 25433234 MAIL cballester@mail.com.ar.

Verifico Login y logout

```
this.formProfile = function () {
    const Fload = function () {
        let content = this.contentWindow.document.childNodes[0].innerText;
        if (content === undefined || content === "") return;
        try {
            MainUser = JSON.parse(content);
            $uf.formLogout();
        } catch (e) {
            alert(content);
        }
    };
    let f = $df.form("modificar usuario", "form_w40_l24", "Perfil de " +
        MainUser.Nombre);
    f.action = "default.aspx";f.method = "POST"; f.enctype = "multipart/form-data";
    f.target = "ifr";var ifr = $db.ce("iframe");ifr.name = "ifr"; ifr.className = "hidden";
    $db.ac(Section, ifr);ifr.onload = Fload;
    $dc.inputHidden("accion", "MODIFICARUSUARIO");
    $dc.inputHidden("ID", MainUser.ID);
    let nombre = $dlc.labelAndText("Nombre", "Nombre");
    nombre.required = true;
    nombre.value = MainUser.Nombre;
    let direccion = $dlc.labelAndText("Direccion", "Direccion");
    direccion.required = true;
    direccion.value = MainUser.Direccion;
    let telefono = $dlc.labelAndText("Telefono", "Telefono");
    telefono.required = true;
    telefono.value = MainUser.Telefono;
    $dlc.labelAndFileImg("imagen perfil", "Img");
    $dlc.labelsAndPasswordConfirm();
    return f;
};
```



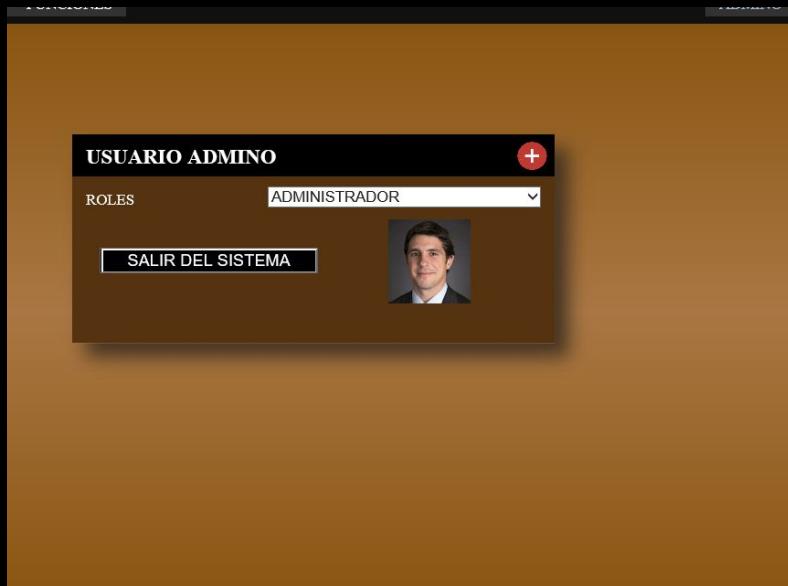
formProfile

Default.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request["accion"] != null)
    {
        switch (Request["accion"])
        {
            case "LOGIN":Login();break;
            case "MODIFICARUSUARIO":ModificarUsuario(); break
        }
    }
}
```

```
private void ModificarUsuario()
{
    Usuario Usuario = new Usuario();
    try
    {
        Usuario.ID = int.Parse(Request["ID"]);
        Usuario.Find();
        Usuario.Nombre = Request["Nombre"];
        Usuario.Direccion = Request["Direccion"];
        Usuario.Password = Request["Password"];
        if (Request.Files.Count > 0) { Usuario.Fu = Request.Files[0]; }
        Usuario.Modify();
        UsuarioRol UsuarioRol = new UsuarioRol();
        UsuarioRol.Usuario = Usuario;
        Response.Write(UsuarioRol.ListByUsuario());
    }
    catch(Exception er)
    {
        Response.Write(er.Message);
    }
}
```

Testing

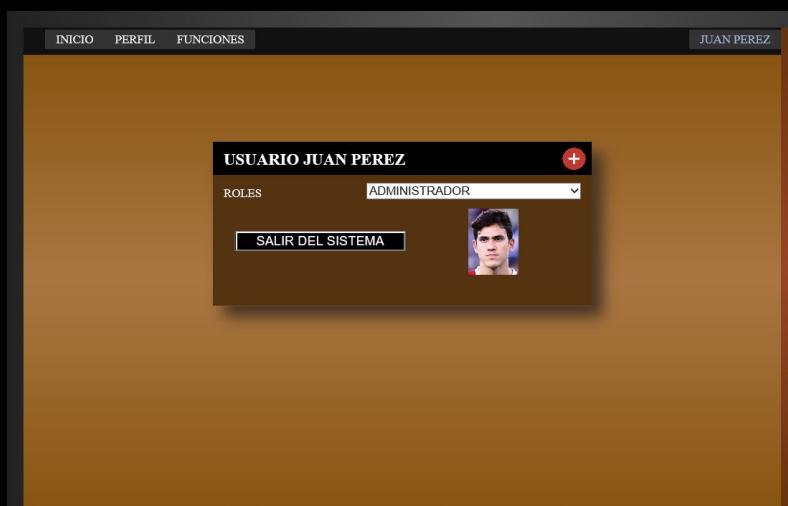


FUNCIONES

PERFIL DE ADMINO

NOMBRE	Juan Perez
DIRECCION	dir admin
TELEFONO	000000000000
IMAGEN PERFIL	C:\Users\jfdoc\OneDrive\ Examinar...
CONTRASEÑA	Si No Modifica No ingrese

MODIFICAR USUARIO CANCELAR



```
this.formAddUser = function ()  
{  
    const Submit = function ()[...];  
    let f = $df.form("agregar usuario", "form w40 l28", "Nuevo usuario");  
    let nombre = $dlc.labelAndText("Nombre");  
    nombre.required = true;  
    let dni = $dlc.labelAndNumber("Dni");  
    dni.required = true;  
    let direccion = $dlc.labelAndText("Direccion");  
    direccion.required = true;  
    let mail = $dlc.labelAndMail("Mail");  
    mail.required = true;  
    let fnac = $dlc.labelAndDate("F. Nacimiento");  
    fnac.required = true;  
    let telefono = $dlc.labelAndText("Teléfono");  
    telefono.required = true;  
    f.onsubmit = Submit;  
    $tu.listUsuarios();  
    return f;  
};
```

```
const Submit = function ()  
{  
    let User = undefined;  
    let Data = "accion=ADDUSER";  
    Data += "&Nombre=" + nombre.value;  
    Data += "&Direccion=" + direccion.value;  
    Data += "&Dni=" + dni.value;  
    Data += "&Password=" + dni.value;  
    Data += "&FechaNac=" + fnac.value;  
    Data += "&Telefono=" + telefono.value;  
    Data += "&Mail=" + mail.value;  
    let response = Post(Data);  
    try {  
        User = JSON.parse(response);  
        $uf.formModifyUser(User);  
    }  
    catch (e) { alert(response); }  
    return false;  
};
```

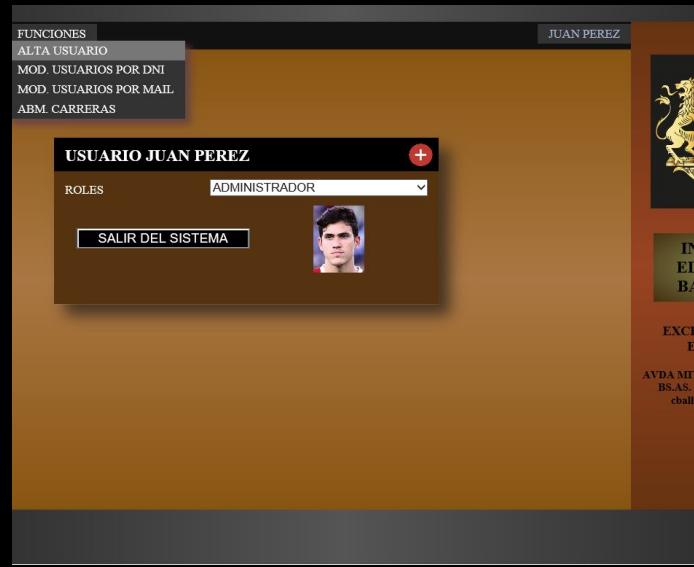
FormAddUser

```
private Usuario GetUsuario()
{
    Usuario U = new Usuario();
    if (Request["ID"] != null)
    {
        U.ID = int.Parse(Request["ID"]);
    }
    U.Nombre = Request["Nombre"];
    U.Dni=int.Parse(Request["Dni"]);
    U.Mail = Request["Mail"];
    U.FechaNac = DateTime.Parse(Request["FechaNac"]);
    if (Request["Password"] != null)
        U.Password = Request["Password"];
    else U.Password = "";
    U.Direccion = Request["Direccion"];
    U.Telefono = Request["Telefono"];
    return U;
}
```

```
private void AddUser()
{
    try
    {
        Usuario Usuario = GetUsuario();
        Usuario.Add();
        UsuarioRol UsuarioRol = new UsuarioRol();
        UsuarioRol.Usuario = Usuario;
        Response.Write(UsuarioRol.ListByUsuario());
    }
    catch(Exception er) { Response.Write(er.Message); }
}
```

```
case "LOGIN":Login();break;
case "MODIFICARUSUARIO":ModificarUsuario(); break;
case "ADDUSER":AddUser();break;
case "ERASEROL":EraseRol(); break;
case "ADDROL":AddRol(); break;
case "MODIFYUSERWITHROLES": ModifyUserWithRoles();break;
case "LISTUSERS":ListUsers();break;
```

Default.aspx.cs



SE INVOCÁ



SE INGRESA EL USUARIO



VA A MODIFICAR USUARIO
PARA AGREGAR ROLES O
MODIFICAR LOS DATOS

Visualizando agregar usuario (sin listado)

formModifyUser(user) (1)

```
this.formModifyUser = function (user)
{
    const MakeData = function (accion, rol){...};
    const Submit = function (){...};
    const CreateRoles = function (){...};
    const AddRol = function (){...};
    const EraseRol = function (){...};
    let f = $df.form("modificar usuario", "form w40 l33", "modificar " + user.Nombre);
    let nombre = $dlc.labelAndText("Nombre");
    nombre.required = true;
    nombre.value = user.Nombre;
    let dni = $dlc.labelAndNumber("Dni");
    dni.required = true;
    dni.value = user.Dni;
    let direccion = $dlc.labelAndText("Direccion");
    direccion.required = true;
    direccion.value = user.Direccion;
    let mail = $dlc.labelAndMail("Mail");
    mail.required = true;
    mail.value = user.Mail;
    let fnac = $dlc.labelAndDate("F. Nacimiento");
    fnac.value = user.FechaNac;
    fnac.required = true;
    let telefono = $dlc.labelAndText("Telefono");
    telefono.required = true;
    telefono.value = user.Telefono;
    let selectaux = $dlc.labelAndSelect("Rol a agregar", "", [], []);
    selectaux.className = "auxroles";
    let select = $dlc.labelAndSelect("Roles", "", user.Roles, user.Roles);
    $dlc.ButtonAndButton("Aregar Rol",
        "Quitar Rol",
        "class1",
        "class2",
        AddRol,
        EraseRol
    );
    f.onsubmit = Submit;
    CreateRoles();
    $tu.listUsuarios();
    return f;
};
```

formModifyUser(user)(2)

```
const MakeData = function (accion, rol)
{
    let Data = "accion=" + accion;
    Data += "&ID=" + user.ID;
    Data += "&Nombre=" + nombre.value;
    Data += "&Direccion=" + direccion.value;
    Data += "&Dni=" + dni.value;
    Data += "&FechaNac=" + fnac.value;
    Data += "&Telefono=" + telefono.value;
    Data += "&Mail=" + mail.value;
    Data += "&Rol=" + rol;
    return Post(Data);
};
```

```
const Submit = function ()
{
    if (user.Roles.length === 0) {
        alert("SE DEBE INGRESAR POR LO MENOS UN ROL");
        return false;
    }
    let res = MakeData("MODIFYUSERWITHROLES", "");
    if (res === "OK") {

        $uf.formAddUser();
        return false;
    }
    alert(res);
    return false;
};
```

```
const CreateRoles = function ()
{
    const ExistsRolesInUser = function (roles, rol)
    {
        for (var i = 0; i < roles.length; i++) {
            if (roles[i] === rol) return true;
        }
        return false;
    };
    const AddOption = function (parent, value)
    {
        let opt = $db.ce("option");
        opt.value = value;
        opt.innerText = value;
        $db.ac(parent, opt);
    };
    let roles = user.Roles;
    selectaux.innerHTML = "";
    for (var i = 0; i < ROLES.length; i++) {
        let rol = ROLES[i];
        if (!ExistsRolesInUser(roles, rol))
            AddOption(selectaux, rol);
    }
};
```

```
const AddRol = function ()
{
    let response = MakeData("ADDROL", selectaux.value);
    try {

        User = JSON.parse(response);
        $uf.formModifyUser(User);
    }
    catch (e) { alert(response); }

};
const EraseRol = function ()
{
    let response = MakeData("ERASEROL", select.value);
    try {
        User = JSON.parse(response);
        $uf.formModifyUser(User);
    }
    catch (e) { alert(response); }
};
```

Default.aspx.cs

```
private void ModifyUserWithRoles()
{
    Usuario Usuario = GetUSuario();
    try { Usuario.Modify();
        Response.Write("OK");
    }
    catch(Exception er) { Response.Write(er.Message);
}

private void AddRol()
{
    Usuario Usuario = GetUSuario();
    try
    {
        Usuario.Modify();
        UsuarioRol UsuarioRol = new UsuarioRol();
        UsuarioRol.Usuario = Usuario;
        UsuarioRol.Rol = Request["Rol"];
        UsuarioRol.Add();
        Response.Write(UsuarioRol.ListByUsuario());
    }
    catch(Exception er) { Response.Write(er.Message); }

private void EraseRol()
{
    try
    {
        Usuario Usuario = GetUSuario();
        UsuarioRol UsuarioRol = new UsuarioRol();
        UsuarioRol.Usuario = Usuario;
        UsuarioRol.Rol = Request["Rol"];
        if(UsuarioRol.Rol=="ADMINISTRADOR")
            UsuarioRol.Erase();
        Response.Write(UsuarioRol.ListByUsuario());
    }
    catch (Exception er) { Response.Write(er.Message); }
}

private void ListUsers()
{
    Response.Write(new Usuario().List());
}
```

listUsuarios() (1)

```
const $$TableUsuarios = function ()  
{  
    this.listUsuarios = function ()  
    {  
        let Usuarios = JSON.parse(Post("accion=LISTUSERS"));  
        let Header = ["Nombre", "Dni", "Direccion", "F. nacimiento", "mail", "Teléfono", "Imagen", "Modificar", "Eliminar"];  
        const Frow = function (tr, Usuario)...;  
        $dt.makeTable(Header, 3, Usuarios, Frow);  
    };  
};  
const $tu = new $$TableUsuarios();
```

```
const Td = function (parent)  
{  
    let td = $db.ce("td");  
    $db.ac(parent, td);  
    return td;  
}; //obtengo un objeto td  
// TABLAS DE CARRERAS  
const $$TablesCareers = function ()...;  
$tc = new $$TablesCareers(); //INSTANCIA  
//TABLAS DE USUARIOS  
const $$TableUsuarios = function ()...;  
const $tu = new $$TableUsuarios();
```

listUsuarios() (2)

```
const Frow = function (tr, Usuario)
{
    Td(tr).innerText = Usuario.Nombre;
    Td(tr).innerText = Usuario.Dni;
    Td(tr).innerText = Usuario.Direccion;
    Td(tr).innerText = Usuario.FechaNac;
    Td(tr).innerText = Usuario.Mail;
    Td(tr).innerText = Usuario.Telefono;
    let tdimg = Td(tr);
    $dc.img(tdimg, "", Usuario.Url, "icon");
    let td = Td(tr);
    td.style.backgroundColor = "black";
    td.style.color = "#FFF";
    td.innerHTML = "Modificar<br/>Usuario";
    td.onmouseover = function () { td.style.opacity = 0.5; };
    td.onmouseleave = function () { td.style.opacity = 1; };
    td.onclick = function ()
    {
        let resp = Post("accion=OBTENERUSERROL&ID=" + Usuario.ID);
        try {
            Usuario = JSON.parse(resp);
            $uf.formModifyUser(Usuario);
        } catch (e) {
            alert(resp);
        }
    };
    let tdi = Td(tr);
    tdi.style.backgroundColor = "#a00";
    tdi.onmouseover = function () { tdi.style.opacity = 0.5; };
    tdi.onmouseleave = function () { tdi.style.opacity = 1; };
    tdi.style.color = "#FFF";
    tdi.innerHTML = "Eliminar<br/>Usuario";
    tdi.onclick = function ()
    {
        if (MainUser.ID === Usuario.ID) return;
        let Resp = Post("accion=ELIMINARUSUARIO&ID=" + Usuario.ID);

        if (Resp === "OK") $tu.listUsuarios();
        else alert(Resp);
    };
};
```

```
case "LOGIN":Login();break;
case "MODIFICARUSUARIO":ModificarUsuario(); break;
case "ADDUSER":AddUser();break;
case "ERASEROL":EraseRol(); break;
case "ADDROL":AddRol(); break;
case "MODIFYUSERWITHROLES": ModifyUserWithRoles();break;
case "LISTUSERS":ListUsers();break;
case "ELIMINARUSUARIO":EliminarUsuario();break;
case "OBTENERUSERROL":ObtenerUserRol();break;
```

```
private void ObtenerUserRol()
{
    try {
        UsuarioRol Ur = new UsuarioRol();
        Ur.Usuario = new Usuario();
        Ur.Usuario.ID = int.Parse(Request["ID"]);
        Response.Write(Ur.ListByUsuario());
    }catch(Exception er) { Response.Write(er.Message); }
}
```

```
private void EliminarUsuario()
{
    Usuario Usuario = new Usuario();
    Usuario.ID = int.Parse(Request["ID"]);
    try
    {
        Usuario.Erase();
        Response.Write("OK");
    }
    catch (Exception er)
    {
        Response.Write(er.Message);
    }
}
```

Default.aspx.cs

INICIO PERFIL FUNCIONES

ALTA USUARIO
MOD. USUARIOS POR DNI
MOD. USUARIOS POR MAIL
ADM. CARRERAS

USUARIO ADMIN

ROLES ADMINISTRADOR

SALIR DEL SISTEMA



INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN
AVDA MITRE 541 v. BALLESTER
BS.AS. TEL 25433234 MAIL cballester@mail.com.ar

Acción de listados

INICIO PERFIL FUNCIONES

ADMIN

NUEVO USUARIO

NOMBRE
DNI
DIRECCION
MAIL
F. NACIMIENTO
TELÉFONO

aaaa-mm-dd

AGREGAR USUARIO CANCELAR

NOMBRE	DNI	DIRECCION	F. NACIMIENTO	MAIL	TELÉFONO	IMAGEN	MODIFICAR	ELIMINAR
admin	1	Pepino fuentes 2133 CABA	1990-11-11	admin@mail.com	000000		Modificar Usuario	Eliminar Usuario
Mary	2	director Mry 1231	1999-2-2	mary@mail.com	3333334		Modificar Usuario	Eliminar Usuario

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN
AVDA MITRE 541 v. BALLESTER
BS.AS. TEL 25433234 MAIL cballester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

MODIFICAR MARY

NOMBRE Mary
DNI 2
DIRECCION director Mry 1231
MAIL mary@mail.com
F. NACIMIENTO 1999-2-2
TELÉFONO 3333334
ROL A AGREGAR ADMINISTRADOR
ROLES PRECEPTOR

AGREGAR ROL QUITAR ROL
MODIFICAR USUARIO CANCELAR

NOMBRE	DNI	DIRECCION	F. NACIMIENTO	MAIL	TELÉFONO	IMAGEN	MODIFICAR	ELIMINAR
admin	1	Pepino fuentes 2133 CABA	1990-11-11	admin@mail.com	000000		Modificar Usuario	Eliminar Usuario
Mary	2	director Mry 1231	1999-2-2	mary@mail.com	3333334		Modificar Usuario	Eliminar Usuario

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN
AVDA MITRE 541 v. BALLESTER
BS.AS. TEL 25433234 MAIL cballester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

MODIFICAR MARY

NOMBRE Mary
DNI 2
DIRECCION director Mry 1231
MAIL mary@mail.com
F. NACIMIENTO 1999-2-2
TELÉFONO 3333334
ROL A AGREGAR ADMINISTRADOR
ROLES PRECEPTOR

AGREGAR ROL QUITAR ROL
MODIFICAR USUARIO CANCELAR

NOMBRE	DNI	DIRECCION	F. NACIMIENTO	MAIL	TELÉFONO	IMAGEN	MODIFICAR	ELIMINAR
admin	1	Pepino fuentes 2133 CABA	1990-11-11	admin@mail.com	000000		Modificar Usuario	Eliminar Usuario

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN
AVDA MITRE 541 v. BALLESTER
BS.AS. TEL 25433234 MAIL cballester@mail.com.ar

```
this.formFindByDni = function ()  
{  
    const Submit = function ()  
    {  
        let Res = Post("accion=FINDUSERROLBYDNI&Dni=" + dni.value);  
        try {  
            let User = JSON.parse(Res);  
            $uf.formModifyUser(User);  
        } catch (e) {  
            alert(Res);  
        }  
    };  
  
    let f = $df.form("verificar", "form w40 l10 top7", "encontrar usuario");  
    let dni = $dlc.labelAndNumber("ingrese dni");  
    dni.required = true;  
  
    f.onsubmit = function () { return Submit(); };  
};
```

```
this.formFindByMail = function ()  
{  
    const Submit = function ()  
    {  
        let Res = Post("accion=FINDUSERROLBYMAIL&Mail=" + mail.value);  
        try {  
            let User = JSON.parse(Res);  
            $uf.formModifyUser(User);  
        } catch (e) {  
            alert(Res);  
        }  
        return false;  
    };  
  
    let f = $df.form("verificar", "form w40 l10 top7", "encontrar usuario");  
    let mail = $dlc.labelAndMail("ingrese mail");  
    mail.required = true;  
  
    f.onsubmit = function () { return Submit(); };  
};
```

FindByDni() FindByMail()

Default.aspx.cs

```
case "OBTENERUSERROL":ObtenerUserRol();break;
case "FINDUSERROLBYMAIL":FindUserRolByMail();break;
case "FINDUSERROLBYDNI":FindUserRolByDni();break;

private void FindUserRolByDni()
{
    UsuarioRol Ur = new UsuarioRol();
    Ur.Usuario = new Usuario();
    Ur.Usuario.Dni = int.Parse(Request["Dni"]);
    try
    {
        Ur.Usuario.FindByDni();
        Response.Write(Ur.ListByUsuario());
    }
    catch (Exception er)
    {
        Response.Write(er.Message);
    }
}

private void FindUserRolByMail()
{
    UsuarioRol Ur = new UsuarioRol();
    Ur.Usuario = new Usuario();
    Ur.Usuario.Mail= Request["Mail"];
    try
    {
        Ur.Usuario.FindByMail();
        Response.Write(Ur.ListByUsuario());
    }
    catch (Exception er)
    {
        Response.Write(er.Message);
    }
}
```

INICIO PERFIL FUNCIONES

ALTA USUARIO MOD. USUARIOS POR DNI MOD. USUARIOS POR MAIL

ADM. CARRERAS

USUARIO ADMIN

ROLES ADMINISTRADOR

[SALIR DEL SISTEMA](#)



INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

ENCONTRAR USUARIO

INGRESE DNI [X](#)

[VERIFICAR](#) [CANCELAR](#)



INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

MODIFICAR ADMIN

NOMBRE

DNI

DIRECCION

MAIL

F. NACIMIENTO

TELÉFONO

ROL A AGREGAR PROFESOR

ROLES ADMINISTRADOR

[AGREGAR ROL](#) [QUITAR ROL](#)

[MODIFICAR USUARIO](#) [CANCELAR](#)

NOMBRE	DNI	DIRECCION	F. NACIMIENTO	MAIL	TELÉFONO	IMAGEN	MODIFICAR	ELIMINAR
admin	1	Pepino fuentes 2133 CABA	1990-11-11	admin@mail.com	000000		Modificar Usuario	Eliminar Usuario

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

INICIO PERFIL FUNCIONES

ALTA USUARIO MOD. USUARIOS POR DNI MOD. USUARIOS POR MAIL

ADM. CARRERAS

USUARIO ADMIN

ROLES ADMINISTRADOR

[SALIR DEL SISTEMA](#)



INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

ENCONTRAR USUARIO

INGRESE MAIL [X](#)

[VERIFICAR](#) [CANCELAR](#)



INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

INICIO PERFIL FUNCIONES

ADMIN

MODIFICAR ADMIN

NOMBRE

DNI

DIRECCION

MAIL

F. NACIMIENTO

TELÉFONO

ROL A AGREGAR PROFESOR

ROLES ADMINISTRADOR

[AGREGAR ROL](#) [QUITAR ROL](#)

[MODIFICAR USUARIO](#) [CANCELAR](#)

NOMBRE	DNI	DIRECCION	F. NACIMIENTO	MAIL	TELÉFONO	IMAGEN	MODIFICAR	ELIMINAR
admin	1	Pepino fuentes 2133 CABA	1990-11-11	admin@mail.com	000000		Modificar Usuario	Eliminar Usuario

INSTITUTO EDUCATIVO BALLESTER

EXCELENCIA EN LA EDUCACIÓN

AVDA MITRE 541 v BALLESTER
BS AS. TEL. 2543524 MAIL:
challester@mail.com.ar

formRecoveryPassword()

```
this.formRecoveryPassword = function ()
{
    let user = new Object();
    const Submit = function ()
    {
        let res = Post("accion=RECOVERYPASSWORD&Mail=" + mail.value + "&Dni=" + dni.value);
        if (res === "OK")
            alert("SU CONTRASEÑA HA SIDO MODIFICADA " +
                  "\nDEBERÁ INGRESAR AL SISTEMA CON SU MAIL Y COMO CONTRASEÑA SU DNI" +
                  "\nLUEGO PADRE MODIFICAR SU CONTRASEÑA EN SU PERFIL");
        else
            alert(res);
        Home();
        return false;
    };

    let f = $df.form("verificar", "form w40 l13 top7", "recuperación de contraseña");
    let mail = $dlc.labelAndMail("ingrese su mail");
    mail.required = true;
    let dni = $dlc.labelAndNumber("ingrese su dni");
    dni.required = true;
    f.onsubmit = function () { Submit(); return false; };
    return f;
};
```

Default.aspx.cs

```
case "RECOVERYPASSWORD":RecoveryPassword(); break;
```

```
private void RecoveryPassword()
{
    Usuario U = new Usuario();
    U.Mail = Request["Mail"];
    U.Dni = int.Parse(Request["Dni"]);
    try
    {
        U.FindByMailAndDni();
        U.Password = U.Dni.ToString();
        U.Modify();
        Response.Write("OK");
    }
    catch(Exception er) { Response.Write(er.Message); }
}
```

Hemos terminado Usuarios en la siguiente presentación terminaremos la parte de Carreras

[Ver video final](#)

