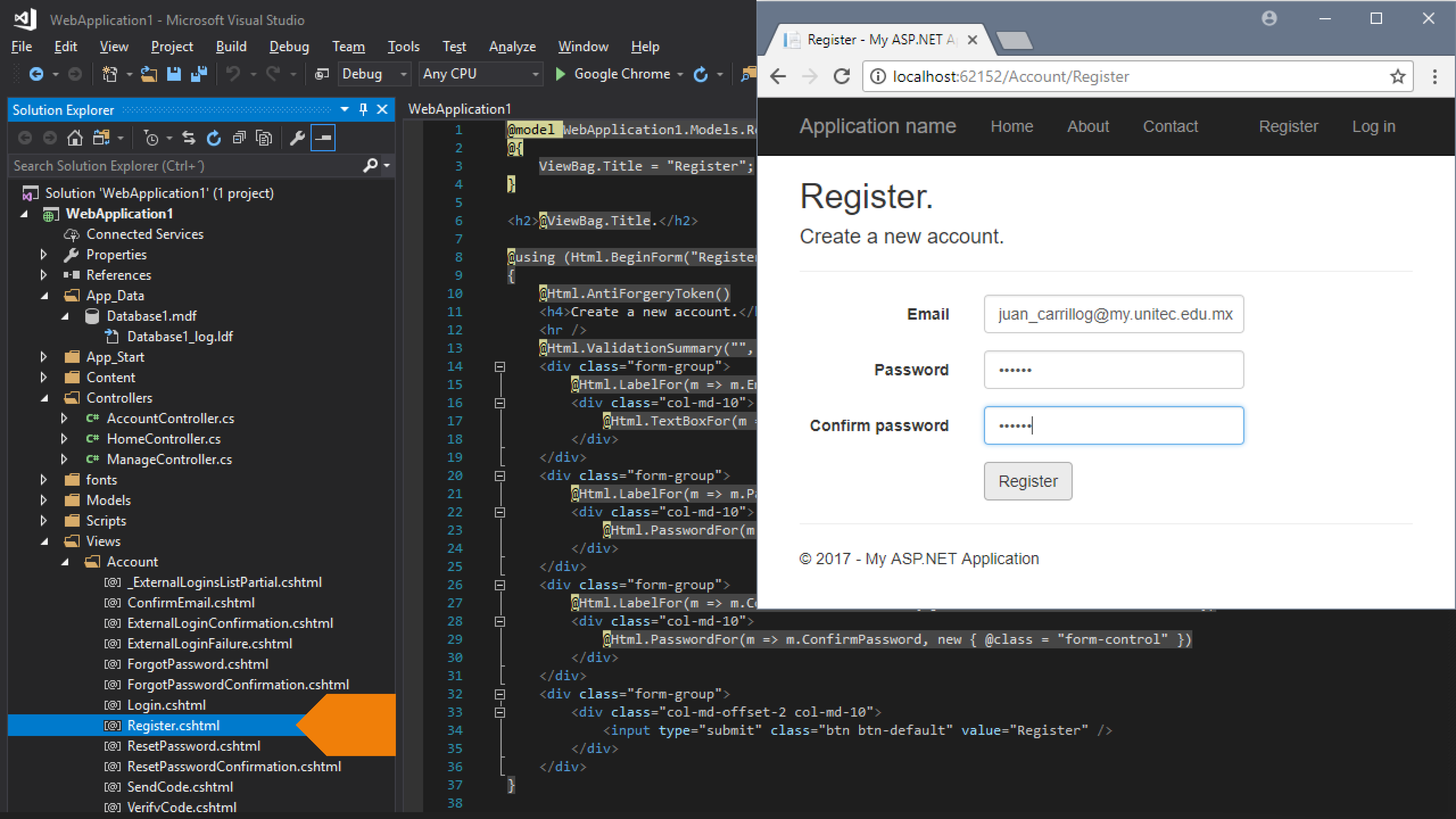
Autenticación

ASP.NET

Juan Luis Carrillo Garcia | juan_carrillog@my.unitec.edu.mx | @ jyacomp | linkedin.com/in/jlcarrillog/



WebApplication1 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Google Chrome
Solution Explorer
Search Solution Explorer (Ctrl+ ;)
WebApplication1

```
Solution 'WebApplication1' (1 project)
  WebApplication1
    Connected Services
    Properties
    References
    App_Data
      Database1.mdf
        Database1_log.ldf
    App_Start
    Content
    Controllers
      AccountController.cs
      HomeController.cs
      ManageController.cs
    fonts
    Models
    Scripts
    Views
      Account
        _ExternalLoginsListPartial.cshtml
        ConfirmEmail.cshtml
        ExternalLoginConfirmation.cshtml
        ExternalLoginFailure.cshtml
        ForgotPassword.cshtml
        ForgotPasswordConfirmation.cshtml
        Login.cshtml
        Register.cshtml
        ResetPassword.cshtml
        ResetPasswordConfirmation.cshtml
        SendCode.cshtml
        VerifyCode.cshtml
```

```
@model WebApplication1.Models.R
@{
    ViewBag.Title = "Register";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("Registe
{
    @Html.AntiForgeryToken()
    <h4>Create a new account.</
    <hr />
    @Html.ValidationSummary("",
    <div class="form-group">
        @Html.LabelFor(m => m.E
        <div class="col-md-10">
            @Html.TextBoxFor(m =
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.P
        <div class="col-md-10">
            @Html.PasswordFor(m
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.C
        <div class="col-md-10">
            @Html.PasswordFor(m => m.ConfirmPassword, new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" class="btn btn-default" value="Register" />
        </div>
    </div>
}
```

Register.

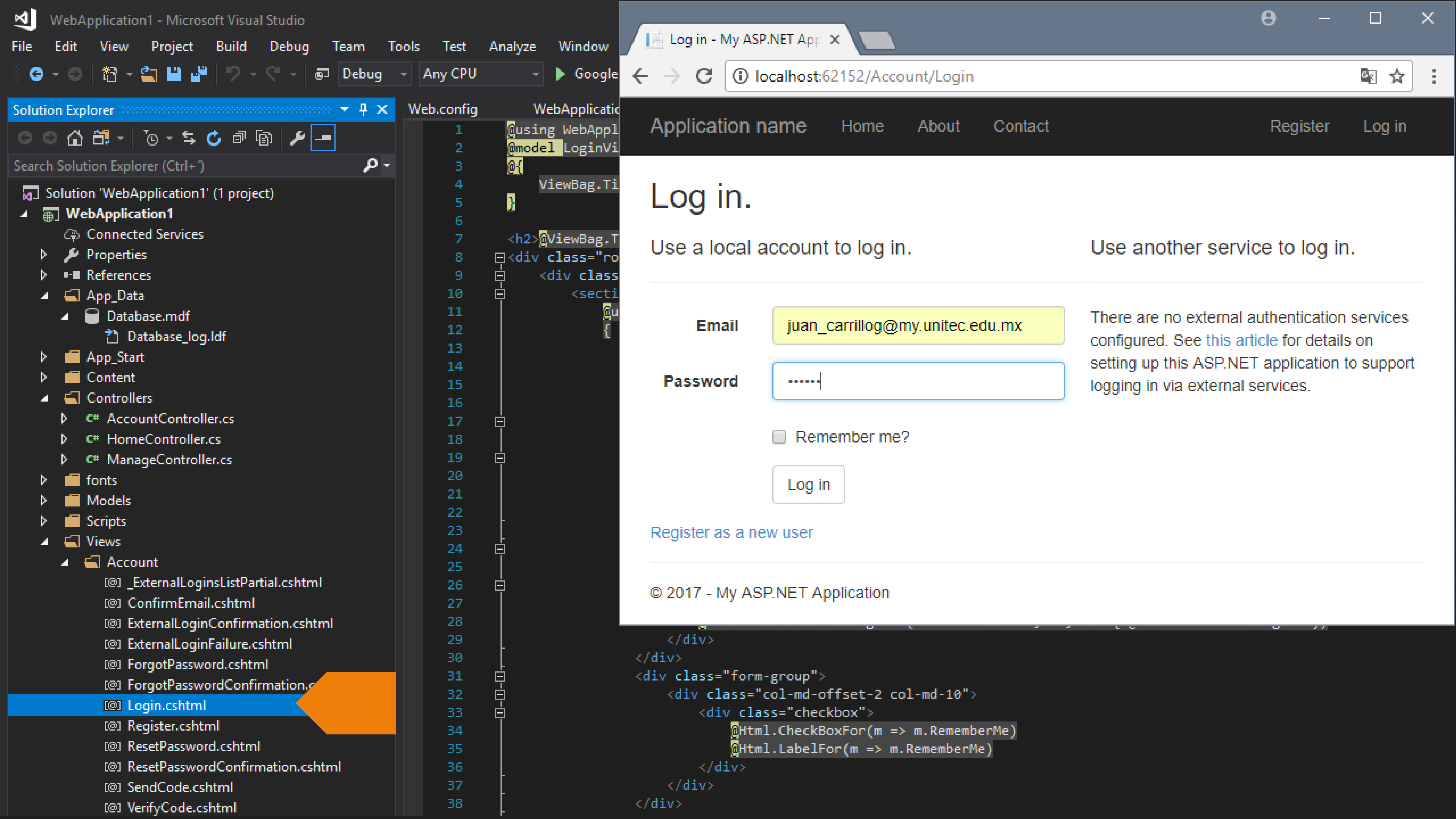
Create a new account.

Email juan_carrillog@my.unitec.edu.mx

Password ••••••

Confirm password ••••••

© 2017 - My ASP.NET Application



```
@using WebAppl
@model LoginVi
@{
    ViewBag.Ti
}

<h2>@ViewBag.T
<div class="ro
    <div class
        <secti
            @u
            {
```

Log in.

Use a local account to log in.

Email: juan_carrillog@my.unitec.edu.mx

Password: ••••••

☐ Remember me?

Log in

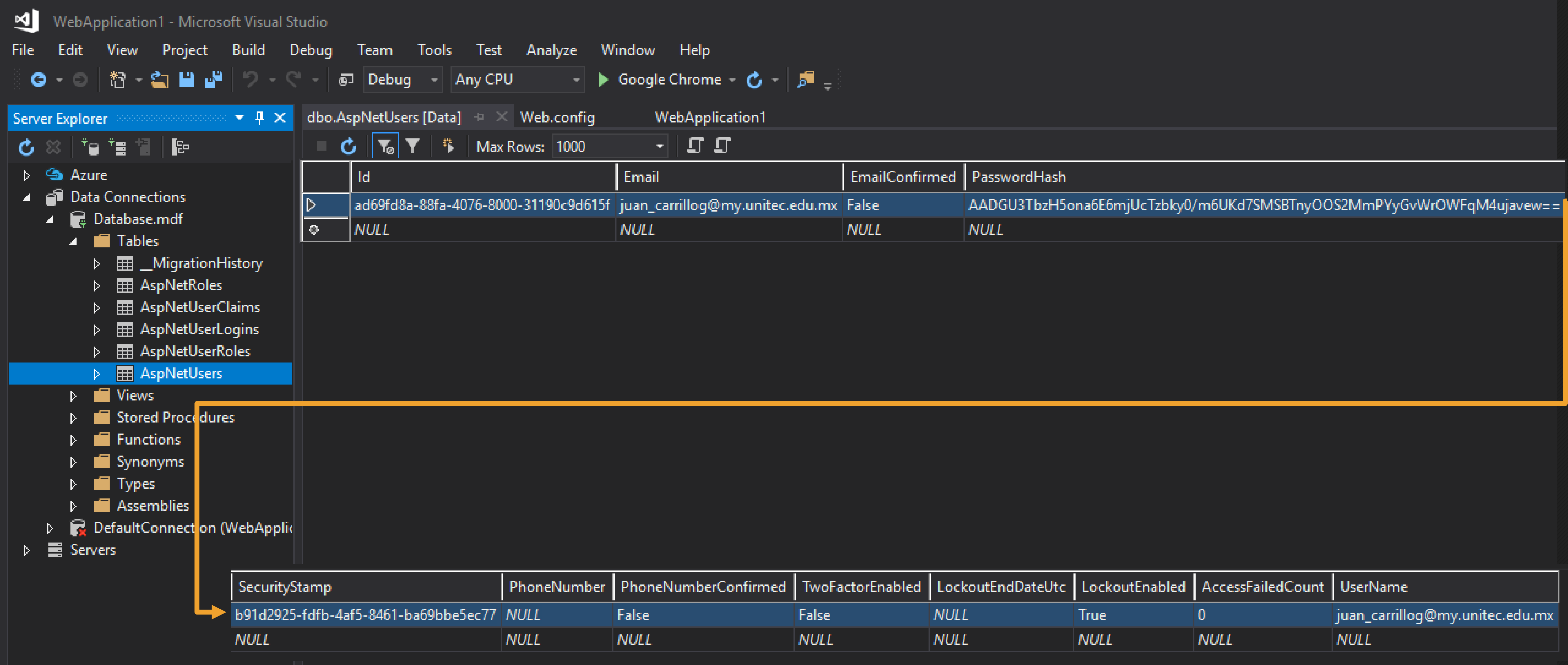
Register as a new user

Use another service to log in.

There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

© 2017 - My ASP.NET Application

```
                </div>
            </div>
            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <div class="checkbox">
                        @Html.CheckBoxFor(m => m.RememberMe)
                        @Html.LabelFor(m => m.RememberMe)
                    </div>
                </div>
            </div>
```

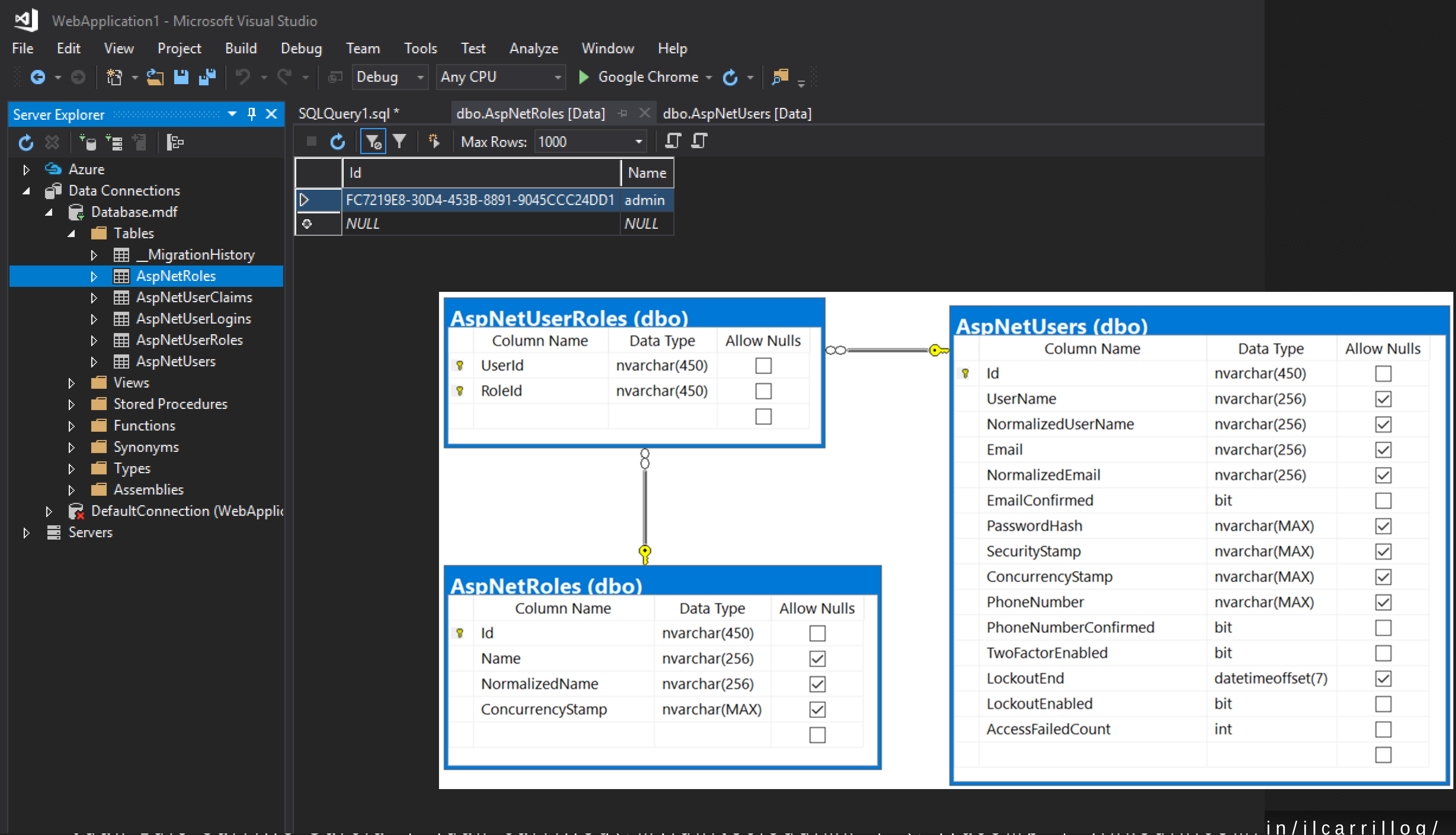


Max Rows: 1000

Id	Email	EmailConfirmed	PasswordHash
ad69fd8a-88fa-4076-8000-31190c9d615f	juan_carrillog@my.unitec.edu.mx	False	AADGU3TbzH5ona6E6mjUcTzbky0/m6UKd7SMSBTnyOOS2MmPYyGvWrOWFqM4ujavew==
NULL	NULL	NULL	NULL

SecurityStamp	PhoneNumber	PhoneNumberConfirmed	TwoFactorEnabled	LockoutEndDateUtc	LockoutEnabled	AccessFailedCount	UserName
b91d2925-fdfb-4af5-8461-ba69bbe5ec77	NULL	False	False	NULL	True	0	juan_carrillog@my.unitec.edu.mx
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

En criptografía, MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.



Id	Name
FC7219E8-30D4-453B-8891-9045CCC24DD1	admin
NULL	*NULL*

AspNetUserRoles (dbo)

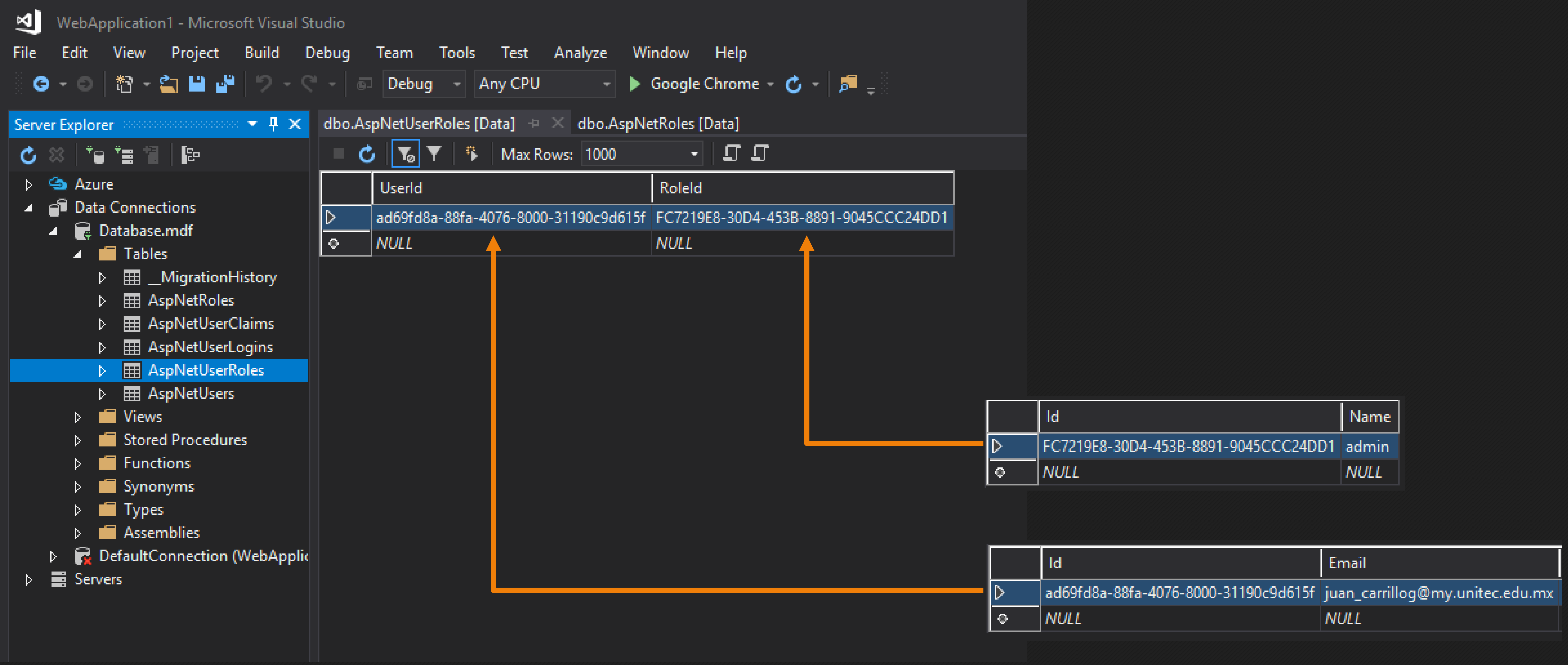
Column Name	Data Type	Allow Nulls
UserId	nvarchar(450)	□
RoleId	nvarchar(450)	□
		□

AspNetRoles (dbo)

Column Name	Data Type	Allow Nulls
Id	nvarchar(450)	□
Name	nvarchar(256)	☑
NormalizedName	nvarchar(256)	☑
ConcurrencyStamp	nvarchar(MAX)	☑
		□

AspNetUsers (dbo)

Column Name	Data Type	Allow Nulls
Id	nvarchar(450)	□
UserName	nvarchar(256)	☑
NormalizedUserName	nvarchar(256)	☑
Email	nvarchar(256)	☑
NormalizedEmail	nvarchar(256)	☑
EmailConfirmed	bit	□
PasswordHash	nvarchar(MAX)	☑
SecurityStamp	nvarchar(MAX)	☑
ConcurrencyStamp	nvarchar(MAX)	☑
PhoneNumber	nvarchar(MAX)	☑
PhoneNumberConfirmed	bit	□
TwoFactorEnabled	bit	□
LockoutEnd	datetimeoffset(7)	☑
LockoutEnabled	bit	□
AccessFailedCount	int	□
		□



UserId	RoleId
ad69fd8a-88fa-4076-8000-31190c9d615f	FC7219E8-30D4-453B-8891-9045CCC24DD1
NULL	*NULL*

Id	Name
FC7219E8-30D4-453B-8891-9045CCC24DD1	admin
NULL	*NULL*

Id	Email
ad69fd8a-88fa-4076-8000-31190c9d615f	juan_carrillog@my.unitec.edu.mx
NULL	*NULL*

Autorización

ASP.NET

Autorización en ASP.NET a nivel de controlador

```
[Authorize]
public class HomeController : Controller
{

    public ActionResult Index()
    {
        return View();
    }

    public ActionResult About()
    {
        return View();
    }

    public ActionResult Contact()
    {
        return View();
    }
}
```

Solo usuarios autenticados

Autorización en ASP.NET a nivel de acción

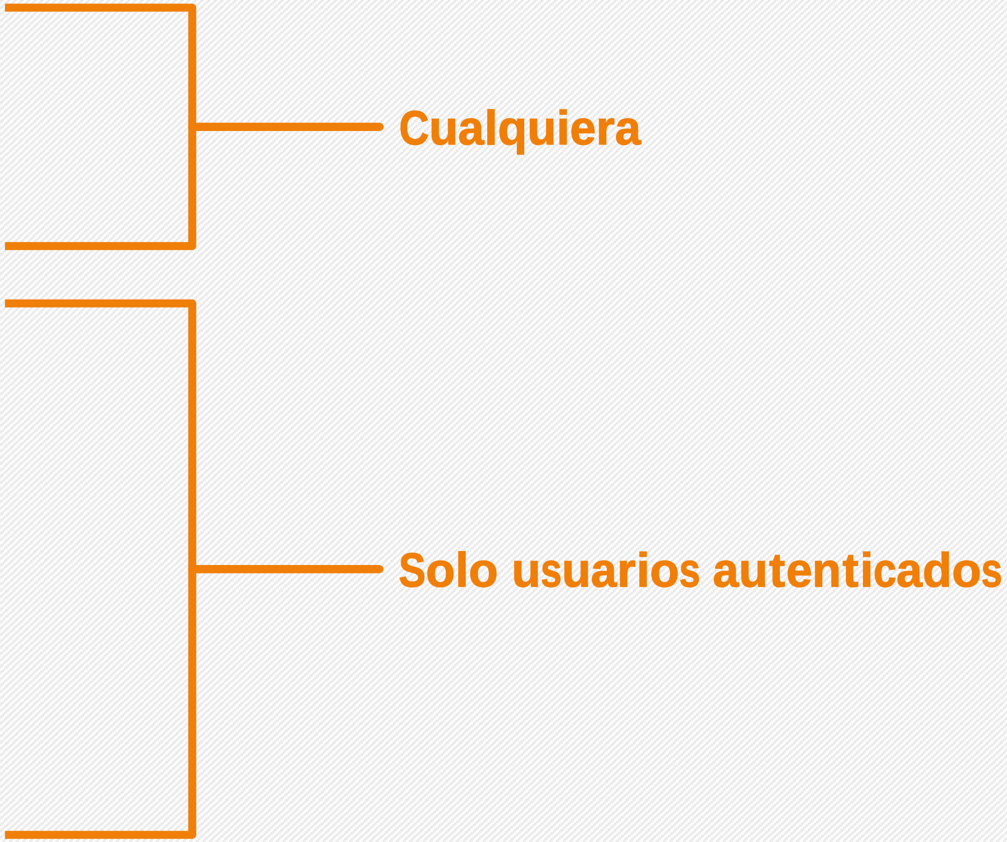
```
public class HomeController : Controller
{

    public ActionResult Index()
    {
        return View();
    }
    [Authorize]
    public ActionResult About()
    {
        return View();
    }
    [Authorize]
    public ActionResult Contact()
    {
        return View();
    }
}
```

Cualquiera

Solo usuarios autenticados

Autorización en ASP.NET sobrescribiendo reglas

```
[Authorize]
public class HomeController : Controller
{
   [AllowAnonymous]
   public ActionResult Index()
   {
      return View();
   }

   public ActionResult About()
   {
      return View();
   }

   public ActionResult Contact()
   {
      return View();
   }
}
```



Cualquiera

Solo usuarios autenticados

Autorización en ASP.NET definiendo usuarios y roles

```
public class HomeController : Controller
{
   [Authorize(Users = "luis@mail.com,pedro@mail.com")]
   public ActionResult Index()
   {
      return View();
   }
   [Authorize(Roles = "Admin")]
   public ActionResult About()
   {
      return View();
   }
   [Authorize(Roles = "Editor,Admin")]
   public ActionResult Contact()
   {
      return View();
   }
}
```

Solo usuarios:

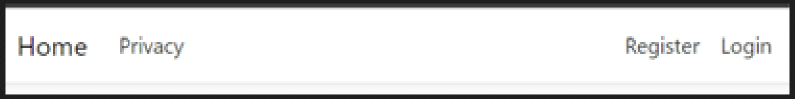
- luis@mail.com
- pedro@mail.com

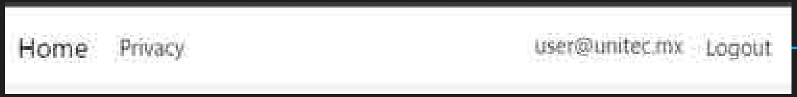
Solo usuarios con el rol:

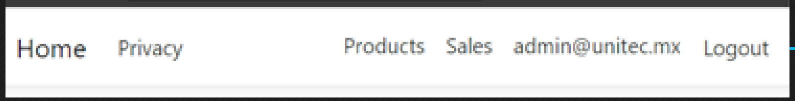
- admin

Solo usuarios los roles:

- editor
- admin







```
@* .\Views\Shared\_LoginPartial.cshtml *@

@using Microsoft.AspNetCore.Identity
@inject SignInManager<IdentityUser> SignInManager
@inject UserManager<IdentityUser> UserManager


<ul class="navbar-nav">
    @if (SignInManager.IsSignedIn(User))
    {
        if (User.IsInRole("Admin"))
        {
            <li class="nav-item">
                <a class="nav-link text-dark" asp-area="" asp-controller="Products" asp-action="Index">Products</a>
            </li>
            <li class="nav-item">
                <a class="nav-link text-dark" asp-area="" asp-controller="Sales" asp-action="Index">Sales</a>
            </li>
        }
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage">@User.Identity.Name</a>
        </li>
        <li class="nav-item">
            <form class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">
                <button type="submit" class="nav-link btn btn-link text-dark">Logout</button>
            </form>
        </li>
    }
    else
    {
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Register">Register</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Login">Login</a>
        </li>
    }
</ul>
```

Usuario	Rol	Contraseña
admin@unitec.mx	Admin	Admin123
user@unitec.mx		User123

Crear usuarios en la aplicación:

Ejercicio

Implementar la autorización a los controladores y las vistas necesarias:

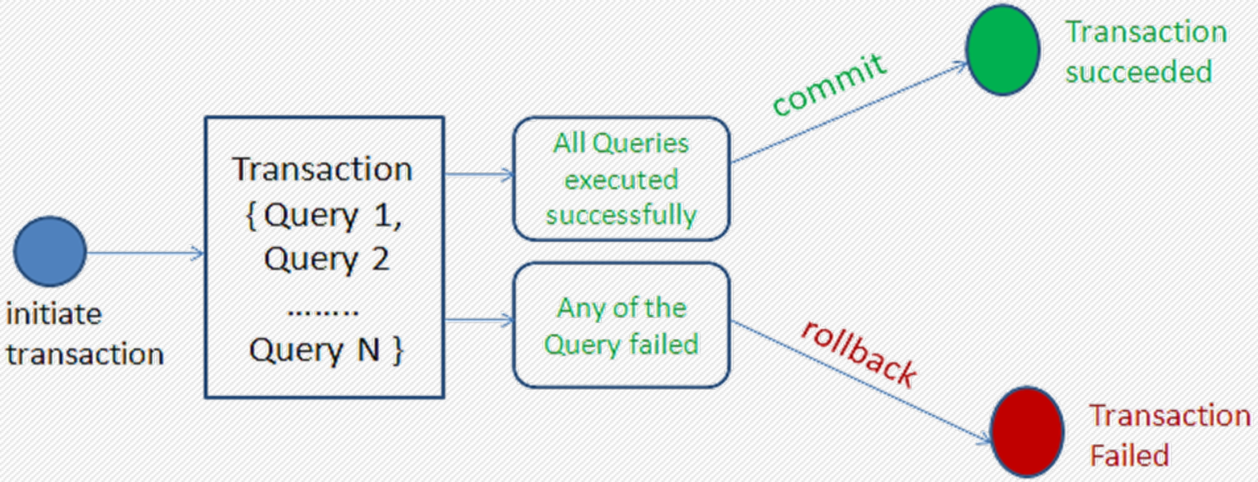
```
[Authorize]
[Authorize(Roles = "Admin")]

if (User.IsInRole("Admin"))
{

}
```

Transaccion

ACID

Conjunto de propiedades que garantizan que las transacciones de una Base de Datos se procesan de manera fiable:

- **Atomicidad:** un cambio debe **completarse en su totalidad**, es decir, si consiste en una serie de pasos, o se ejecutan todos o ninguno.
- **Consistencia (Integridad):** asegura que sólo se empieza aquello que se puede acabar, es decir, cualquier cambio debe conducir de un estado válido a otro estado válido.
- **Aislamiento (Isolation):** un cambio no debe afectar a otros cambios que se estén ejecutando al mismo tiempo, es decir, dos transacciones sobre la misma información deben ser independientes y no generen ningún tipo de error.
- **Durabilidad (Persistencia):** una vez completado el cambio, éste debe conservarse, aunque se produzcan fallos en la base de datos o el sistema completo.



Models>UserProducts.cs

```
/*
CREATE TABLE [UserProducts](
    [Id] [uniqueidentifier] PRIMARY KEY NOT NULL,

    [UserId] [nvarchar](450) NOT NULL,
    [UserName] [nvarchar](256) NULL,

    [ProductId] [uniqueidentifier] NOT NULL,
    [Name] [nvarchar](128) NOT NULL,

    [Date] [smalldatetime] NOT NULL,

    FOREIGN KEY([ProductId]) REFERENCES [Products] ([Id]),
    FOREIGN KEY([UserId]) REFERENCES [AspNetUsers] ([Id])
);
*/



using System;

public class UserProducts
{
    public Guid Id { get; set; }

    public string UserId { get; set; }
    public string UserName { get; set; }

    public Guid ProductId { get; set; }
    public string Name { get; set; }

    public DateTime Date { get; set; }
}
```

```
CREATE TABLE [AspNetUsers](
      [Id] [nvarchar](450) PRIMARY KEY NOT NULL,
      [UserName] [nvarchar](256) NULL,

      [NormalizedUserName] [nvarchar](256) NULL,
      [Email] [nvarchar](256) NULL,
      [NormalizedEmail] [nvarchar](256) NULL,
      [EmailConfirmed] [bit] NOT NULL,
      [PasswordHash] [nvarchar](max) NULL,
      [SecurityStamp] [nvarchar](max) NULL,
      [ConcurrencyStamp] [nvarchar](max) NULL,
      [PhoneNumber] [nvarchar](max) NULL,
      [PhoneNumberConfirmed] [bit] NOT NULL,
      [TwoFactorEnabled] [bit] NOT NULL,
      [LockoutEnd] [datetimeoffset](7) NULL,
      [LockoutEnabled] [bit] NOT NULL,
      [AccessFailedCount] [int] NOT NULL
);

CREATE TABLE [Products] (
    [Id] [uniqueidentifier] PRIMARY KEY NOT NULL,
    [Name] [nvarchar](128) NOT NULL,

    [Price] [float] NOT NULL,
    [Stock] [int] NOT NULL DEFAULT(0),
    [Photo] [varchar](max) NULL
);
```

Controllers>HomeController.cs

```
public IActionResult Purchase(Guid id)
{
    var data = new Products();

    var con = new SqlConnection(_connectionString);
    con.Open();

    var sqlt = con.BeginTransaction();

    try
    {
        using (var cmd = new SqlCommand("SELECT [Id],[Name],[Price],[Stock],[Photo] FROM [Products] WHERE [Id] = @id", con, sqlt))
        {
            cmd.Parameters.Add("@id", SqlDbType.UniqueIdentifier).Value = id;
            var dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                data.Id = (Guid)dr["Id"];
                data.Name = (string)dr["Name"];
                data.Price = (double)dr["Price"];
                data.Stock = (int)dr["Stock"];
                data.Photo = (string)dr["Photo"];
            }
            dr.Close();
            if (data.Stock <= 0)
            {
                return BadRequest();
            }
        }

        using (var cmd = new SqlCommand(@"UPDATE [Products] SET [Stock] = @stock WHERE [Id] = @id;", con, sqlt))
        {
            cmd.Parameters.Add("@id", SqlDbType.UniqueIdentifier).Value = id;
            cmd.Parameters.Add("@stock", SqlDbType.Int).Value = data.Stock - 1;
            cmd.ExecuteNonQuery();
        }

        sqlt.Commit();

        return View(data);
    }
    catch (Exception)
    {
        sqlt.Rollback();
        throw;
    }
    finally { con.Close(); }
}
```

Controllers>HomeController.cs

```csharp
public IActionResult Purchase(Guid id)
{
    var data = new Products();
    var userId = new Guid(User.FindFirst(ClaimTypes.NameIdentifier)?.Value);
    var userName = User.Identity.Name;
    var con = new SqlConnection(_connectionString);
    con.Open();
    var sqlt = con.BeginTransaction();
    try
    {
        using (var cmd = new SqlCommand("SELECT [Id],[Name],[Price],[Stock],[Photo] FROM [Products] WHERE [Id] = @id", con, sqlt))
        {
            cmd.Parameters.Add("@id", SqlDbType.UniqueIdentifier).Value = id;
            var dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                data.Id = (Guid)dr["Id"];
                data.Name = (string)dr["Name"];
                data.Price = (double)dr["Price"];
                data.Stock = (int)dr["Stock"];
                data.Photo = (string)dr["Photo"];
            }
            dr.Close();
            if (data.Stock <= 0)
            {
                return BadRequest();
            }
        }
        using (var cmd = new SqlCommand(@"UPDATE [Products] SET [Stock] = @stock WHERE [Id] = @id;", con, sqlt))
        {
            cmd.Parameters.Add("@id", SqlDbType.UniqueIdentifier).Value = id;
            cmd.Parameters.Add("@stock", SqlDbType.Int).Value = data.Stock - 1;
            cmd.ExecuteNonQuery();
        }
        using (var cmd = new SqlCommand(@"INSERT INTO [UserProducts] ([Id],[UserId],[UserName],[ProductId],[Name],[Date]) VALUES (NEWID(),@userId, @userName, @productId, @name, @date);", con, sqlt))
        {
            cmd.Parameters.Add("@userId", SqlDbType.UniqueIdentifier).Value = userId;
            cmd.Parameters.Add("@userName", SqlDbType.NVarChar, 450).Value = userName;
            cmd.Parameters.Add("@productId", SqlDbType.UniqueIdentifier).Value = id;
            cmd.Parameters.Add("@name", SqlDbType.NVarChar, 128).Value = data.Name;
            cmd.Parameters.Add("@date", SqlDbType.SmallDateTime).Value = DateTime.Now;
            cmd.ExecuteNonQuery();
        }
        SendMail.Send(userName, "Compra realizada", "Gracias por la compra");
        sqlt.Commit();
        return View(data);
    }
    catch (Exception)
    {
        sqlt.Rollback();
        throw;
    }
    finally { con.Close(); }
}
```

Proyecto Bimestral

Primer entregable Bimestral

https://github.com/jlcarrillog/im221bim