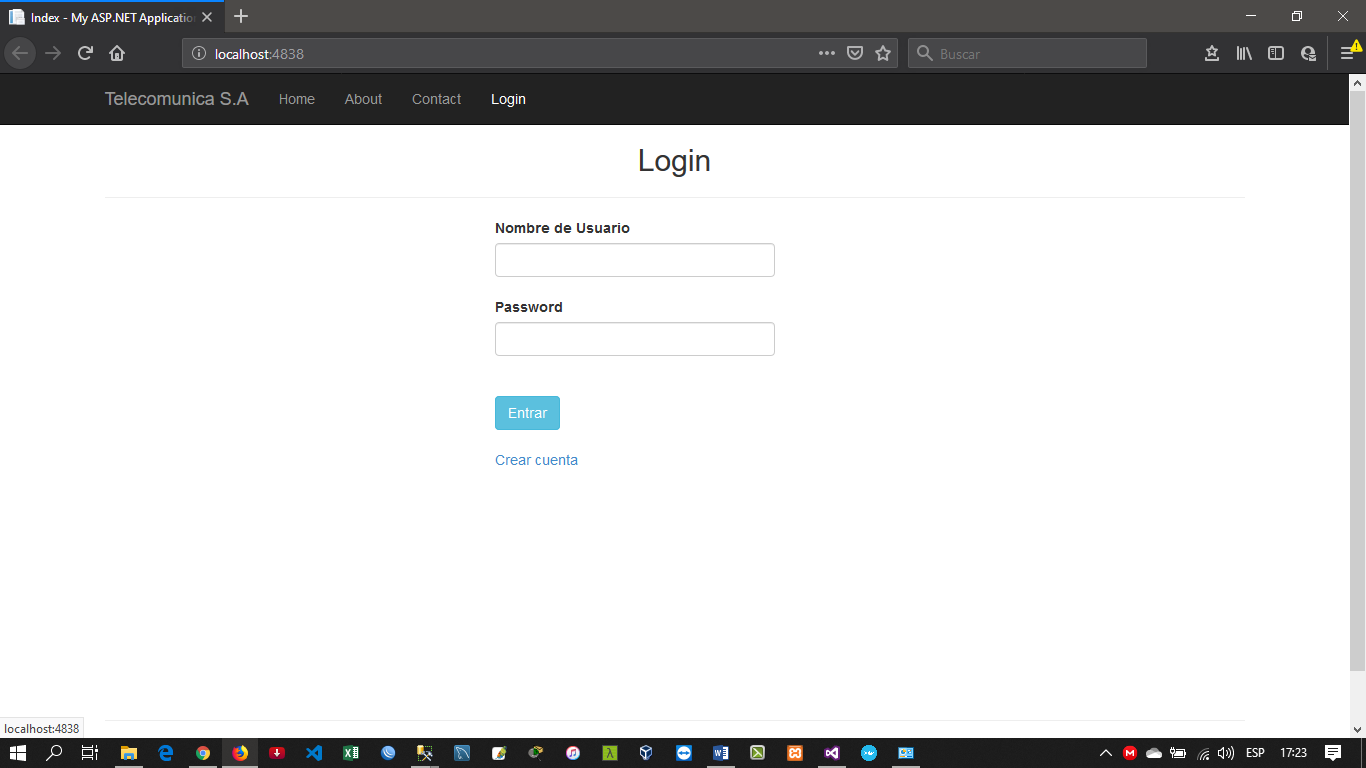
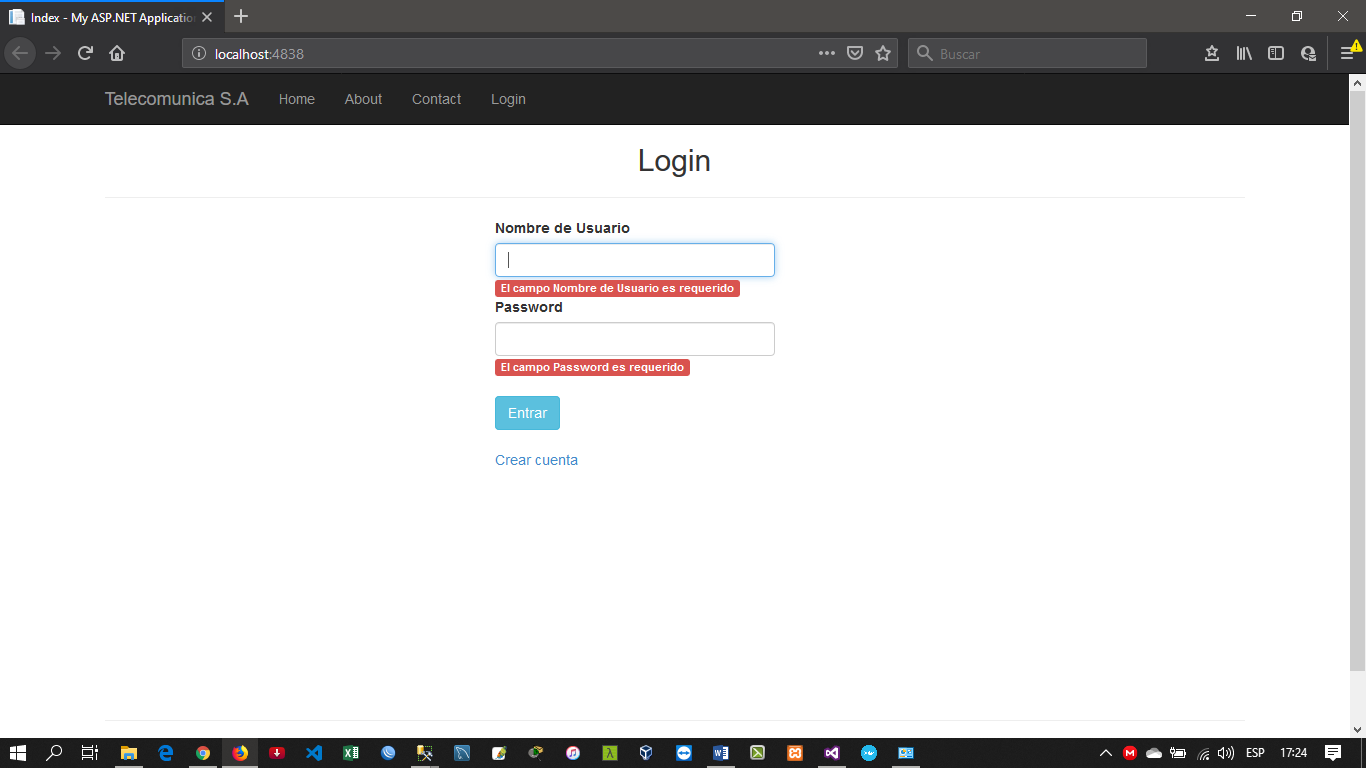
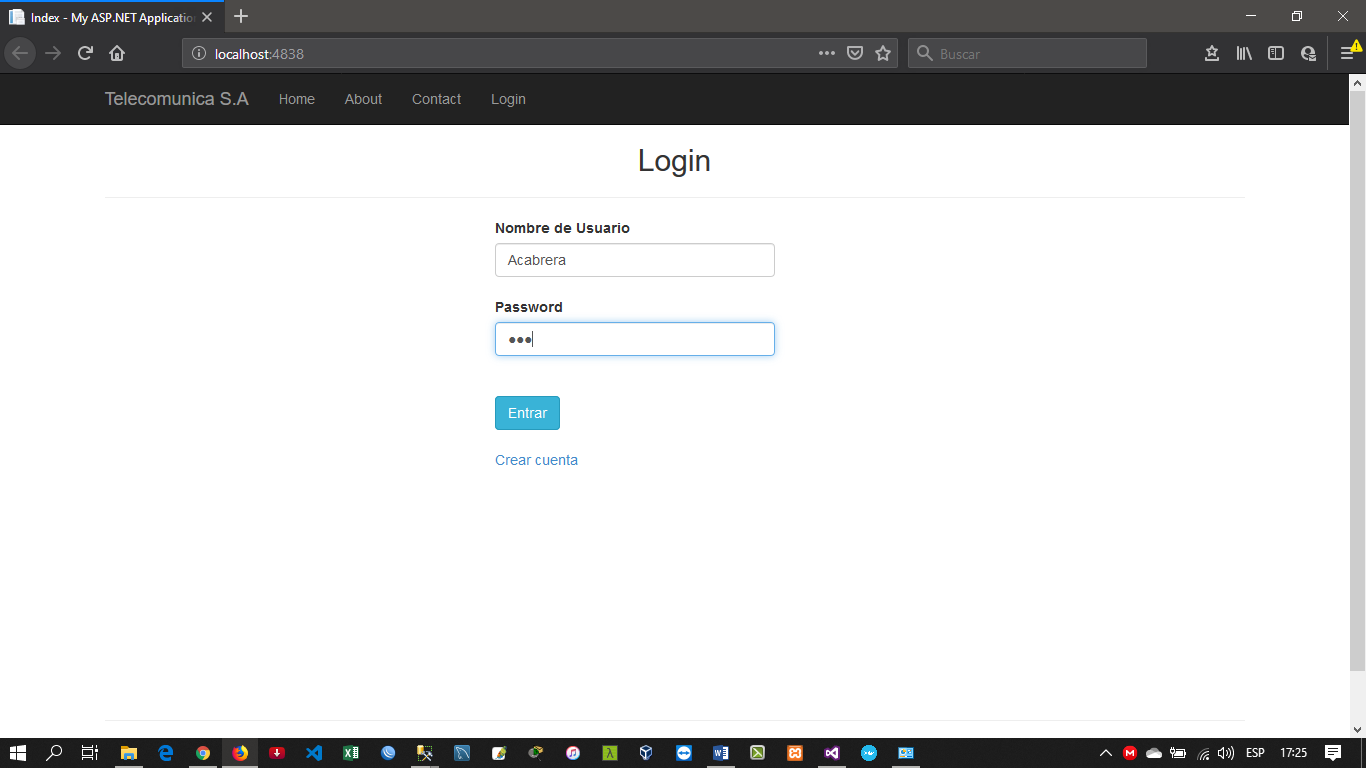
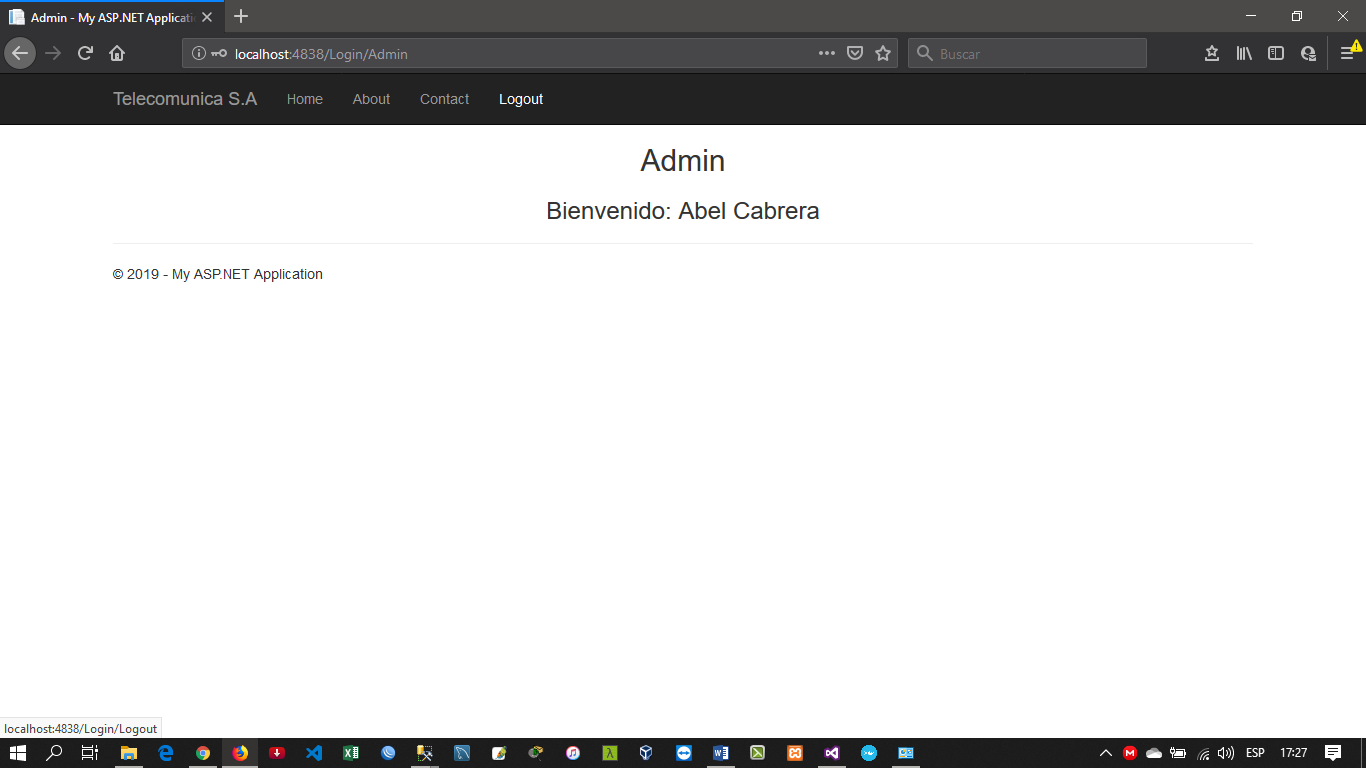
**SISTEMA WEB FUNCIONAL**

Validación De Campos





Usuario Autenticado, con Rol Administrativo



Datos del usuario en la base de datos SQLServer

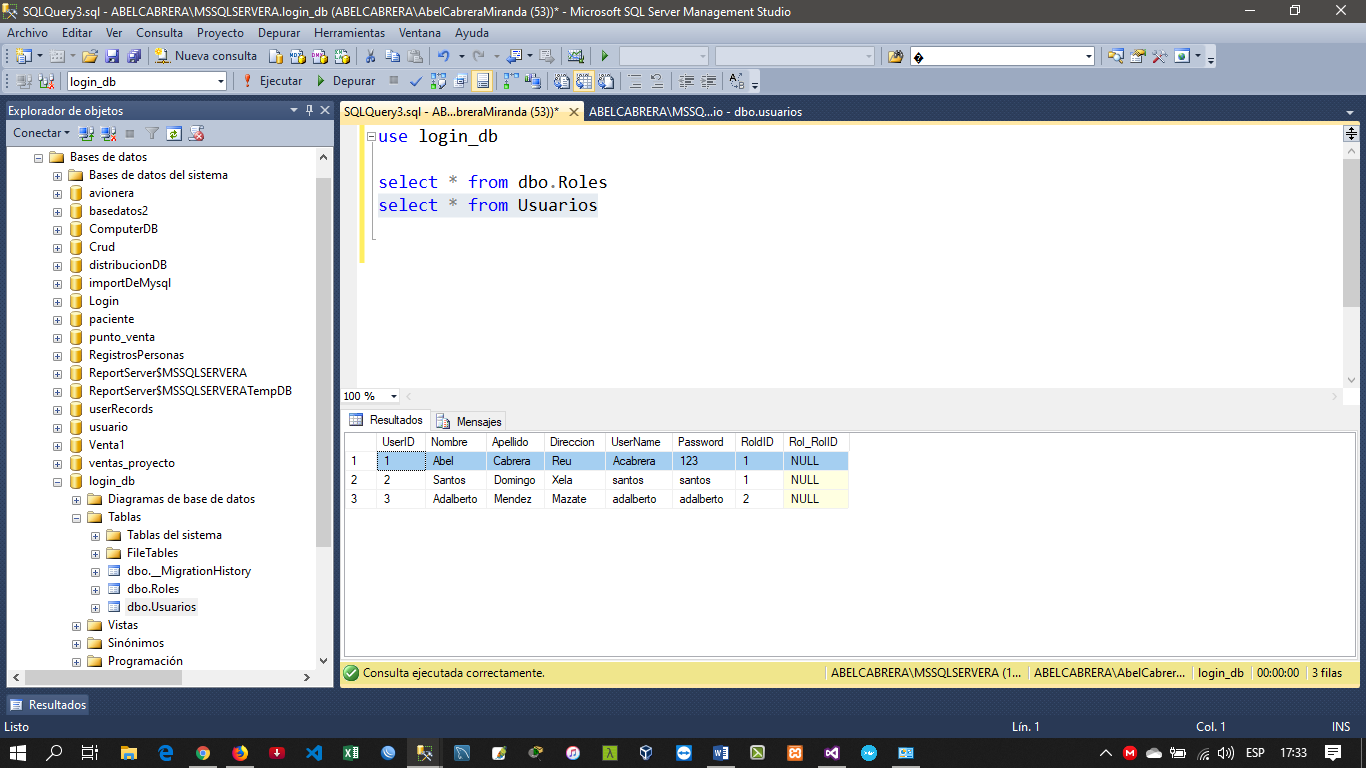
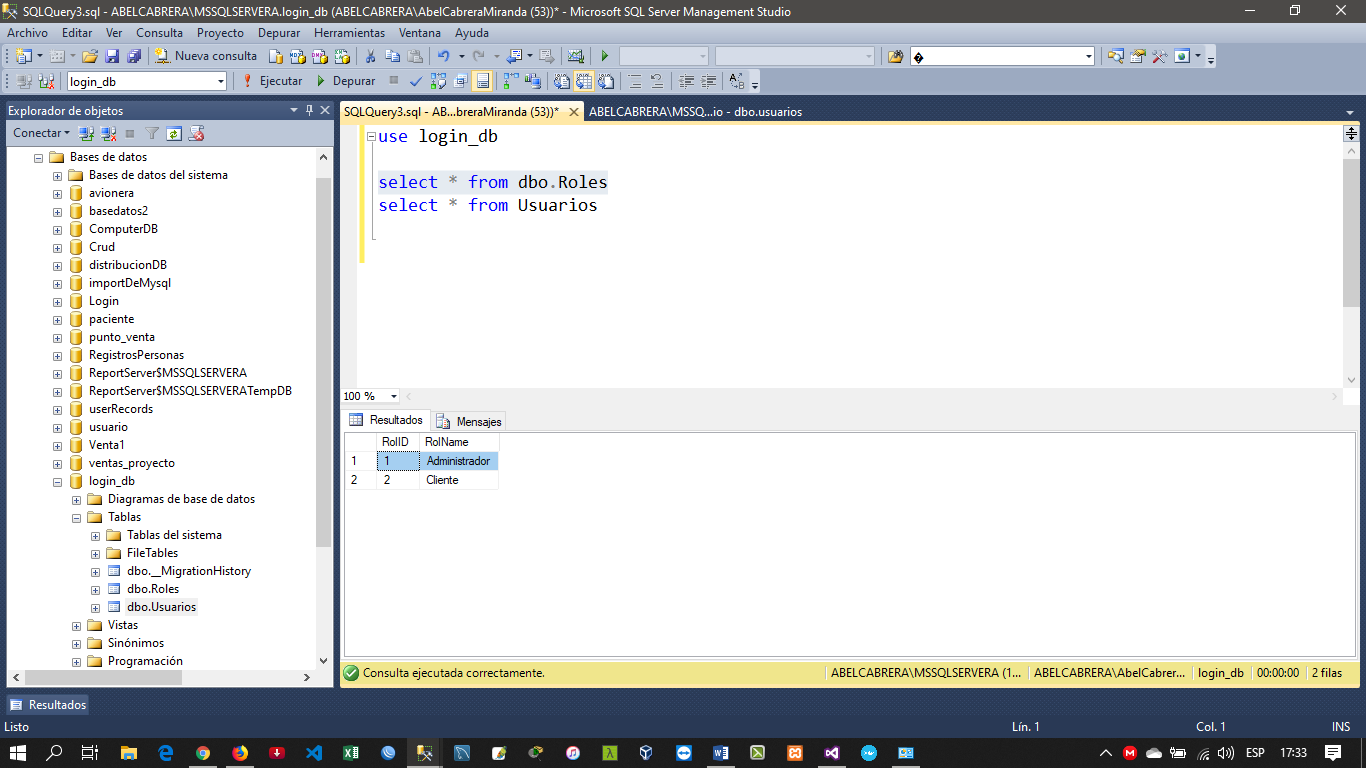
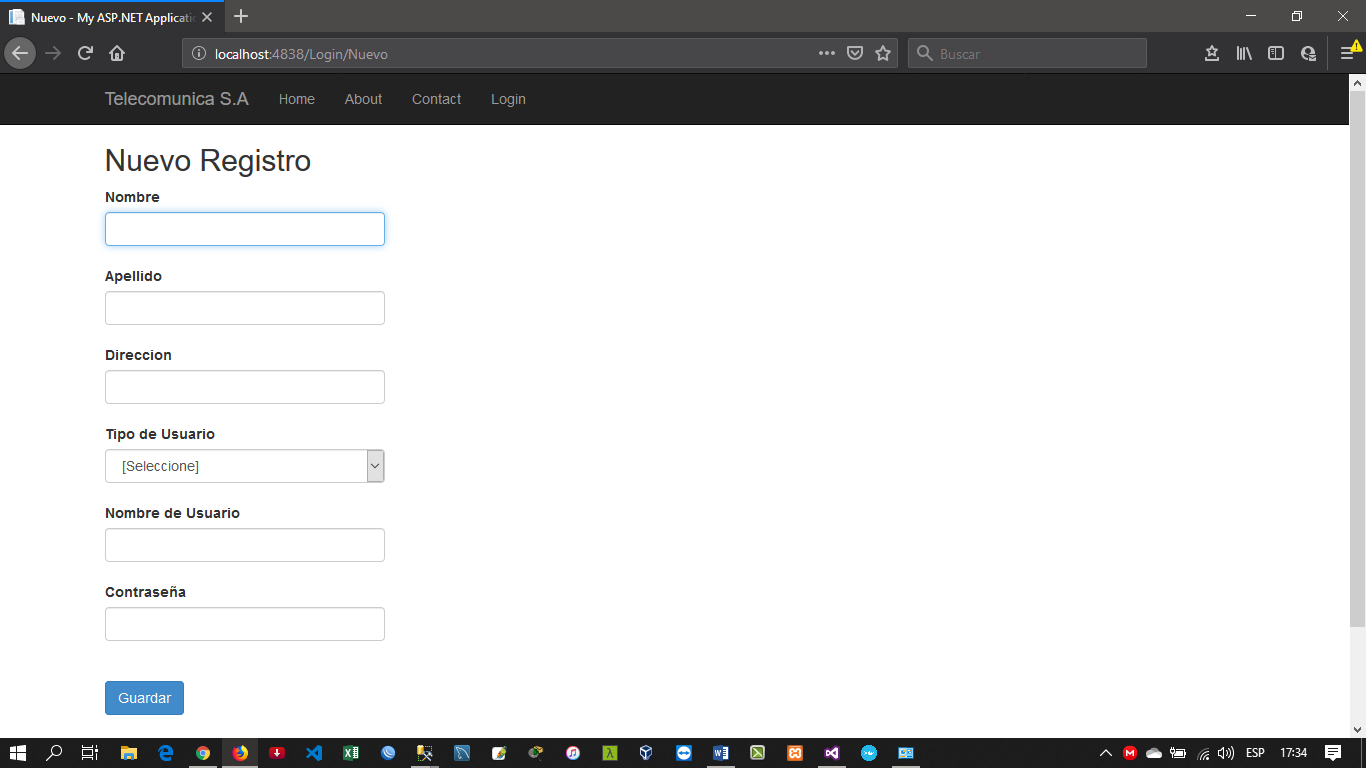


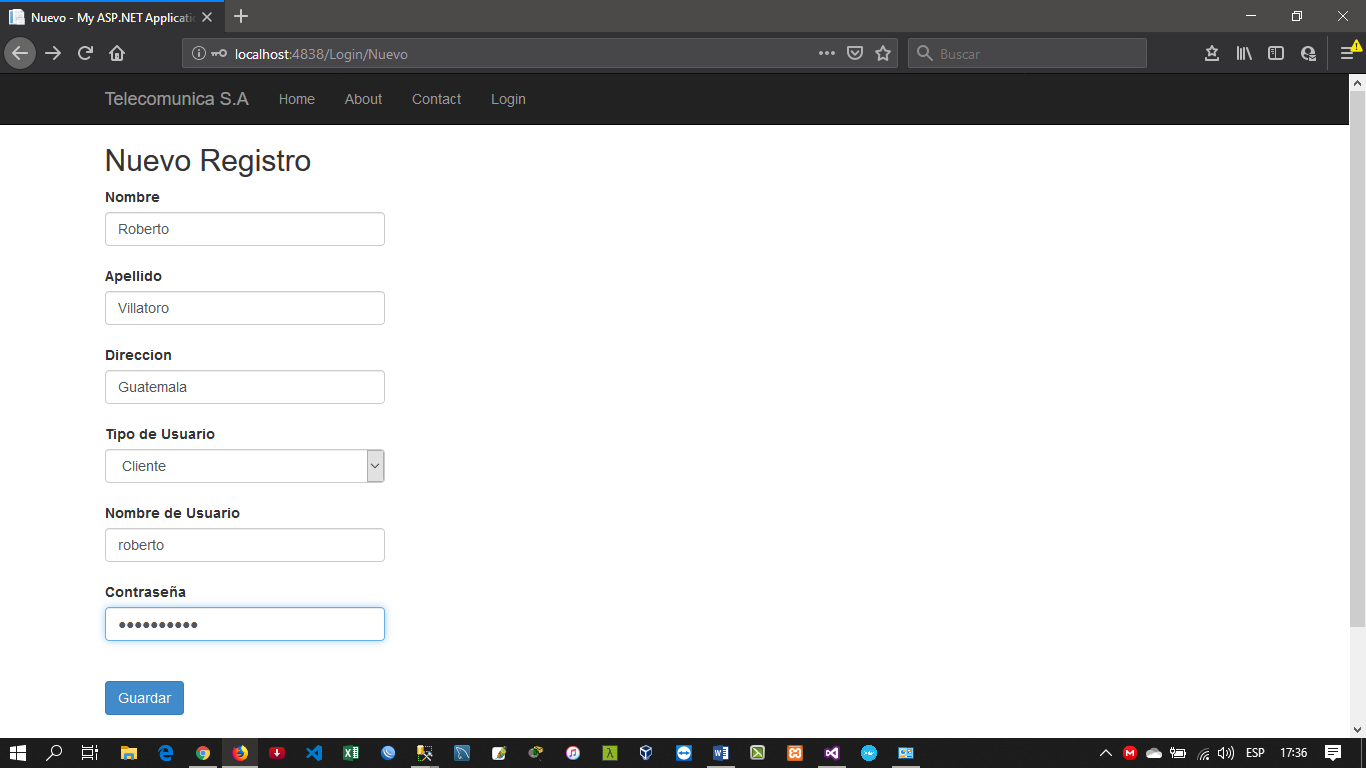
Tabla de Roles



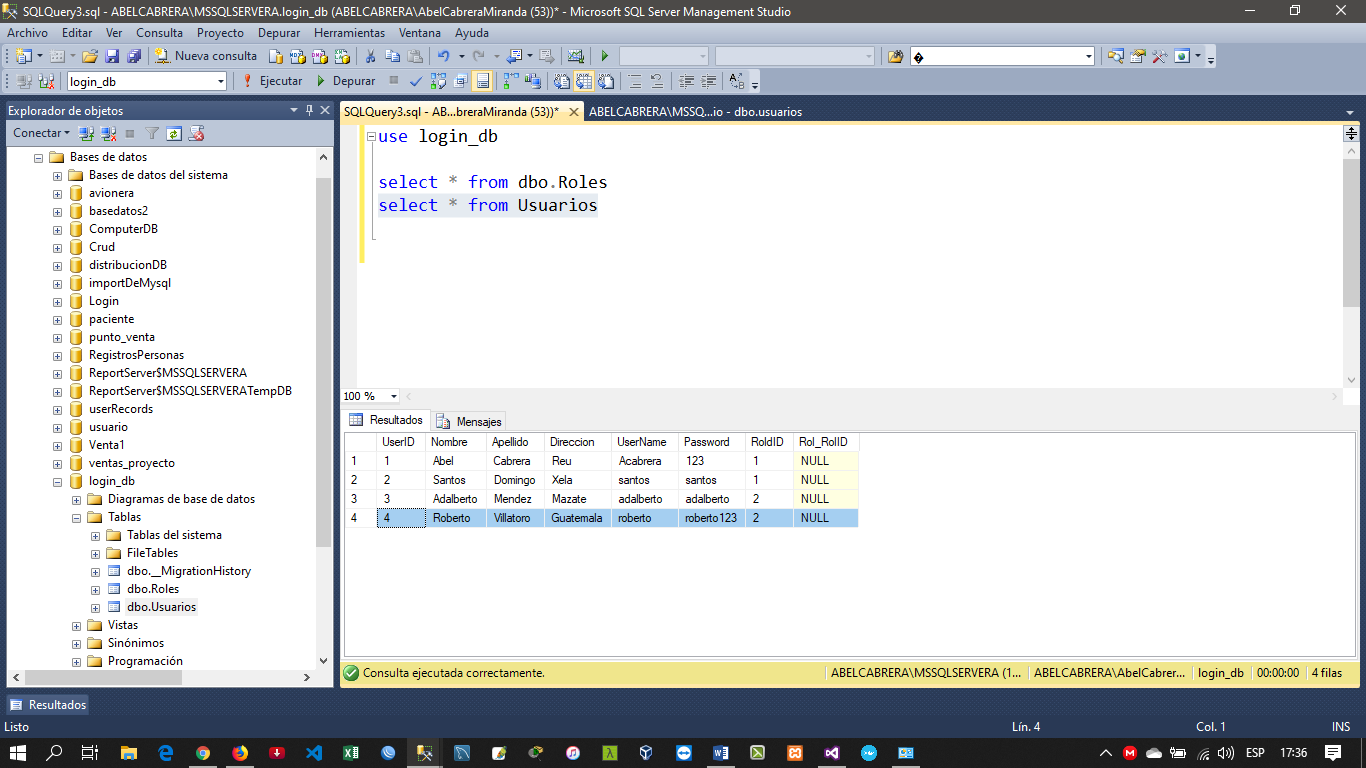
**REGISTRO DE UNA NUEVA CUENTA**

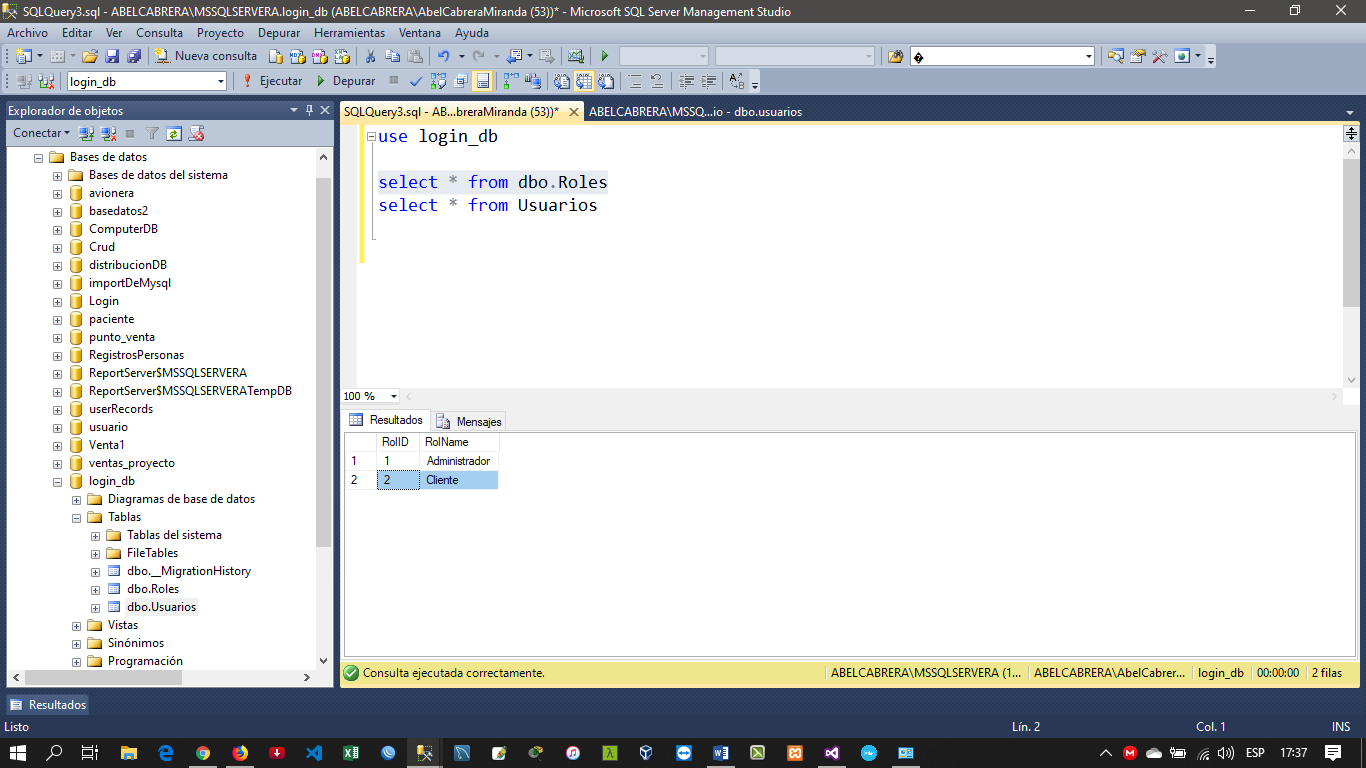
Usuario Nuevo



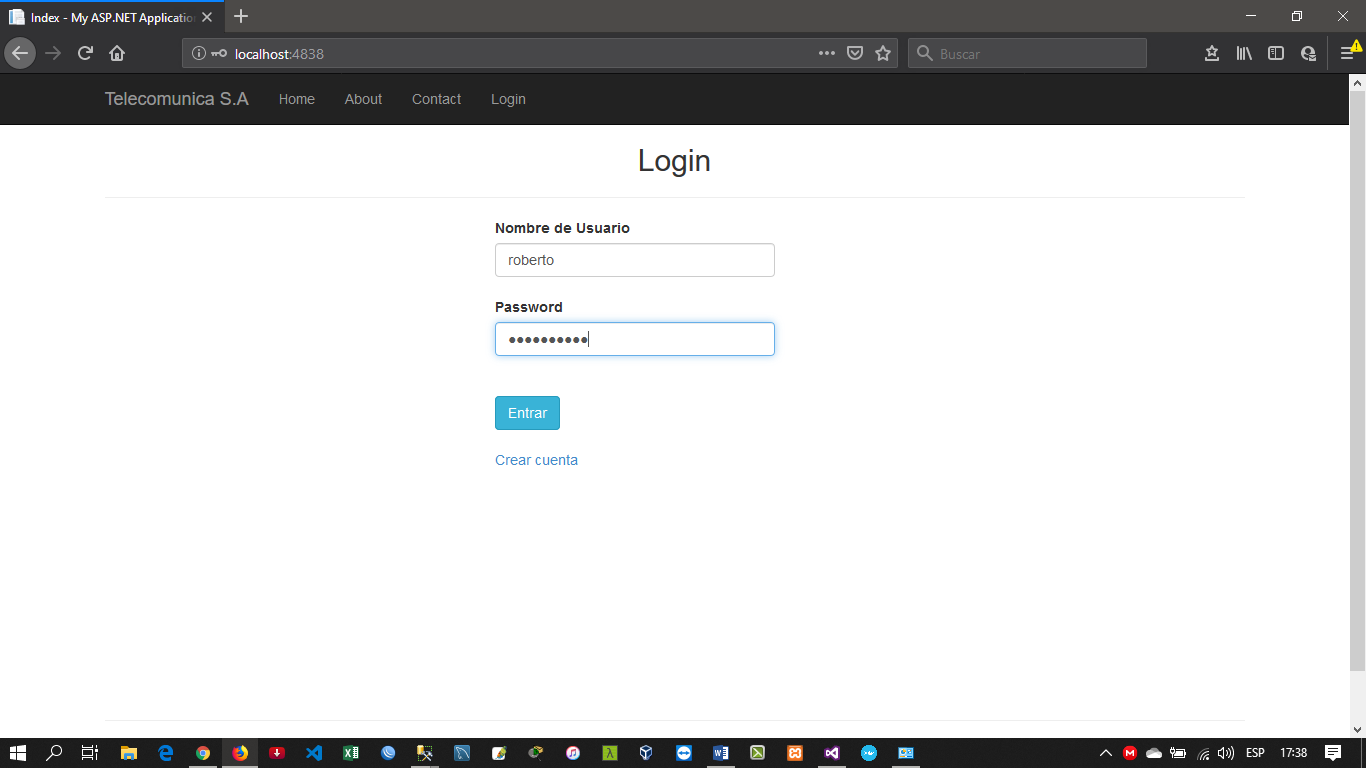


Datos del usuario almacenado en la Base de Datos

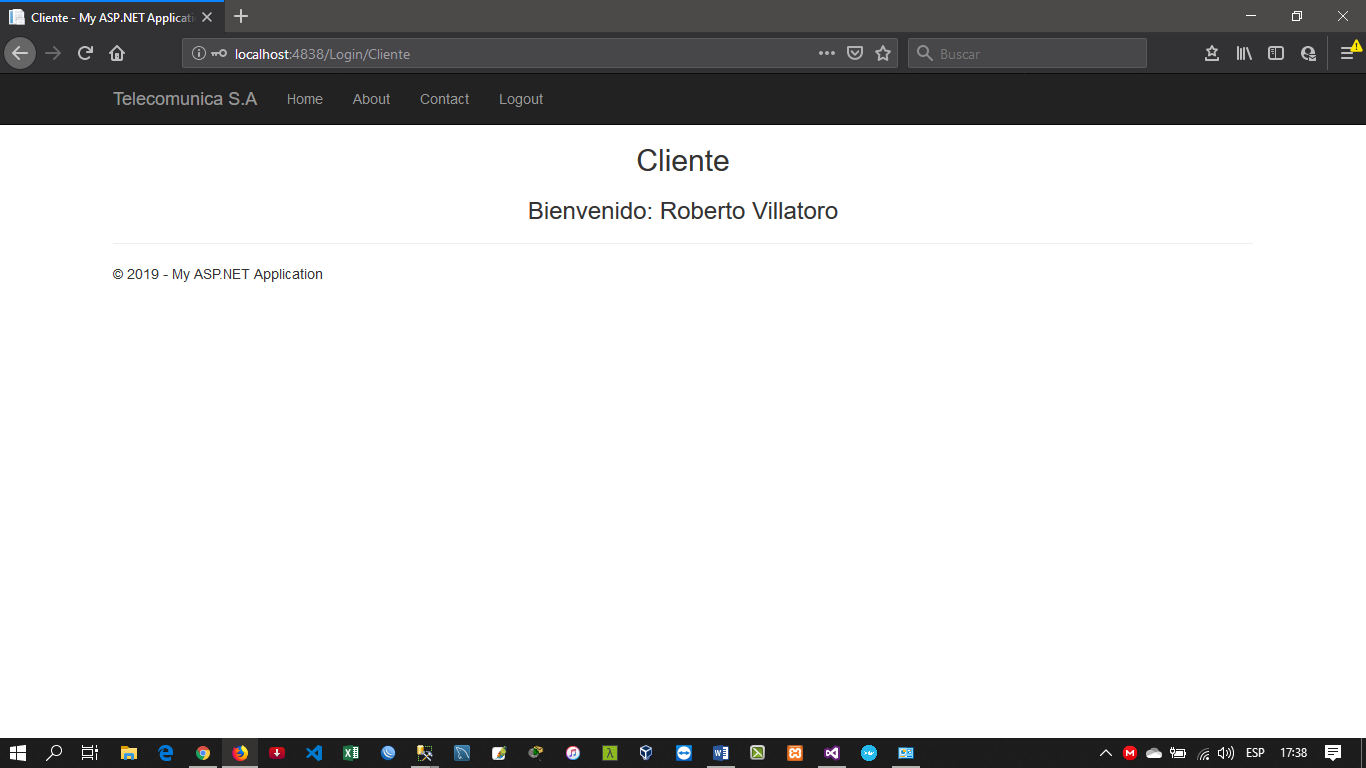




Inicio de Sesión del usuario nuevo



Usuario Logueado en el sistema



El sistema está programado para identificar el tipo de usuario, para luego redireccionarlo a la página que le corresponde.

**Admin**: es una página para administrador

**Cliente**: es una página para el cliente

CODIGO C# MVC4

**VISTAS**

**Vista: Index.cshtml**

@model LoginProject.ViewModels.LoginViewModel

@{

ViewBag.Title = "Index";

}

<h2 class="text-center">Login</h2>

<hr />

@using (Html.BeginForm("Index", "Login", FormMethod.Post))

{

<div class="row" style="margin-bottom:250px">

<div class="col-md-4 col-md-offset-4 col-xs-12">

@Html.LabelFor(d => d.UserName)

@Html.TextBoxFor(d => d.UserName, "", new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.UserName, "", new { @class = "label label-danger"})

<br />

@Html.LabelFor(d => d.Password)

@Html.PasswordFor(d => d.Password, new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.Password, "", new { @class = "label label-danger" })

<br /><br />

<button type="submit" class="btn btn-info">Entrar</button> <br />

<br>

<a href="/Login/Nuevo">Crear cuenta</a>

</div>

</div>

}

@section Scripts{

@Scripts.Render("~/bundles/jqueryval")

}

**Vista: Nuevo.cshtml**

@model LoginProject.Models.Usuario

@{

ViewBag.Title = "Nuevo";

}

<!-- nuevo usuario -->

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

<!-- Formulario, datos del nuevo usuario -->

@using (Html.BeginForm("Nuevo", "Login", FormMethod.Post))

{

<div class="row">

<div class="col-md-12">

<!-- Nombre -->

@Html.LabelFor(d => d.Nombre )

@Html.TextBoxFor(d => d.Nombre, "", new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.Nombre, "", new { @class = "label label-danger" })

<br />

<!-- Apelido -->

@Html.LabelFor(d => d.Apellido)

@Html.TextBoxFor(d => d.Apellido, "", new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.Apellido, "", new { @class = "label label-danger" })

<br />

<!-- Direccion -->

@Html.LabelFor(d => d.Direccion)

@Html.TextBoxFor(d => d.Direccion, "", new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.Direccion, "", new { @class = "label label-danger" })

<br />

<!-- Tipo de Usuario(Roles) -->

@Html.LabelFor(d => d.RoldID)

@Html.DropDownListFor(d => d.RoldID, (SelectList)ViewBag.Roles,"[Seleccione]" ,new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.RoldID, "", new { @class = "label label-danger" })

<br />

<!-- Nombre de Usuario -->

@Html.LabelFor(d => d.UserName)

@Html.TextBoxFor(d => d.UserName, "", new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.UserName, "", new { @class = "label label-danger" })

<br />

<!-- Contraseña del Usuario -->

@Html.LabelFor(d => d.Password)

@Html.PasswordFor(d => d.Password, new { @class = "form-control" })

@Html.ValidationMessageFor(d => d.Password, "", new { @class = "label label-danger" })

<br />

<br />

<!-- Botón Guardar datos -->

<div>

<input value="Guardar" class="btn btn-primary" type="submit" />

</div><br />

<a href="Index">Iniciar Sesión</a>

</div>

</div>

}

<!-- Scripts -->

@section Scripts{

@Scripts.Render("~/bundles/jqueryval")

}

**Vista: Cliente.cshtml**

@{

ViewBag.Title = "Cliente";

}

<h2 class="text-center">Cliente</h2>

<!—Usuario CLiente Logueado en el sistema web-->

<h3 class="text-center">Bienvenido: @ViewBag.User</h3>

**Vista: Admin.cshtml**

@{

ViewBag.Title = "Admin";

}

<h2 class="text-center">Admin</h2>

<!-- Usuario Administrador Logueado en el sistema web-->

<h3 class="text-center">Bienvenido: @ViewBag.User</h3>

**Vista: \_Layout.cshtml**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - My ASP.NET Application</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

@Html.ActionLink("Telecomunica S.A", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

</div>

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li>@Html.ActionLink("Home", "Index", "Home")</li>

<li>@Html.ActionLink("About", "About", "Home")</li>

<li>@Html.ActionLink("Contact", "Contact", "Home")</li>

<!--Autenticación de usuario, si a iniciado sesión al sistema-->

@if (HttpContext.Current.Session["User"] == null)

{

<!--Se habilita la opcion Login si el usuario no ha iniciado sesión-->

<li class="text-left">@Html.ActionLink("Login", "Index", "Login")</li>

}

else

{

<!--Se habilita la opcion Logout para cerrar sesión si el usuario se ha logueado-->

<li class="text-left">@Html.ActionLink("Logout", "Logout","Login")</li>

}

</ul>

</div>

</div>

</div>

<div class="container body-content">

@RenderBody()

<hr />

<footer>

<p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

<script src="~/Scripts/jquery.validate.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.js"></script>

</body>

</html>

**MODELOS DE DATOS**

**Modelo: Usuario.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Web;

namespace LoginProject.Models

{

[Table("Usuarios")]

public class Usuario

{

//Modelo de dato de Usuario

[Key]

public int UserID { get; set; }

[Required(ErrorMessage = "El campo {0} es requerido")]

public string Nombre { get; set; }

[Required(ErrorMessage = "El campo {0} es requerido")]

public string Apellido { get; set; }

[Required(ErrorMessage = "El campo {0} es requerido")]

public string Direccion { get; set; }

[Required(ErrorMessage = "El campo {0} es requerido")]

[Display(Name = "Nombre de Usuario")]

public string UserName { get; set; }

[Required(ErrorMessage = "El campo {0} es requerido")]

[Display(Name = "Contraseña")]

public string Password { get; set; }

[Display(Name = "Tipo de Usuario")]

public int RoldID { get; set; }

public virtual Rol Rol { get; set; }

}

}

**Modelo: LoginContext.cs**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Web;

namespace LoginProject.Models

{

public class LoginContext : DbContext

{

public LoginContext(): base("login\_db") { }

public DbSet<Usuario> Usuarios { get; set; }

public DbSet<Rol> Roles { get; set; }

}

}

Modelo: Rol.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Web;

namespace LoginProject.Models

{

[Table("Roles")]

public class Rol

{

//Modelo de datos del ROl

[Key]

public int RolID { get; set; }

public string RolName { get; set; }

public virtual IEnumerable<Usuario> Usuarios { get; set; }

public Rol()

{

this.Usuarios = new HashSet<Usuario>();

}

}

}

**CONTROLADORES**

**LoginController.cs**

using LoginProject.Models;

using LoginProject.ViewModels;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Claims;

using System.Web;

using System.Web.Mvc;

namespace LoginProject.Controllers

{

public class LoginController : Controller

{

private readonly string rolAdmin = "Administrador";

private readonly string rolCliente = "Cliente";

// GET: Usuarios

[HttpGet]

public ActionResult Index()

{

return View();

}

//Metodo Index

[HttpPost]

public ActionResult Index(LoginViewModel model)

{

if (model != null)

{

using (var context = new LoginContext())

{

var user = context.Usuarios

.Where(x => x.UserName == model.UserName && x.Password == model.Password)

.FirstOrDefault();

if (user != null)

{

var rol = context.Roles.FirstOrDefault(x => x.RolID == user.RoldID);

if(rol != null)

{

Session["User"] = $"{user.Nombre} {user.Apellido}";

Session["Rol"] = rol.RolName;

if (rol.RolName.Equals("Administrador"))

{

return Redirect("/Login/Admin");

}

else

{

return Redirect("/Login/Cliente");

}

}

}

else

return View(model);

}

}

return View(model);

}

//Metodo Admin

public ActionResult Admin()

{

var session = Session["User"];

if (session == null)

{

return RedirectToAction("Index");

}

if (Session["Rol"].ToString().Equals(this.rolAdmin))

{

ViewBag.User = Session["User"];

return View();

}

else

{

return RedirectToAction("Cliente");

}

}

//Metodo Cliente

public ActionResult Cliente()

{

var session = Session["User"];

if (session == null)

{

return RedirectToAction("Index");

}

if (Session["Rol"].ToString().Equals(this.rolCliente))

{

ViewBag.User = Session["User"];

return View();

}

else

{

return RedirectToAction("Admin");

}

}

//Metodo Logout

public ActionResult Logout()

{

Session["User"] = null;

return RedirectToAction("Index");

}

//Metodo Nuevo

[HttpGet]

public ActionResult Nuevo()

{

var roles = new List<Rol>();

using (var context = new LoginContext())

{

roles = context.Roles.ToList();

}

ViewBag.Roles = new SelectList(roles, "RolID", "RolName");

return View();

}

//Metodo Nuevo

[HttpPost]

public ActionResult Nuevo(Usuario model)

{

try

{

if (ModelState.IsValid)

{

using (LoginContext db = new LoginContext())

{

var oUsu = new Usuario();

oUsu.Nombre = model.Nombre;

oUsu.Apellido = model.Apellido;

oUsu.Direccion = model.Direccion;

oUsu.RoldID = model.RoldID;

oUsu.UserName = model.UserName;

oUsu.Password = model.Password;

db.Usuarios.Add(oUsu);

db.SaveChanges();

}

return Redirect("/");

}

return View(model);

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

}

}

**CADENA DE CONEXIÓN A LA BASE DE DATOS SQLSERVER**

Archivo: Web.config

<connectionStrings><!---Configuracion de la cadena de conexion con la base de datos-->

<add name="login\_db" connectionString="Server=ABELCABRERA\MSSQLSERVERA; Database=login\_db; Integrated Security=True" providerName="System.Data.SqlClient" />

</connectionStrings>

Nota: esta es la parte de la conexión a la base de datos y el resto de código se encuentra en el archivo: Web.config