Segunda prueba de actuación en clase M2C – 09/12/2020

Tiempo – 90 minutos.

Nombre: Christian Alico

Numero: 154519

Dado el siguiente modelo y sus constructores:

namespace prueba2.Models

{

public class Cliente

{

private static int ultimoId = 1;

public int Id { get; set; }

public string Nombre { get; set; }

public string Apellido { get; set; }

public string Cedula { get; set; }

public bool Estado { get; set; }

private List<string> generosFavoritos listaGenerosFavoritos = new List<String>

public Cliente()

{

}

public Cliente(string nombre, string apellido, string cedula)

{

Id =ultimoId;

ultimoId++;

Nombre = nombre;

Apellido = apellido;

Cedula = cedula;

Estado = true;

}

}

}

**1 – Realizar vista de alta de Cliente, puede utilizar Html o Razor.**

|  |
| --- |
| **@model prueba2.Models.Usuario**  **@{**  **ViewBag.Title = "AltaCliente";**  **}**  **<h2Alta Cliente</h2>**  **@using (Html.BeginForm())**  **{**  **@Html.AntiForgeryToken()**  **<div class="form-horizontal">**  **<h4>Cliente</h4>**  **<hr />**  **@Html.ValidationSummary(true, "", new { @class = "text-danger" })**  **<div class="form-group">**  **@Html.LabelFor(model => model.Nombre, htmlAttributes: new { @class = "control-label col-md-2" })**  **<div class="col-md-10">**  **@Html.EditorFor(model => model.Nombre, new { htmlAttributes = new { @class = "form-control" } })**  **@Html.ValidationMessageFor(model => model.Nombre, "", new { @class = "text-danger" })**  **</div>**  **</div>**  **<div class="form-group">**  **@Html.LabelFor(model => model.Apellido, htmlAttributes: new { @class = "control-label col-md-2" })**  **<div class="col-md-10">**  **@Html.EditorFor(model => model.Apellido, new { htmlAttributes = new { @class = "form-control" } })**  **@Html.ValidationMessageFor(model => model.Apellido, "", new { @class = "text-danger" })**  **</div>**  **</div>**  **<div class="form-group">**  **@Html.LabelFor(model => model.Cedula, htmlAttributes: new { @class = "control-label col-md-2" })**  **<div class="col-md-10">**  **@Html.EditorFor(model => model.Cedula, new { htmlAttributes = new { @class = "form-control" } })**  **@Html.ValidationMessageFor(model => model.Cedula, "", new { @class = "text-danger" })**  **</div>**  **</div>**  **</div>**  **<div class="form-group">**  **<div class="col-md-offset-2 col-md-10">**  **<input type="submit" value="Alta" class="btn btn-default" required />**  **</div>**  **</div>**  **</div>**  **}**  **<div>**  **@Html.ActionLink("Volver a la pagina principal", "../Home/Index")**  **</div>**  **@section Scripts {**  **@Scripts.Render("~/bundles/jqueryval")**  **}** |

**2 – Realizar métodos del controller que permitan conectar el formulario de la vista con el modelo. El alta debe utilizar el contstructor con parámetros y retornar un mensaje de alta satisfactoria a través de un viewBag.**

|  |
| --- |
| Sistema S = Sistema.GetInstancia();  // POST: Cliente/AltaCliente  [HttpPost]  public ActionResult AltaCliente(Usuario u)  {  try  {  Usuario Usu = S.agregarCliente(u.Nombre, u.Apellido,u.Cedula,);  a.agregarCliente(u.Nombre, u.Apellido, u.Cedula);  if (Usu != null)  {    ViewBag.Msg = "Cliente agregado: " + Usu.Nombre + " " + Usu.Apellido ";  }  else {  ViewBag.Msg = "No se pudo Registrar el Cliente, verifique sus datos";  }    return View("../Home/Index");    }  catch  {  return View();  }  } |

**3 – El siguiente actionLink es para hacer un borrado lógico de un cliente.**

@Html.ActionLink("Delete", "Delete", new { id = item.Id })

Realizar el método por get que retorne a la vista el cliente que se desea borrar. Se debe validar que únicamente un usuario logueado con rol “Admin” pueda realizar esta operación, sino debe ser redirigido o avisado de alguna forma. Podemos asumir que el sistema cuenta con un método getCliente(id) y no es necesario implementarlo.

|  |
| --- |
| Sistema S = Sistema.GetInstancia();  public ActionResult BorrarCliente(int id)  {  if ((string)Session["logueadoRol"] == "Admin")  {  Cliente buscado = S.getCliente(id);    ViewBag.ClienteBuscado = buscado;  return View("../Cliente/BorrarCliente");  }  else {  ViewBag.MSG = "Para acceder a este sitio debe estar logeado como Administrador";  return View("../Home/Index");  }  } |

**4 – El siguiente código pertenece a una vista que se llama busquedaCliente.**

<h3>FORMULARIO</h3>

@using (Html.BeginForm())

{

<input type="text" id="nombre" name="nombre"><br />

<input type="text" id="apellido" name="apellido"><br />

<input type="submit" value="buscar">

}

**Escribir el método POST que permita buscar un cliente por nombre y apellido. Podemos asumir que existe un método en el sistema que se llama getCliente(string n, string a) y retorna una lista de clientes .**

|  |
| --- |
| Sistema S = Sistema.GetInstancia();  **[HttpPost]**  **public ActionResult busquedaCliente (string nombre,string apellido) {**    **ViewBag.ClientesBuscados = S.getCliente(nombre,apellido)**  **return View(“../Cliente/busquedaCliente”);**  **}** |

**5 - Dado el siguiente código de la vista busquedaPorFechas**

<h3>FORMULARIO</h3>

@using (Html.BeginForm())

{

<input type="text" id="categoria" name="categoria"><br />

<input type="Date" id="fch1" name=" fch1"><br />

<input type="Date" id="fch2" name=" fch2"><br />

<input type="submit" value="buscar">

}

**Escribir únicamente el nombre y los parámetros del método actionResult.**

|  |
| --- |
| [HttpPost]  public ActionResult busquedaPorFechas(string categoría,DateTime fch1,DateTime fch2){  return View();  } |

**6 – Nos llega un objeto Cliente a nuestra vista a través de un @ViewBag.cliente**

**Listar todos sus datos incluyendo su lista de géneros Favoritos. (Asumo que Cliente tiene un método getFavoritos() para obtenerla lista de generosFavoritos**

|  |
| --- |
| @{  ViewBag.Title = "ListarCliente";  }  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h2>Cliente</h2>  <hr />  <h2>Detalles del Cliente</h2>  <br />  <h3>Id</h3>  @ViewBag.cliente.Id  <h3>Nombre</h3>  @ViewBag.cliente.Nombre  <h3>Apellido</h3>  @ViewBag.cliente.Apellido  <h3>Cedula</h3>  @ViewBag.cliente.Cedula  <h3> Estado </h3>  @if(ViewBag.cliente.Estado == true){  <p>Activo</p>  }else{  <p>Inactivo</p>  }    <h3>Géneros Favoritos</h3>  @foreach (var Genero in ViewBag.cliente.getFavoritos())  {  <h4>Genero</h4>  <p>@Genero.Nombre</p>    } |

**7 – Al loguearnos hemos almacenado en variables de sesión los datos del usuario para poder operar. Realizar el método del controller que permita hacer el logout del sistema y envíe al usuario al home. Este método se ejecuta a partir del siguiente action link:**

**@Html.ActionLink("Cerrar sesion", "Salir", "Home")**

|  |
| --- |
| **//El Método se encuentra en HomeController**  **public ActionResult Salir()**  **{**  **If((string) Session[“logueadoRol”] != null){**  **Session["logueadoRol"] = null;**  **Session["logueadoNombre"] = null;**  **Session["logueadoApellido"] = null;**    **ViewBag.Msg = "Inicie Session";**    **}**  **return View("Index");**  **}** |