

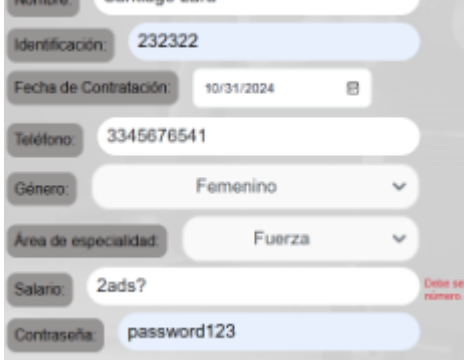


- PRIMER RELEASE
- 1. E1-HU1: LOGIN
  - 2. E1 - HU1: REGISTRAR CLIENTES
  - 3. E1 - HU2: ASIGNAR MEMBRESÍA
  - 4. E2 - HU1: REGISTRAR ENTRENADORES

ID	Componente	Valores	Tipo	Resultado Esperado	Resultado Real																										
E2-H U1-C P1	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) {     Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value;     HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString());     if (ModelState.IsValid)     {         try         {             if (Cliente.refMembresia == "Premium")             {                 return Json(new { mensaje = "Membresia Premium no disponible." }, JsonRequestBehavior.AllowGet);             }             else             {                 return Json(new { mensaje = "Membresia General disponible." }, JsonRequestBehavior.AllowGet);             }         }         catch { }     }     return Json(new { mensaje = "Error al registrar cliente." }, JsonRequestBehavior.AllowGet); }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.ClienteModel}</td></tr><tr><td>DiasRestantes</td><td>0</td></tr><tr><td>FechaNacimiento</td><td>{11/01/2005 0:00:00}</td></tr><tr><td>Genero</td><td>"M"</td></tr><tr><td>IdEntrenador</td><td>0</td></tr><tr><td>IdSede</td><td>"1"</td></tr><tr><td>Identificacion</td><td>"80"</td></tr><tr><td>Nombre</td><td>"sofi"</td></tr><tr><td>Telefono</td><td>"3106683378"</td></tr><tr><td>fechaMembresia</td><td>null</td></tr><tr><td>refMembresia</td><td>"General"</td></tr></table>	Cliente	{NET_MVC.Models.ClienteModel}	DiasRestantes	0	FechaNacimiento	{11/01/2005 0:00:00}	Genero	"M"	IdEntrenador	0	IdSede	"1"	Identificacion	"80"	Nombre	"sofi"	Telefono	"3106683378"	fechaMembresia	null	refMembresia	"General"	+	Respuesta de admPersona: true Respuesta de admCliente: true	<table><tr><td>respuesta</td><td>true</td></tr><tr><td>respuesta2</td><td>true</td></tr></table>	respuesta	true	respuesta2	true
Cliente	{NET_MVC.Models.ClienteModel}																														
DiasRestantes	0																														
FechaNacimiento	{11/01/2005 0:00:00}																														
Genero	"M"																														
IdEntrenador	0																														
IdSede	"1"																														
Identificacion	"80"																														
Nombre	"sofi"																														
Telefono	"3106683378"																														
fechaMembresia	null																														
refMembresia	"General"																														
respuesta	true																														
respuesta2	true																														
E2-H U1-C P2	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) {     Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value;     HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString());     if (ModelState.IsValid)     {         try         {             if (Cliente.refMembresia == "Premium")             {                 return Json(new { mensaje = "Membresia Premium no disponible." }, JsonRequestBehavior.AllowGet);             }             else             {                 return Json(new { mensaje = "Membresia General disponible." }, JsonRequestBehavior.AllowGet);             }         }         catch { }     }     return Json(new { mensaje = "Error al registrar cliente." }, JsonRequestBehavior.AllowGet); }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.ClienteModel}</td></tr><tr><td>DiasRest...</td><td>0</td></tr><tr><td>FechaNa...</td><td>{11/01/1999 0:00:00}</td></tr><tr><td>Genero</td><td>"F"</td></tr><tr><td>IdEntren...</td><td>0</td></tr><tr><td>IdSede</td><td>"1"</td></tr><tr><td>Identifica...</td><td>"111"</td></tr><tr><td>Nombre</td><td>"sofi"</td></tr><tr><td>Telefono</td><td>"3106683378"</td></tr><tr><td>fechaMe...</td><td>null</td></tr><tr><td>refMemb...</td><td>"General"</td></tr></table>	Cliente	{NET_MVC.Models.ClienteModel}	DiasRest...	0	FechaNa...	{11/01/1999 0:00:00}	Genero	"F"	IdEntren...	0	IdSede	"1"	Identifica...	"111"	Nombre	"sofi"	Telefono	"3106683378"	fechaMe...	null	refMemb...	"General"	+	1. Respuesta de admPersona: false Respuesta de admCliente: false 2.Excepción personalizada de oracle indicando que la persona existe	<table><tr><td>respuesta</td><td>false</td></tr><tr><td>respuesta2</td><td>false</td></tr></table> <pre>{"Error en Oracle: ORA-20002: Persona existente\nC"}</pre>	respuesta	false	respuesta2	false
Cliente	{NET_MVC.Models.ClienteModel}																														
DiasRest...	0																														
FechaNa...	{11/01/1999 0:00:00}																														
Genero	"F"																														
IdEntren...	0																														
IdSede	"1"																														
Identifica...	"111"																														
Nombre	"sofi"																														
Telefono	"3106683378"																														
fechaMe...	null																														
refMemb...	"General"																														
respuesta	false																														
respuesta2	false																														

E2-H U1-C P3	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) {     Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value;     HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString());     if (ModelState.IsValid)     {         try         {             if (Cliente.refMembresia == "Premium")             {                 // ...             }         }     } }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.Cl</td></tr><tr><td>DiasRest...</td><td>0</td></tr><tr><td>FechaNa...</td><td>{11/01/1999 0:00:00}</td></tr><tr><td>Genero</td><td>"F"</td></tr><tr><td>IdEntren...</td><td>0</td></tr><tr><td>IdSede</td><td>"1"</td></tr><tr><td>Identifica...</td><td>"1111"</td></tr><tr><td>Nombre</td><td>"sofi"</td></tr><tr><td>Telefono</td><td>"3106683378"</td></tr><tr><td>fechaMe...</td><td>null</td></tr><tr><td>refMemb...</td><td>"Premium"</td></tr></table>	Cliente	{NET_MVC.Models.Cl	DiasRest...	0	FechaNa...	{11/01/1999 0:00:00}	Genero	"F"	IdEntren...	0	IdSede	"1"	Identifica...	"1111"	Nombre	"sofi"	Telefono	"3106683378"	fechaMe...	null	refMemb...	"Premium"	+	Respuesta positiva y redirección al módulo asignar entrenador	<table><tr><td>Request</td><td>GET http://localhost:5009/Cliente/AsignarEntrenadorCliente</td></tr><tr><td>Response</td><td>200 OK</td></tr></table>	Request	GET http://localhost:5009/Cliente/AsignarEntrenadorCliente	Response	200 OK														
Cliente	{NET_MVC.Models.Cl																																												
DiasRest...	0																																												
FechaNa...	{11/01/1999 0:00:00}																																												
Genero	"F"																																												
IdEntren...	0																																												
IdSede	"1"																																												
Identifica...	"1111"																																												
Nombre	"sofi"																																												
Telefono	"3106683378"																																												
fechaMe...	null																																												
refMemb...	"Premium"																																												
Request	GET http://localhost:5009/Cliente/AsignarEntrenadorCliente																																												
Response	200 OK																																												
E2-H U1-C P4	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) {     Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value;     HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString());     if (ModelState.IsValid)     {         try         {             if (Cliente.refMembresia == "Premium")             {                 // ...             }         }     } }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.C</td></tr><tr><td>DiasRestantes</td><td>0</td></tr><tr><td>FechaNacimie...</td><td>{11/01/1999 0:00:00}</td></tr><tr><td>Genero</td><td>"F"</td></tr><tr><td>IdEntrenador</td><td>0</td></tr><tr><td>IdSede</td><td>"1"</td></tr><tr><td>Identificacion</td><td>"111"</td></tr><tr><td>Nombre</td><td>"SOFI"</td></tr><tr><td>Telefono</td><td>"3106683378"</td></tr><tr><td>fechaMembre...</td><td>null</td></tr><tr><td>refMembresia</td><td>"General"</td></tr></table>	Cliente	{NET_MVC.Models.C	DiasRestantes	0	FechaNacimie...	{11/01/1999 0:00:00}	Genero	"F"	IdEntrenador	0	IdSede	"1"	Identificacion	"111"	Nombre	"SOFI"	Telefono	"3106683378"	fechaMembre...	null	refMembresia	"General"	+	1. Respuesta de admPersona: false Respuesta de admCliente: false 2.Excepción personalizada de oracle indicando que el cliente existe	<table><tr><td>respuesta</td><td>false</td></tr><tr><td>respuesta2</td><td>false</td></tr></table> <pre>{ "Error en Oracle: ORA-20002: Persona existente\nC</pre>	respuesta	false	respuesta2	false														
Cliente	{NET_MVC.Models.C																																												
DiasRestantes	0																																												
FechaNacimie...	{11/01/1999 0:00:00}																																												
Genero	"F"																																												
IdEntrenador	0																																												
IdSede	"1"																																												
Identificacion	"111"																																												
Nombre	"SOFI"																																												
Telefono	"3106683378"																																												
fechaMembre...	null																																												
refMembresia	"General"																																												
respuesta	false																																												
respuesta2	false																																												
E2-HU 1-CA1- CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)     {         // ...     } }</pre>	<table><tr><td>Nombre:</td><td></td></tr><tr><td>Identificación:</td><td></td></tr><tr><td>Fecha de Contratación:</td><td>mm/dd/yyyy</td></tr><tr><td>Teléfono:</td><td></td></tr><tr><td>Género:</td><td>Seleccione Género</td></tr><tr><td>Área de especialidad:</td><td>Seleccione Especialidad</td></tr><tr><td>Salario:</td><td></td></tr><tr><td>Contraseña:</td><td></td></tr></table>	Nombre:		Identificación:		Fecha de Contratación:	mm/dd/yyyy	Teléfono:		Género:	Seleccione Género	Área de especialidad:	Seleccione Especialidad	Salario:		Contraseña:		+	<div>\$( '#nombre-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#identificacion-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#fechaInicioContrato-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#fechaInicioContrato-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#telefono-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#salario-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#contraseña-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#genero-error' ).text('Campo obligatorio. ');[False]</div> <div>\$( '#especialidad-error' ).text('Campo obligatorio. ');[False]</div>	<table><tr><td>Nombre:</td><td></td><td>Campo obligatorio.</td></tr><tr><td>Identificación:</td><td></td><td>Campo obligatorio.</td></tr><tr><td>Fecha de Contratación:</td><td>mm/dd/yyyy</td><td>Campo obligatorio.</td></tr><tr><td>Teléfono:</td><td></td><td>Campo obligatorio.</td></tr><tr><td>Género:</td><td>Seleccione Género</td><td>Campo obligatorio.</td></tr><tr><td>Área de especialidad:</td><td>Seleccione Especialidad</td><td>Campo obligatorio.</td></tr><tr><td>Salario:</td><td></td><td>Campo obligatorio.</td></tr><tr><td>Contraseña:</td><td></td><td>Campo obligatorio.</td></tr></table>	Nombre:		Campo obligatorio.	Identificación:		Campo obligatorio.	Fecha de Contratación:	mm/dd/yyyy	Campo obligatorio.	Teléfono:		Campo obligatorio.	Género:	Seleccione Género	Campo obligatorio.	Área de especialidad:	Seleccione Especialidad	Campo obligatorio.	Salario:		Campo obligatorio.	Contraseña:		Campo obligatorio.
Nombre:																																													
Identificación:																																													
Fecha de Contratación:	mm/dd/yyyy																																												
Teléfono:																																													
Género:	Seleccione Género																																												
Área de especialidad:	Seleccione Especialidad																																												
Salario:																																													
Contraseña:																																													
Nombre:		Campo obligatorio.																																											
Identificación:		Campo obligatorio.																																											
Fecha de Contratación:	mm/dd/yyyy	Campo obligatorio.																																											
Teléfono:		Campo obligatorio.																																											
Género:	Seleccione Género	Campo obligatorio.																																											
Área de especialidad:	Seleccione Especialidad	Campo obligatorio.																																											
Salario:		Campo obligatorio.																																											
Contraseña:		Campo obligatorio.																																											

				('Campo obligatorio.');	
E1-HU 1-CA3-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] // references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		+	\$('#nombre-error').text('Campo obligatorio.');	
E2-HU 1-CA1 0 & E2-HU 1-CA1 2-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] // references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		+	\$('#nombre-error').text('El nombre no puede contener caracteres especiales ni números.');	
E2-HU 1-CA1 1-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] // references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		+	\$('#nombre-error').text('No puede exceder los 50 caracteres.');	
E2-HU 1-CA1 3-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] // references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		+	\$('#identificacion-error').text('No puede tener más de 10 dígitos.');	

E2-HU1-CA14-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	<p>\$('#telefono-error').text('Debe ser un número.');</p> <p>[FALSE]</p>	
E2-HU1-CA15-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	<p>\$('#telefono-error').text('Debe tener exactamente 10 dígitos.');</p> <p>[FALSE]</p>	
E2-HU1-CA16-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	<p>\$('#telefono-error').text('El número de teléfono debe comenzar con 3.');</p> <p>[FALSE]</p>	
E2-HU1-CA17-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	<p>\$('#salario-error').text('Debe ser un número.');</p> <p>[FALSE]</p>	

E2-HU1-CA18-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	$\$(\text{'#salario-error'}).text(\text{'El salario no puede superar los 10,000,000.'});$ [FALSE]	
E2-HU1-CA19-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	$\$(\text{'#fechaInicioContrato-error'}).text(\text{'Fecha inválida. No puede ser más de una semana.'});$ [FALSE]	
E2-HU1-CA20-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	$\$(\text{'#fechaInicioContrato-error'}).text(\text{'Fecha inválida. No puede ser una fecha pasada.'});$ [FALSE]	
E2-HU1-CA21-CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;      if (ModelState.IsValid)</pre>		(+)	$\$(\text{'#entrenador-existe'}).text(\text{'Entrenador existente.'});$ [FALSE]	



E2-HU1 -CA22- CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 references public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) {     // Obtener el usuario actual     string idsede = User.FindFirst(ClaimTypes.Name)?.Value;     Entrenador.IdSede = idsede;     if (ModelState.IsValid)</pre>		(+)	Respuesta y Respuesta2 = true	
-------------------------	--	--	-----	-------------------------------	--

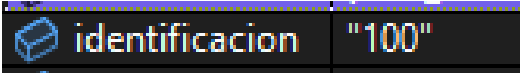
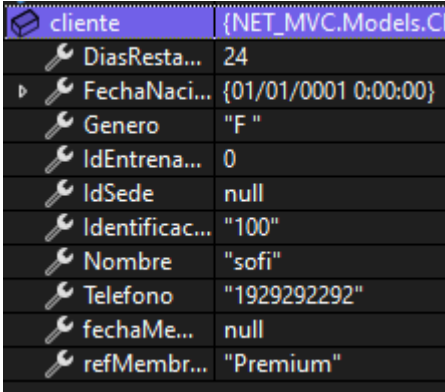
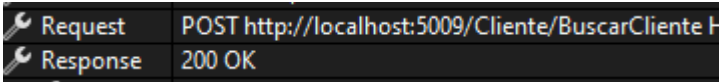
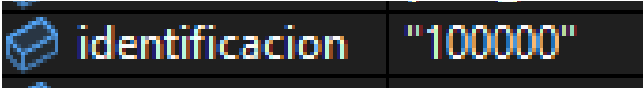
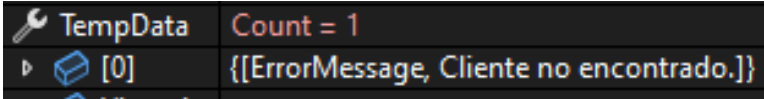
Total de casos: 18  
% de casos aprobados: 100%

SEGUNDO RELEASE

- 1. E1 - HU3: Consultar información de los entrenadores disponibles
- 2. E1 - HU5: Consultar información personal de clientes del gimnasio

ID	Componente	Valores	Tipo (+/-)	Resultado Esperado	Resultado Real
E1-H U5-C P1	<pre>[HttpPost] 0 referencias public IActionResult BuscarCliente(string identificacion) {     // Verificar que la identificación no esté vacía     if (string.IsNullOrEmpty(identificacion))     {         TempData["ErrorMessage"] = "La identificación no puede estar vacía.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página     } }</pre>		+	Mensaje mandado por la TempData que indique que la identificación no puede estar vacía.	

E1-H U5-C P2	<pre>[HttpPost] 0 referencias public IActionResult BuscarCliente(string identificacion) {     // Verificar que la identificación no esté vacía     if (string.IsNullOrEmpty(identificacion))     {         TempData["ErrorMessage"] = "La identificación no puede estar vacía.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página de información del cliente     }      // Verificar que la identificación no tenga más de 10 dígitos     if (identificacion.Length &gt; 10)     {         TempData["ErrorMessage"] = "La identificación no puede tener más de 10 dígitos.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página donde se muestra el formulario     } }</pre>		+	Mensaje mandado por la TempData que indique que la identificación no puede tener más de 10 dígitos.	
E1-H U5-C P3	<pre>[HttpPost] 0 referencias public IActionResult BuscarCliente(string identificacion) {     // Verificar que la identificación no esté vacía     if (string.IsNullOrEmpty(identificacion))     {         TempData["ErrorMessage"] = "La identificación no puede estar vacía.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página de información del cliente     } }</pre>		+	Mensaje mandado por la TempData que indique que la identificación debe ser un número válido.	

E1-H U5-C P4	<pre>[HttpPost] 0 referencias public IActionResult BuscarCliente(string identificacion) {     // Verificar que la identificación no esté vacía     if (string.IsNullOrEmpty(identificacion))     {         TempData["ErrorMessage"] = "La identificación no puede estar vacía.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página de información del cliente     } }</pre>		+	Se trae la información del cliente existente y se redirige al módulo de información cliente específico	 
E1-H U5-C P5	<pre>[HttpPost] 0 referencias public IActionResult BuscarCliente(string identificacion) {     // Verificar que la identificación no esté vacía     if (string.IsNullOrEmpty(identificacion))     {         TempData["ErrorMessage"] = "La identificación no puede estar vacía.";         return RedirectToAction("InformacionCliente"); // Redirigir a la página de información del cliente     } }</pre>		+	Mensaje mandado por la TempData que indique que el cliente no fue encontrado.	
			+		
			+		
			+		
			+		



			+		
--	--	--	---	--	--

Total de casos: 5  
% de casos aprobados: 100%