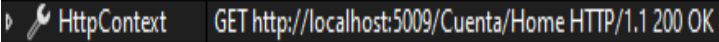

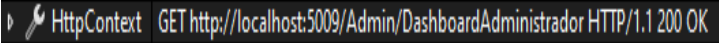
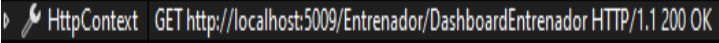


PRIMER RELEASE (SPRINT 1 Y 2)

- 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	Criticidad	Resultado Real								
E4-H U1-C A3-C P1	<pre>public async Task<IActionResult> Login(LoginModel model) { if (ModelState.IsValid) { if (!EsUsuarioValido(model.Usuario, model.Contraseña, out string rol) { ViewBag.MensajeError = "Identificación o contraseña erróneos"; return View(model); } var claims = new List<Claim> { new Claim(ClaimTypes.Name, model.Usuario), // Nombre del usuario new Claim(ClaimTypes.Role, rol), // Rol del usuario (rol) }; } }</pre>	<table><tr><td>Contraseña</td><td>"password123"</td></tr><tr><td>EsAdministrador</td><td>false</td></tr><tr><td>EsEntrenador</td><td>false</td></tr><tr><td>Usuario</td><td>"1"</td></tr></table>	Contraseña	"password123"	EsAdministrador	false	EsEntrenador	false	Usuario	"1"	+	Asignación de rol Administrador y redireccionamiento al dashboard de administrador, ya que este usuario existe en la base de datos	Alta	<div>rol"Administrador"</div> <div>ControllerContext → NET_MVC.Controllers.AdminController.DashboardAdministrador (NET_MVC)</div> <div>HttpContext GET http://localhost:5009/Admin/DashboardAdministrador HTTP/1.1 200 OK</div>
Contraseña	"password123"													
EsAdministrador	false													
EsEntrenador	false													
Usuario	"1"													
E4-H U1-C A4-C P1	<pre>public async Task<IActionResult> Login(LoginModel model) { if (ModelState.IsValid) { if (!EsUsuarioValido(model.Usuario, model.Contraseña, out string rol) { ViewBag.MensajeError = "Identificación o contraseña erróneos"; return View(model); } var claims = new List<Claim> { new Claim(ClaimTypes.Name, model.Usuario), // Nombre del usuario new Claim(ClaimTypes.Role, rol), // Rol del usuario (rol) }; } }</pre>	<table><tr><td>Contraseña</td><td>"entrenador123"</td></tr><tr><td>EsAdministrador</td><td>false</td></tr><tr><td>EsEntrenador</td><td>false</td></tr><tr><td>Usuario</td><td>"1"</td></tr></table>	Contraseña	"entrenador123"	EsAdministrador	false	EsEntrenador	false	Usuario	"1"	+	Asignación de rol Entrenador y redireccionamiento al dashboard del entrenador, ya que existe en la base de datos	Alta	<div>HttpContext GET http://localhost:5009/Entrenador/DashboardEntrenador HTTP/1.1 200 OK</div> <div>ModelState → Entries = 0, IsValid = true</div>
Contraseña	"entrenador123"													
EsAdministrador	false													
EsEntrenador	false													
Usuario	"1"													
E4-H U1-C A6-C P1	<pre>public async Task<IActionResult> Logout() { await HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme); return RedirectToAction("Login", "Cuenta"); }</pre>		+	Cambiar la autenticidad de la cuenta, colocando que no está autenticado y redirigiendo a el apartado de login	Alta	<div>Context GET http://localhost:5009/Cuenta/Login HTTP/1.1 200 OK</div> <div>UserIsAuthenticated = false</div>								

E4-H U1-C A7-C P1	<pre>public IActionResult Home() { return RedirectToAction("DashboardAdministrador", "Admin") }</pre>		+	cambia el url de contexto y lo redirige a el dashboard del admin cuando es llamado con el botón	Alta	
E4-H U1-C A9-C P1	<pre>public async Task<IActionResult> Login(LoginModel model) { if (ModelState.IsValid) { if (!EsUsuarioValido(model.Usuario, model.Contraseña, out string rol)) { ViewBag.MensajeError = "Identificación o contraseña erróneos"; return View(model); } var claims = new List<Claim> { new Claim(ClaimTypes.Name, model.Usuario), // Nombre del usuario new Claim(ClaimTypes.Role, rol), // Rol del usuario (Admin o Entrenador) }; } }</pre>	http://localhost:5009/Ciente/Registrar	+	Como se está intentando acceder a una ventana de admin sin iniciar sesión como admin, nunca se habrá ejecutado el método y por lo tanto no estará autenticado	Alta	
E4-H U1-C A11-C P1	<pre>public IActionResult Login() { var rolClaim = User.FindFirst(ClaimTypes.Role)?.Value; if (User.Identity.IsAuthenticated && rolClaim == "Administrador") { return RedirectToAction("DashboardAdministrador", "Admin"); } if (User.Identity.IsAuthenticated && rolClaim == "Entrenador") { return RedirectToAction("DashboardEntrenador", "Entrenador"); } return View(); // Devuelve la vista Login.cshtml }</pre>	ClaimTypes.Role = “Administrador”; click en página anterior del navegador	+	lo redirige a la página principal del administrador	Alta	
E4-H U1-C A11-C P2	<pre>public IActionResult Login() { var rolClaim = User.FindFirst(ClaimTypes.Role)?.Value; if (User.Identity.IsAuthenticated && rolClaim == "Administrador") { return RedirectToAction("DashboardAdministrador", "Admin"); } if (User.Identity.IsAuthenticated && rolClaim == "Entrenador") { return RedirectToAction("DashboardEntrenador", "Entrenador"); } return View(); // Devuelve la vista Login.cshtml }</pre>	ClaimTypes.Role = “Entrenador”; click en página anterior del navegador	+	lo redirige a la página principal del administrador	Alta	

E1-H U1-C A18- CP1	<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.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	Alta	<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																															
E1-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") { // ... } } } }</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	Alta	<table><tr><td>respuesta</td><td>false</td></tr><tr><td>respuesta2</td><td>false</td></tr></table> <div>["Error en Oracle: ORA-20002: Persona existente\n"]</div>	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																															

E1-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.CI</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.CI	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	Alta	<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.CI																															
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																															
E1-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.CI</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.CI	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	Alta	<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.CI																															
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																															


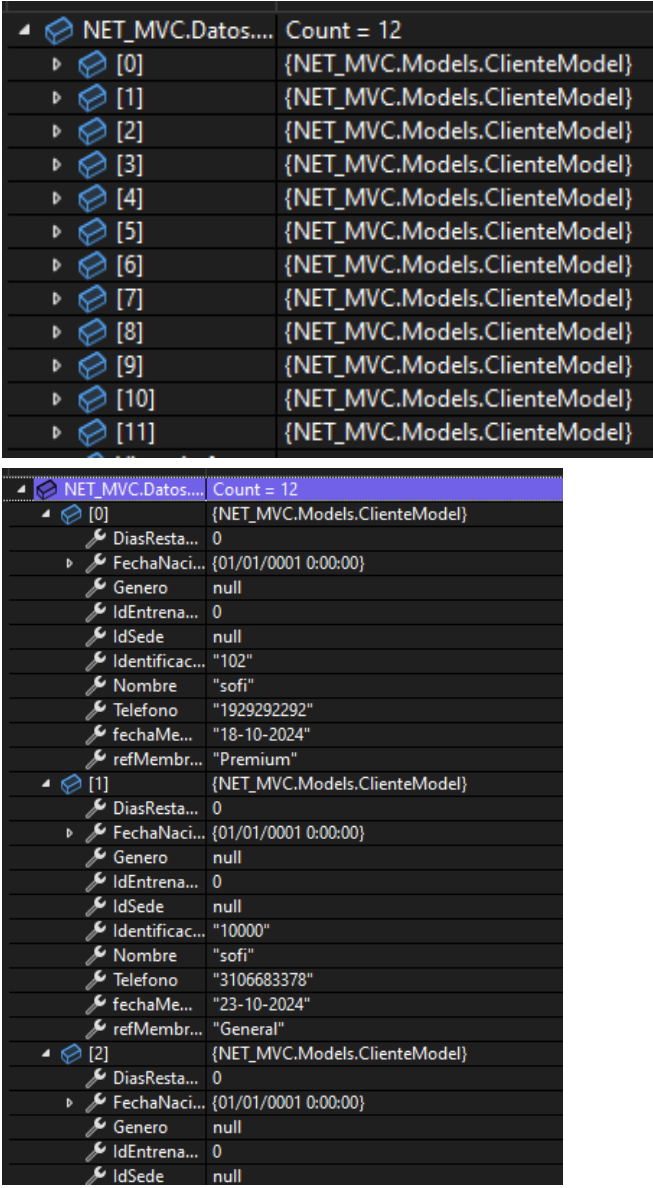
E1-H U2-C A1-C P1	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString()); if (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium") { // ... } } } }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.Cli</td></tr><tr><td>DiasRestantes</td><td>0</td></tr><tr><td>FechaNacimiento</td><td>{11/01/2006 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>"51"</td></tr><tr><td>Nombre</td><td>"Sofia"</td></tr><tr><td>Telefono</td><td>"3203020302"</td></tr><tr><td>fechaMembresia</td><td>null</td></tr><tr><td>refMembresia</td><td>"General"</td></tr></table> <table><tr><td>refMembresia</td><td>"General"</td></tr></table>	Cliente	{NET_MVC.Models.Cli	DiasRestantes	0	FechaNacimiento	{11/01/2006 0:00:00}	Genero	"F"	IdEntrenador	0	IdSede	"1"	Identificacion	"51"	Nombre	"Sofia"	Telefono	"3203020302"	fechaMembresia	null	refMembresia	"General"	refMembresia	"General"	+	El sistema deberá guardar el registro del cliente, dirigirá a la página de inicio, eliminar la información ingresada y mostrar el mensaje "Cliente registrado correctamente".	Alta	<table><tr><td>TempData</td><td>Count = 1</td></tr><tr><td>[0]</td><td>{[SuccessMessage, Cliente registrado correctamente]}</td></tr><tr><td>Vista sin form...</td><td></td></tr></table> <table><tr><td>Microsoft.AspNetCo...</td><td>"/Admin/DashboardAdministrador"</td></tr></table>	TempData	Count = 1	[0]	{[SuccessMessage, Cliente registrado correctamente]}	Vista sin form...		Microsoft.AspNetCo...	"/Admin/DashboardAdministrador"				
Cliente	{NET_MVC.Models.Cli																																									
DiasRestantes	0																																									
FechaNacimiento	{11/01/2006 0:00:00}																																									
Genero	"F"																																									
IdEntrenador	0																																									
IdSede	"1"																																									
Identificacion	"51"																																									
Nombre	"Sofia"																																									
Telefono	"3203020302"																																									
fechaMembresia	null																																									
refMembresia	"General"																																									
refMembresia	"General"																																									
TempData	Count = 1																																									
[0]	{[SuccessMessage, Cliente registrado correctamente]}																																									
Vista sin form...																																										
Microsoft.AspNetCo...	"/Admin/DashboardAdministrador"																																									
E1-H U2-C A2-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") { // ... } } } }</pre>	<table><tr><td>Cliente</td><td>{NET_MVC.Models.Cli</td></tr><tr><td>DiasRestantes</td><td>0</td></tr><tr><td>FechaNacimiento</td><td>{11/01/2006 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>"53"</td></tr><tr><td>Nombre</td><td>"sofia"</td></tr><tr><td>Telefono</td><td>"3203020302"</td></tr><tr><td>fechaMembresia</td><td>null</td></tr><tr><td>refMembresia</td><td>"Premium"</td></tr></table> <table><tr><td>refMembresia</td><td>"Premium"</td></tr></table>	Cliente	{NET_MVC.Models.Cli	DiasRestantes	0	FechaNacimiento	{11/01/2006 0:00:00}	Genero	"F"	IdEntrenador	0	IdSede	"1"	Identificacion	"53"	Nombre	"sofia"	Telefono	"3203020302"	fechaMembresia	null	refMembresia	"Premium"	refMembresia	"Premium"	+	Entonces el sistema deberá mantener la información ingresada y pasar a el módulo asignar entrenador a cliente.	Alta	<table><tr><td>Newtonsoft.Json.JsonConvert.SerializeObject devuelto</td><td></td></tr></table> <table><tr><td>Expresión:</td><td>\$ReturnValue2</td></tr><tr><td>Manipulación de cadenas:</td><td>Ninguno</td></tr><tr><td>Valor:</td><td>{ "FechaNacimiento": "2006-01-11T00:00:00", "IdEntrenador": 0, "refMembresia": "Premium", "fechaMembresia": null, "DiasRestantes": 0, "IdSede": "1", "Nombre": "sofia", "Identificacion": "53", "Telefono": "3203020302" }</td></tr></table> <table><tr><td>Microsoft.AspNetCore.Mvc.UrlHelperExtensions.Action devuelto</td><td></td></tr><tr><td>devuelto</td><td>"/Cliente/AsignarEntrenadorCliente"</td></tr></table>	Newtonsoft.Json.JsonConvert.SerializeObject devuelto		Expresión:	\$ReturnValue2	Manipulación de cadenas:	Ninguno	Valor:	{ "FechaNacimiento": "2006-01-11T00:00:00", "IdEntrenador": 0, "refMembresia": "Premium", "fechaMembresia": null, "DiasRestantes": 0, "IdSede": "1", "Nombre": "sofia", "Identificacion": "53", "Telefono": "3203020302" }	Microsoft.AspNetCore.Mvc.UrlHelperExtensions.Action devuelto		devuelto	"/Cliente/AsignarEntrenadorCliente"
Cliente	{NET_MVC.Models.Cli																																									
DiasRestantes	0																																									
FechaNacimiento	{11/01/2006 0:00:00}																																									
Genero	"F"																																									
IdEntrenador	0																																									
IdSede	"1"																																									
Identificacion	"53"																																									
Nombre	"sofia"																																									
Telefono	"3203020302"																																									
fechaMembresia	null																																									
refMembresia	"Premium"																																									
refMembresia	"Premium"																																									
Newtonsoft.Json.JsonConvert.SerializeObject devuelto																																										
Expresión:	\$ReturnValue2																																									
Manipulación de cadenas:	Ninguno																																									
Valor:	{ "FechaNacimiento": "2006-01-11T00:00:00", "IdEntrenador": 0, "refMembresia": "Premium", "fechaMembresia": null, "DiasRestantes": 0, "IdSede": "1", "Nombre": "sofia", "Identificacion": "53", "Telefono": "3203020302" }																																									
Microsoft.AspNetCore.Mvc.UrlHelperExtensions.Action devuelto																																										
devuelto	"/Cliente/AsignarEntrenadorCliente"																																									

E2-H U1-C A22	<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) { try { var respuesta = consulta.RegistrarPersona(Entrenador); var respuesta2 = consultaEntrenador.RegistrarEntrenador(Entrenador); if (respuesta && respuesta2) { TempData["SuccessMessage"] = "Entrenador registrado correctamente"; // Si se necesita redirigir a otra vista, como lo haría el controlador return Json(new { success = true, redirectUrl = "/Inicio" }); } } catch { } } }</pre>	<table><tr><td>Entrenador</td><td>{NET_MVC.Models.EntrenadorModel}</td></tr><tr><td>ClientesAsignados</td><td>0</td></tr><tr><td>Contraseña</td><td>"1234566"</td></tr><tr><td>Especialidad</td><td>"Fuerza"</td></tr><tr><td>Genero</td><td>"F"</td></tr><tr><td>IdSede</td><td>null</td></tr><tr><td>Identificacion</td><td>"7000"</td></tr><tr><td>Nombre</td><td>"Sofia Arango"</td></tr><tr><td>Salario</td><td>"10000"</td></tr><tr><td>Telefono</td><td>"3203020302"</td></tr><tr><td>fechaInicioContrato</td><td>{27/10/2024 0:00:00}</td></tr></table>	Entrenador	{NET_MVC.Models.EntrenadorModel}	ClientesAsignados	0	Contraseña	"1234566"	Especialidad	"Fuerza"	Genero	"F"	IdSede	null	Identificacion	"7000"	Nombre	"Sofia Arango"	Salario	"10000"	Telefono	"3203020302"	fechaInicioContrato	{27/10/2024 0:00:00}	+	El sistema deberá guardar el registro del entrenador, dirigirá a la página de inicio, limpiar la información y mostrar el mensaje "Entrenador registrado correctamente"	Alta	<table><tr><td>TempData</td><td>Count = 1</td></tr><tr><td>[0]</td><td>{{SuccessMessage, Entrenador registrado correctamente}}</td></tr></table>	TempData	Count = 1	[0]	{{SuccessMessage, Entrenador registrado correctamente}}
Entrenador	{NET_MVC.Models.EntrenadorModel}																															
ClientesAsignados	0																															
Contraseña	"1234566"																															
Especialidad	"Fuerza"																															
Genero	"F"																															
IdSede	null																															
Identificacion	"7000"																															
Nombre	"Sofia Arango"																															
Salario	"10000"																															
Telefono	"3203020302"																															
fechaInicioContrato	{27/10/2024 0:00:00}																															
TempData	Count = 1																															
[0]	{{SuccessMessage, Entrenador registrado correctamente}}																															


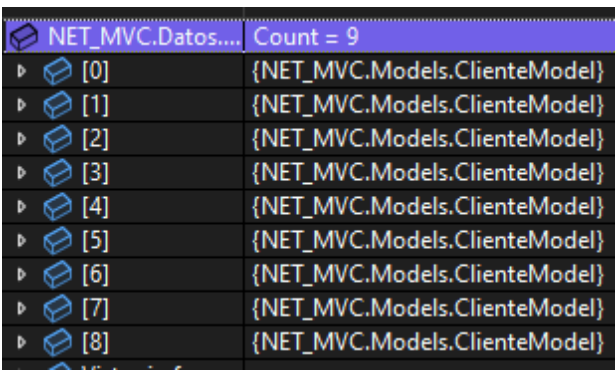
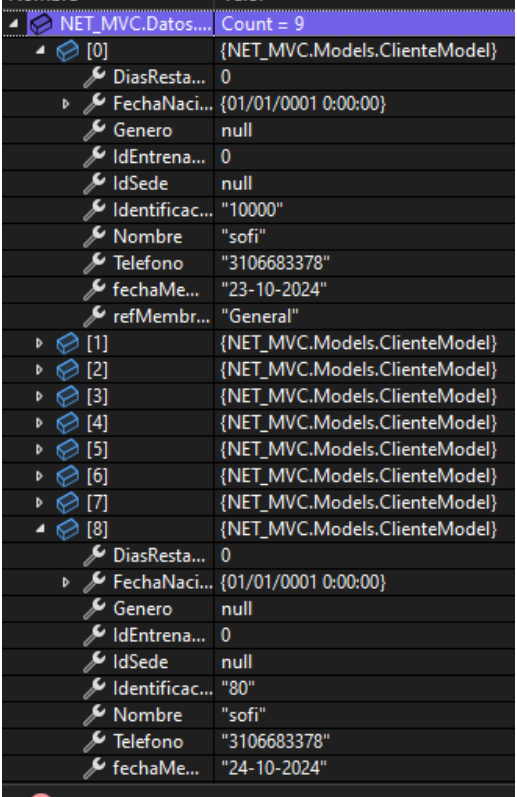
Total de casos: 14
% de casos aprobados: 100%

SEGUNDO RELEASE (SPRINT 3 Y 4)

- 1. E1 - HU3: Consultar información de los entrenadores disponibles
- 2. E1 - HU5: Consultar información personal de clientes del gimnasio
- 3. E1 - HU6: Consultar información personal de un cliente del gimnasio

ID	Componente	Valores	Tipo (+/-)	Resultado Esperado	Criticidad	Resultado Real
E1-H U5-C A1-C P1	<pre>[HttpPost] 0 referencias public JsonResult Filtrar(string filter) { List<ClienteModel> clientesFiltrados = new List<ClienteModel>(); String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // La consulta SQL se basa en el filtro seleccionado string sql = "SELECT c.id_cliente, MAX(p.nombre_persona) AS 'FROM cliente c INNER JOIN persona p ON c.id_cliente = p.id_cliente 'INNER JOIN membresia m ON c.id_cliente = m.id_cliente 'WHERE p.id_sede = " + IdSede + " 'GROUP BY c.id_cliente"; switch (filter) { case "all": break; case "premium": sql = cadenasql2(1, IdSede); // Clientes premium break; case "general": sql = cadenasql2(2, IdSede); // Clientes generales break; case "masculino": break; } }</pre>		+	Tabla donde se muestran todos los clientes registrados.	Alta	

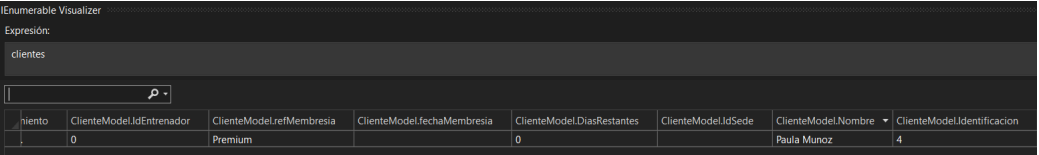
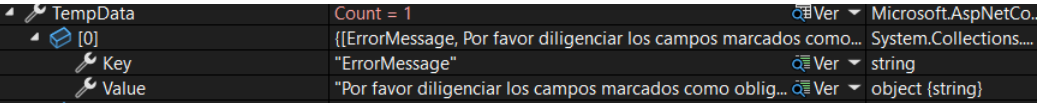
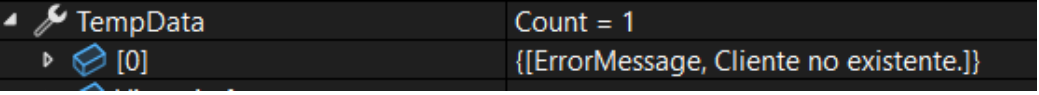
						<div><div>[10]</div><div>{NET_MVC.Models.ClienteModel}</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div><div><div>Identificac...</div><div>"30"</div></div><div><div>Nombre</div><div>"sofi"</div></div><div><div>Telefono</div><div>"3106683378"</div></div><div><div>fechaMe...</div><div>"24-10-2024"</div></div><div><div>refMembr...</div><div>"General"</div></div></div> <div><div>[11]</div><div>{NET_MVC.Models.ClienteModel}</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div><div><div>Identificac...</div><div>"80"</div></div></div>
E1-H U5-C A2-C P1	<div><div>[HttpPost]</div><div>0 referencias</div><div>public JsonResult Filtrar(string filter)</div><div>{</div><div><div>List<ClienteModel> clientesFiltrados = new List<ClienteModel></div><div>String IdSede = User.FindFirst(ClaimTypes.Name)?.Value;</div><div><div>// La consulta SQL se basa en el filtro seleccionado</div><div>string sql = "SELECT c.id_cliente, MAX(p.nombre_persona) AS</div><div>"FROM cliente c INNER JOIN persona p ON c.id_client</div><div>"INNER JOIN membresia m ON c.id_cliente = m.id_clie</div><div>"WHERE p.id_sede = " + IdSede +</div><div>"GROUP BY c.id_cliente";</div></div><div><div>switch (filter)</div><div>{</div><div><div>case "all":</div><div>break;</div><div>case "premium":</div><div>sql = cadenasql2(1, IdSede); // Clientes premium</div><div>break;</div><div>case "general":</div><div>sql = cadenasql2(2, IdSede); // Clientes generales</div><div>break;</div><div>case "masculino":</div><div></div></div><div>}</div></div></div></div>	<div><div>filter</div><div>"premium"</div></div>	+	Tabla donde se muestran todos los clientes premium registrados.	Alta	<div><div>NET_MVC.Datos....</div><div>Count = 3</div><div><div>[0]</div><div>{NET_MVC.Models.ClienteModel}</div></div><div><div>[1]</div><div>{NET_MVC.Models.ClienteModel}</div></div><div><div>[2]</div><div>{NET_MVC.Models.ClienteModel}</div></div></div> <div><div>NET_MVC.Datos....</div><div>Count = 3</div><div><div>[0]</div><div>{NET_MVC.Models.ClienteModel}</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div><div><div>Identificac...</div><div>"102"</div></div><div><div>Nombre</div><div>"sofi"</div></div><div><div>Telefono</div><div>"1929292292"</div></div><div><div>fechaMe...</div><div>"18-10-2024"</div></div><div><div>refMembr...</div><div>"Premium"</div></div></div><div><div>[1]</div><div>{NET_MVC.Models.ClienteModel}</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div><div><div>Identificac...</div><div>"100"</div></div><div><div>Nombre</div><div>"sofi"</div></div><div><div>Telefono</div><div>"1929292292"</div></div><div><div>fechaMe...</div><div>"17-10-2024"</div></div><div><div>refMembr...</div><div>"Premium"</div></div></div><div><div>[2]</div><div>{NET_MVC.Models.ClienteModel}</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div></div></div>

E1-H U5-C A3-C P1	<pre>[HttpPost] 0 referencias public JsonResult Filtrar(string filter) { List<ClienteModel> clientesFiltrados = new List<ClienteModel>(); String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // La consulta SQL se basa en el filtro seleccionado string sql = "SELECT c.id_cliente, MAX(p.nombre_persona) AS "FROM cliente c INNER JOIN persona p ON c.id_cliente = p.id_persona "INNER JOIN membresia m ON c.id_cliente = m.id_cliente "WHERE p.id_sede = " + IdSede + " "GROUP BY c.id_cliente"; switch (filter) { case "all": break; case "premium": sql = cadenasql2(1, IdSede); // Clientes premium break; case "general": sql = cadenasql2(2, IdSede); // Clientes generales break; case "masculino": break; } }</pre>		+	Tabla donde se muestran todos los clientes generales registrados.	Alta	 
----------------------------	--	---	---	---	------	---

E1-H U5-C A4-C P1	<pre>[HttpPost] 0 referencias public JsonResult Filtrar(string filter) { List<ClienteModel> clientesFiltrados = new List<ClienteModel>(); String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // La consulta SQL se basa en el filtro seleccionado string sql = "SELECT c.id_cliente, MAX(p.nombre_persona) AS "FROM cliente c INNER JOIN persona p ON c.id_cliente = p.id_persona "INNER JOIN membresia m ON c.id_cliente = m.id_cliente "WHERE p.id_sede = " + IdSede + " "GROUP BY c.id_cliente"; switch (filter) { case "all": break; case "premium": sql = cadenasql2(1, IdSede); // Clientes premium break; case "general": sql = cadenasql2(2, IdSede); // Clientes generales break; case "masculino": break; } }</pre>	<div>filter</div> <div>"masculino"</div>	+	Tabla donde se muestran todos los clientes con género masculino registrados.	Alta	<div>NET_MVC.Datos... Count = 4</div> <div><div>[0]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[1]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[2]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[3]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div>NET_MVC.Datos... Count = 4</div> <div><div>[0]</div><div><div>DiasResta...</div><div>0</div></div><div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div><div><div>Genero</div><div>null</div></div><div><div>IdEntrena...</div><div>0</div></div><div><div>IdSede</div><div>null</div></div><div><div>Identificac...</div><div>"80"</div></div><div><div>Nombre</div><div>"sofi"</div></div><div><div>Telefono</div><div>"3106683378"</div></div><div><div>fechaMe...</div><div>"24-10-2024"</div></div><div><div>refMembr...</div><div>"General"</div></div></div> <div>[1]</div> <div>{NET_MVC.Models.ClienteModel}</div> <div><div>[2]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[3]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>DiasResta...</div><div>0</div></div> <div><div>FechaNaci...</div><div>{01/01/0001 0:00:00}</div></div> <div><div>Genero</div><div>null</div></div> <div><div>IdEntrena...</div><div>0</div></div> <div><div>IdSede</div><div>null</div></div> <div><div>Identificac...</div><div>"20"</div></div> <div><div>Nombre</div><div>"sofi"</div></div> <div><div>Telefono</div><div>"3106683378"</div></div> <div><div>fechaMe...</div><div>"24-10-2024"</div></div> <div><div>refMembr...</div><div>"General"</div></div>
E1-H U5-C A5-C P1	<pre>[HttpPost] 0 referencias public JsonResult Filtrar(string filter) { List<ClienteModel> clientesFiltrados = new List<ClienteModel>(); String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // La consulta SQL se basa en el filtro seleccionado string sql = "SELECT c.id_cliente, MAX(p.nombre_persona) AS "FROM cliente c INNER JOIN persona p ON c.id_cliente = p.id_persona "INNER JOIN membresia m ON c.id_cliente = m.id_cliente "WHERE p.id_sede = " + IdSede + " "GROUP BY c.id_cliente"; switch (filter) { case "all": break; case "premium": sql = cadenasql2(1, IdSede); // Clientes premium break; case "general": sql = cadenasql2(2, IdSede); // Clientes generales break; case "masculino": break; } }</pre>	<div>filter</div> <div>"femenino"</div>	+	Tabla donde se muestran todos los clientes con género femenino registrados.	Alta	<div>NET_MVC.Datos... Count = 7</div> <div><div>[0]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[1]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[2]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[3]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[4]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[5]</div><div>{NET_MVC.Models.ClienteModel}</div></div> <div><div>[6]</div><div>{NET_MVC.Models.ClienteModel}</div></div>

Total de casos: 12
% de casos aprobados: 100%

- TERCER RELEASE (SPRINT 5 Y 6)**
- 1. **E1-HU4: Asignar entrenador**
 - 2. **E2 - HU2: Consultar información personal de entrenadores**
 - 3. **E2 - HU3: Consultar información personal de un entrenador del gimnasio**
 - 4. **E3 - HU1: Consultar la lista de clientes premium asignados por entrenador**
 - 5. **E3 - HU4: Agregar ejercicios**

E3-H U1-C A1-C P1	<pre>[Authorize(Roles = "Entrenador")] 1 referencia private List<ClienteModel> ObtenerClientesAsignados() { String IdEntrenador = User.FindFirst(ClaimTypes.Name)?.Value; List<ClienteModel> clientes = consultaCliente.ListarClientesAsignados(IdEntrenador); if (clientes.Count() > 0) { return clientes; } else { // Mensaje no hay entrenadores disponibles return new List<ClienteModel> { }; } }</pre>	Id_Entrenador que inicio sesión: 1 Click : Clientes Asignados	+	Obtener Lista de Clientes (List<CLie nteModel>) que le correspond an al entrenador	Media	
E3-H U1-C A2-C P1	<pre>[HttpPost] [Authorize(Roles = "Administrador, Entrenador")] 0 referencias public IActionResult BuscarCliente(string identificacion) { String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // Verificar que la identificación no esté vacía if (string.IsNullOrEmpty(identificacion)) { if (User.IsInRole("Administrador")) { TempData["ErrorMessage"] = "La identificación no puede estar vacía."; } else if (User.IsInRole("Entrenador")) { TempData["ErrorMessage"] = "Por favor diligenciar los campos marcados como obligatorios."; } return RedirectToAction("InformacionCliente"); // Redirigir a la página donde se muestra el formulario } }</pre>	Identificación =	+	Almacena en TempData un mensaje de error que dice “Por favor diligenciar los campos obligatorio s” para ser mostrado	Media	
E3-H U1-C A3-C P1	<pre>[HttpPost] [Authorize(Roles = "Administrador, Entrenador")] 0 referencias public IActionResult BuscarCliente(string identificacion) { String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // Verificar que la identificación no esté vacía if (string.IsNullOrEmpty(identificacion)) { if (User.IsInRole("Administrador")) { TempData["ErrorMessage"] = "La identificación no puede estar vacía."; } else if (User.IsInRole("Entrenador")) { TempData["ErrorMessage"] = "Por favor diligenciar los campos marcados como obligatorios."; } return RedirectToAction("InformacionCliente"); // Redirigir a la página donde se muestra el formulario } }</pre>	Identificación =666	+	Se almacena en TempData el mensaje “Cliente no existente”	Media	
E3-H U1-C A4-C P1	<pre>[HttpPost] [Authorize(Roles = "Administrador, Entrenador")] 0 referencias public IActionResult BuscarCliente(string identificacion) { String IdSede = User.FindFirst(ClaimTypes.Name)?.Value; // Verificar que la identificación no esté vacía if (string.IsNullOrEmpty(identificacion)) { if (User.IsInRole("Administrador")) { TempData["ErrorMessage"] = "La identificación no puede estar vacía."; } else if (User.IsInRole("Entrenador")) { TempData["ErrorMessage"] = "Por favor diligenciar los campos marcados como obligatorios."; } return RedirectToAction("InformacionCliente"); // Redirigir a la página donde se muestra el formulario } }</pre>	Identificación = 4	+	Se redirige a la vista Informacio nClienteAs ignado con el modelo del cliene ClienteMo del	Media	