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	Crticidad	Resultado Real					
E4-H U1-C A1-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 (}; }</claim></iactionresult></pre>	Identificación = Contraseña =	+	guardar mensaje de error en ViewBag para ser mostrado	Alta	<pre>▼ <div claims="new" class="text-center error-message" contraseña="" erróneos";="" identificación="" list<claim="" o="" return="" style="visibility: visible;</td></tr><tr><td>E4-H
U1-C
A2-C
P1</td><td><pre>public async Task<IActionResult> Login(LoginModel model) { if (ModelState.IsValid) { if (!EsUsuarioValido(model Usuario, model.Contraseña, out string rol { ViewBag.MensajeError = " var="" view(model);="" }=""> { new Claim(ClaimTypes.Name, model.Usuario), // Nombre del usuario new Claim(ClaimTypes.Role, rol), // Rol del usuario (}; }</div></pre>	Identificación = asdasd Contraseña = asdads	+	guardar mensaje de error de "Contraseñ a o usuario incorrecto" para ser mostrado	Alta	r <div class="text-center error-message" style="visibility: visible; color: red; text-align: start; margin-top: 10px; "> "Identificación o contraseña erróneos " ViewBag.MensajeError "Identificación o contraseña erróneos"</div>
E1-H U1-C A1-C P1	O referencias public JsonResult RegistrarCliente(ClienteModel { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStri if (ModelState.IsValid) {	Nombre = Identificación = Teléfono = Género = fechaNacimiento membresía =	+	Las Variables que almacenan los datos del formulario están vacías y se muestra el mensaje de campo obligatorio en los campos vacíos. Además, el sistema no dejará enviar el formulario	Alta	Fecha de Nacimiento ingresada: Error: Fecha de nacimiento vacía. Nombre ingresado: Error: Campo nombre vacío. Identificación ingresada: Error: Campo identificación vacío. Teléfono ingresado: Error: Campo teléfono vacío. Género seleccionado: Error: Género no seleccionado. Membresía seleccionada: Error: Membresía no seleccionada. Error en el formulario, no enviado.					

				1	1	
				de registro.		
E1-H U1-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.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } <head></head></pre>	membresía = General	+	La variable que almacena la fecha de nacimiento esté vacía y se muestre el mensaje de campo obligatorio en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: Error: Fecha de nacimiento vacía. Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A3-C P1	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Identificación = 1111 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	La variable que almacena el Nombre esté vacía y se muestre el mensaje de campo obligatorio en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Error: Campo nombre vacío. Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

E1-H U1-C A4*C P1	<pre>[HttpPost]</pre>	Nombre = Sofia Arango Identificación = Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	La variable que almacena la Identificaci ón esté vacía y se muestre el mensaje de campo obligatorio en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: Error: Campo identificación vacío. Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A5-C P1	<pre>[HttpPost] O referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtrif } </pre>	Nombre = Sofia Arango Identificación = 1111 Teléfono = Género = F fechaNacimiento = 11/01/2006 membresía = General	+	La variable que almacena el Teléfono esté vacía y se muestre el mensaje de campo obligatorio en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: Error: Campo teléfono vacío. Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

E1-H U1-C A6-C P1	[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel Cliente) [Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStri if (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } <hr/> <head></head>	Nombre = Sofia Arango Identificación = 1111 Teléfono = 3106683378 Género = fechaNacimiento = 11/01/2006 membresía = General	+	La variable que almacena el género esté vacía y se muestre el mensaje de campo obligatorio en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: Error: Género no seleccionado. Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A7-C P1	<pre>[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStri if (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	membresía =	+	La variable que almacena el Membresia esté vacía y se muestre el mensaje de "Campo obligatorio" en el respectivo cambio. Además, el sistema no dejará enviar el formulario de registro.		Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: Error: Membresía no seleccionada. Error en el formulario, no enviado.
E1-H U1-C A8-C P1	<pre>[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStri if (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Sofia Arango Identificación = 1111 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2025 membresía = General	+	Se muestra el mensaje de "No puede ser en el futuro". Además, el sistema no dejará enviar el	Media	Fecha de Nacimiento ingresada: 2025-01-11 Error: Fecha de nacimiento en el futuro. Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

	<pre>@model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } <<head></head></pre>			formulario de registro.		
E1-H U1-C A9-C P1	Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStri	Nombre = Sofia Arango Identificación = 1111 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/1800 membresía = General	+	Se muestra el mensaje de "La edad no puede ser superior a 100 años". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 1800-01-11 Edad calculada: 224 Error: Edad mayor a 100 años. Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A10- CP1		Nombre = %&\$\$#912221 Identificación = 1111 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "El nombre no puede contener caracteres especiales ni números". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: %&\$\$#912221 Error: Nombre contiene caracteres no permitidos. Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

E1-H U1-C A11-C P1	<pre>[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Sofiaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	+	Se muestra el mensaje de "No puede exceder los 50 caracteres" . Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofiasasasasasasasasasasasasasasasasasasas
E1-H U1-C A12- CP1	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Sofia Arango Identificación = %󞭝 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "Debe ser un número entero". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: %󞭝 Error: Identificación no es numérica. Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A13- CP1	<pre>[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStriif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Sofia Arango Identificación = 1111111111 Teléfono = 3106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "No puede tener más de 10 dígitos". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111111111 Error: Identificación excede 10 dígitos. Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

E1-H U1-C A14- CP1	[HttpPost] Oreferencis public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStrinif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")	Nombre = Sofia Arango Identificación = 1111 Teléfono = 3106683\$\$378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "Debe ser un número". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683\$\$378 Error: Teléfono no es numérico. Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A15- CP1	[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStrinif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } <hr/> <head></head>	Nombre = Sofia Arango Identificación = 1111 Teléfono = 310 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "Debe tener exactamen te 10 dígitos". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 310 Error: Teléfono no tiene 10 dígitos. Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A16- CP1	<pre>[HttpPost] Oreferencias public JsonResult RegistrarCliente(ClienteModel { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToString("ModelState.IsValid)) { try { if (Cliente.refMembresia == "Premium") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } </pre>	Nombre = Sofia Arango Identificación = 1111 Teléfono = 2106683378 Género = F fechaNacimiento = 11/01/2006 membresía = General	+	Se muestra el mensaje de "El número de teléfono debe comenzar con 3". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2006-01-11 Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 2106683378 Error: Teléfono no comienza con 3. Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.

E1-H U1-C A17- CP1	<pre>[HttpPost] O referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStrinif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	membresía = General	+	Se muestra el mensaje de "Cliente existente". Además, el sistema no dejará enviar el formulario de registro.	Media	Identificación ingresada: 111 Identificación es numérica Respuesta de VerificarClienteExistente: ➤ {existe: true} Cliente existente.
E1-H U1-C A19- CP1	<pre>[HttpPost] 0 referencias public JsonResult RegistrarCliente(ClienteModel Cliente) { Cliente.IdSede = User.FindFirst(ClaimTypes.Name)?.Value; HttpContext.Session.SetString("ClienteId", Cliente.Identificacion.ToStrinif (ModelState.IsValid) { try { if (Cliente.refMembresia == "Premium")</pre>	Nombre = Sofia Arango Identificación = 1111 Teléfono = 2106683378 Género = F fechaNacimiento = 11/01/2020 membresía = General	+	Se muestra el mensaje de "Debe tener más de 14 años ". Además, el sistema no dejará enviar el formulario de registro.	Media	Fecha de Nacimiento ingresada: 2020-01-11 Edad calculada: 4 Error: Edad menor a 14 años. Nombre ingresado: Sofia Arango Identificación ingresada: 1111 Teléfono ingresado: 3106683378 Género seleccionado: F Membresía seleccionada: General Error en el formulario, no enviado.
E1-H U1-C A21- 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>	Identificación ingresada: 100000 Teléfono ingresado: 3102929292 Género seleccionado: F Membresía seleccionada: General	+	Al recargar la página quedan vacíos los campos del módulo registrar cliente.	Media	Identificación ingresada: undefined Fecha de Nacimiento ingresada: undefined Nombre ingresado: undefined Teléfono ingresado: undefined Género seleccionado: undefined Membresía seleccionada: undefined

E1-H U1-C A22	<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") @model ClienteModel @{ ViewData["Title"] = "Registrar Cliente"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtm } </pre>	Fecha de Nacimiento ingresada: 2006-01- Edad calculada: 18 Nombre ingresado: Sofia Arango Identificación ingresada: 1111111 Teléfono ingresado: 3202020202 Género seleccionado: F Membresía seleccionada: General Formulario enviado correctamente.	+	Al salir del módulo se borran los campos del módulo registrar cliente.	Media	Identificación ingresada: undefined Fecha de Nacimiento ingresada: undefined Nombre ingresado: undefined Teléfono ingresado: undefined Género seleccionado: undefined Membresía seleccionada: undefined
E2-H U1-C A1-C P1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] O referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenad var respuesta2 = consultaEntrenador.RegistrarEntre if (respuesta && respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }</pre>	Nombre = Identificación = Fecha de Contratación = Teléfono = Género = Área de especialidad = Salario = Contraseña =	+	El sistema muestra el mensaje "Campo obligatorio." en cada uno de los campos no diligenciado s. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Error: Nombre es obligatorio. Valor de Identificación: Error: Identificación es obligatoria. Valor de Fecha de Contrato: Error: Fecha de contrato es obligatoria. Valor de Teléfono: Error: Teléfono es obligatorio. Valor de Salario: Error: Salario es obligatorio. Valor de Contraseña: Error: Contraseña es obligatoria. Valor de Género: Error: Género es obligatorio. Valor de Especialidad: Error: Especialidad es obligatoria. Error: No todos los campos son válidos.

E2-H U1-C A2-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClainTypes Name)? Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenador Var respuesta) @model EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Identificación = 60 Fecha de Contratación = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566 ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Nombre. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Error: Nombre es obligatorio. Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A3-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Orefrencias public JsonResult RegistrarEntrenador(EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel Entrenador.IdSede = User.FindFirst(ClaimTypes Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(EntrenadorModel EntrenadorModel EntrenadorModel EntrenadorModel	+	Entonces el sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Identificació n. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: Error: Identificación es obligatoria. Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A4-C P1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenad var respuesta2 = consultaEntrenador.RegistrarEntre if (respuesta && respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Teléfono. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: Error: Teléfono es obligatorio. Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A5-C P1	<pre>[HttpPost] [Authorize(Roles ="Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // 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.RegistrarEntre if (respuesta && respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Género. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: Error: Género es obligatorio. Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A6-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrena { // 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) emodel EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 3203020302 Género = F Área de especialidad = Salario = 10000 Contraseña = 1234566	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Área de especialida d. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Error: Especialidad es obligatoria. Error: No todos los campos son válidos.
E2-H U1-C A7-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // 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.RegistrarEntre if (respuesta & respuesta2) @model EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 3203020302 Género = F Área de especialidad = Fuerza Salario = Contraseña = 1234566	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Salario. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: Error: Salario es obligatorio. Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A8-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Orferencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)? Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenad var respuesta2 = consultaEntrenador.RegistrarEntre if (respuesta 56 respuesta2) @model EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Fecha de contratación . Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: Error: Fecha de contrato es obligatoria. Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A9-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // 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) emodel EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = (+	El sistema deberá mostrar el mensaje "Campo obligatorio." en el campo Contraseña. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: Error: Contraseña es obligatoria. Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A10- CP1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrena { // 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.RegistrarEntre if (respuesta & respuesta2) @model EntrenadorModel ef ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }	+	El sistema deberá mostrar el mensaje "El nombre no puede contener caracteres especiales ni números." en el campo Nombre. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: %%((121 Error: Nombre contiene caracteres no permitidos. Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A11-C P1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrenadorada aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	+	Entonces el sistema deberá mostrar el mensaje "No puede exceder los 50 caracteres." en el campo Nombre. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofiasasasasasasasasasasasasasasasasasasas

E2-H U1-C A12- CP1	[HttpPost] [Authorize(Roles = "Administrador")] Orderendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrena { // 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.RegistrarEntre if (respuesta 5% respuesta2) @model EntrenadorModel e{ ViewData[*Title*] = "Registrar Entrenador.cshtml*; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml*; }	+	Entonces el sistema deberá mostrar el mensaje "Debe ser un número entero." en el campo Identificació n. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: Sofia Error: Identificación no es numérica. Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A13- CP1	[HttpPost] [Authorize(Roles = "Administrador")] Orferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) {	+	Entonces el sistema deberá mostrar el mensaje "No puede tener más de 10 dígitos." en el campo Identificació n. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 6000000000000000000000000000000000000

E2-H U1-C A14- CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenadorar respuesta2 = consultaEntrenador.RegistrarEntre if (respuesta && respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = Sofia Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "Debe ser un número." en el campo Teléfono. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: Sofia Error: Teléfono no es numérico. Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A15- CP1	<pre>[HttpPost] [Authorize(Roles ="Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenad</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302999 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "Debe tener exactament e 10 dígitos." en el campo Teléfono . Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 60 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302999 Error: Teléfono no tiene 10 dígitos. Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A16- CP1	[HttpPost] [Authorize(Roles = "Administrador")] Orefrencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User FindFirst(ClaimTypes Name)? Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) { try { var respuesta = consulta.RegistrarPersona(Entrenad var respuesta2 = consultaEntrenador.RegistrarEntre if (respuesta && respuesta2) @model EntrenadorModel e{ ViewData["Title"] = "Registrar Entrenador"; Layout = "-/Views/Admin/_LayoutAdministrador.cshtml"; } } Nombre = %%((121 Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "El número de teléfono debe comenzar con 3." en el campo Teléfono. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 70 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 2203020302 Error: 'El número de teléfono debe comenzar con 3 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A17- CP1	[HttpPost] [Authorize(Roles = "Administrador")] Oreferendas 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) emodel EntrenadorNodel et ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = Sofia Contraseña = 1234566 ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; }	+	Entonces el sistema deberá mostrar el mensaje "Debe ser un número." en el campo Salario. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 70 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: Sofia Error: Salario no es numérico. Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A18- CP1	<pre>[Authorize(Roles = "Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede;</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 1000000000000000000000000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "El salario no puede superar los 10,000,000." en el campo Salario. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 70 Valor de Fecha de Contrato: 2024-10-27 Valor de Teléfono: 3203020302 Valor de Salario: 1000000000000000000000000000000000000
E2-H U1-C A19- CP1	<pre>[Authorize(Roles = "Administrador")] 0 referencias public JsonResult RegistrarEntrenador(EntrenadorModel Entrenad { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede;</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/11/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "Fecha inválida. No puede ser más de una semana." en el campo Fecha de contratación . Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 70 Valor de Fecha de Contrato: 2024-11-27 Error: Fecha de contrato es más de una semana en el futuro. Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.

E2-H U1-C A20- CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] O 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 if (respuesta & respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml" }</pre>		+	Entonces el sistema deberá mostrar el mensaje "Fecha inválida. No puede ser una fecha pasada." en el campo Fecha de contratación . Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Nombre: Sofia Arango Valor de Identificación: 70 Valor de Fecha de Contrato: 2002-11-27 Error: Fecha de contrato es pasada. Valor de Teléfono: 3203020302 Valor de Salario: 10000 Valor de Contraseña: 1234566 Valor de Género: F Valor de Especialidad: Fuerza Error: No todos los campos son válidos.
E2-H U1-C A21- CP1	<pre>[HttpPost] [Authorize(Roles = "Administrador")] Oreferencias 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 if (respuesta && respuesta2) @model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } }</pre>	Nombre = Sofia Arango Identificación = 1 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "Entrenador existente." en el campo Identificació n. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Identificación: 1 La identificación es numérica, verificando entrenador Error: Entrenador existente.
E2-H U1-C A21- CP2	<pre>[HttpPost] [Authorize(Roles = "Administrador")] Oreferencias 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); var respuesta & respuesta2)</pre>	Nombre = Sofia Arango Identificación = 5 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	Entonces el sistema deberá mostrar el mensaje "Persona existente." en el campo Identificació n. Además, el sistema no dejará enviar el formulario de registro.	Media	Valor de Identificación: 5 La identificación es numérica, verificando entrenador No es entrenador, verificando si la persona existe Error: Persona existente.

E2-H U1-C A23- CP1	<pre>@model EntrenadorModel @{ ViewData["Title"] = "Registrar Entrenador"; Layout = "~/Views/Admin/_LayoutAdministrador.cshtml"; } [HttpPost] [Authorize(Roles ="Administrador")] Oreferendas public JsonResult RegistrarEntrenador(EntrenadorModel Entrenador) { // Obtener el usuario actual string idsede = User.FindFirst(ClaimTypes.Name)?.Value; Entrenador.IdSede = idsede; if (ModelState.IsValid) {</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	El sistema deberá eliminar la información ingresada,	Ваја	<pre></pre>
E2-H U1-C A24- 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) { try { var respuesta = consulta.RegistrarPersona(Entrenador); var respuesta2 = consultaEntrenador.RegistrarEntrenado if (respuesta && respuesta2)</pre>	Nombre = Sofia Arango Identificación = 60 Fecha de Contratación = 2024/10/27 Teléfono = 3203020302 Género = F Área de especialidad = Fuerza Salario = 10000 Contraseña = 1234566	+	El sistema deberá eliminar la información ingresada.	Baja	<pre> RegistrarEntrena * {Nombre: '', Identificacion: '', fechaInicioContrato: '', Telefono: '', Genero: '', _} \ Contraseña: "" Especialidad: "" Genero: "" Identificacion: "" Nombre: "" Salario: "" Telefono: "" fechaInicioContrato: ""</pre>

Total de casos: 47

% de casos aprobados: 100%