



## HISTORIA DE USUARIO

<b>Nombre de la HU:</b>		Registro básico de pacientes y mascotas
<b>Objetivo de la HU</b>		Como <b>administrador de la clínica</b> , quiero <b>registrar y consultar pacientes desde consola</b> para poder tener un control inicial de la información básica.
TASK1	<b>Instalar el entorno de desarrollo (.NET SDK + editor).</b>	
	<ul style="list-style-type: none"><li>• Descargar e instalar el <b>.NET SDK</b> compatible con la versión de C#.</li><li>• Configurar el editor recomendado (Visual Studio o VS Code).</li><li>• Crear un primer programa de consola para verificar que el entorno funciona correctamente (Console.WriteLine("Hola Clínica Salud+")).</li></ul>	
TASK2	<b>Crear el proyecto base de consola en C#.</b>	
	<ul style="list-style-type: none"><li>• Iniciar un nuevo proyecto con dotnet new console -n ClinicaSalud.</li><li>• Organizar la estructura de carpetas para futuros archivos (ej. Models para clases).</li><li>• Configurar el Program.cs como punto de entrada.</li></ul>	
TASK3	<b>Definir la clase Paciente</b>	
	<ul style="list-style-type: none"><li>• Crear un archivo Paciente.cs dentro de la carpeta Models.</li><li>• Incluir propiedades básicas: Id (int), Nombre (string), Edad (int), Sintoma (string).</li><li>• Usar propiedades automáticas (public string Nombre { get; set; }).</li><li>• Probar la creación de objetos Paciente en el Program.cs.</li></ul>	
TASK4	<b>Diseñar el menú principal en consola.</b>	
	<ul style="list-style-type: none"><li>• Mostrar opciones en pantalla (ej. 1. Registrar paciente, 2. Listar pacientes, 3. Buscar paciente, 4. Salir).</li><li>• Usar un bucle while para que el menú se repita hasta que el usuario elija salir.</li><li>• Implementar la navegación entre opciones con switch-case.</li></ul>	

TASK 5	<b>Desarrollar métodos para gestionar pacientes</b>
	<ul style="list-style-type: none"> <li>• RegistrarPaciente(List&lt;Paciente&gt; lista): Solicitar datos por consola y agregar un nuevo paciente a la lista.</li> <li>• ListarPacientes(List&lt;Paciente&gt; lista): Recorrer la lista e imprimir cada paciente en pantalla.</li> <li>• BuscarPacientePorNombre(List&lt;Paciente&gt; lista, string nombre): Encontrar un paciente que coincida con el nombre ingresado y mostrarlo en consola.</li> <li>• Colocar estos métodos en una clase auxiliar llamada PacienteService para empezar a separar responsabilidades.</li> </ul>
TASK 6	<b>Almacenar pacientes en una colección.</b>
	<ul style="list-style-type: none"> <li>• Declarar una lista en Program.cs: List&lt;Paciente&gt; pacientes = new List&lt;Paciente&gt;();.</li> <li>• Usar esta lista para pasarla como parámetro a los métodos (RegistrarPaciente, ListarPacientes, etc.).</li> </ul>
TASK 7	<b>Implementar manejo básico de errores con try-catch.</b>
	<ul style="list-style-type: none"> <li>• Validar que la edad ingresada sea un número entero.</li> <li>• Capturar excepciones al convertir datos (int.Parse).</li> <li>• Mostrar mensajes amigables al usuario si la entrada es incorrecta.</li> </ul>
<b>Criterios de aceptación.</b> <ul style="list-style-type: none"> <li>• El programa compila y se ejecuta sin errores.</li> <li>• El usuario puede ingresar datos del paciente y de la mascota a través de la consola.</li> <li>• La información ingresada se valida correctamente (no se permiten campos vacíos ni valores inválidos).</li> <li>• Los datos quedan almacenados en variables o colecciones.</li> <li>• El sistema muestra un resumen claro con la información registrada.</li> <li>• La lógica está organizada en métodos bien definidos, evitando repetir código.</li> <li>• El manejo de errores es adecuado: el programa no se interrumpe ante entradas incorrectas y muestra mensajes claros.</li> </ul>	
<b>History points:</b> 20 puntos	
<b>Cierre de actividad:</b>	



Al finalizar esta primera semana, serás capaz de construir una aplicación de consola en C# que registre y valide la información de un paciente y su mascota. Habrás organizado tu lógica en métodos, manejo de errores de manera adecuada y dado estabilidad a tu programa. Con esto, entenderás mejor la estructura básica de C# y estarás preparado para dar el siguiente paso: trabajar con colecciones y consultas más avanzadas.