




## HISTORIA DE USUARIO

<b>Nombre de la HU:</b>		Gestión de colecciones y consultas con LINQ
<b>Objetivo de la HU</b>		Como asistente de la clínica veterinaria, quiero poder consultar, filtrar y organizar la información de los pacientes y sus mascotas usando colecciones y LINQ, para trabajar con los datos de manera más rápida y ordenada.
TASK1	<b>Reforzar el uso de colecciones en C#.</b>	
	<ul style="list-style-type: none"><li>• Crear una lista (List&lt;T&gt;) que almacene los pacientes y sus mascotas en forma de objetos o estructuras.</li><li>• Explorar cómo agregar, eliminar y modificar elementos en la lista.</li><li>• Usar diccionarios (Dictionary&lt;TKey, TValue&gt;) para asociar el ID del paciente con su información, garantizando acceso rápido a los datos.</li></ul>	
TASK2	<b>Practicar la diferencia entre sintaxis de consulta y sintaxis de métodos en LINQ.</b>	
	<ul style="list-style-type: none"><li>• Usar Where para filtrar pacientes por edad o especie de la mascota.</li><li>• Usar Select para proyectar solo ciertos datos (ejemplo: nombres de pacientes).</li><li>• Ordenar con OrderBy y OrderByDescending por nombre, edad o especie.</li><li>• Agrupar con GroupBy para obtener pacientes organizados por especie de mascota.</li><li>• Probar métodos como First, FirstOrDefault, Any, All y Count en consultas concretas.</li></ul>	
TASK3	<b>Aplicar los métodos fundamentales de LINQ.</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>	

	
TASK 4	<b>Encadenar consultas LINQ para obtener resultados más expresivos.</b>
	<ul style="list-style-type: none"> <li>• Combinar filtros, ordenamientos y selecciones en una sola consulta.</li> <li>• Por ejemplo: obtener todos los pacientes con mascotas de especie "Perro", ordenados por edad, y mostrar solo nombre y teléfono del dueño.</li> </ul>
TASK 5	<b>Resolver problemas prácticos con LINQ.</b>
	<ul style="list-style-type: none"> <li>• Encontrar el paciente más joven o el de mayor edad.</li> <li>• Contar cuántas mascotas hay de cada especie.</li> <li>• Verificar si existe al menos un paciente con determinada condición (ejemplo: mascota sin raza definida).</li> <li>• Listar todos los nombres de pacientes en mayúsculas, ordenados alfabéticamente.</li> </ul>
<b>Criterios de aceptación.</b> <ul style="list-style-type: none"> <li>• Los pacientes y sus mascotas están organizados en colecciones (List o Dictionary).</li> <li>• Se pueden realizar consultas usando LINQ en ambas sintaxis (consulta y métodos).</li> <li>• Las consultas permiten filtrar, ordenar, agrupar y proyectar datos de manera correcta.</li> <li>• Se han utilizado métodos clave de LINQ (Where, Select, OrderBy, GroupBy, First, Any, Count).</li> <li>• Existen ejemplos de consultas encadenadas que devuelven resultados complejos.</li> <li>• Los problemas prácticos planteados se resuelven con consultas LINQ que funcionan sin errores.</li> <li>• El código es legible y cuenta con comentarios que explican cada consulta.</li> </ul>	
<b>History points:</b> 20 puntos	
<b>Cierre de actividad:</b> Al finalizar esta segunda semana, serás capaz de manejar colecciones en C# y aplicar consultas LINQ para filtrar, ordenar, agrupar y transformar datos de pacientes y mascotas. Habrás practicado tanto la sintaxis de consulta como la de métodos, comprendiendo en qué situaciones conviene cada una. Con este avance, estarás preparado para dar el siguiente paso: diseñar clases y relaciones con POO, utilizando UML para planificar la estructura del sistema.	

