

# TRABAJO PRÁCTICO INTEGRADOR

Clínica Veterinaria



Berghella Paula, Blanco Luis, Silva Paiva Alexis  
LABORATORIO DE SOFTWARE – UNPAZ

2023

## Documento de Requerimientos de Software

### I. Introducción

En el presente trabajo se explicarán los requerimientos de la entidad elegida para el desarrollo de nuestro trabajo. El mismo, contempla el caso de una hipotética clínica veterinaria, cuya entidad principal es el **cliente** que se registra en la misma.

El principal objetivo de este documento es definir una especificación clara y precisa de las funciones y restricciones que nuestro sistema necesita para su construcción.

Dado que para el desarrollo del sistema se siguieron los conceptos DDD aprendidos, nuestro punto de partida fue la *capa de dominio*.

Se definió la *entidad cliente*, dada la siguiente historia de usuario denominada “**Administrar Clientes**”:

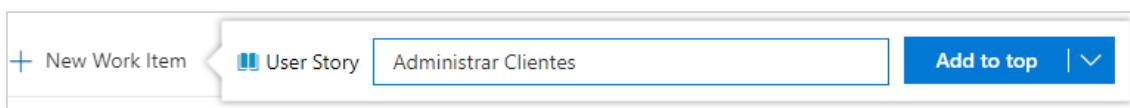


Figura 1. Agregando Historia de Usuario a nuestro backlog.

En su **descripción** se detalló:

*Como empleado de la veterinaria, quiero poder administrar la información de los clientes para mantenerla actualizada y eliminar registros que se consideren obsoletos.*

Los **criterios de aceptación** incluyen lo siguiente:

- *Como usuario, quiero poder registrar un nuevo cliente para poder gestionarlo luego. Quiero que cuente con la siguiente información de contacto: nombre, apellido, correo electrónico.*
- *Como usuario, quiero poder guardar la información del cliente en el sistema para persistir sus datos.*
- *Como usuario, quiero ver una lista de todos los clientes registrados en el sistema para una mejor gestión.*
- *Como usuario, quiero seleccionar un cliente de la lista para ver y actualizar su información.*
- *Como usuario, quiero poder modificar el correo electrónico de un cliente existente para mantener sus datos actualizados.*
- *Como usuario, quiero poder eliminar un cliente de la lista para reflejar el vínculo no existente con la organización.*

State	● Active	Area	Laboratorio de Software
Reason	🔒 Implementation s...	Iteration	Trabajo Integrador\Sprint 1

<b>Description</b>	<b>Planning</b>
Como empleado de la veterinaria, quiero poder administrar la información de los clientes para mantenerla actualizada y eliminar registros que se consideren obsoletos.	Story Points
<b>Acceptance Criteria</b>	Priority
<ul style="list-style-type: none"> <li>• Como usuario, quiero poder registrar un nuevo cliente para poder gestionarlo luego. Quiero que cuente con la siguiente información de contacto: nombre, apellido, correo electrónico.</li> <li>• Como usuario, quiero poder guardar la información del cliente en el sistema para persistir sus datos.</li> <li>• Como usuario, quiero ver una lista de todos los clientes registrados en el sistema para una mejor gestión.</li> <li>• Como usuario, quiero seleccionar un cliente de la lista para ver y actualizar su información.</li> <li>• Como usuario, quiero poder modificar el correo electrónico de un cliente existente para mantener sus datos actualizados.</li> <li>• Como usuario, quiero poder eliminar un cliente de la lista para reflejar el vínculo no existente con la organización.</li> </ul>	1
	Risk
	<b>Classification</b>
	Value area
	Business

Figura 2. Detalles de nuestra Historia de Usuario a desarrollar.

## 2. Descripción General

En esta sección, se detallará una breve descripción general del sistema a desarrollar para la Clínica Veterinaria.

### 2.1. Función

El sistema permite gestionar clientes para la organización, es decir, permite crear nuevos clientes, actualizar existentes (correo electrónico) y la posibilidad de eliminarlos, de ser necesario.

## 3. Requerimientos Funcionales

A partir de lo descripto, se obtuvieron los siguientes requerimientos a contemplar:

- **REQ-001: Crear cliente.** Permite agregar nuevos clientes a la base de datos.
- **REQ-002: Leer cliente.** Permite ver información de un cliente ya existente.
- **REQ-003: Actualizar cliente.** Permite modificar el correo electrónico de un cliente ya existente.
- **REQ-004: Eliminar cliente.** Permite eliminar un cliente de la base de datos.

## 4. Requerimientos No Funcionales

Los requerimientos no funcionales capturados fueron los siguientes:

- **Usabilidad.** La interfaz de usuario debe ser intuitiva y fácil de usar.
- **Flexibilidad.** El sistema debe permitir incorporar fácilmente cambios y nuevas funcionalidades, asumiendo que el software cambia y evoluciona constantemente.
- **Mantenibilidad.** Debe ofrecer facilidad de mantenimiento, entendiendo por ello modificaciones o ampliaciones de carácter evolutivo.
- **Rendimiento.** La aplicación debe responder rápida y eficientemente.
- **Lenguaje.** El lenguaje a utilizar debe ser C#.
- **Persistencia de datos.** Debe poder conectarse tanto a bases de datos relacionales como no relacionales.

## Documento de Instrucciones de Procesos para el Despliegue

En esta sección, se proporcionará una serie de instrucciones básicas sobre cómo desplegar la aplicación en un entorno de prueba.

### 1. Requisitos Previos

- Base de datos SQL Server instalada y configurada.
- Base de datos MongoDB instalada y configurada.
- Lenguaje de programación C#, junto a sus dependencias correspondientes instaladas.
- Código fuente de la aplicación (obtenido del repositorio).

### 2. Configuración de la Base de Datos SQL

- Iniciar sesión en el sistema de administración de SQL Server usando las credenciales correspondientes.
- Crear una base de datos.
- Insertar la cadena de conexión referente al usuario para los repositorios relacionados a esta base de datos.
- Ejecutar el script para la creación de la tabla y relaciones proporcionadas en la aplicación para configurar el esquema de la base de datos.

### 3. Configuración de la Base de Datos MongoDB

- Iniciar el servicio de MongoDB.

### 4. Instalación de dependencias

- Usar el administrador de paquetes NuGet para restaurar las dependencias requeridas por la aplicación.

- Asegurarse de que todas las dependencias estén instaladas correctamente y actualizadas, según corresponda.

## 5. Despliegue de la Aplicación

- Si todo está correctamente configurado, la aplicación debería correr y estar lista para su uso.

## 6. Proceso de Pruebas

- Realizar las pruebas necesarias en la aplicación para verificar su correcto funcionamiento.
- Verificar que las funcionalidades se realizan correctamente.
- Comprobar la validez de los datos ingresados.
- Realizar pruebas adicionales, según sea necesario.