**FARMING - COR**

****

**APRENDICES:**

**SERGIO ALEJANDRO CALDERON PERDOMO**

**ENYEER MANUEL GRANADOS MARDINIS**

**FICHA: 2558104**

**SERVICIO NACIONAL DE APRENDIZAJE - SENA**

**CENTRO DE ELECTRICIDAD, ELECTRÓNICA Y TELECOMUNICACIONES - CEET**

**INSTRUCTORA: MIGUEL ANGEL LOPEZ**

**BOGOTÁ D.C**

****

FARMING - COR

**MANUAL TÉCNICO**

**Manual Tecnico**

**1. Introducción**

Farming Cor es un sistema de información web desarrollado para facilitar la donación de productos agrícolas como frutas y verduras. Este manual técnico describe la arquitectura del sistema, las tecnologías utilizadas y las instrucciones para su instalación, configuración y mantenimiento.

**2. Manual**

Este manual proporciona información detallada sobre la arquitectura, tecnologías y configuraciones utilizadas en el desarrollo de Farming Cor, nuestra plataforma de donación de productos.

**3. Alcance del documento**

Este documento abarca la descripción técnica detallada de Farming Cor, incluyendo su arquitectura, tecnologías empleadas, así como instrucciones específicas para su instalación, configuración y mantenimiento.

**4. Proyecto Farming-Cor**

Farming-Cor es un sistema de información basado en plataforma web cuyo objetivo principal es facilitar la donación de productos, como frutas y verduras, provenientes del sector de Corabastos.

**5. Objetivo del Manual**

El objetivo de este manual es proporcionar a los usuarios una comprensión completa de la arquitectura, tecnologías utilizadas y procedimientos necesarios para operar Farming Cor de manera efectiva, garantizando así su funcionamiento óptimo y su mantenimiento adecuado.

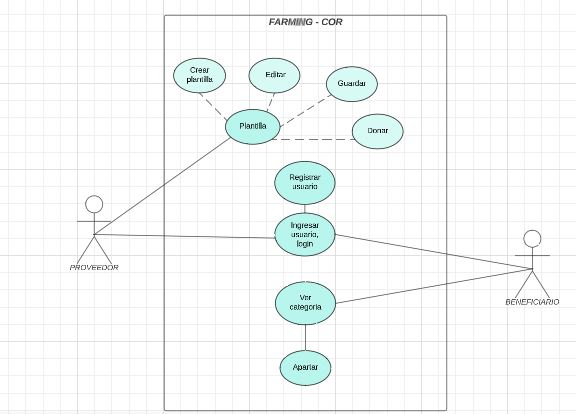
**6. Arquitectura del Sistema:**

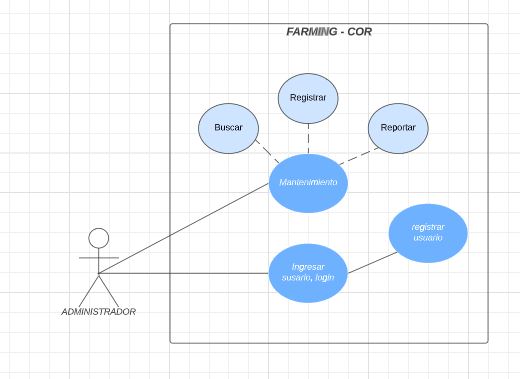
Farming Cor está desarrollado como una aplicación web utilizando una arquitectura cliente-servidor.

* **Cliente:** La interfaz de usuario de Farming Cor está implementada utilizando Angular, un framework de desarrollo frontend basado en TypeScript. Angular proporciona una estructura sólida y modular para el desarrollo de aplicaciones web, facilitando la creación de interfaces de usuario interactivas y receptivas. La comunicación entre el cliente y el servidor se realiza a través de solicitudes HTTP para recuperar datos y realizar acciones.
* **Servidor:** El servidor backend está construido utilizando tecnologías basadas en Node.js, cómo Express.js. Node.js permite la creación de servidores web altamente escalables y eficientes, mientras que Express.js proporciona un marco flexible para el desarrollo de aplicaciones web en Node.js. El servidor maneja las solicitudes de los clientes, procesa la lógica empresarial y se comunica con la base de datos.
* **Base de Datos:** Utilizamos MySQL como sistema de gestión de bases de datos relacional para almacenar información sobre usuarios, productos, reservas y otras entidades relacionadas con la aplicación. MySQL es conocido por su confiabilidad y rendimiento en entornos web, y ofrece una estructura de tabla relacional para organizar y relacionar los datos de manera eficiente.

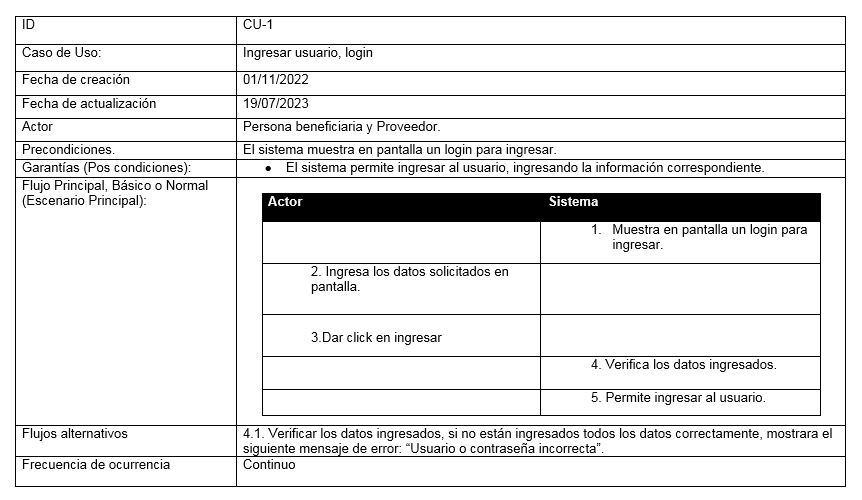
### **7. Caso de uso**

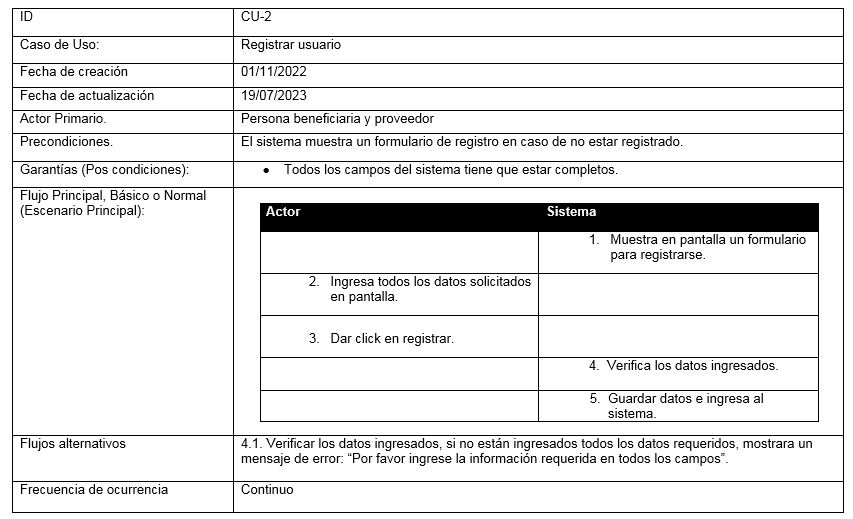
**7.1 Diagrama de caso de uso**

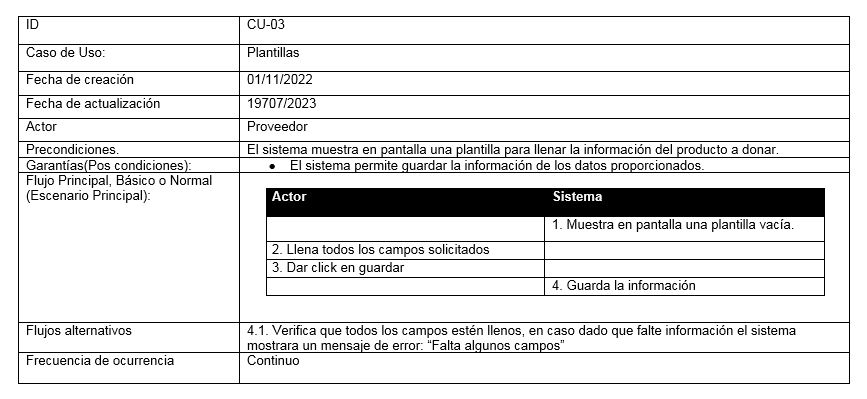
****

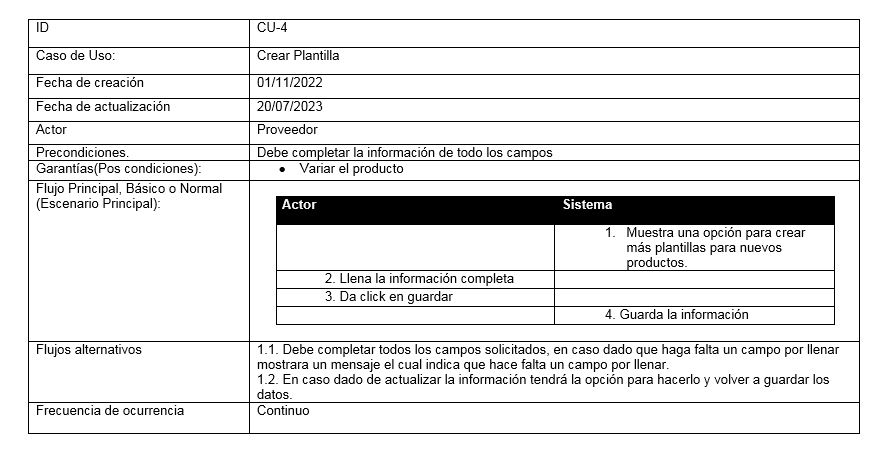
****

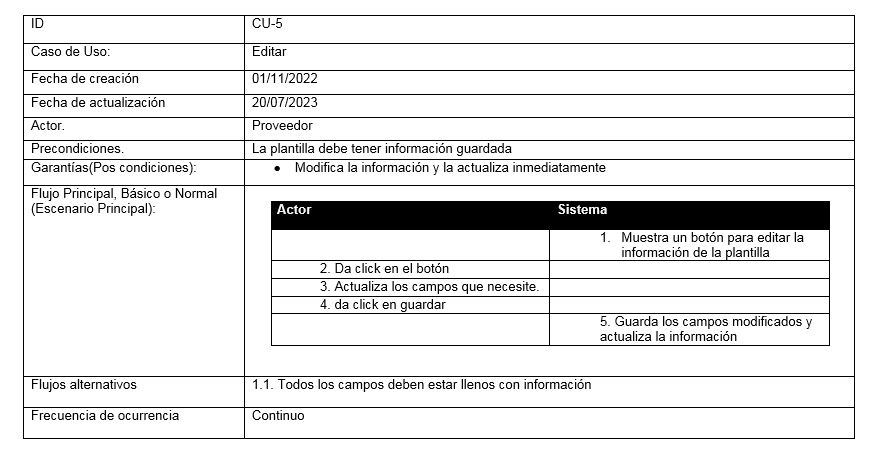
**8. Caso de uso extendido**

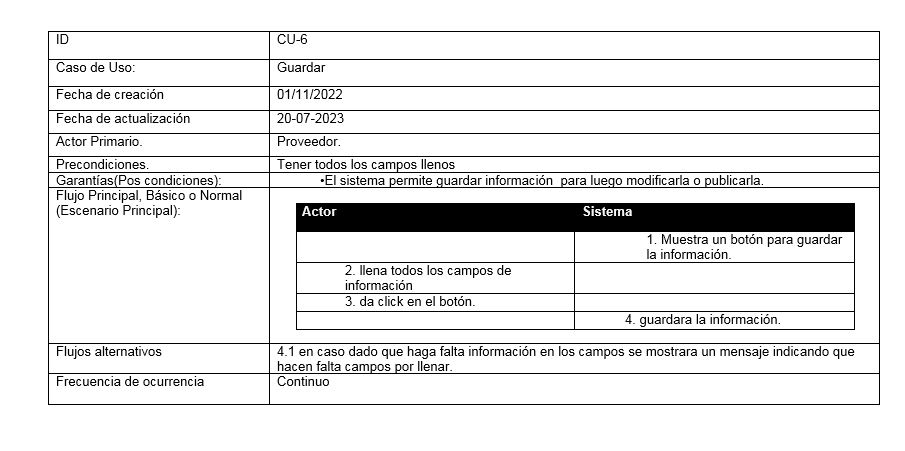
****

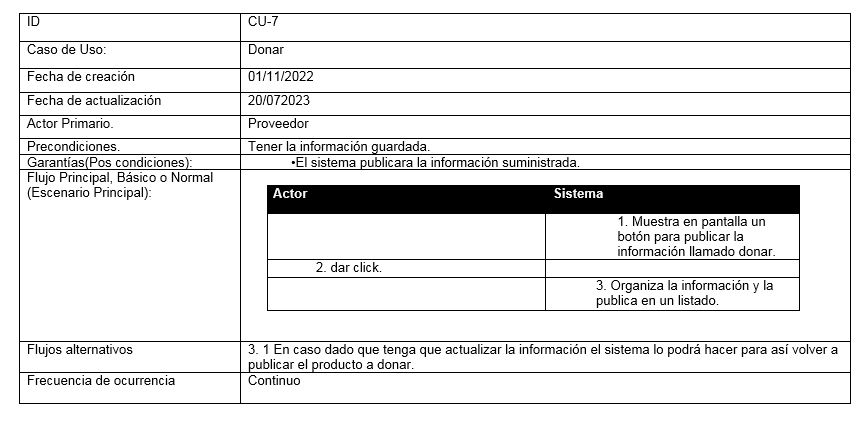
****

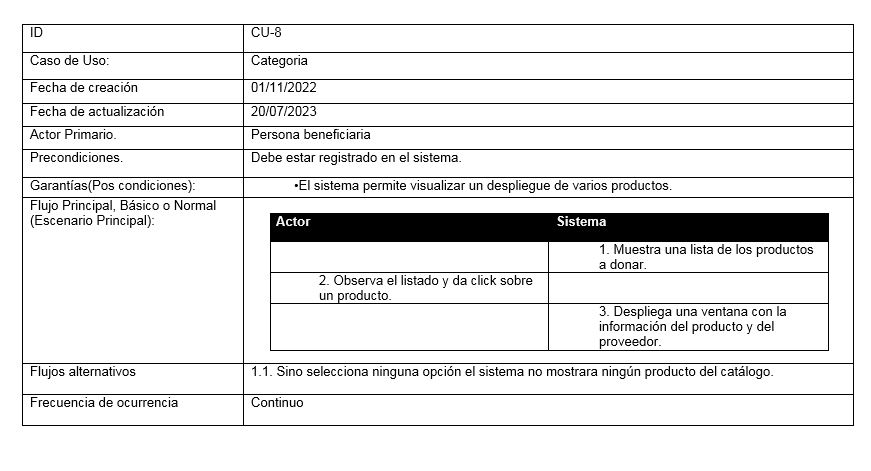
****

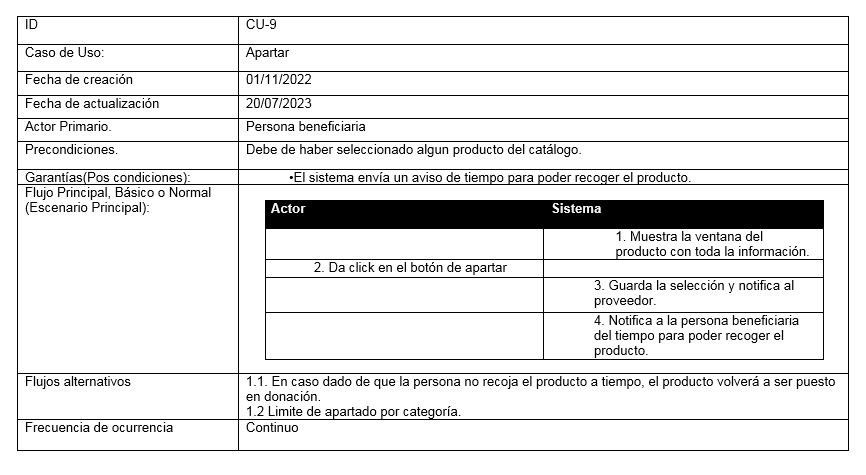
****

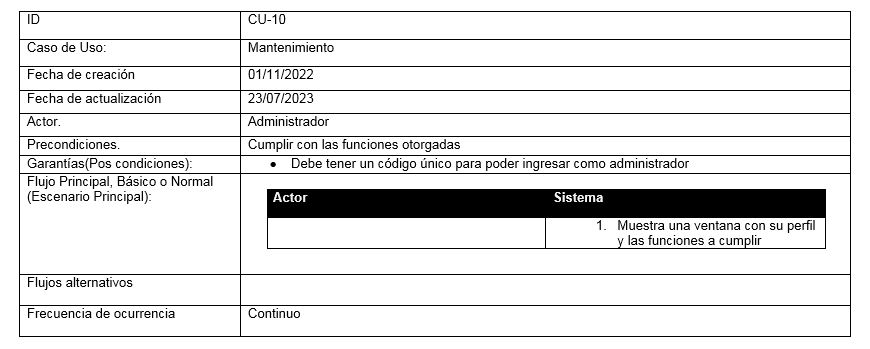
****

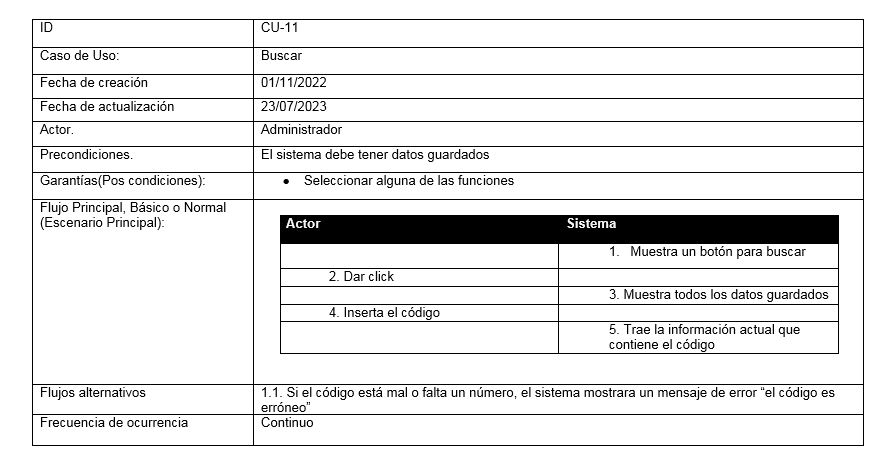
****

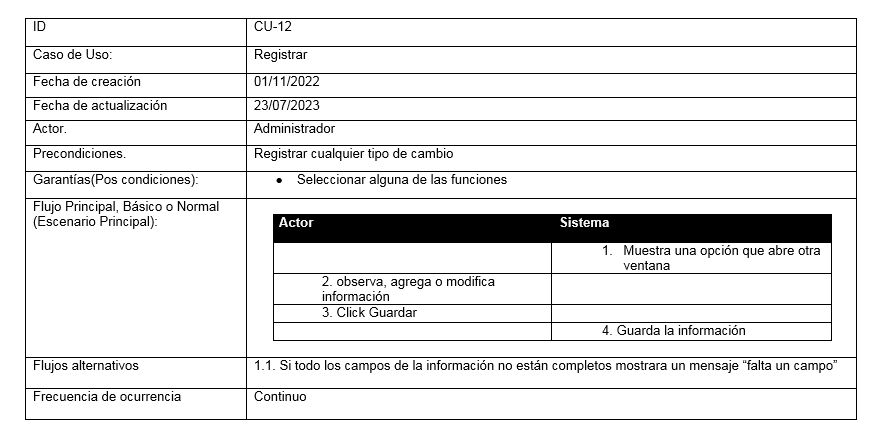
****

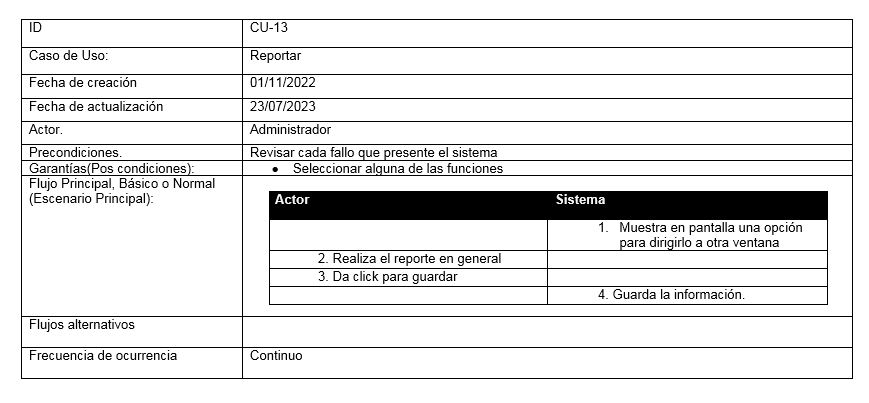
****

****

****

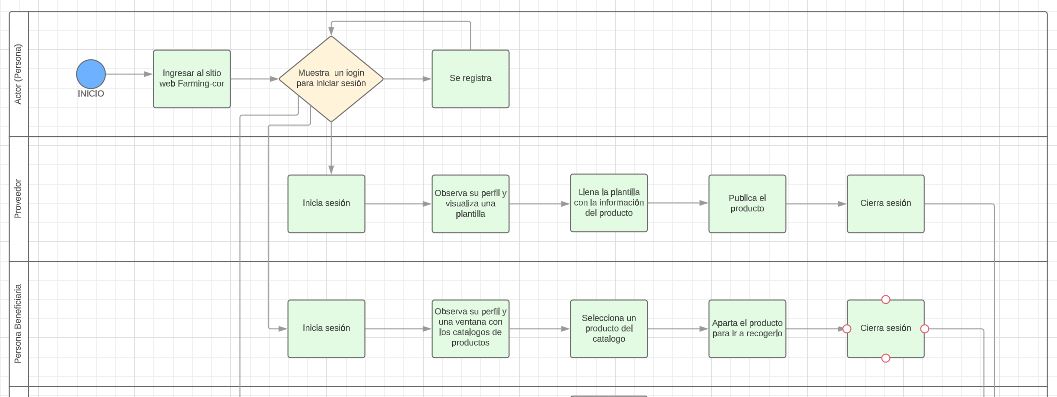
****

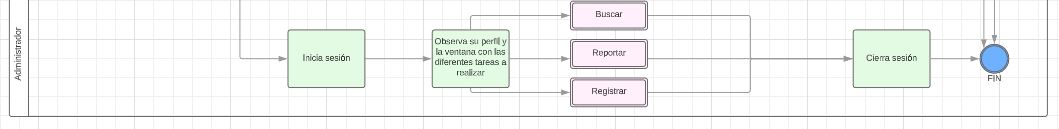
****

****

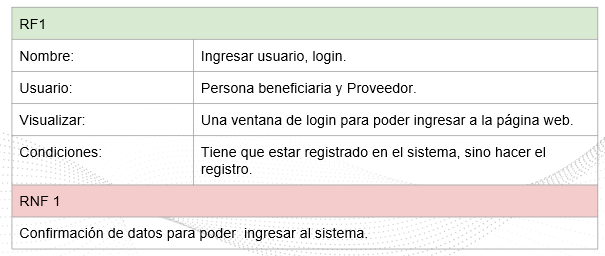
### **9. Procesos del sistema**

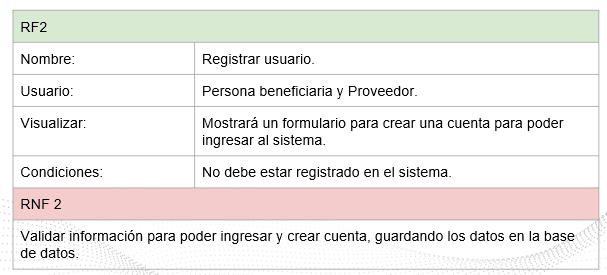
**9.1 Mapa de procesos (BPMN)**

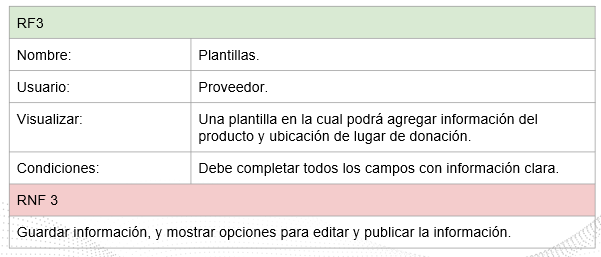


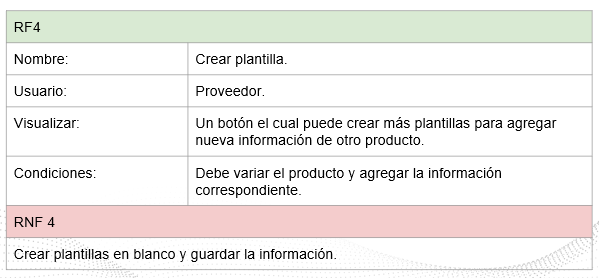


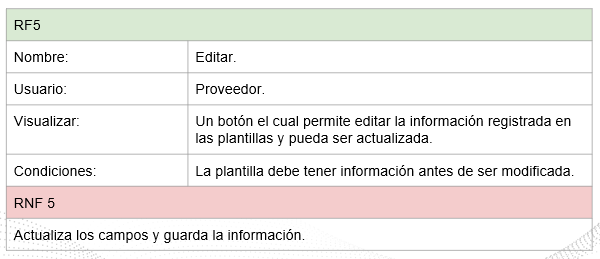
**10. Requerimientos funcionales y no funcionales**

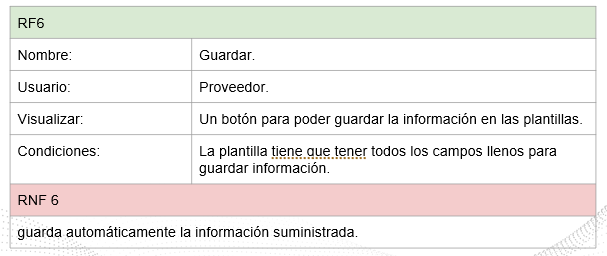
****

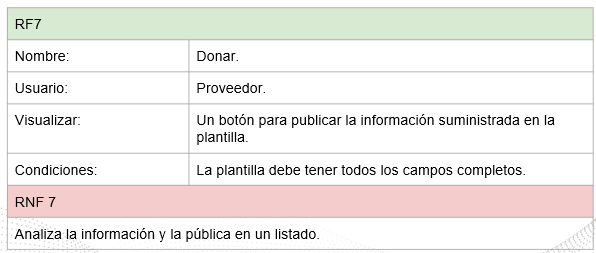
****

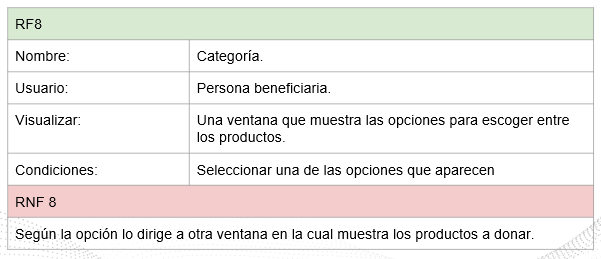
****

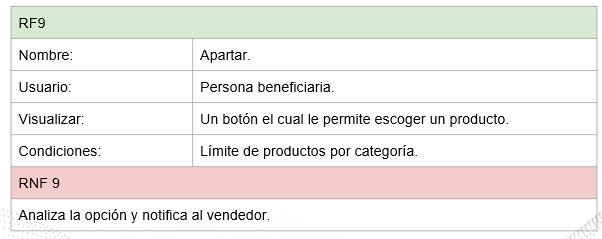
****

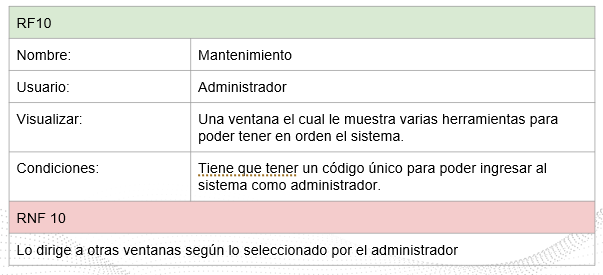
****

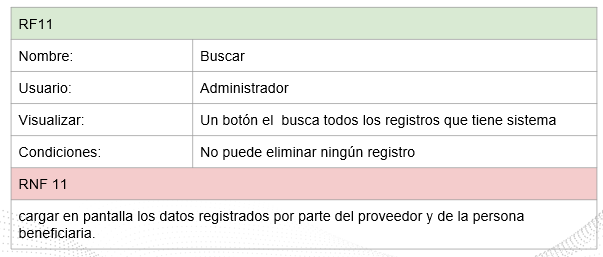
****

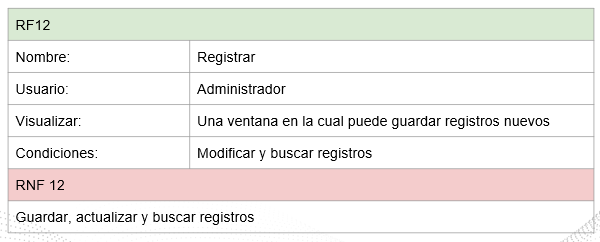
****

****

****

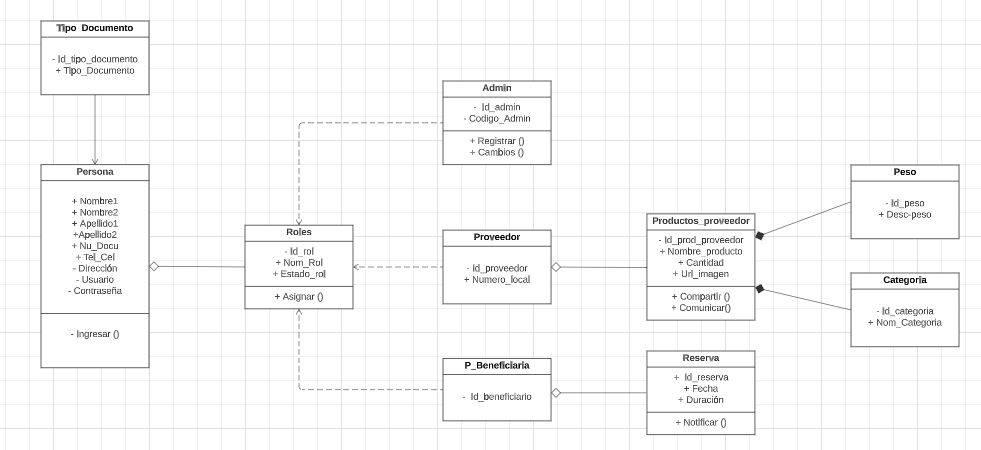
****

****

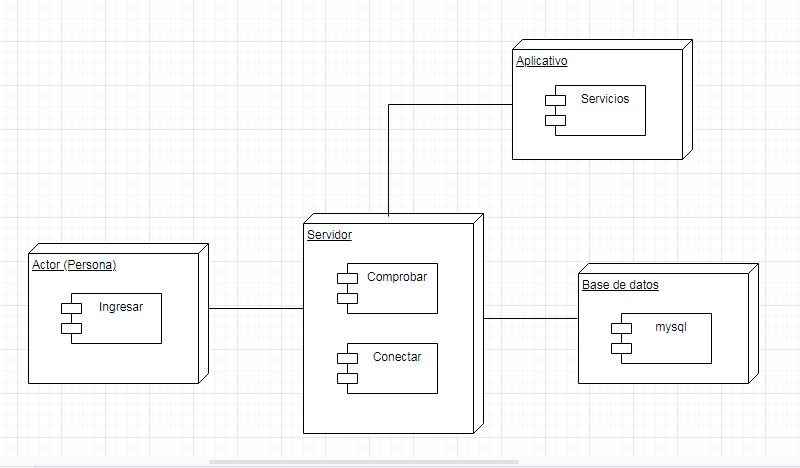
****

**11. Diagrama UML**

**11.1 Diagrama de clases**

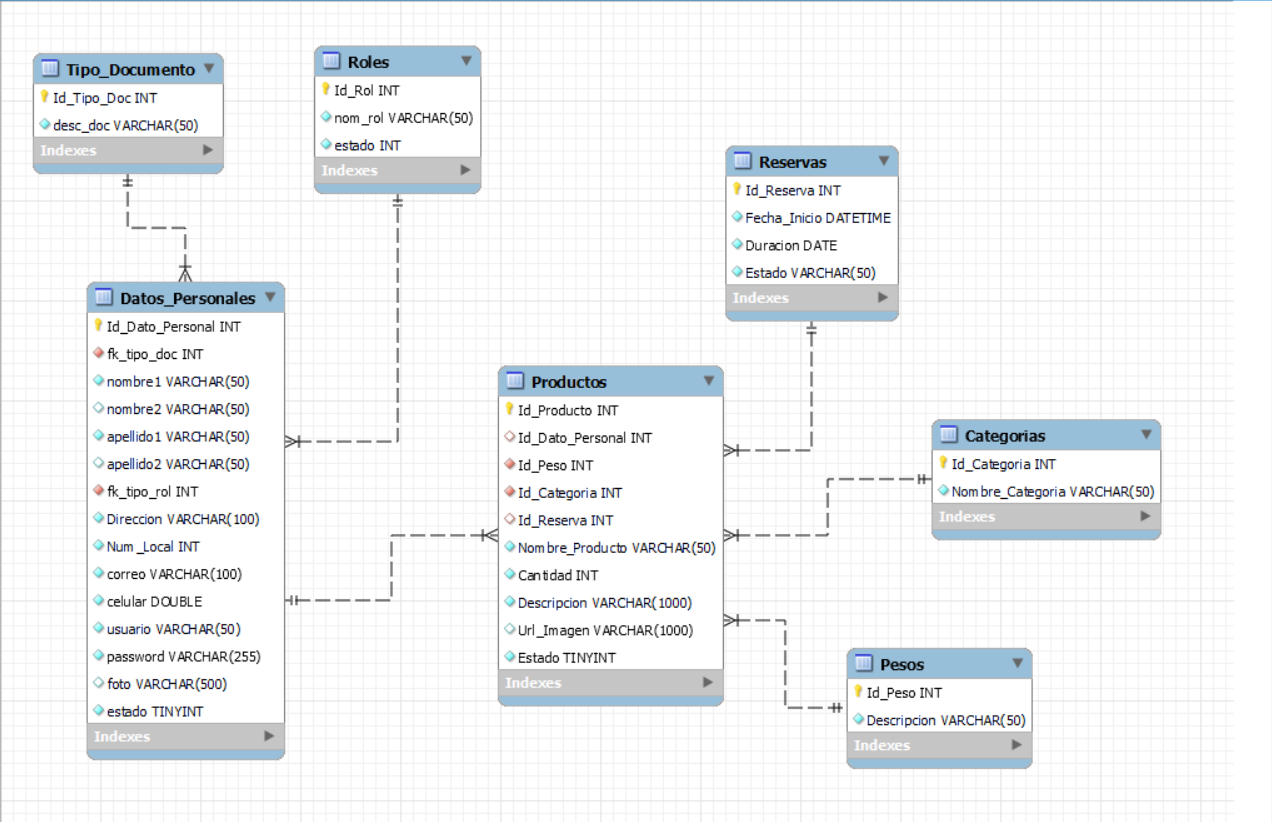
****

**12. Diagrama de despliegue**

****

**13. Base de datos**

**13.1 Modelo relacional**

****

### 

### **14. Tecnologías Utilizadas:**

En el desarrollo de Farming Cor, se emplearon las siguientes tecnologías:

**Lenguajes de Programación:**

* + TypeScript para el desarrollo del frontend utilizando el framework Angular.
  + JavaScript para lógica adicional en el frontend.
  + Node.js para el desarrollo del backend.

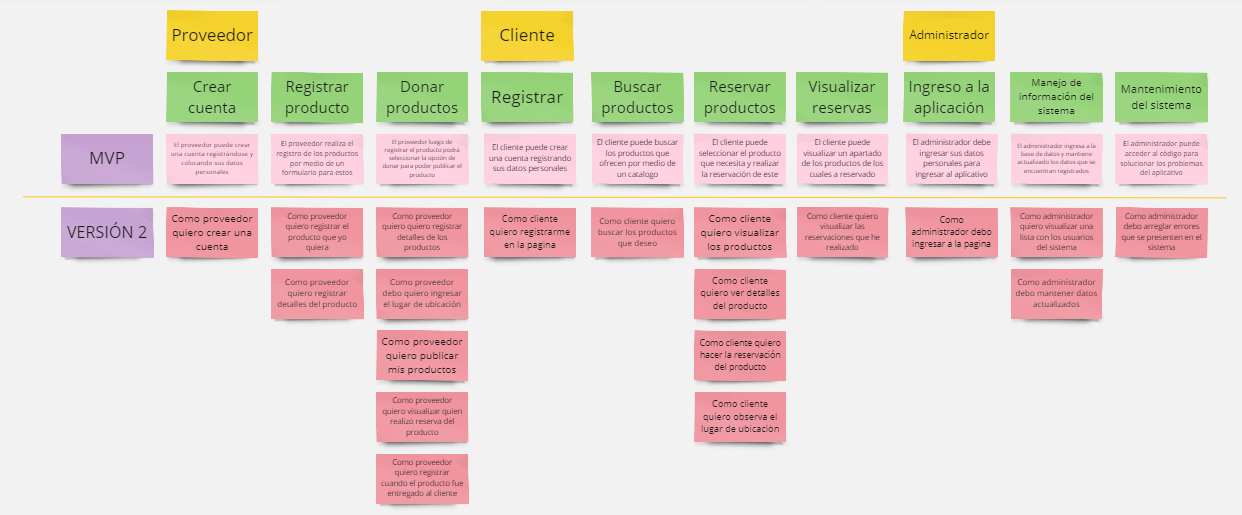
**Base de Datos:**

* + MySQL para el almacenamiento de datos.

**Herramientas y Frameworks:**

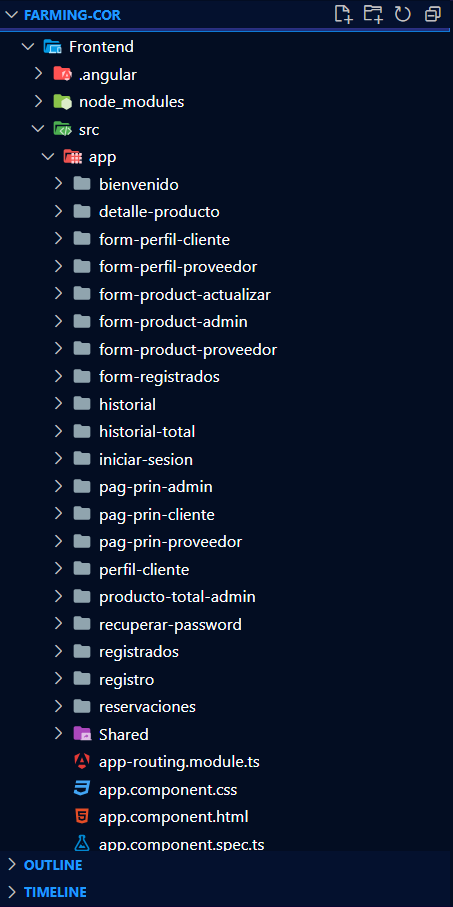
* + XAMPP se utilizó para proporcionar un entorno de servidor local durante el desarrollo.
  + Bootstrap se empleó para el diseño responsivo y la maquetación de la interfaz de usuario.
  + Angular Material para elementos de diseño y UI en el frontend.
  + Express.js como framework de aplicación web para el backend.

### **15 Mapa historia de usuario**

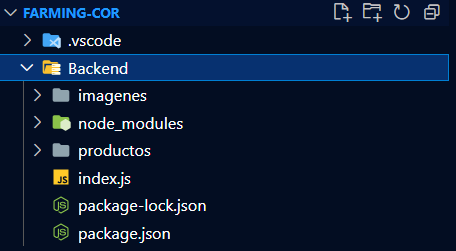


### **16. Documento Fuente**

**16.1 Estructura frontend**

****

**16.2 Estructura Backend**

****

### **17. Configuración del Entorno de Desarrollo:**

Para configurar un entorno de desarrollo similar al utilizado para desarrollar Farming Cor, sigue estos pasos:

* + Instalar Node.js en tu sistema operativo Windows 10 para tener un entorno de ejecución de JavaScript.
  + Configura una instancia de MySQL en tu sistema o utiliza un servicio de base de datos en la nube compatible con MySQL para almacenar los datos de Farming Cor.
  + Clona el repositorio de código fuente de Farming Cor en tu sistema local.
  + Configura la conexión a la base de datos en el archivo de configuración del servidor backend, utilizando los datos de conexión proporcionados por tu instancia de MySQL.
  + Utiliza un editor de texto o un IDE compatible con TypeScript para modificar y trabajar en el código fuente del frontend y el backend según sea necesario.

### **18. Integración de Funcionalidades:**

* Registro e Inicio de Sesión: Utiliza tokens de autenticación para gestionar el estado de inicio de sesión del usuario, asegurando un acceso seguro a la plataforma.
* Visualización de Productos: Los datos de los productos se recuperan de la base de datos utilizando consultas específicas y se muestran dinámicamente en el frontend, permitiendo a los usuarios explorar los productos disponibles de manera eficiente.
* Reservación de Productos: Las reservas de productos se procesan mediante solicitudes HTTP, utilizando un sistema de gestión de pedidos para mantener un registro actualizado de las reservas realizadas por los usuarios. Estas actualizaciones se reflejan automáticamente en la base de datos.
* Funciones del Proveedor y del Administrador: Se implementa una lógica específica para gestionar las acciones y privilegios de los proveedores y administradores. Esto incluye la capacidad de publicar y gestionar productos para los proveedores, así como la administración de usuarios y productos para los administradores.

### 

### **19. Requisitos del Sistema:**

* Sistema Operativo: Compatible con Windows 10.
* Servidor Web: XAMPP u otro servidor web compatible con MySQL.
* Navegador Web: Se recomienda Google Chrome, Mozilla Firefox o Microsoft Edge para una experiencia óptima.