| **DUOC UC - Escuela de informática y telecomunicaciones** |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: Garden Store* |
|  |
| **Revisión*: [01]*** |
| **[Seleccionar fecha]** |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[**Ficha del documento 3**](#_heading=h.gjdgxs)

[**1. Introducción 4**](#_heading=h.30j0zll)

[1.1.](#_heading=h.1fob9te) Propósito 4

[1.2.](#_heading=h.3znysh7) Ámbito del Sistema 4

[1.3.](#_heading=h.2et92p0) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_heading=h.tyjcwt) Referencias 4

[1.5.](#_heading=h.3dy6vkm) Visión General del Documento 4

[**2.**](#_heading=h.1t3h5sf) **Descripción General 5**

[2.1.](#_heading=h.4d34og8) Perspectiva del Producto 5

[2.2.](#_heading=h.2s8eyo1) Funciones del Producto 5

[2.3.](#_heading=h.17dp8vu) Características de los Usuarios 5

[2.4.](#_heading=h.3rdcrjn) Restricciones 5

[2.5.](#_heading=h.26in1rg) Suposiciones y Dependencias 6

[2.6.](#_heading=h.lnxbz9) Requisitos Futuros 6

[**3.**](#_heading=h.35nkun2) **Requisitos Específicos 7**

[3.1](#_heading=h.1ksv4uv) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.44sinio) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.2jxsxqh) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.z337ya) *Interfaces de software 8*

[*3.1.4*](#_heading=h.3j2qqm3) *Interfaces de comunicación 8*

[3.2](#_heading=h.1y810tw) Requisitos funcionales 8

[3.3](#_heading=h.4i7ojhp) Requisitos no funcionales 9

[*3.3.1*](#_heading=h.2xcytpi) *Requisitos de rendimiento 9*

[*3.3.2*](#_heading=h.1ci93xb) *Seguridad 9*

[*3.3.3*](#_heading=h.3whwml4) *Fiabilidad 10*

[*3.3.4*](#_heading=h.2bn6wsx) *Disponibilidad 10*

[*3.3.5*](#_heading=h.qsh70q) *Mantenibilidad 10*

[*3.3.6*](#_heading=h.3as4poj) *Portabilidad 10*

[3.4](#_heading=h.1pxezwc) Otros Requisitos 10

[**4. Propuesta de Planificación 11**](#_heading=h.49x2ik5)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.2p2csry)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.147n2zr)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.3o7alnk)

[*4.1.4 Diagrama EDT 11*](#_heading=h.23ckvvd)

[*4.1.5 Carta Gantt 11*](#_heading=h.ihv636)

[*4.1.6 Resumen Costos del Desarrollo del Proyecto 11*](#_heading=h.32hioqz)

[4.2 Plan de Control de Cambio 12](#_heading=h.1hmsyys)

[5. Anexos 12](#_heading=h.41mghml)

[*5.1 Acta de Proyecto 12*](#_heading=h.2grqrue)

[*5.2 Matriz Especificación de Requerimientos 12*](#_heading=h.vx1227)

[*5.3 Diagrama de Casos de Uso General 12*](#_heading=h.3fwokq0)

[*5.4 Planilla Casos de Uso 12*](#_heading=h.1v1yuxt)

[*5.5 Prototipado de Software 13*](#_heading=h.4f1mdlm)

[*5.6 Resultado Análisis de Calidad Diagramas Modelamiento 13*](#_heading=h.2u6wntf)

[*5.7 Resultado Análisis de Calidad Prototipo No funcional del Sistema 13*](#_heading=h.19c6y18)

[*5.8 Planilla entregables del Proyecto 13*](#_heading=h.3tbugp1)

[*5.9 Matriz de Control de Cambios 13*](#_heading=h.28h4qwu)

[*5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 13*](#_heading=h.nmf14n)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Felipe Sanchez* | *Lider de Proyecto* |
| *Ignacio Jimenez* | *Programador de Aplicaciones Móviles* |
| *Diego Barrera* | *QA* |
| *Nestor Aviles* | *Arquitecto de Software* |
| *Carlos Jimenez* | *Cliente Final* |

**1. Introducción**

En este documento se especificarán los requerimientos funcionales y no funcionales que tendrá el proyecto que iremos desarrollando durante el semestre, “Garden Store” está dirigido a ayudar a un pequeño negocio que quiere emprenderse al mundo de la jardinería por medio de internet, haciendo uso de la informática y el conocimiento de los ingenieros de Software.

Se explicará y describirá cada uno de los requerimientos específicos que tendrá el proyecto, detallando cada uno de ellos, explicando el propósito de cada uno de ellos.

**1.1. Propósito**

El propósito de este documento es la de detallar y describir detalladamente los requerimientos que tendrá el proyecto, dando un descripción general del producto, la perspectiva y funcionalidades del producto, restricciones, suposiciones, dependencias, requerimientos funcionales y no funcionales. Este documento va dirigido principalmente al cliente final.

**1.2. Ámbito del Sistema**

**Nombre del proyecto**: Garden Store

**Funciones y limitaciones**: Como principales funcionalidades que tendrá el sistema es el poder realizar compras de productos de jardinería, realizar descuentos u ofertas. También se podrán realizar subscripciones sin fines de lucro con el fin de incentivar a los usuarios a realizar pequeñas donaciones para recibir un beneficio a cambio. Todos los registros tanto como autenticación de usuario, facturas de compras serán privados y no se revelará ninguna información personal de los clientes para así mantener la integridad y seguridad de los datos. Tendrá la funcionalidad de gestionar y administrar todos los productos, usuarios, compras, envíos, etc. Y la visualización de datos estadísticos para facilitar la toma de decisiones del negocio.

**Beneficios, objetivos y metas**: El principal beneficio de este sistema, son las ganancias que se obtendrían a través de las compras de los usuarios a través de la aplicación, también de las donaciones que lleguen de los usuarios. Como principal objetivo es expandir la base de clientes, con el fin de aumentar la base de usuarios o clientes, mediante una estrategia de marketing digital planificada anteriormente.

**1.3. Definiciones, Acrónimos y Abreviaturas**

En esta subsección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

**GS:** Garden Store.

**ERS:** Especificación de Requerimientos de Software.

**Requerimiento:** Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

**1.4. Referencias**

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

* Planilla de requerimientos

**1.5. Visión General del Documento**

Este documento cuenta con una descripción general del producto con subsecciones que nos definirán las distintas funcionalidades que tiene el sistema, también cuenta con la especificación de requerimientos detallados por cada tipo de requerimiento dispuesto en el sistema..

**2. Descripción General**

Este proyecto busca resolver, satisfacer las necesidades y poder expandir el negocio de garden store por el limitante que tenían que solo realizaban ventas en su sector o a sus vecinos más cercanos y por no contar un software que gestione y controle los usuarios, productos, ventas y compra de productos en el sistema. Para poder integrar el negocio al ámbito virtual para poder tener un mayor alcance o influencia de clientes.

El propósito de estas funcionalidades generales es el resolver el problema o necesidad que tiene la empresa Garden Store, principalmente la gestión y control de los usuarios, productos y ventas. Y la de poder expandir la base de usuarios que tiene la empresa y poder aumentar sus ventas

**2.1. Perspectiva del Producto**

La perspectiva hacia futuro del proyecto, se ve con algo de dificultades al momento de entrar en el mercado, ya que, hay varios servicios o herramientas que cumplen con los requerimientos que tenemos, pero con el paso del tiempo se le puede llegar a implementar muchas más funcionalidades únicas al software para añadirla valor agregado al proyecto.

Este proyecto no es totalmente independiente, requiere de que los interesados y los accionistas o patrocinadores invierten en este proyecto, para poder desarrollarlo e implementarlo, también requiere de que el cliente se encuentre interesado en el producto o solución, ya que, hay la posibilidad de que el cliente hasta incluso proponga utilizar otro producto o servicio para resolver su problema o necesidad.

**2.2. Funciones del Producto**

El producto tendrá la capacidad de realizar compra y ventas de productos de jardinería de manera online. También deberá gestionar los usuarios, productos, ventas y compras, realizando estadística o visualización de los datos, toda esta información estará almacenada y registrada en una base de datos SQL.

Algunas de las funcionalidades que tendrá el sistema son los siguientes:

* **Gestionar productos y clientes:** El sistema podrá gestionar los productos y usuarios
* **Autenticación de dos factores:** El sistema pedirá un código de verificación al momento de acceder los administradores al panel de control.
* **Contenido Educativo:** El sistema tendrá un catálogo de recursos educativos y consejos para los entusiastas en la jardinería.
* **Sistema de suscripciones:** El sistema tendrá suscripciones por nivel sin fines de lucro
* **Búsqueda Avanzada:** El sistema tendrá una busqueda de productos a través de distintos filtros y opciones
* **Visualización de Datos:** El sistema tendrá un apartado para la visualización de datos estadísticos de la plataforma.
* **Identificación de Transacciones Fraudulentas:** El sistema tendrá un apartado donde realizará un análisis e identificación de transacciones fraudulentas
* **Asistente Virtual:** El panel de control tipo Dashboard tendrá integrado un asistente virtual que ayudará en algunas funcionalidades del sitio, y proporcionará información para ayudar a las decisiones de negocio
* **Sistema de Reviews:** El sistema permitirá a los usuarios crear reseñas a los distintos productos, con un rating de estrellas y un comentario

**2.3. Características de los Usuarios**

La principal característica de los usuarios son, que sean entusiastas a la jardinería, tengan algo de conocimientos sobre el internet, pero muy básico. Además tener conocimientos básicos del manejo de interfaz básica para poder utilizar el producto de manera más rápida, para no tener dificultad de manejar el producto. Tener la capacidad de leer y escribir, educación básica.

**2.4. Restricciones**

* Solo contamos con un presupuesto de 8 millones de pesos
* Tenemos un plazo habilitado de 5 meses de desarrollo y entrega del producto
* El proyecto utilizará tecnologías específicas, Typescript/Javascript,
* Limitaciones de disponibilidad del equipo de trabajo, solo podemos estar a ciertas horas limitadas del día
* Diseño responsivo obligatorio
* Experiencia de usuario simplificada
* Sistema de Transacciones de Transbank y Mercado Pago
* Disponibilidad regional limitada, solo estará disponible en las regiones centro-sur y sur del país
* Cumplimiento de normas económicas locales

**2.5. Suposiciones y Dependencias**

**Suposiciones:**

* Compatibilidad con distintos dispositivos
* Conocimiento básico de tecnología
* Todos los usuarios tienen acceso a internet
* Tendremos mayor seguridad en datos muy sensibles
* Tenemos un financiamiento asegurado por los patrocinadores
* Diferenciación del producto o valor agregado

**Dependencias:**

**Frontend:**

* Electron: Herramienta que se utilizara para la creación de la aplicación de escritorio
* Axios: Librería para el manejo de solicitudes HTTP.
* Leaflet: Para el uso de mapas interactivos para el manejo de ubicaciones en tiempo real
* D3.js: Para la visualización de datos, como gráficos del nivel de variación de precios de los productos relacionados a la jardinería
* Angular: Framework frontend de javascript, para la creación de aplicaciones de una sola página

**Backend:**

* Nestjs: Framework de javascript mucho más robusto que express, para la creación de APIs, Websockets o microservicios
* Express: Framework de javascript backend, para la creación de APIs
* Node JS: Entorno de ejecución de javascript, para ejecutar scripts o programas que estén escritos en javascript

**Red Neuronal:**

* Tensorflow: Para el manejo de modelos de aprendizaje automático
* Keras: Para el manejo y uso de modelos de aprendizaje profundo o redes neuronales
* Pandas: Para la manipulación y el análisis de los datos
* Numpy: Para la realización de cálculos y el manejo de grandes volúmenes de datos
* Matplotlib: Librería para la creación de gráficos en 2 dimensiones
* Seaborn: Librería para la creación de gráficos de más alto nivel

**Testing:**

* Jest: Biblioteca para la creación, ejecución y estructuración de pruebas o testing
* Jasmine: Framework para la creación y ejecución de pruebas unitarias
* Karma: Herramienta para la creación y ejecución de pruebas End-To-End

**Seguridad:**

* Json Web Token: Para el manejo de la autenticación a través de tokens de autenticación
* Cors-Headers: Para el manejo de los cors en lo navegadores y que permite que la API sea consumida desde especificos dominios
* Bcrypt: Librería para la encriptación y validación de cadenas de texto, en base a saltos
* Crypto: Librería para la encriptación

**2.6. Requisitos Futuros**

* La integración de un asistente virtual, para el panel de administración, en la ayuda de la toma de decisiones de negocio
* La implementación de un chatbot de quejas o de autoayuda
* La implementación de una aplicación móvil de escaneo de QR
* Integración de un generador de diseños personalizados de plantas sintéticas

**3. Requisitos Específicos**

El producto tendrá la capacidad de realizar compra y ventas de productos de jardinería de manera online. También deberá gestionar los usuarios, productos, ventas y compras, realizando estadística o visualización de los datos, toda esta información estará almacenada y registrada en una base de datos SQL.

Algunas de las funcionalidades que tendrá el sistema son los siguientes:

* **Gestionar productos y clientes:** El sistema deberá gestionar a los usuarios y productos que se encuentren registrados en la base de datos, también debe poder editarlos y eliminarlos cuando sea estrictamente necesario.
* **Autenticación de dos factores:** El sistema pedirá un código de verificación al momento de acceder los administradores al panel de control.
* **Contenido Educativo:** El sistema tendrá un catálogo de recursos educativos y consejos para los entusiastas en la jardinería.
* **Sistema de suscripciones:** El sistema tendrá suscripciones por nivel sin fines de lucro
* **Búsqueda Avanzada:** El sistema tendrá una búsqueda de productos a través de distintos filtros y opciones.
* **Visualización de Datos:** El sistema tendrá un apartado para la visualización de datos estadísticos de la plataforma.
* **Identificación de Transacciones Fraudulentas:** El sistema tendrá un apartado donde realizará un análisis e identificación de transacciones fraudulentas.
* **Asistente Virtual:** El panel de control tipo Dashboard tendrá integrado un asistente virtual que ayudará en algunas funcionalidades del sitio, y proporcionará información para ayudar a las decisiones de negocio.
* **Creación de Posts:** El sistema contará con un apartado para la creación y vista de Posts de la comunidad relacionado al tema de la jardinería, donde los usuarios podrán subir sus propios Posts, también se integrará algún servicio externo para las noticias relacionadas con la jardinería.
* **Sistema de Reviews:** El sistema permitirá a los usuarios crear reseñas a los distintos productos, con un rating de estrellas y un comentario.

**3.1 Requisitos comunes de las interfaces**

Los requerimientos comunes que deberán contar las interfases es que el formato de letra y el tamaño de la letra sea lo más legible y legible para la mayoría de usuarios.

El tamaño de los botones de las interfaces no deben de ser muy pequeños ni muy grandes, para que no haya problemas al momento de ocupar una funcionalidad.

El sistema debe contar con un modo oscuro o modo dark, para poder ayudar a relajar la vista si se está mucho tiempo en frente de una pantalla.

**3.1.1 Interfaces de usuario**

El diseño de la interfaz debe estar acorde a los colores de la empresa y relacionado con las plantas o jardinería, como colores relacionados con la naturaleza, como el color verde y algo de azul. Resaltando en las funciones internas del programa las cuales estarán resaltadas presentadas en la parte principal de la página.

En el apartado del panel de control y gestión se encontrará con datos estadísticos de la plataforma, con gráficos interactivos para una mejor visualización de los datos. El diseño estará acorde a la empresa, no tendrá colores relacionados a la jardinería.

**3.1.2 Interfaces de hardware**

* El hardware debe ser un computador con un procesador de uso general con una potencia de 2 Ghz, ya sea Intel o AMD
* El dispositivo debe contar con un conector Ethernet para la conexión de internet.
* La computadora debe contar con 30GB de almacenamiento en disco.
* La computadora debe contar con mínimo de 4gb de memoria ram. Recomendado tener 8gb

**3.1.3 Interfaces de software**

* Navegador predeterminado: Se requerirá el uso de un navegador para el ingreso y poder utilizar la página web.
* Sistema operativo: Se requerirá tener algún sistema operativo para el funcionamiento de la máquina y de la compatibilidad con la aplicación, el sistema operativo tiene que ser Windows 10 u 11, sistemas basados en UNIX o Linux, Android e IOS.

**3.1.4 Interfaces de comunicación**

La plataforma se comunicará entre los distintos sistemas utilizando el protocolo HTTPS, Hyper Text Transfer Protocol, este protocolo nos servirá para el envío seguro de la información entre las distintas aplicaciones que tengamos desplegada. También se utilizará el protocolo SMTP para el envío de correos electrónicos a través de gmail.

**3.2 Requisitos funcionales**

**Como principales funcionalidades que tendrá nuestro sistema serán:**

**1.- Autenticación de doble factor**

**Descripción**: Para los usuarios de administración se les pedirá una clave de doble factor que se le enviará por vía SMS o por correo electrónico

**2.- Crear Post**

**Descripción**: Los usuarios que estén registrados en la plataforma podrán crear post y publicarlos cuando quieran

**3.- Crear Curso**

**Descripción**: El sistema tendrá un apartado para la creación de cursos o recursos de aprendizajes para los aficionados a la jardinería, este apartado sólo será permitido a los usuarios con el permiso o rol de profesor

**4.- Identificación de Transacciones Fraudulentas**

**Descripción**: El sistema tendrá un apartado para analizar las transacciones de los clientes que hayan realizado, en busca de transacciones fraudulentas, esta funcionalidad solo está permitida a los usuarios de administración

**5.- Escaneo QR**

**Descripción**: El sistema contará con un apartado donde se escanear un QR para la validación de entrega, para así no solicitar el rut, o algún dato personal del cliente.

**6.- Validación de Entrega**

**Descripción**: El sistema validará a través del escaneo de un QR, si los datos del cliente y de la orden de compra son correctos.

**7.- Exportar PDF**

**Descripción**: El sistema tendrá la opción de exportar la información del escaneo de las transacciones mediante PDF

**3.3 Requisitos no funcionales**

**3.3.1 Requisitos de rendimiento**

* El sistema debe demorar no más de 15 segundos al momento de cargar la información necesaria para el funcionamiento.
* El sistema debe soportar una cantidad mínima de 100.000 usuarios activos instantáneamente, sin tener pérdidas de rendimiento.
* El sistema debe soportar una cantidad mínima de 10.000 transacciones por minutos, sin tener pérdidas de rendimiento
* El sistema para no llegar a su máxima capacidad y presentar problemas de rendimiento elimina algunas cuentas de usuarios que no tengan actividad por más de un año..

**3.3.2 Seguridad**

* El sistema implementa un sistema de tokens con Json Web Token para la autenticación de los usuarios a través de tokens de autenticación.
* El sistema contará con permisos y roles para que cada uno tenga accesos a ciertas funcionalidades del sistema.
* El sistema contará con autenticación de dos factores para los usuarios administradores de la aplicación.
* Para la conexión y transferencia de datos entre servicios o servidores se utiliza la criptografía asimétrica para la autenticación entre servidores.
* Se realizará una validación y comprobación de las transacciones realizadas en la plataforma, con el fin de identificar transacciones fraudulentas o fuera de lo normal
* Las contraseñas de los usuarios se guardarán de manera encriptada en la base de datos, ya que, se considera información sensible.

**3.3.3 Fiabilidad**

El software debe tener la capacidad de funcionar correctamente de manera consistente y sin fallos durante un periodo de tiempo prolongado. Asegurando así a los usuarios puedan confiar en su rendimiento.

Para ello el software se debe desarrollar de manera que sea tolerable a los fallos, al momento de ocurrir un error, no se tenga que detener el sistema, ya que, conlleva requerir personal disponible, recursos, tiempo y costos.

Para comprobar que el software cumpla con los requerimientos solicitados se realizarán pruebas unitarias y pruebas End-to-End. Estas pruebas verifican si el software está cumpliendo con sus funciones correspondientes correctamente.

**3.3.4 Disponibilidad**.

La plataforma deberá funcionar correctamente la mayora parte del tiempo, es decir, estar disponible las 24 horas del día y los 7 días de la semana, aproximadamente un 95% de disponibilidad. A menos que ocurra algún fallo o por mantenimiento se tenga que detener o apagar el sistema.

**3.3.5 Mantenibilidad**

En el caso del mantenimiento, se realizará cada mes, para revisar el estado del software si no se encontró algún bug o problema de rendimiento o disponibilidad. Estas tareas serán realizadas cuando ocurra algún problema importante o se llegue a la fecha indicada de cada mes.

El mantenimiento lo realizará un grupo de desarrolladores para ver el estado de los servicios y si están cumpliendo con lo que fueron especificados.

El sistema estará diseñado para que sea escalable en un futuro, esto mejorará el nivel de vida del software, así también mejorará la mantención del sistema teniendo el código mucho mejor estructurado y refactorizado.

**3.3.6 Portabilidad**

* **Porcentaje de componentes dependientes del servidor:** El software debe minimizar la dependencia de componentes específicos del servidor, favoreciendo la modularidad y la capacidad de ejecutar la lógica del negocio en diferentes entornos. Idealmente, no más del 30% de los componentes del servidor deberían depender de una configuración específica, facilitando su migración.
* **Porcentual de código dependiente del servidor:** Se debe mantener un nivel de porcentaje de código lo menos posible, maximizando la reutilización de librerías o funciones entre las plataformas. El servidor no debe superar un nivel de dependencia de no más del 30% del total, permitiendo mayor flexibilidad en la implementación de aplicaciones móviles, web y escritorio.
* **Uso de un determinado lenguaje por su portabilidad:** El lenguaje de programación que se utilizara gracias a su gran portabilidad y compatibilidad con diversos entornos es Typescript, gracias a su gran variedad de frameworks, librerías y herramientas que tiene el ecosistema de Javascript, permite crear aplicaciones web, móviles y escritorio, con una base de código compartida, optimizando la portabilidad del software.
* **Uso de compilador o plataforma de desarrollo:** El software utilizará como entorno de ejecución para las aplicaciones Node JS, este entorno de ejecución es compatible con la mayoría de sistemas operativos que hay actualmente, además el ecosistema de herramientas asociadas se utilizara par asegurar que el software pueda compilarse y ejecutarse en diferentes plataformas con una mínima configuración adicional.

* **Uso de un sistema operativo:** Software será compatible con sistema Windows 10, sistemas UNIX o linux, Android e IOS. La elección de tecnologías multiplataforma permitirá una base de código común, asegurando que el software pueda adaptarse fácilmente a los requisitos de cada sistema operativo

El software contará con un instalador portable en el caso de la aplicación de escritorio, para que sea instalado en los sistemas operativos windows.

Y se contará una aplicación móvil con la APK instalable en un sitio web donde se tiene que escanear el QR para que se descargue el APK y se pueda instalar en el dispositivo móvil. También se tendrá la aplicación en la App Store de Google

**3.4 Otros Requisitos**

Cualquier otro requisito que no encaje en otra sección.

**4. Propuesta de Planificación**

**4.1 Descripción general acerca de la Planificación**

La planificación del proyecto Garden Store estará enfocada en la metodología de trabajo la cual es incremental y esta consta de 4 fases y se seguirá una seguimiento a través del modelo por consiguiente esto se realizará por

1- Fase de Inicio y Planificación: es esencial para garantizar que todos los aspectos del proyecto estén bien pensados y organizados antes de iniciar la ejecución, reduciendo así el riesgo de problemas y asegurando que el proyecto tenga un camino claro hacia el éxito

2- Fase de Análisis y Diseño: es donde se traduce la visión y los requisitos del proyecto en un plan tangible y detallado para la construcción de la solución. Esta fase asegura que todos los aspectos del sistema estén bien definidos antes de que comience la implementación

3- Fase de Desarrollo: es donde se materializa el diseño en un producto funcional. Es crucial que durante esta fase se mantenga un enfoque en la calidad, tanto del código como del sistema en su conjunto

4- Fase de pruebas: para asegurar que el producto final sea robusto, libre de defectos significativos y cumpla con los requisitos especificados. Un proceso de pruebas riguroso reduce el riesgo de problemas en el entorno de producción

Cabe destacar que todas las fases del proyecto estarán guiadas a través de la práctica de la metodología incremental

**4.1.2 Definición del Equipo de Trabajo**

El equipo de trabajo está diseñado para poder cubrir todos los aspectos esenciales del proyecto, desde la planificación y desarrollo hasta el marketing. En donde el equipo estará conformado por el Jefe de Proyecto, QA, Arquitecto de Software y Desarrollador de Aplicaciones Móviles.

Cada uno de estos roles desempeña una función esencial para el desarrollo del proyecto

| **Nº** | **Rol en el Negocio /Cargo** | **Funciones** |
| --- | --- | --- |
| 1 | Jefe de Proyecto | Planificacion y Coordinacion para establecer la hoja de ruta del proyecto y define hitos, Gestión de Recursos y Monitoreo y Control |
| 2 | Arquitecto de Software | Plantea la necesidad de implementar un modelo de calidad, para poder seguir los estándares de calidad del mercado y sacar un producto bien trabajado, aportando información estratégica sobre las necesidades del proyecto y la metodología que se utilizara para el desarrollo |
| 3 | QA | Desarrolla y ejecuta casos de pruebas para asegurar que el proyecto funcione correctamente y cumpla con los requisitos del negocio |
| 4 | Desarrollador de Aplicaciones móviles | Encargado de llevar a cabo la idea de traspaso del proyecto y poder llegar a nuevos usuarios de acuerdo al estándar de dispositivos celulares para los futuros usuarios se espera poder aumentar el rango de accesibilidad de estos |

**4.1.3 Definición de Actividades principales del Proyecto**

Fase de Inicio y Planificación: En este apartado se constaría de actividades de reunión como de programación de itinerarios del equipo de trabajo se realizan diversos documentos los cuales estos se verán reflejados en el cronograma del equipo de trabajo

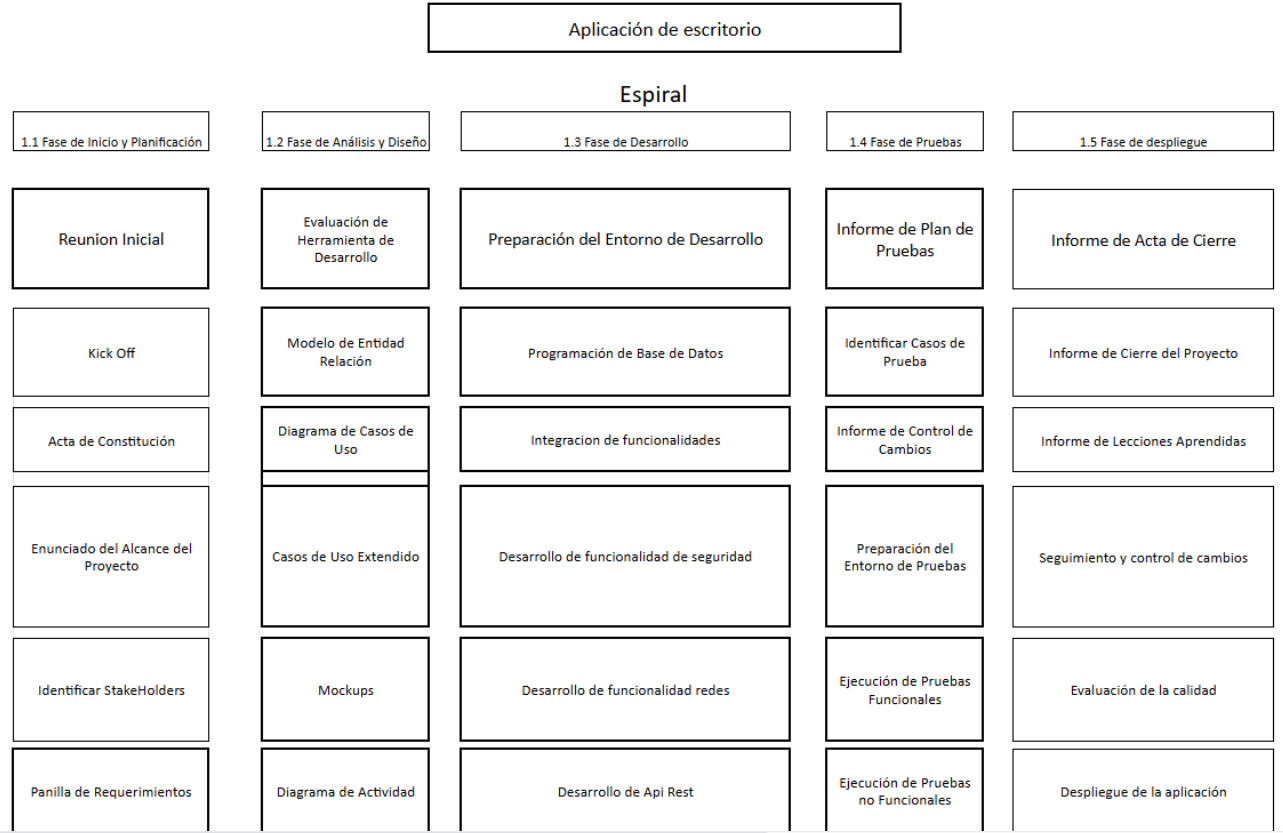
Fase de Análisis y Diseño: Los documentos y actividades que están relacionados con esta área son los diagramas del proyecto junto al manejo de documentos de alto nivel los cuales serán ejecutados o realizados a través de actores del proyecto del area de gestion

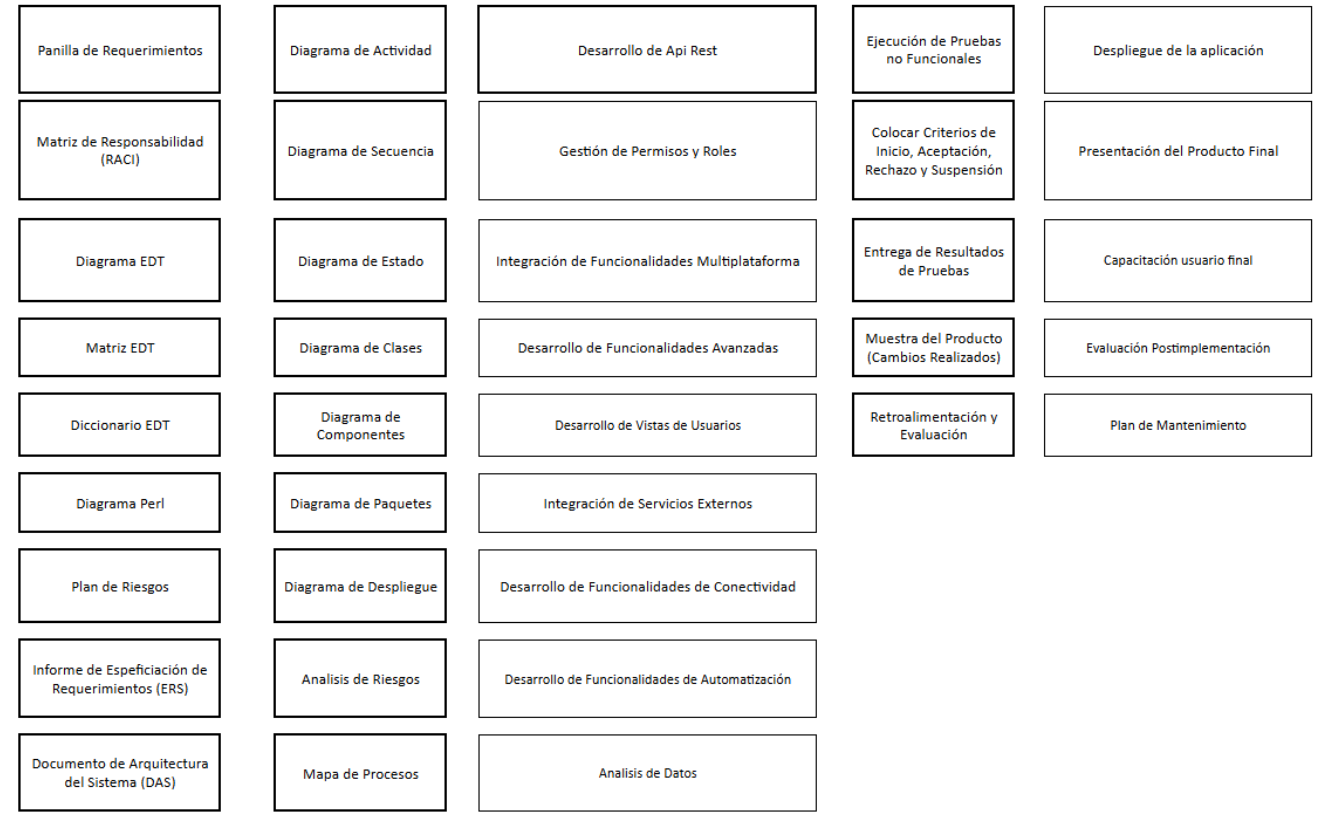
Fase de Desarrollo: Las tareas a realizar de área de desarrollo son creación de api, paneles de administración, Desarrollo de funcionalidades de red y gestión de permisos de red

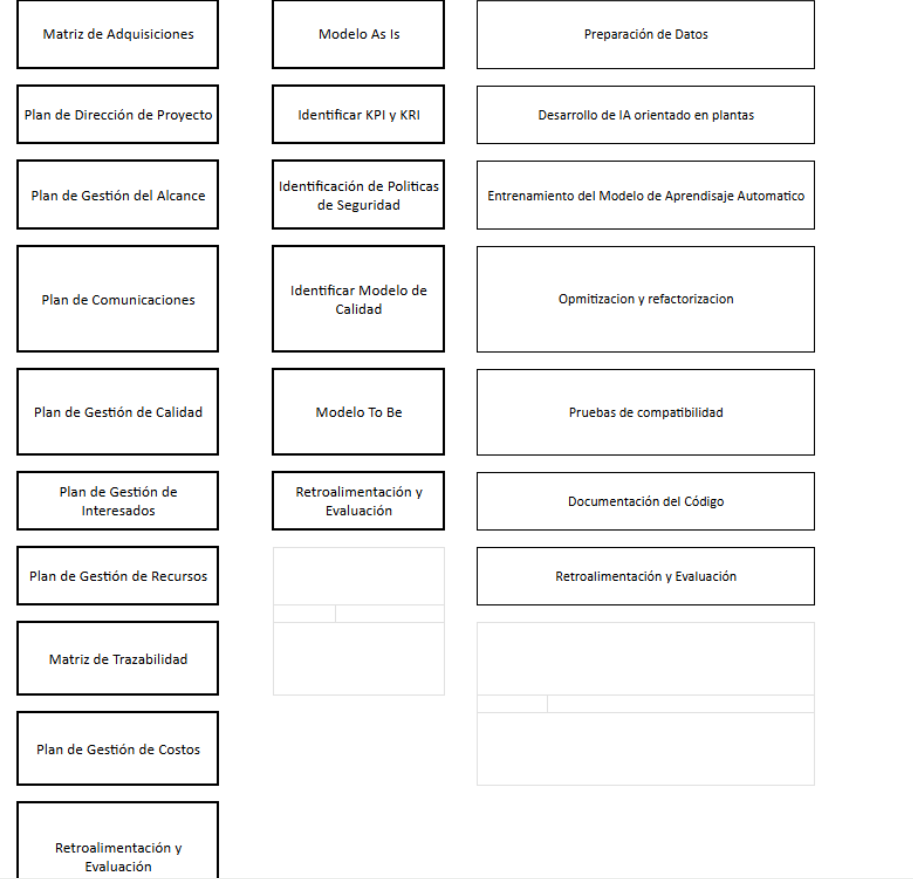
Fase de Pruebas: Se crean informes de plan de pruebas se identificaron casos de pruebas relacionados a las necesidades del proyecto, proceso de criterios aceptación de pruebas además de diversas preparaciones de entorno de pruebas

Fase de Despliegue:

**4.1.4 Diagrama EDT**

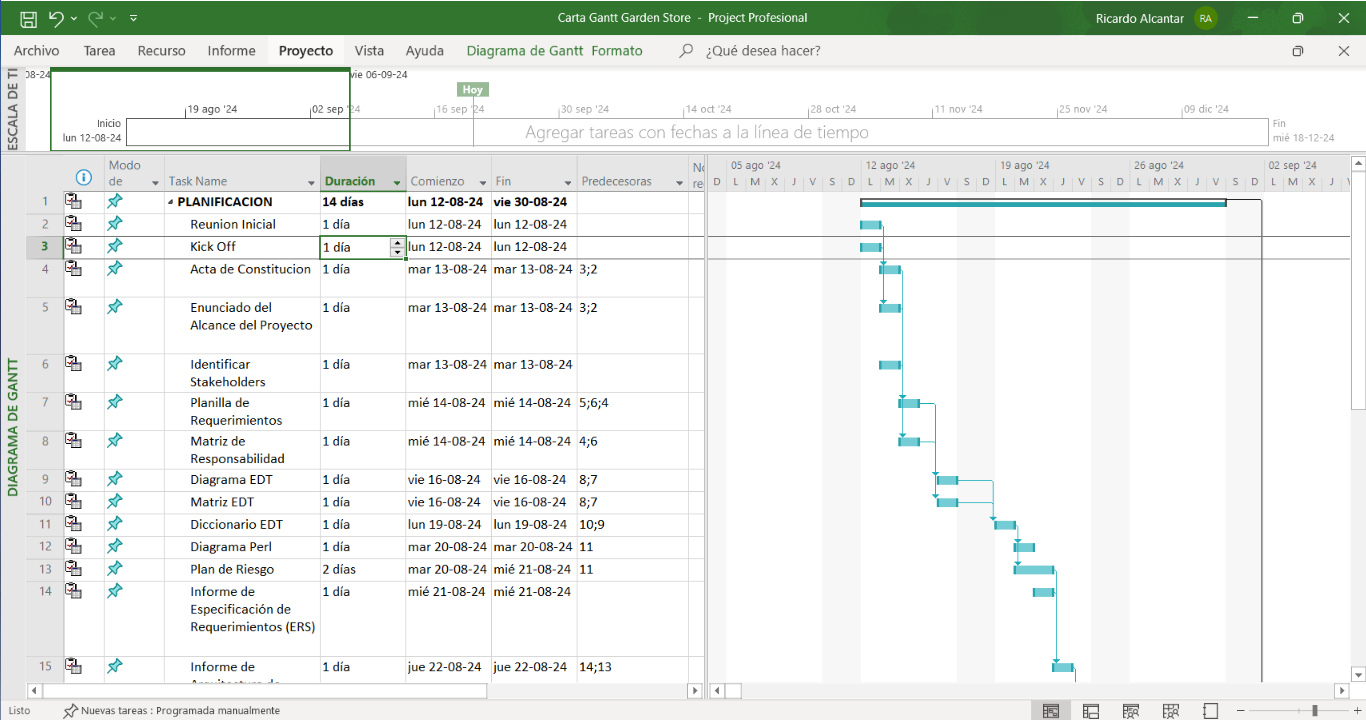


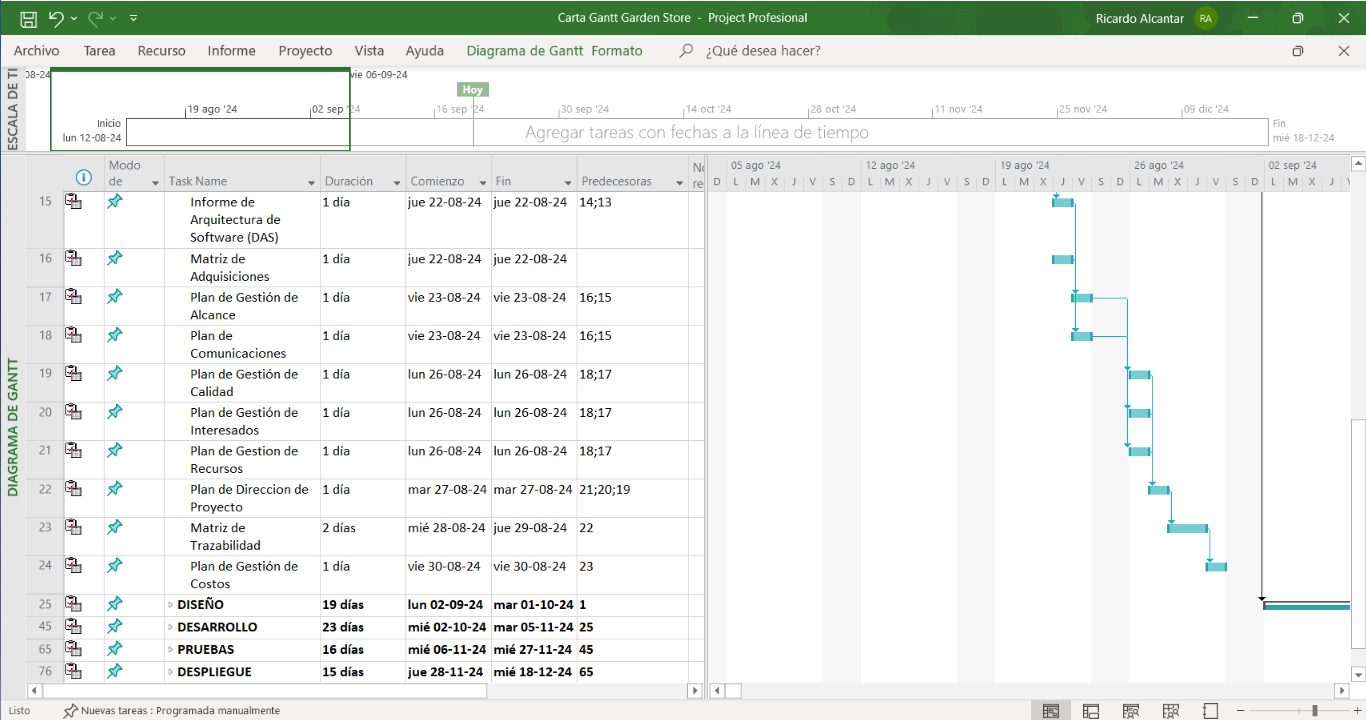


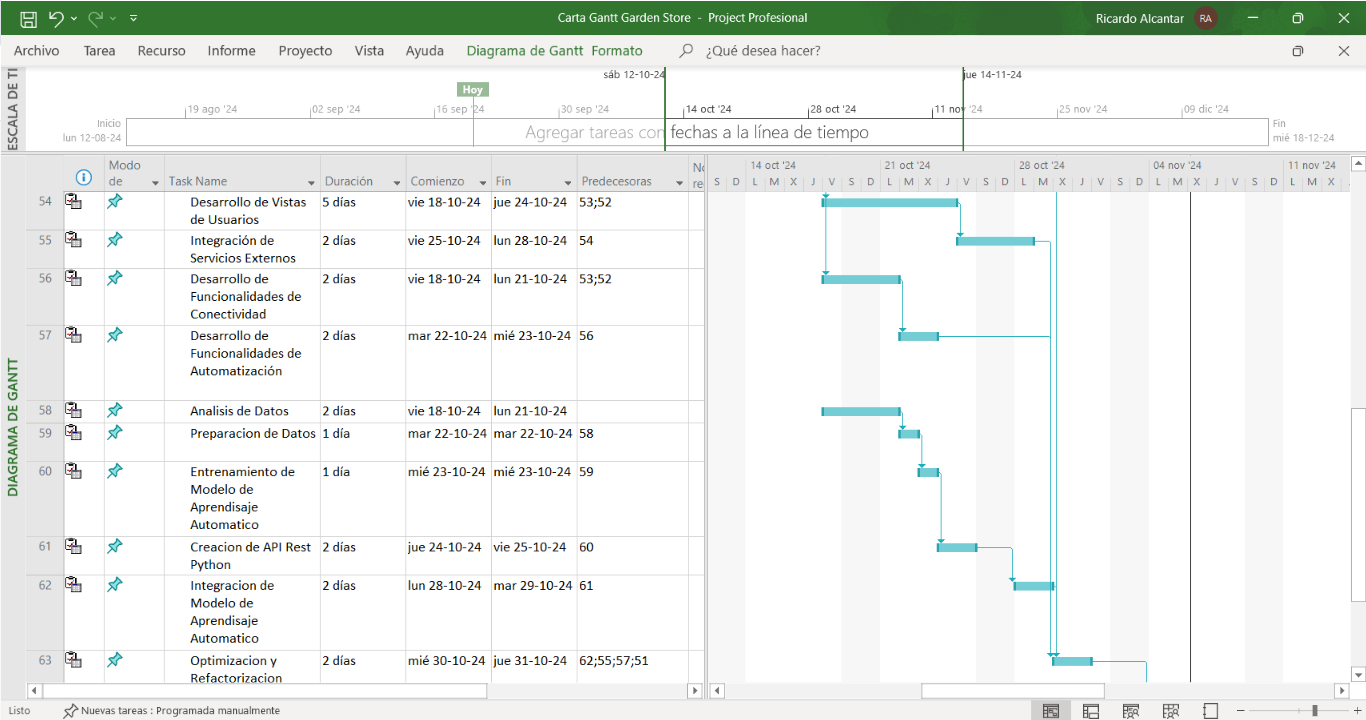


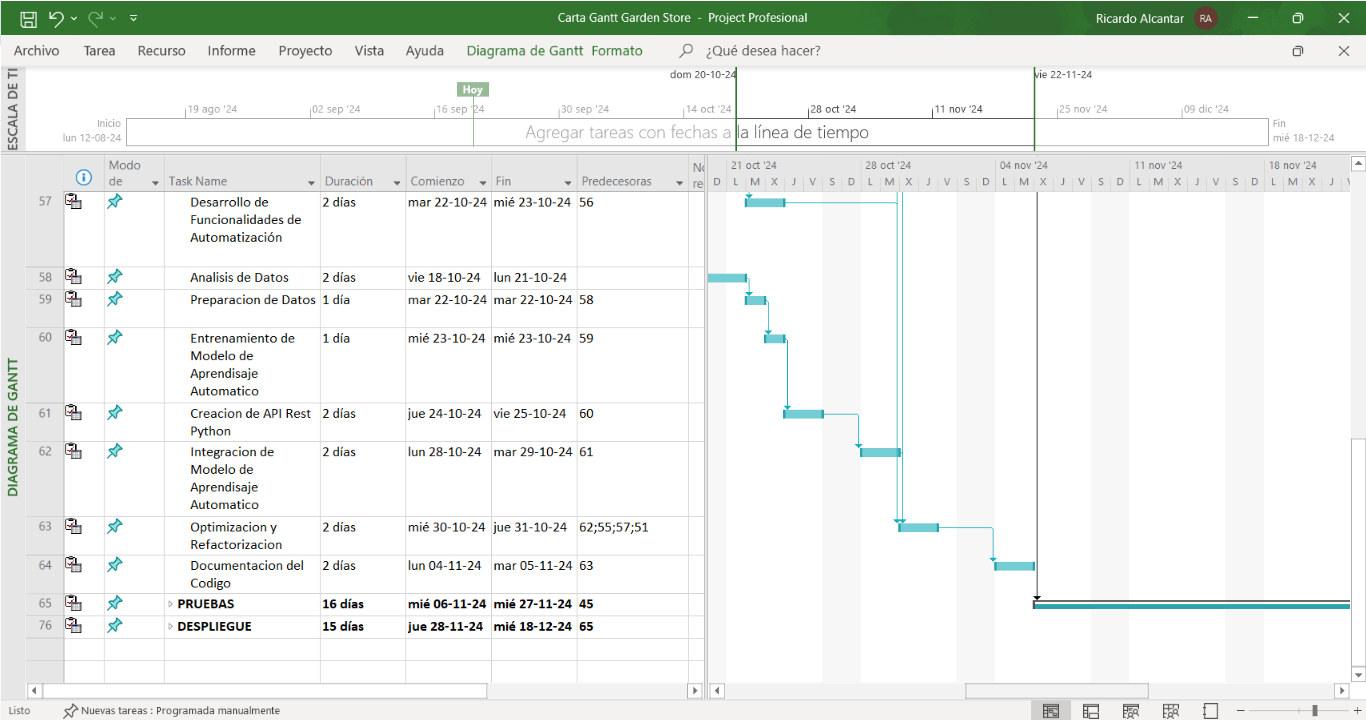
La distribución del diagrama EDT está consolidado por las actividades que se tomaron en cuenta para realizar el proyecto además de seguir una estructura adecuada a las especificaciones realizadas en la matriz EDT

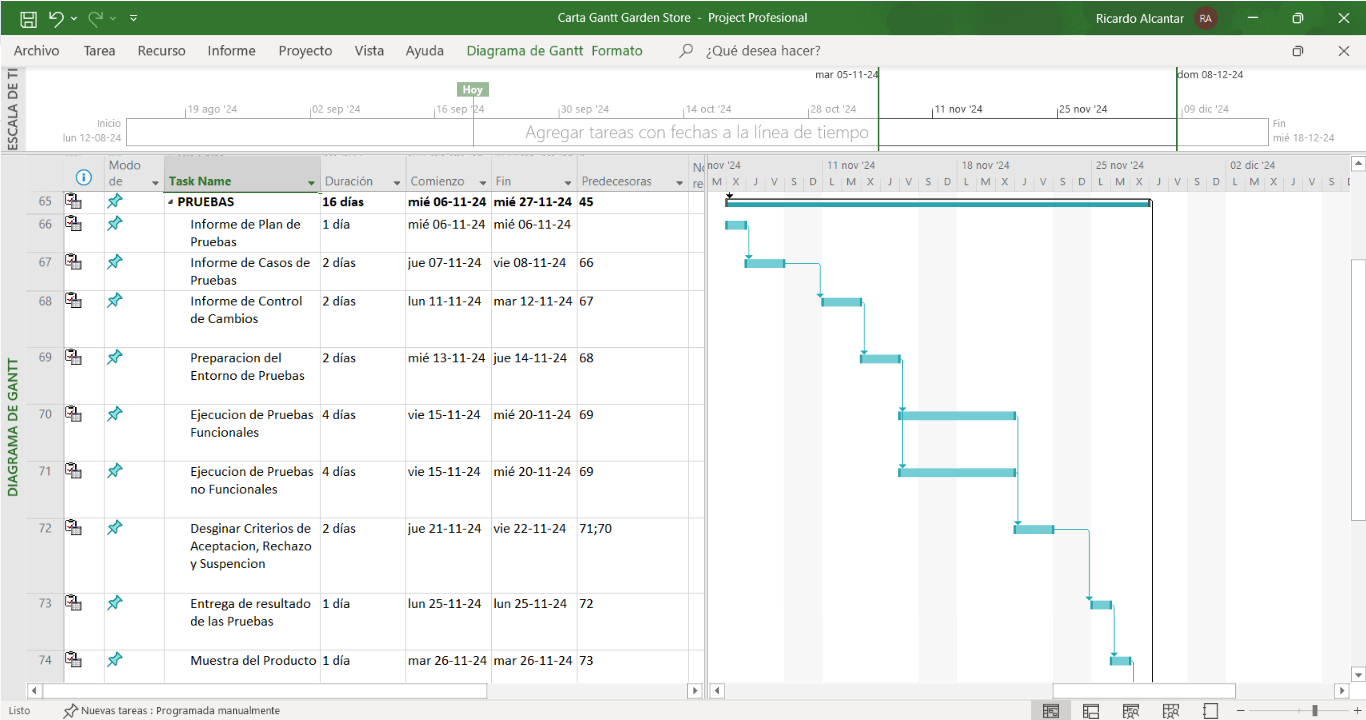
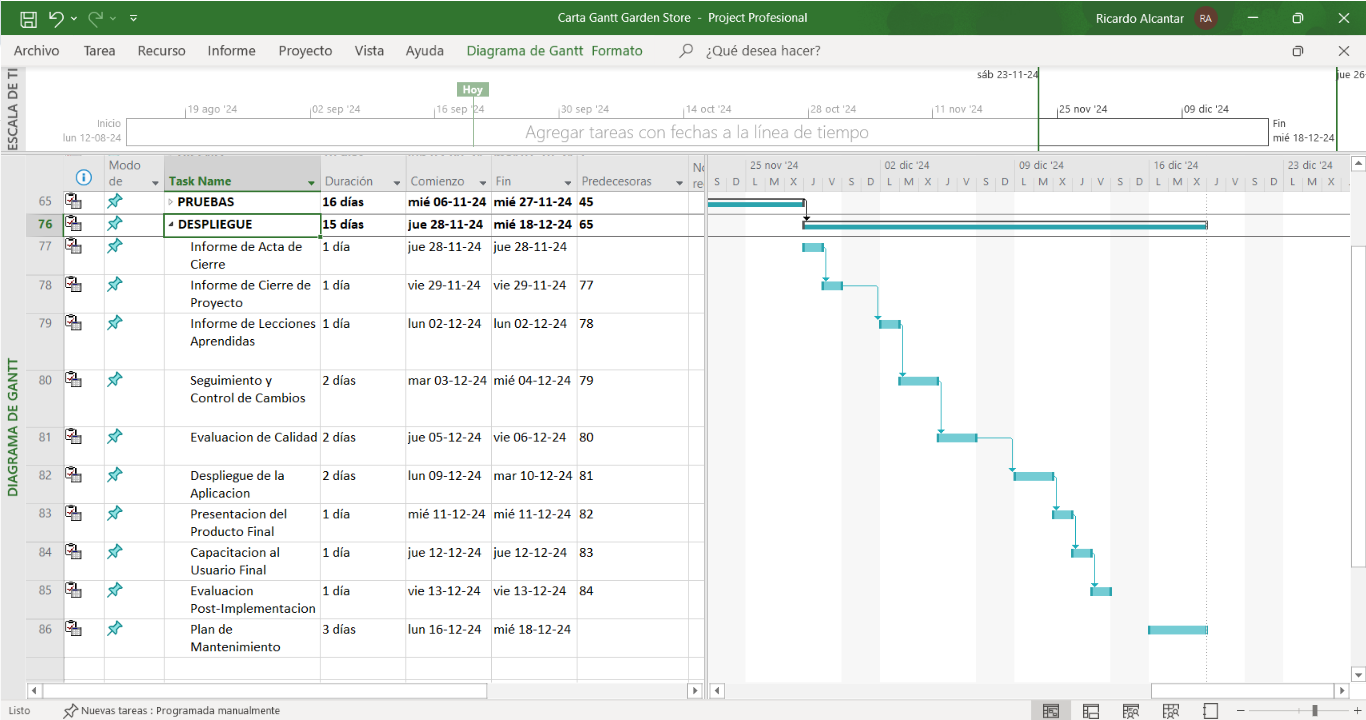
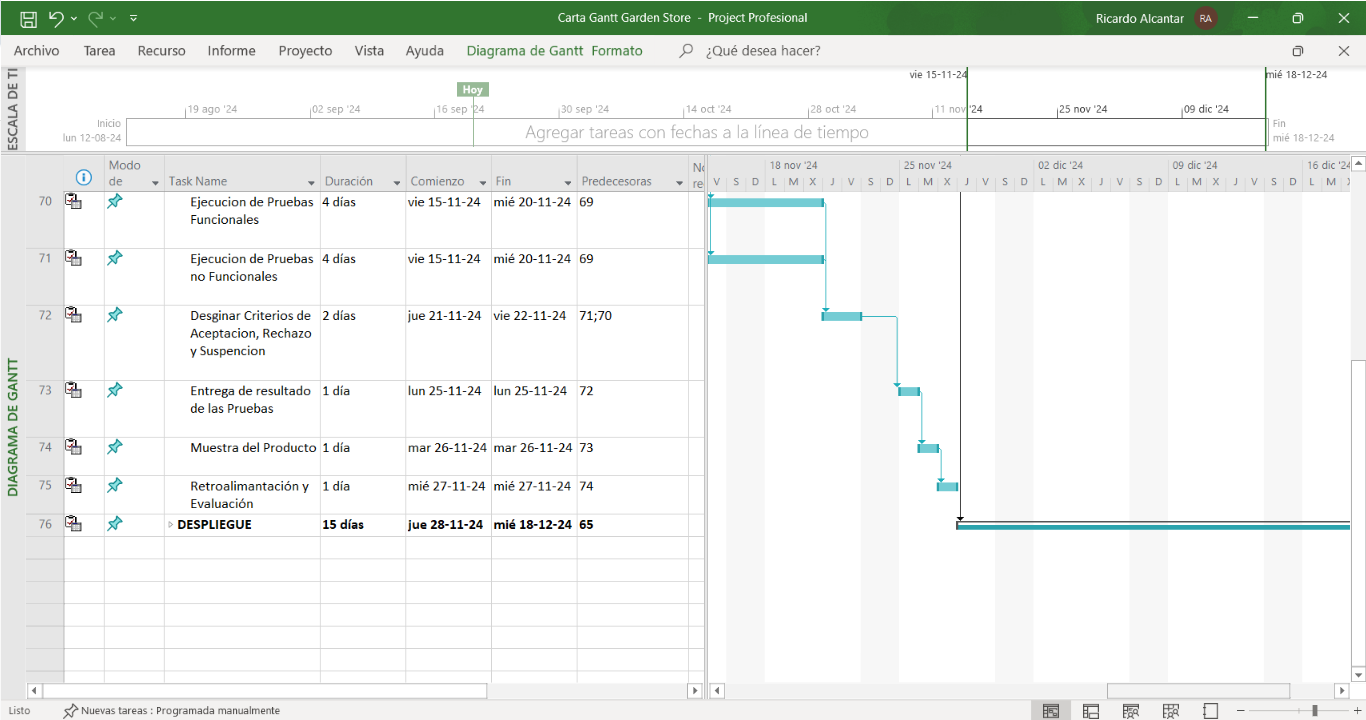
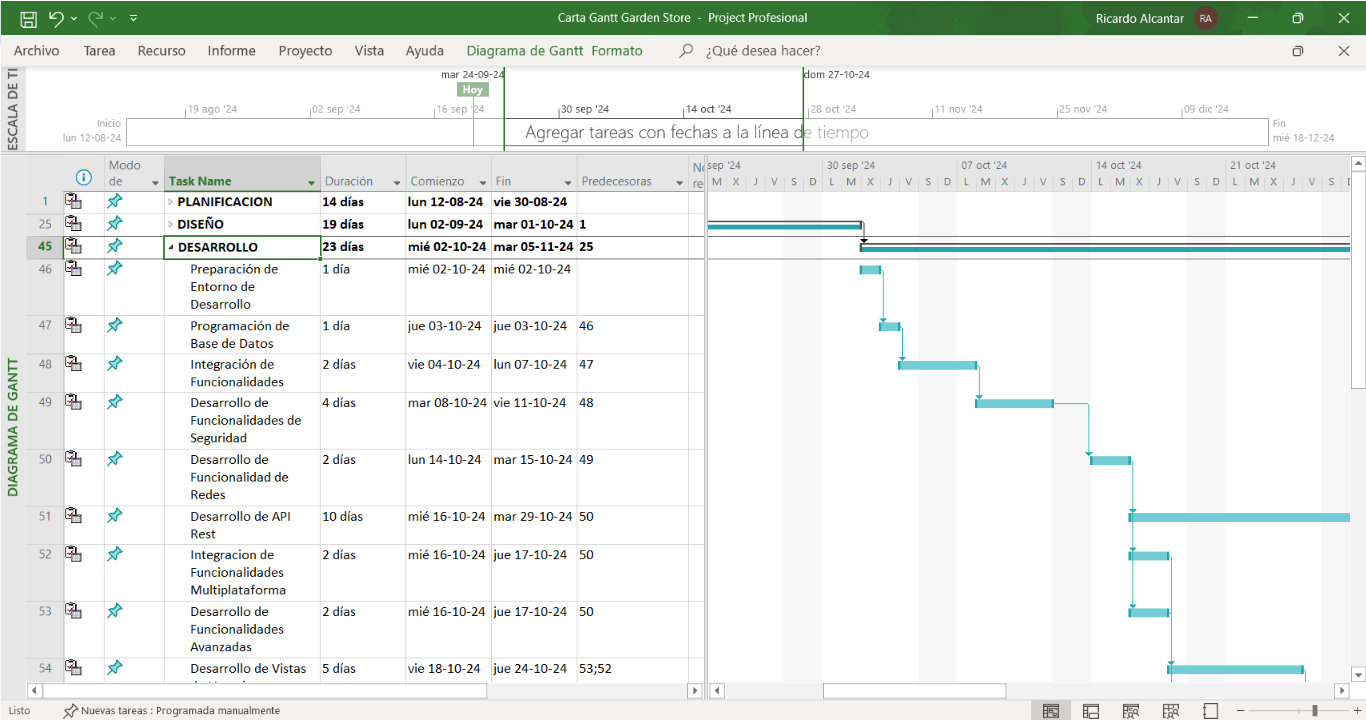
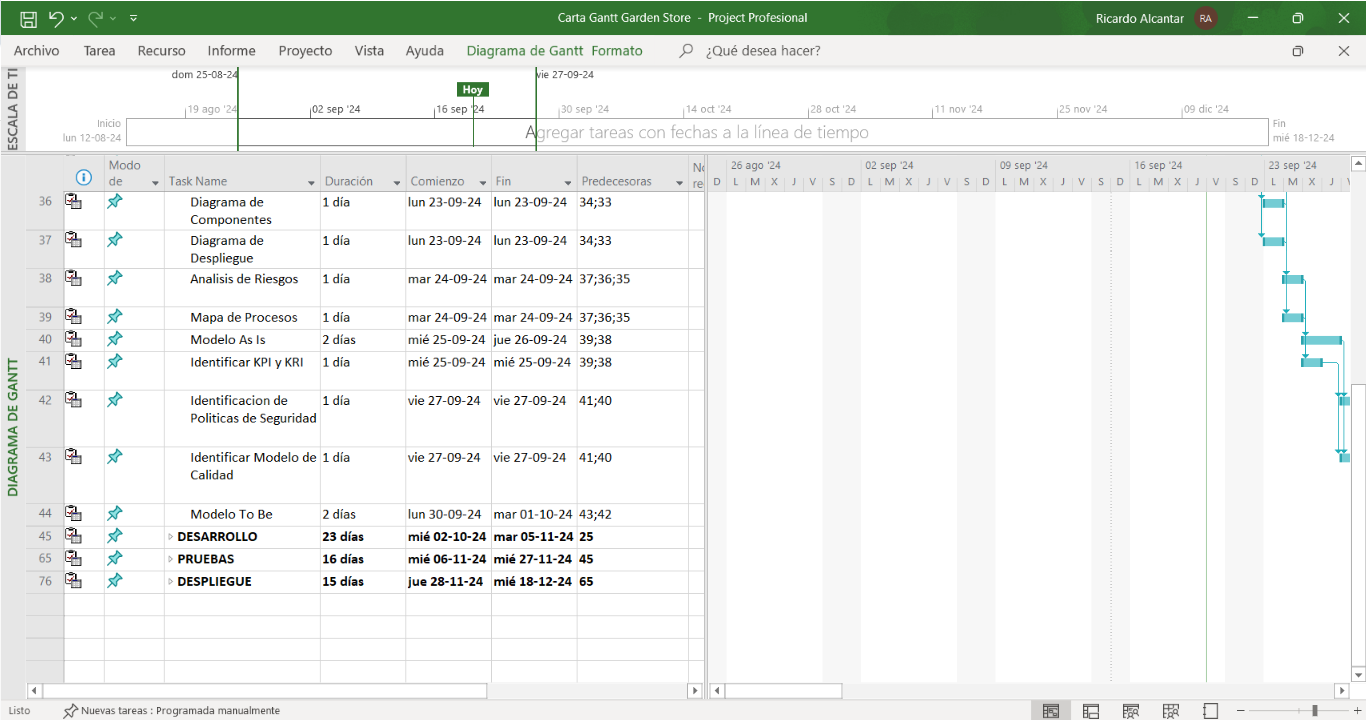
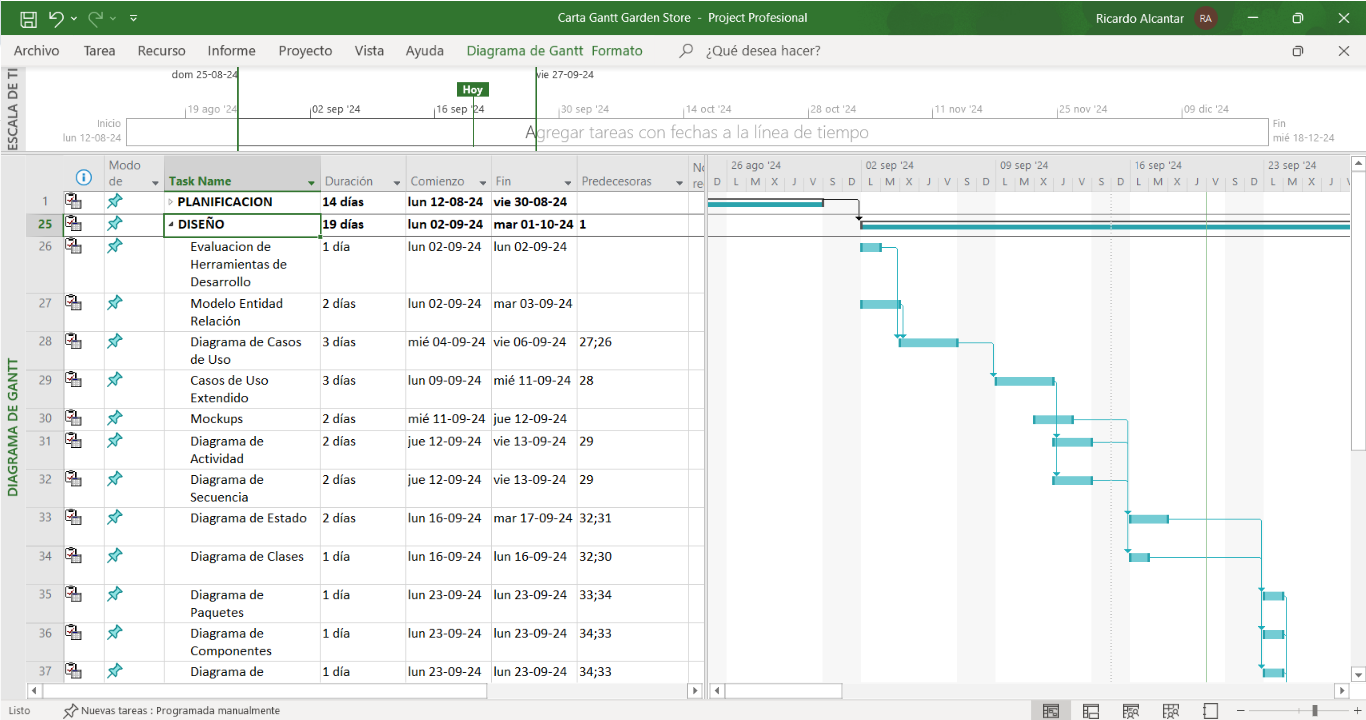
**4.1.5 Carta Gantt**









****

**4.1.6 Resumen Costos del Desarrollo del Proyecto**

[OBS.

Crear una tabla resumen extraída del EDT de cálculo de esfuerzo que desglose los principales costos asociados al proyecto: en base a la Hora hombre y roles profesionales definidos

* Costo total base esfuerzo hora hombre
* Costos por FASE
* Costos por Actor o Rol

]

| Fase de Planificación | $ |
| --- | --- |
| Fase de Diseño | $ |
| Fase de Desarrollo | $ |
| Fase de Pruebas | $ |
| Fase de Despliegue | $ |
| Arquitecto de software / Nestor Aviles | $ 3.300.946 |
| QA / Diego Barrera | $ 900.970 |
| Jefe de proyecto / Felipe Sanches | $ 1.680.042 |
| Desarrollador de aplicaciones / Ignacio Jimenes | $ 763.096 |
| Desarrollador de base de datos / Javier Guzman | $ 276.900 |
| Ingeniero Informático / Matias Arteaga | $ 1.589.652 |
| Desarrollador Frontend / Jose Donoso | $ 112.000 |
| Diseñador / Tomas Campos | $ 175.720 |
| Ingeniero de Redes y Telecomunicaciones / Nicolas Zuñiga | $ 265.842 |
| Desarrollador Backend / Matias Carvagal | $ 2.601.730 |

**4.2 Plan de Control de Cambio**

[Se recomienda primero describir los tipos de cambio que se podrán resolver y sus alcances]

[Insertar Tabla de Control de Cambios]

[ Obs.

Insertar Descripción de los aspectos del desarrollo en los que se permitirá aplicar cambios como parte del Desarrollo del Software definiendo sus alcances y limitaciones asociadas.

El control de cambios es una actividad paralela al desarrollo del proyecto que responde a eventos que surgen del mismo, sea por requerimientos propios del usuario o por mejoras o correcciones detectadas por el mismo equipo del proyecto.

 Se describe de manera independiente de las demás fases de la metodología pues puede ser aplicada indistintamente a proyectos en marcha o proyectos ya implementados, y porque es necesario resaltar su importancia y no relegarla como una actividad posterior al desarrollo, sino reconocerla como una actividad que debe estar definida, presente y es crítica desde el inicio del proyecto.  Deberá describir que tipo aspectos Funcionalidades y no funcionales se podrán modificar con cambio, en que instancia de proyecto se podrán aplicar y que motivos los validarían para ser aplicables y en qué caso no será posible aplicar cambios.

Luego esto se debe complementar con la observación de que en el anexo encontrarán la Planilla de Control de Cambio con los Tipos de Cambio que podrán aplicarse en la cual posteriormente se debe completar la planilla al ejecutarse la instancia. ]

**Documento de Plan de Control de Cambios\***

**5. Anexos**

**5.1 Acta de Proyecto**

Insertar Acta de Constitución del Proyecto

[Acta de Constitución Garden Store.docx](https://docs.google.com/document/d/12eMY0Z86Lo2ibgQ_Ix0tjGh0K-dKNKFS/edit?usp=sharing&ouid=109204094965603786832&rtpof=true&sd=true)

**5.2 Matriz Especificación de Requerimientos**

Matriz en formato planilla sobre la especificación de Requerimientos con su identificador y columnas de datos correspondiente. RF1. O RNF.1

[Requerimientos Garden Store.xlsx](https://docs.google.com/spreadsheets/d/15trf_ctXs_RQOGBebwXqwvK7SR9ipMYq/edit?usp=sharing&ouid=109204094965603786832&rtpof=true&sd=true)

**Matriz de Trazabilidad de Requerimientos\***

**5.3 Diagrama de Casos de Uso General**

Insertar Diagrama de Caso de Uso General.

**Diagrama de Caso de Uso GENERAL\***

**5.4 Planilla Casos de Uso**

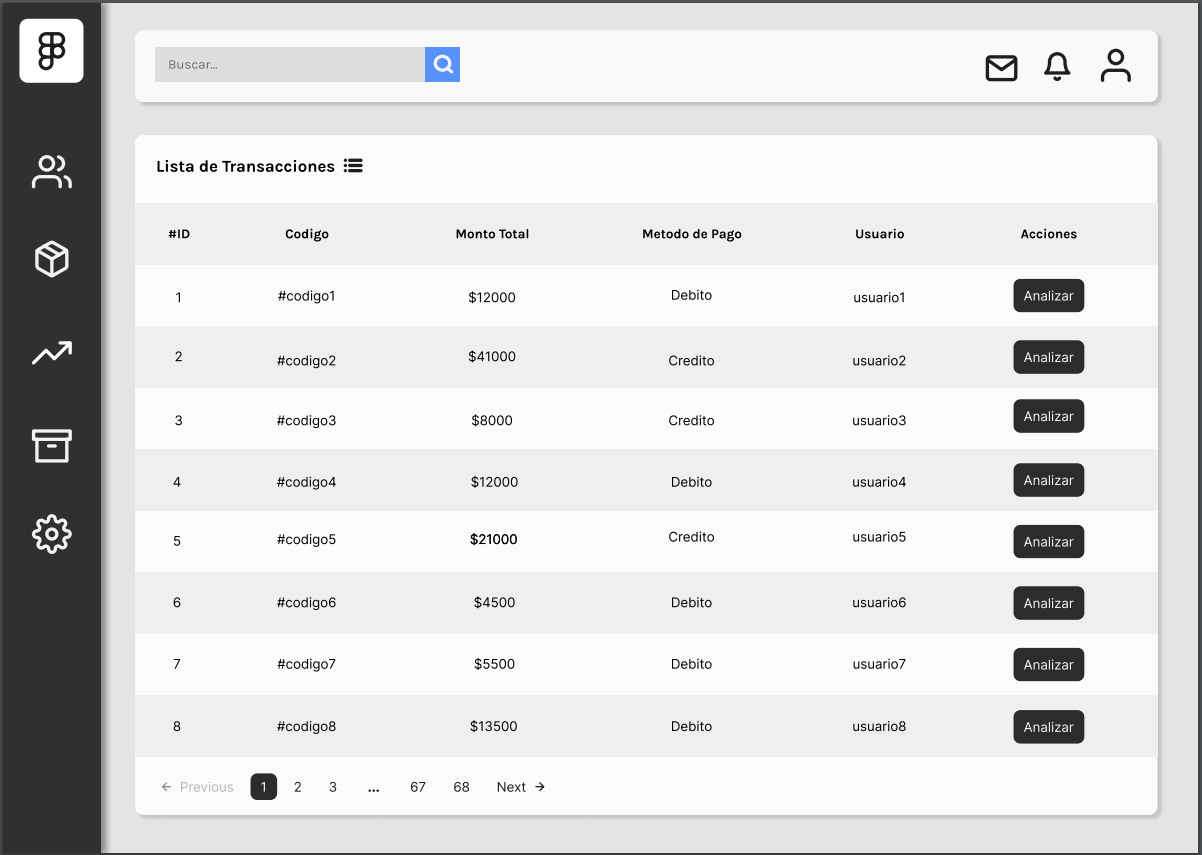
Insertar Planilla detallada de Caso de Uso para cada actor o acción clave del proceso que lleva el sistema.

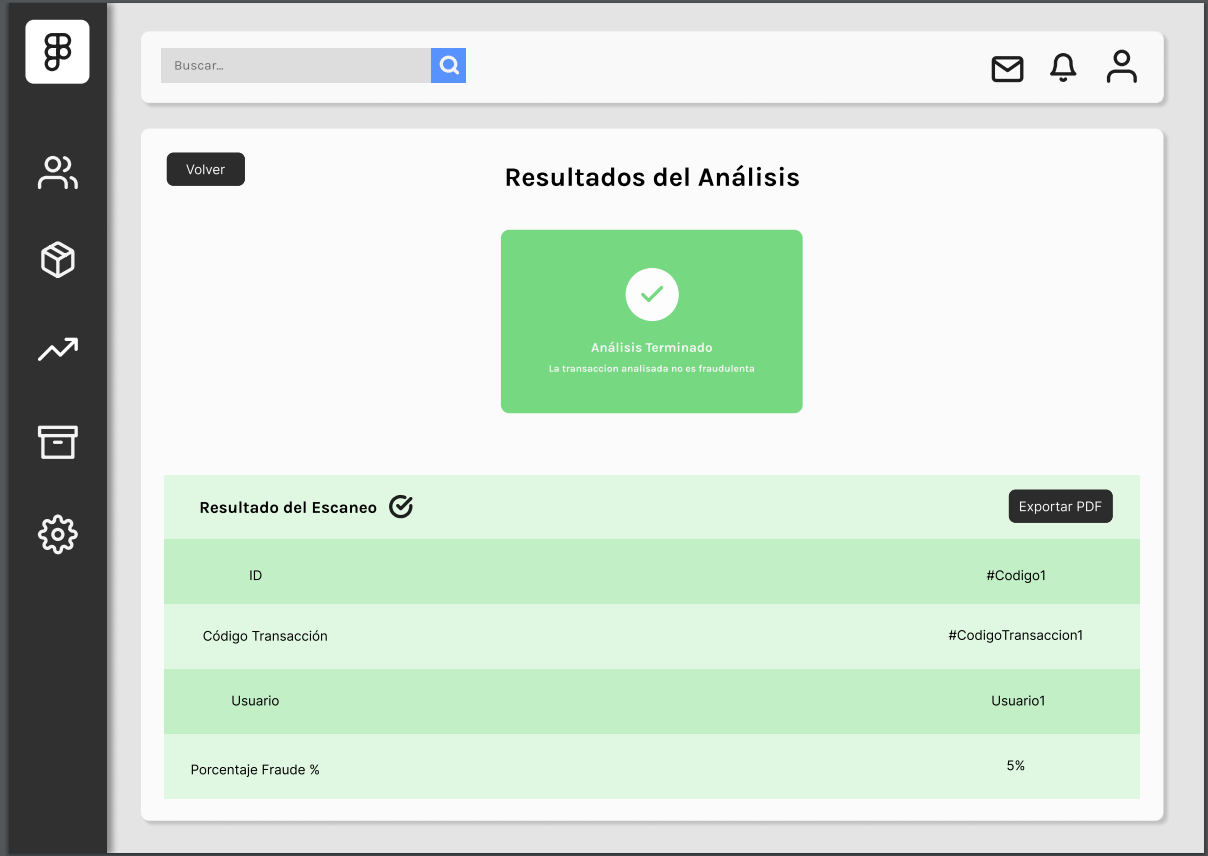
[Casos de uso](https://drive.google.com/drive/folders/15oeNWmc9YJ3iZBer8_8pYDv5EAGgMbGD?usp=sharing)

**5.5 Prototipado de Software**

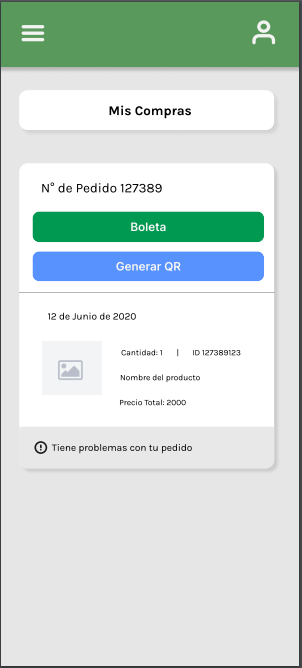
Mockups

1. Sistema de detección de fraude, sistema que analice las transacciones de la aplicación para la detección de fraude utilizando modelos de aprendizaje automático.

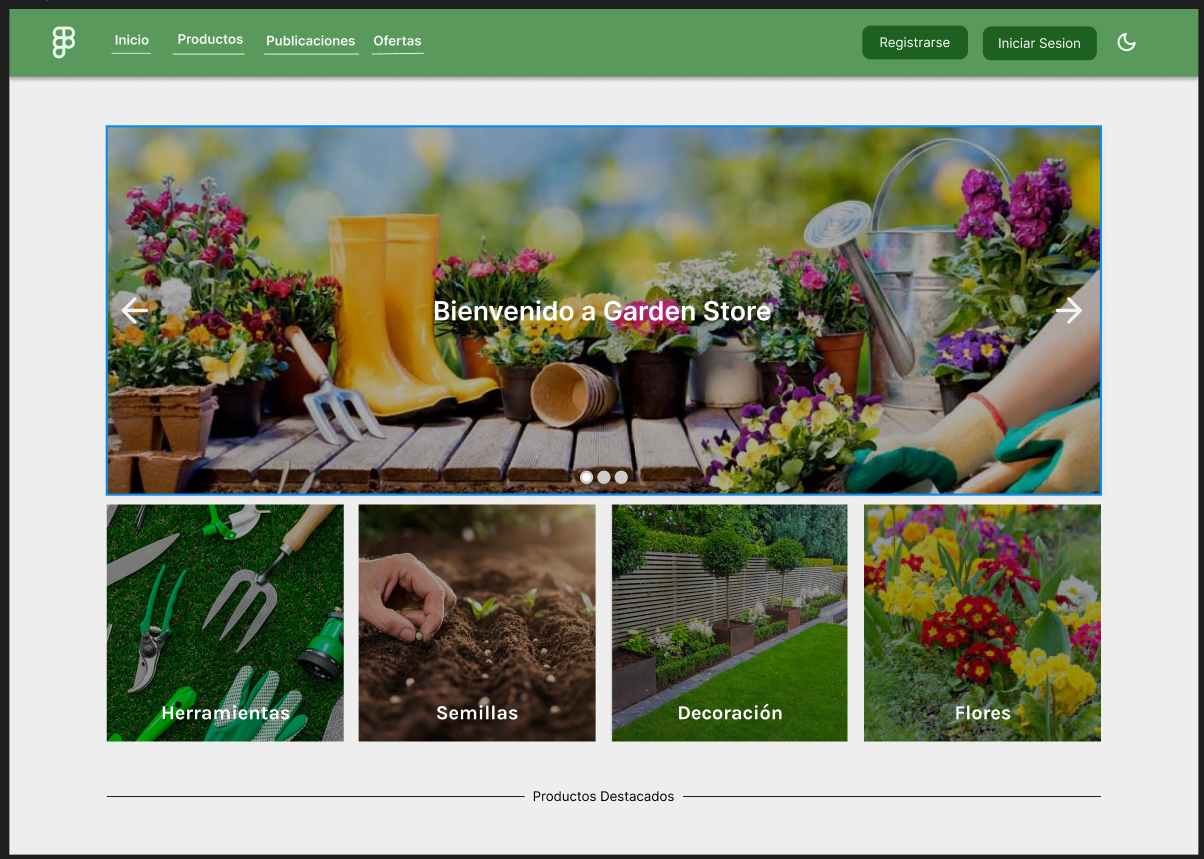


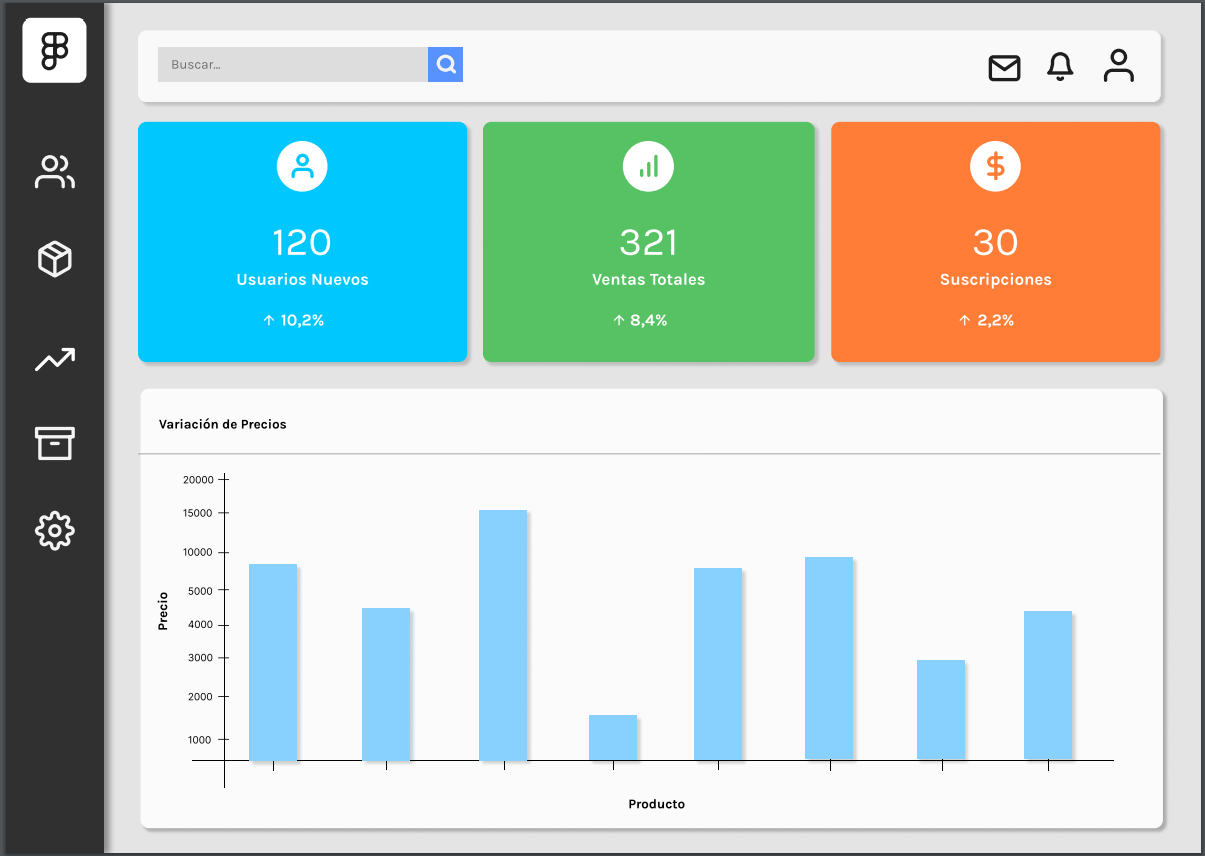


1. Sistema de generador QR dinámico para cada compra de validación de entrega del pedido



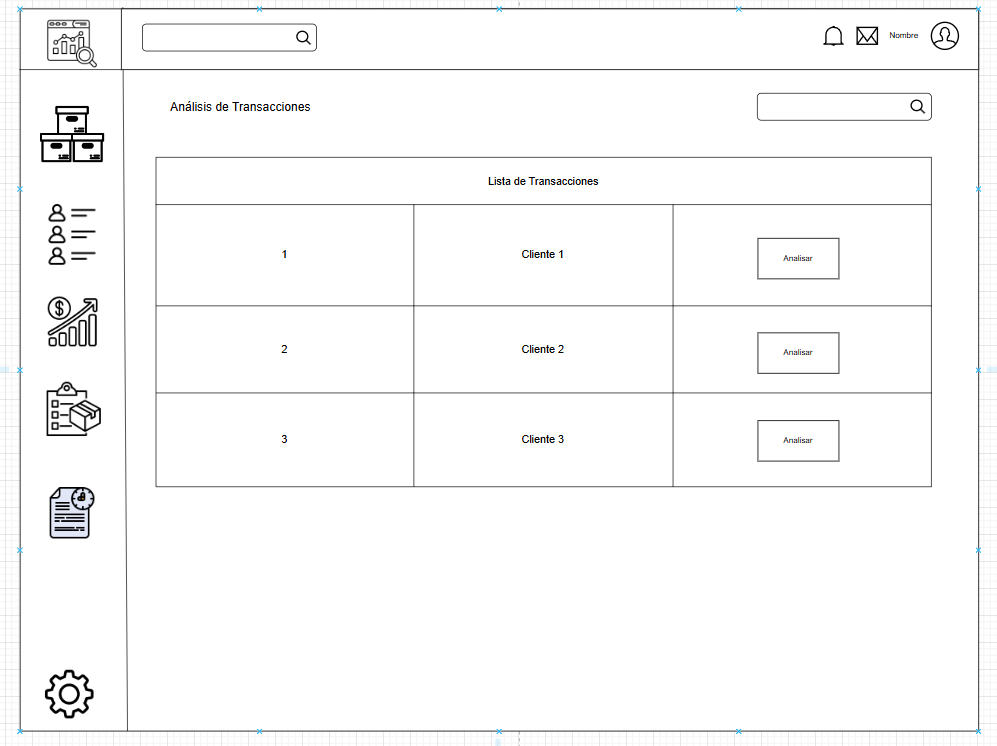




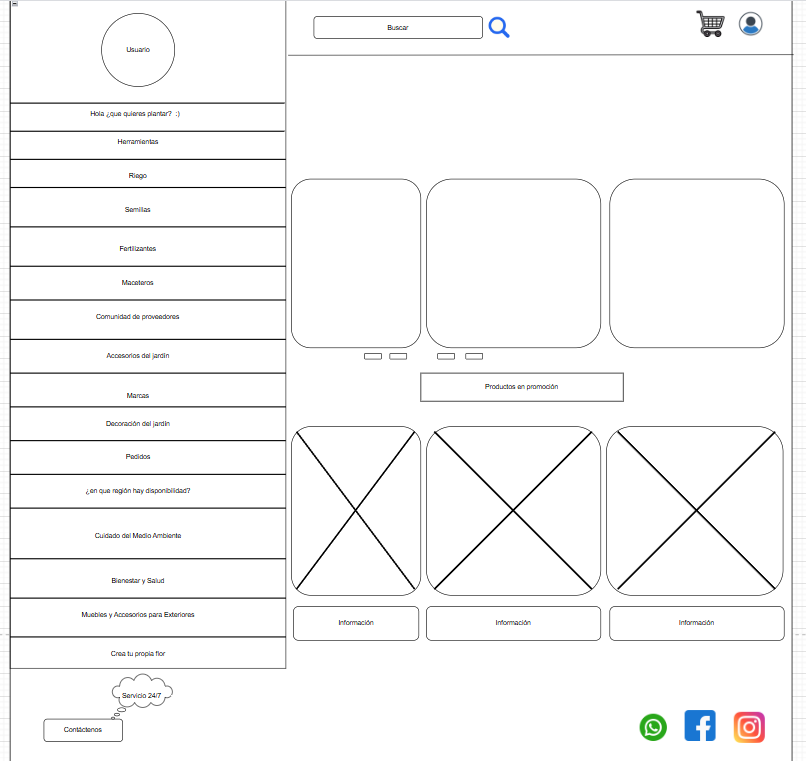


**Wareframe**

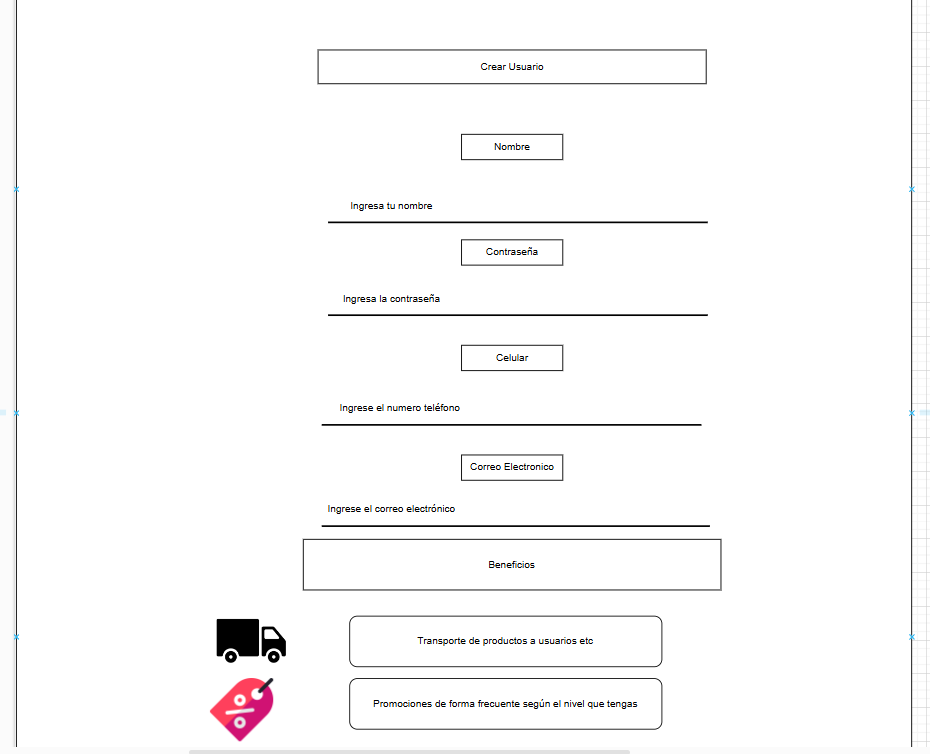
Análisis de transacciones



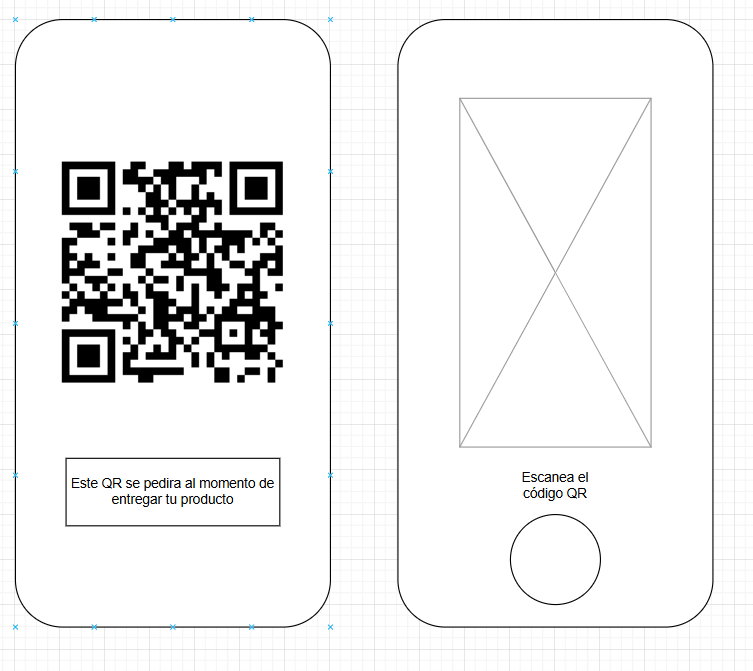
Barra de menú



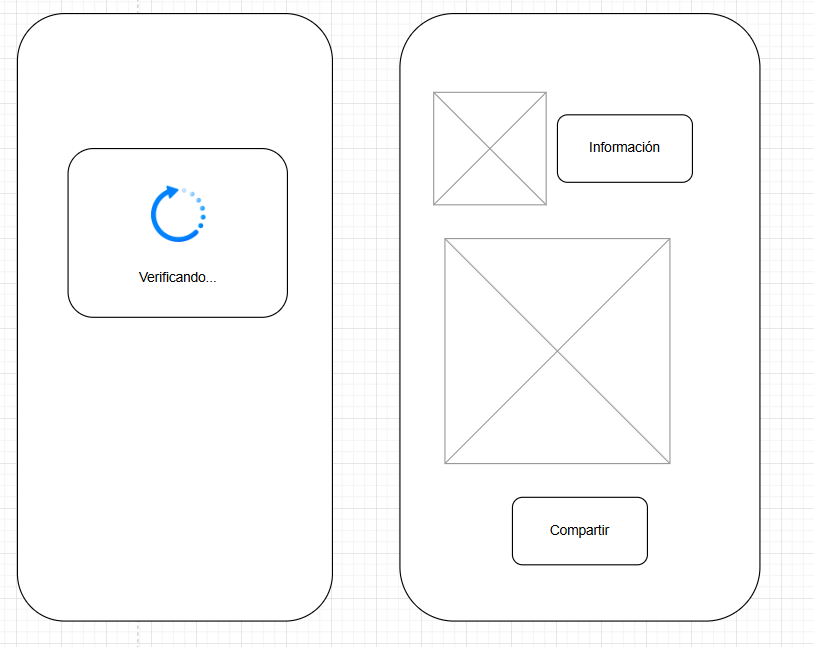
Crear Usuario



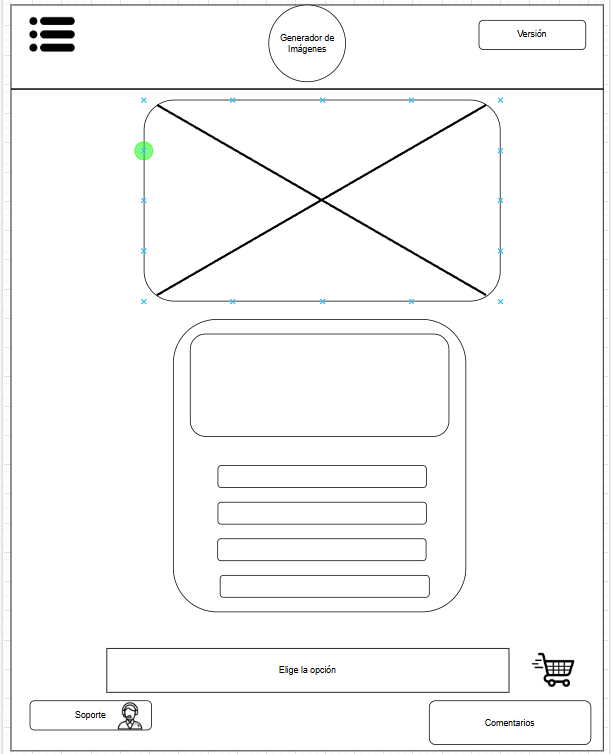
Escaner QR 1



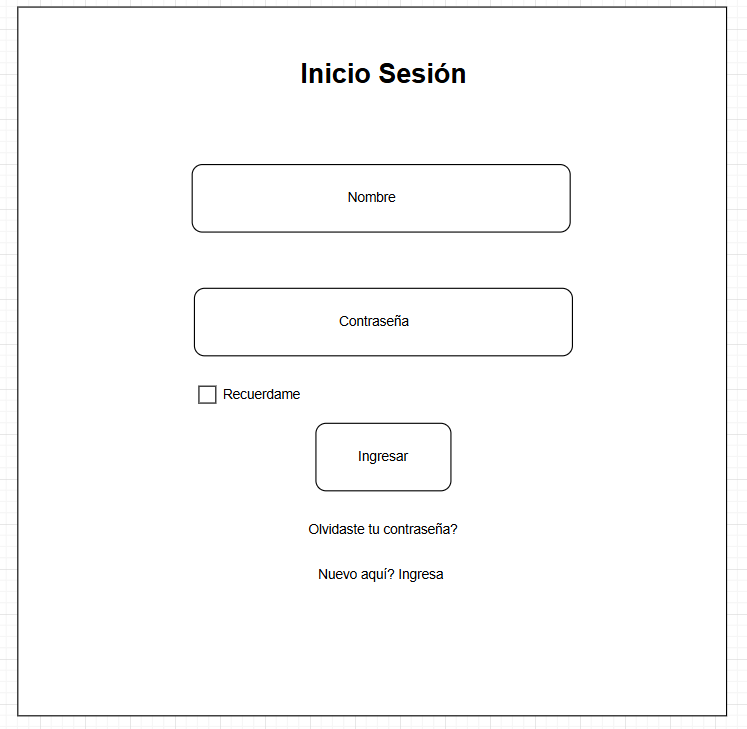
Escáner de QR 2



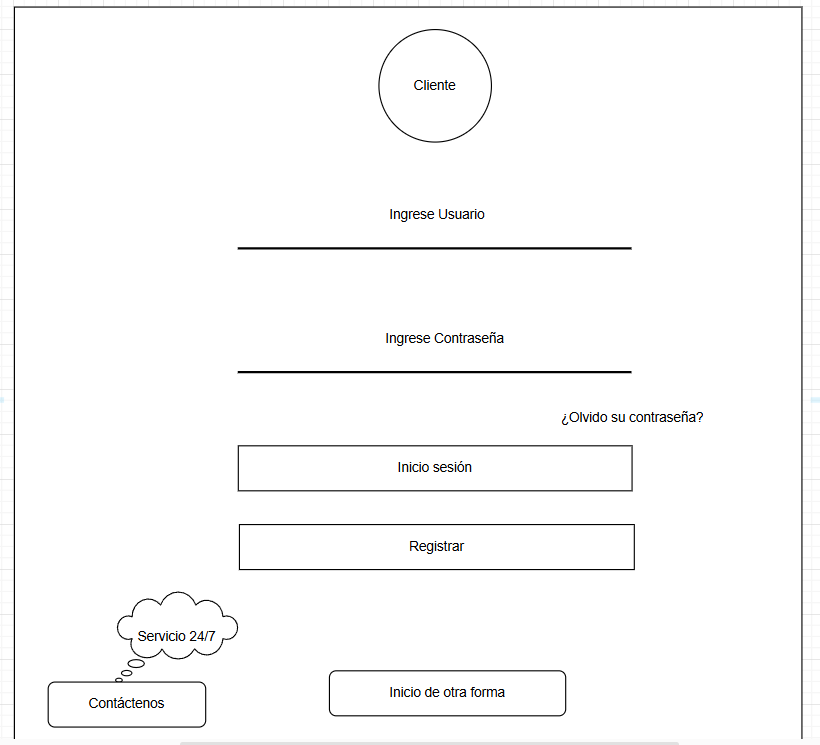
IA de plantas



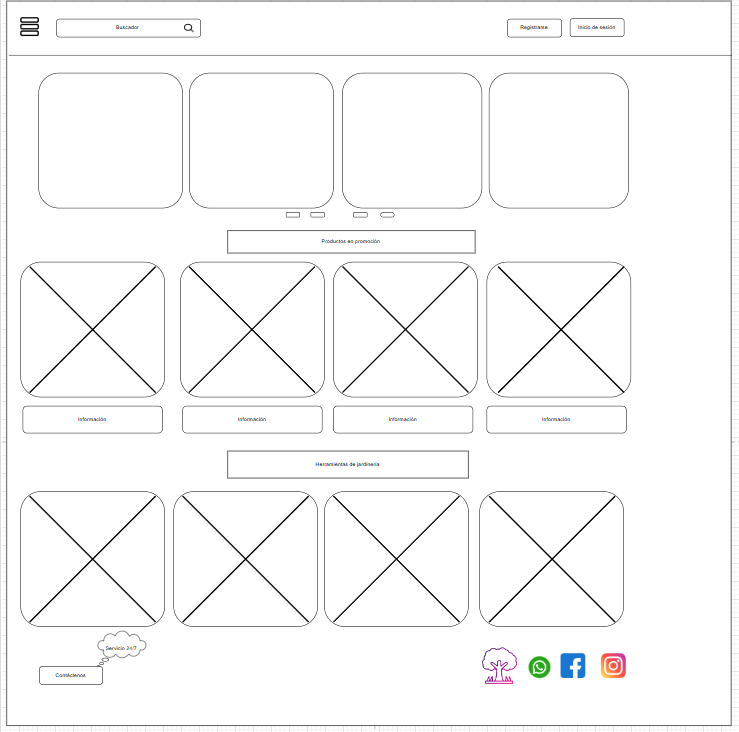
Iniciar Sesión en móvil



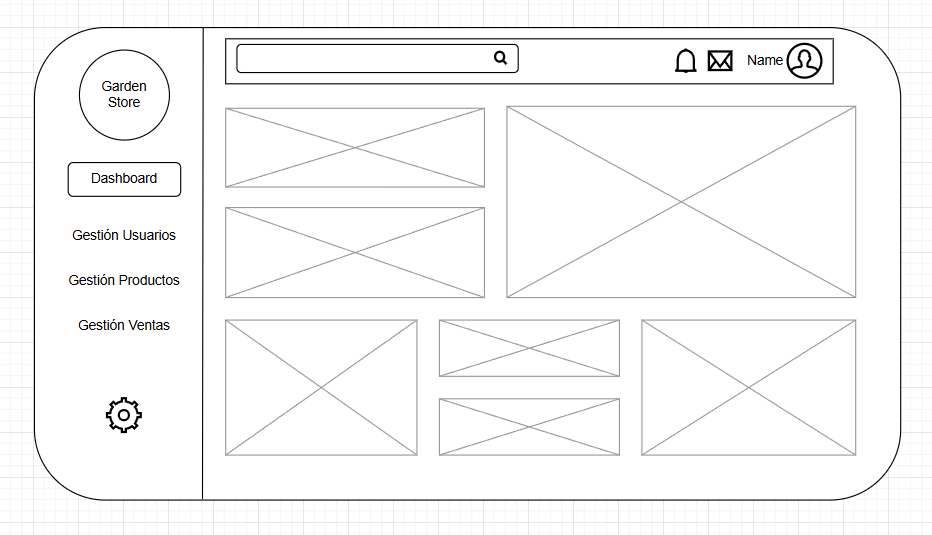
Inicio de sesión web



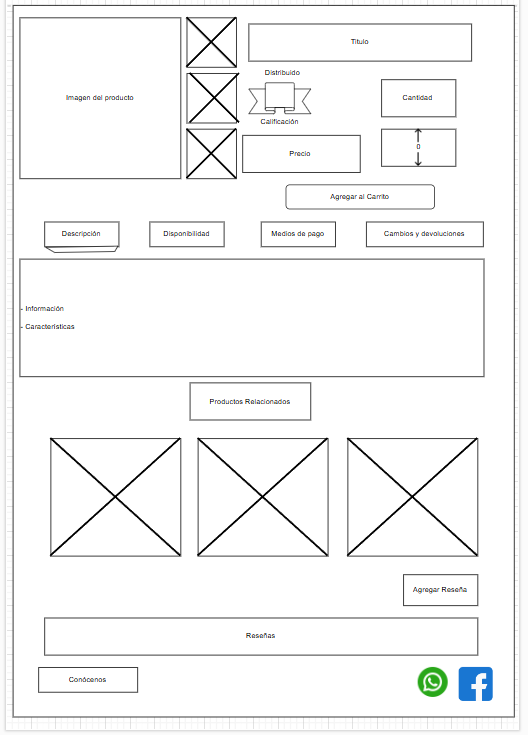
Página de Inicio



Panel administrativo

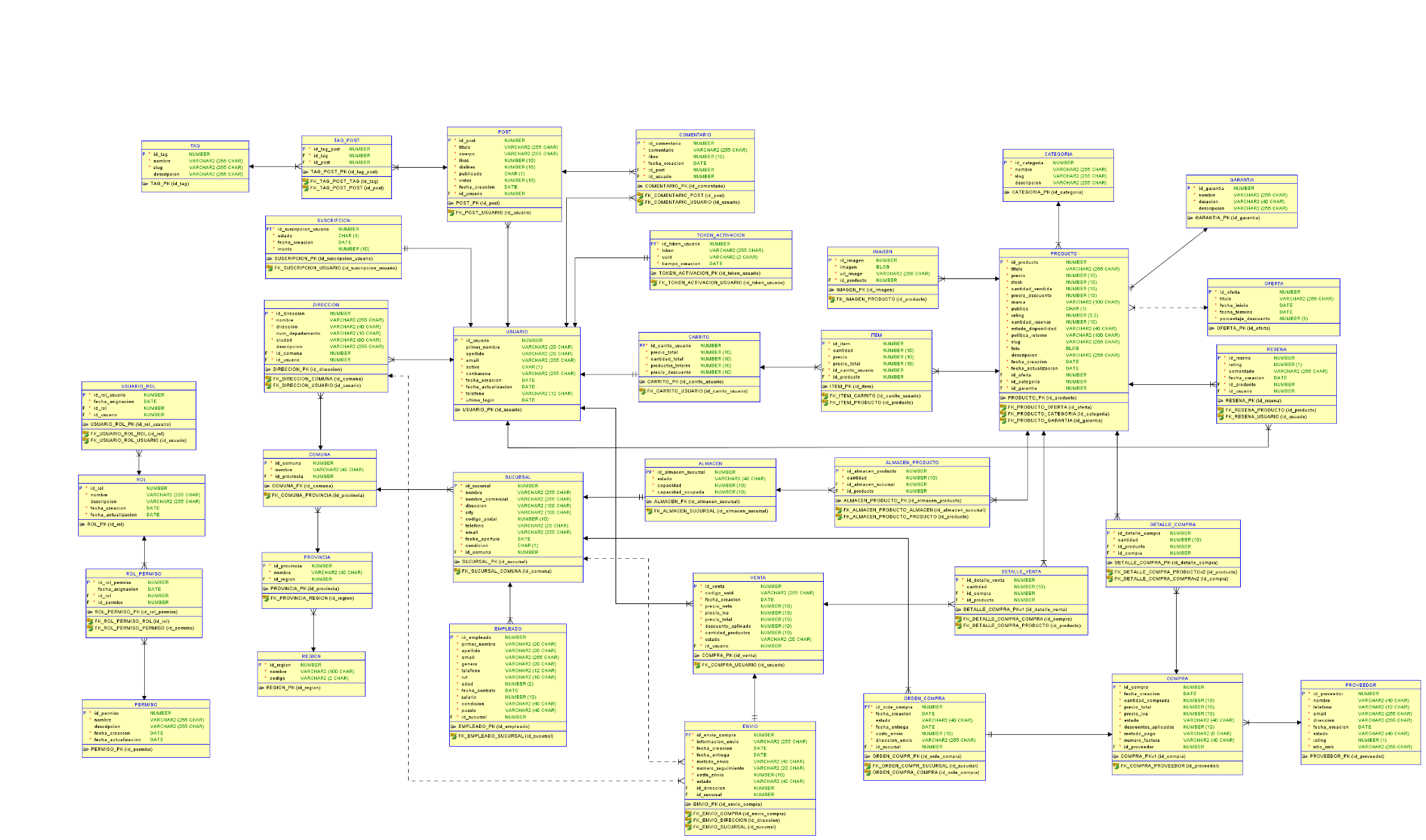


Producto página web



**5.6 Resultado Análisis de Calidad Diagramas Modelamiento**

Insertar Resultado del Análisis de Calidad basado en los estándares y la Planilla de Análisis de Calidad de modelado de Software.



**5.7 Resultado Análisis de Calidad Prototipado No funcional del Sistema**

Insertar Resultado del Análisis de Calidad basado en los estándares y la Planilla de Análisis de Calidad de Prototipo de Interfaz de Usuario.

**5.8 Planilla entregables del Proyecto**

Insertar la Planilla que define los Módulos y Artefactos asociados al Caso de Uso a los que se pueden aplicar cambios en un punto de su desarrollo.

**Matriz EDT\***

**5.9 Matriz de Control de Cambios**

Insertar la Planilla que define los Módulos y Artefactos asociados al Caso de Uso a los que se pueden aplicar cambios en un punto de su desarrollo.

**Matriz de Control de Cambios\***

**5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo**

[Insertar matriz EDT en formato Planilla que nos permite realizar el cálculo de estimación de esfuerzo en base a jornadas laborales.] **Matriz EDT\***