| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto:Gestorio.cl* |
|  |
| **Revisión*: [01]*** |
| **31/08/2024** |

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

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.gjdgxs)

[**Ficha del documento 4**](#_heading=h.1fob9te)

[**1. Introducción 5**](#_heading=h.3znysh7)

[1.1. Propósito 5](#_heading=h.2et92p0)

[1.2. Ámbito del Sistema 5](#_heading=h.tyjcwt)

[1.3. Definiciones, Acrónimos y Abreviaturas 6](#_heading=h.3dy6vkm)

[1.4. Referencias 6](#_heading=h.1t3h5sf)

[1.5. Visión General del Documento 7](#_heading=h.4d34og8)

[**2. Descripción General 7**](#_heading=h.2s8eyo1)

[2.1. Perspectiva del Producto 7](#_heading=h.17dp8vu)

[2.2. Funciones del Producto 7](#_heading=h.3rdcrjn)

[2.3. Características de los Usuarios 8](#_heading=h.26in1rg)

[2.4. Restricciones 9](#_heading=h.lnxbz9)

[2.5. Suposiciones y Dependencias 10](#_heading=h.35nkun2)

[2.6. Requisitos Futuros 10](#_heading=h.1ksv4uv)

[**3. Requisitos Específicos 12**](#_heading=h.44sinio)

[3.1 Requisitos comunes de las interfaces 12](#_heading=h.2jxsxqh)

[*3.1.1 Interfaces de usuario 12*](#_heading=h.z337ya)

[*3.1.2 Interfaces de hardware 12*](#_heading=h.3j2qqm3)

[*3.1.3 Interfaces de software 13*](#_heading=h.1y810tw)

[3.2 Requisitos funcionales 14](#_heading=h.4i7ojhp)

[3.3 Requisitos no funcionales (Organización) y de calidad (Producto) 15](#_heading=h.2xcytpi)

[*3.3.1 Requisitos de Rendimiento o Eficiencia 15*](#_heading=h.1ci93xb)

[*3.3.2 Requisitos de Seguridad 16*](#_heading=h.3whwml4)

[*3.3.3 Requisitos de Usabilidad 16*](#_heading=h.2bn6wsx)

[*3.3.4 Requisitos de Disponibilidad 16*](#_heading=h.qsh70q)

[*3.3.5 Requisitos de Portabilidad 17*](#_heading=h.3as4poj)

[*3.3.6 Requisitos de Mantenibilidad 17*](#_heading=h.1pxezwc)

[*3.3.7 Requisitos de Funcionalidad 17*](#_heading=h.49x2ik5)

[3.4 Requisitos No funcionales Organizacionales 17](#_heading=h.2p2csry)

[**4. Propuesta de Planificación 18**](#_heading=h.147n2zr)

[4.1 Descripción general acerca de la Planificación 18](#_heading=h.3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 19*](#_heading=h.23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 19*](#_heading=h.ihv636)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 19*](#_heading=h.32hioqz)

[*4.1.5 Carta Gantt 20*](#_heading=h.1hmsyys)

[5. Anexos 21](#_heading=h.41mghml)

[*5.1 Acta de Proyecto 21*](#_heading=h.2grqrue)

[*5.2 Matriz Especificación de Requerimientos 25*](#_heading=h.vx1227)

[*5.3 Prototipo de Software 29*](#_heading=h.3fwokq0)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 33*](#_heading=h.1v1yuxt)

[*5.5 Planilla Carta Gantt 34*](#_heading=h.4f1mdlm)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| ***31/08/2024*** | ***01*** | ***Matias Fuentes*** | ***Completar el informe*** |
|  |  |  |  |

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| ***Matias Fuentes*** | ***Desarrollador , Documentación*** |
| ***Marcelo Vivanco*** | ***Desarrollador , Documentación*** |
| ***Bruno Perez*** | ***Lider de proyecto*** |
| ***Ignacio Fonseca*** | ***Desarrollador*** |

# 1. Introducción

El propósito principal de nuestro proyecto es proporcionar a los comercios (Pymes) una herramienta que no solo automatice procesos, sino que también mejore la precisión en la gestión de inventarios y facilite el análisis de ventas. La relevancia de nuestro proyecto es ayudar en las necesidades reales de los negocios, basándose en requerimientos de problemas cotidianos y por nuestra parte también vamos agregar un diseño centrado en el usuario.

Nuestro proyecto abarca principalmente las áreas de ventas, inventario y finanzas. A través de un interfaz que haremos con React, React Native, Firebase y PouchDB, este sistema tiene como objetivo ofrecer una solución accesible y eficaz para mejorar la eficiencia operativa y la toma de decisiones.

## 1.1. Propósito

El propósito de este documento es definir y documentar todos los requisitos técnicos, funcionales y no funcionales de nuestro proyecto antes de iniciar la fase de desarrollo.

Algunos puntos importantes del documento son los siguientes:

* Detallar las características, funcionalidades y restricciones de nuestro sistema para dar claridad entre el equipo de desarrollo e interesados.
* Servir como un documento de referencia para el equipo de desarrollo durante el ciclo de vida del proyecto, asegurando que se cumplan los requisitos especificados.
* Facilitar la gestión de cambios, permitiendo identificar qué requisitos han cambiado y cómo podrían afectar al proyecto.
* Asegurar que el sistema a desarrollar esté alineado con los objetivos y necesidades del negocio o cliente.

## 1.2. Ámbito del Sistema

El sistema se trata de una aplicación de ventas dirigida principalmente para pymes, para este proyecto se espera cumplir con las siguientes funcionalidades:

* Realizar funciones de punto de venta.
* Visualizar estadísticas sobre ventas en forma de gráficos.
* Control sobre las ventas, es decir, ver el valor total y el valor total con descuento
* Ver las ventas totales del día, cancelar ventas, como otras formas de tratar las ventas.
* Control de inventario, es decir, permitirá eliminar, agregar y modificar productos de su inventario.
* Conexión con una base de datos para almacenar la información y datos necesarios, considerando Firestore para este proyecto.
* Ocupar pouchDB para operar sin conexión.
* Generar reportes personalizables y que pueden ser exportados a pdf.
* Visualizar e interactuar con los proveedores, es decir, se podrá cambiar el estado de estado de sus productos, pagados, en espera, etc.
* Gestor de usuarios, asignando roles de usuario(administrador, trabajador, R.R.H.H).

## 1.3. Definiciones, Acrónimos y Abreviaturas

Definiciones:

* **ERP (Enterprise Resource Planning):** Sistema de planificación de recursos empresariales que integra las principales funciones de una empresa en un solo sistema para mejorar la eficiencia y la gestión de los recursos.
* **PYME (Pequeña y Mediana Empresa):** Empresa de tamaño pequeño o mediano que se caracteriza por tener un número limitado de empleados y volumen de ventas moderado.
* **Sistema de Gestión:** Conjunto de procesos y herramientas que una empresa utiliza para dirigir y controlar sus operaciones diarias.
* **Acrónimos y Abreviaturas:**
* **ISO:** Organización Internacional de Normalización (International Organization for Standardization).
* **PMI**: Instituto de Gestión de Proyectos (Project Management Institute).
* **SRS:** Especificación de Requisitos del Software (Software Requirements Specification).
* **UI:** Interfaz de Usuario (User Interface).
* **UX:** Experiencia de Usuario (User Experience).
* **API**: Interfaz de Programación de Aplicaciones (Application Programming Interface).
* **CRUD:** Crear, Leer, Actualizar, Borrar (Create, Read, Update, Delete).
* **BD:** Base de Datos.

## 1.4. Referencias

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

* **Acta de Constitución**
* **Estructura de Desglose del Trabajo(EDT)**
* **Project Requirements (requerimientos)**
* **Definición Proyecto APT**
* **Carta Gantt**

## 1.5. Visión General del Documento

Este documento de especificaciones de requisitos del software (ERS) se organiza en varias secciones que describen en detalle los requisitos del sistema.

Primero, se presenta una descripción general del sistema, incluyendo su contexto, usuarios y restricciones. Luego, se detallan los requisitos funcionales y no funcionales del sistema, junto con las interfaces y dependencias con otros productos o sistemas. Además, se definen las interacciones con hardware y otros sistemas externos, así como los casos de usos principales.

Finalmente, se incluyen las consideraciones sobre el desarrollo, los recursos involucrados y las condiciones necesarias para la correcta implementación del sistema.

# 2. Descripción General

## 2.1. Perspectiva del Producto

El sistema de venta **POS** (Punto de Venta) no es completamente independiente, ya que requiere una base de datos para almacenar y gestionar información crítica del negocio, como ventas, inventarios, usuarios y proveedores. Para este propósito, se ha elegido **Firebase** como plataforma de base de datos en la nube, lo que implica una dependencia directa de este servicio para la gestión de datos y autenticación de usuarios.

#### Interfaces del sistema:

* **Interacción con Firebase:** El sistema se conecta a Firebase para almacenar datos de manera centralizada, gestionar autenticaciones de usuarios y generar reportes en tiempo real.
* **Funcionamiento offline con PouchDB:** En caso de pérdida de conexión, PouchDB permite que el sistema almacene las transacciones y otras operaciones localmente hasta que la conexión se restaure y se puedan sincronizar los datos con Firebase.

## 2.2. Funciones del Producto

El sistema en desarrollo ofrece una solución completa para la gestión de ventas, inventarios y análisis de datos. Permite el procesamiento de pagos en efectivo, tarjeta y pagos móviles, así como la emisión de tickets personalizables y la gestión de artículos en tiempo real (agregar, eliminar, modificar). Además, incluye seguimiento de inventarios en tiempo real con alertas automáticas, integración con códigos de barras y SKU, y análisis de ventas mediante gráficos, tablas e informes detallados. También cuenta con perfiles de clientes, programas de fidelización, control de acceso por roles, y cuadros mensuales y anuales para facilitar el análisis del rendimiento comercial.

## 2.3. Características de los Usuarios

Tipo de Usuario/Administrador

* **Nivel Educacional:** Generalmente no se necesita tener una formación concreta, pero ayuda mucho tener conocimientos de contabilidad o áreas similares.
* **Experiencia:** Experiencia en toma de decisiones. Puede tener conocimientos básicos a intermedios en el uso de software de gestión.
* **Experiencia Técnica:** Familiaridad con herramientas digitales y software básicos. Habilidad para aprender nuevas aplicaciones con soporte mínimo.

Encargado de ventas

* **Nivel Educacional:** Educación técnica, técnicas de ventas, administración de áreas relacionadas.
* **Experiencia:** Experiencia en ventas y atención al cliente. Conocimiento práctico del proceso de ventas y manejo de inventarios.
* **Experiencia Técnicas:** Experiencia en el uso de sistemas de punto de ventas y software de ventas. Necesita una interfaz intuitiva y fácil de usar para el usuario.

Contador/Encargado de Finanzas

* **Nivel educacional:** Formación en contabilidad, finanzas o administración de empresas.
* **Experiencia:** Experiencia en contabilidad y manejo de informes financieros.
* **Experiencia Técnica:** Conocimientos en software y financiero. habilidad para gestionar y analizar grandes volúmenes de datos.

Encargado de Inventario

* **Nivel Educacional:** Educacion tecnica en logistica, administracion de inventarios os similar.
* **Experiencia:** Experiencia en la gestión de inventario y control de stock.
* **Experiencia técnica:** Experiencia en sistemas de gestión de inventarios. Necesita herramientas para seguimientos y actualización de inventarios de manera eficiente

Requisitos en común

* **Acceso a Capacitación:** todos los usuarios deben recibir capacitación adecuada para utilizar el sistema de manera efectiva.
* **Interfaz Intuitiva:** El sistema debe ser intuitivo para facilitar el uso por parte de personas con diversos niveles de experiencia técnica.
* **Soporte Técnico:** Disponibilidad de soporte telecinco para resolver problemas y proporcionar asistencia en el uso del sistema

## 2.4. Restricciones

**Limitaciones del hardware:** el equipo necesario para llevar a cabo el proyecto solo debe tener un computador que tenga acceso a internet y soporte con el IDE Visual Studio Code.

El cliente debe tener especificaciones similares o superiores a las siguientes:

**CPU:** Procesador moderno de x86-64 (Intel or AMD)

**RAM:** Mínimo 4GB DDR4 o superior.

**Almacenamiento:** 20 GB de espacio libre.

Consideraciones adicionales:

* Sistema operativo Windows, macOS, y/o Linux.
* **Interfaces con otras aplicaciones:** La aplicacion tendra de conexion con firebase , pouchDB y framework de typescript como nextUS
* **Operaciones paralelas.**
  + Sincronización en tiempo real con bases de datos externas, como Firebase Firestore, para mantener actualizados los datos de ventas e inventarios.
  + Manejo eficiente de múltiples usuarios, asegurando que varios procesos , como la sincronización de datos o la generación de reportes, se realizan en paralelo sin colisiones ni pérdida de información.
* **Funciones de auditoría:**Para el sistema de auditorías se crearán triggers que se activen al momento de realizar un cambio por ej. cambios de rol en usuarios , retiro de dinero etc,esto para controlar todo lo que pasa y asi saber cuando y quien lo realiz**ó**
* **Funciones de control:**
  + Monitoreo de operación en tiempo real, asegurando que los procesos de ventas, inventario y sincronización de datos se ejecuten sin problemas.
* **Lenguaje(s) de programación:** Debido a que el desarrollo de la aplicación está basado en typescript con el framework Next.js es en cierta forma una limitante, debido a que es un lenguaje y un framework que no tenemos dominado por lo cual hay que aprender desde cero para hacer este proyecto
* **Protocolos de comunicación:** No hubo problemas de comunicación ya que al concretar reuniones para decidir el avance de la semana y como estamos respecto a tiempos del proyecto no hubo mayor complicación
* **Requisitos de habilidad :** Al igual que en lenguajes de programación al no tener el suficiente conocimiento completo de los lenguajes que utilizamos nos limita para hacer el proyecto ya que tenemos que aprender acerca de ciertas cosas que no sabemos del lenguaje que estamos utilizando, dedicando algunos días aprenderlo para poder implementarlo.
* **Criticidad de la aplicación:**La mayor complicación que podríamos llegar a tener sería la integración de pouchDB ya que hasta el momento no hemos encontrado mucha información sobre esta base de datos
* **Consideraciones acerca de la seguridad:** Nosotros para comprometer la seguridad utilizaremos el sistema de roles para que no puedan acceder a zonas que no deban, también utilizaremos patrones de diseño seguros

## 2.5. Suposiciones y Dependencias

Suposiciones técnicas: Se supone que el sistema operará en la infraestructura actual de la pyme, utilizando tecnologías como Next.js, Firebase, PouchDB, Vercel. Cualquier cambio en estas tecnologías o en las plataformas de implementación podría requerir ajustes en los requisitos y el desarrollo.

## 2.6. Requisitos Futuros

A medida que evoluciona y se adapta a las necesidades cambiantes del negocio, se anticipa la necesidad futuras mejoras en el sistema de escritorio.  
Se prevé la posibilidad de integrar el sistema de escritorio con una gama más amplia de servicios externos, como plataformas de pago, y servicios de contabilidad. Esto permitirá a los usuarios gestionar de manera más eficiente las ventas y la contabilidad desde una única interfaz.

En el futuro, se podría incorporar inteligencia artificial para mejorar la gestión del inventario, predecir tendencias de ventas, y proporcionar recomendaciones personalizadas a los usuarios basado en el análisis de datos históricos.

# 3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describe comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.

Deberán aplicarse los siguientes principios:

* El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
* Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
* Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.

## 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### 3.1.1 Interfaces de usuario

* Los colores debaran ser correspondientes al negocio en este caso los colores elegidos son (Naranjo,Negro,Blanco,Amarrilo y verde)
* Los botones deben tener el texto blanco o negro y seran responsibos
* Ell fondo de las paginas contendrá los colores negro y blanco
* se mostrará el nombre de usuario correspondiente
* Las páginas tendrán una barra de navegación a excepción de la página de venta POS
* Las páginas cambian según el rol que tenga el usuario, esto quiere decir que podrían haber más o menos opciones de navegación

### 3.1.2 Interfaces de hardware

**Requerimientos Mínimos:**

* **CPU:** Procesador moderno de x86-64 (Intel or AMD)
* **RAM:** Mínimo 4GB DDR4 o superior.
* **Almacenamiento:** 20 GB de espacio libre.

### 3.1.3 Interfaces de software

* **Sistema Operativo con soporte para Node.js: Windows, macOS, y Linux.**
* **Typescript**
* **framework NextJS**
* **Firebase**

## 3.2 Requisitos funcionales

Los requisitos funcionales deben estar enumerados (diagrama caso uso) y pueden ser divididos en sub-secciones.

**3.2.1 Requisitos Funcional 1: Gestión de Usuarios**

**Descripción:** El sistema debe permitir la gestión completa de usuarios, incluyendo la creación, actualización, modificación, eliminación y asignación de roles.

* **Crear usuario:** El administrador podrá crear nuevos usuarios ingresando información como nombre, correo electrónico, contraseña, y rol asignado (e.g., administrador, vendedor).
* **Actualizar Usuario:** Los administradores podrán modificar la información del usuario existente.
* **Modificación**: El administrador podrá modificar a los usuarios.
* **Eliminar usuario:** El administrador podrá eliminar usuario cuando sea necesario.
* **Asignación de Roles:** Los roles determinan los permisos y accesos a las diferentes funcionalidad del sistemas.

**3.2.2 Requisito Funcional 2: Gestión de inventario**

**Descripción**: El sistema debe permitir la gestión detallada del inventario, incluyendo el control de stock, ingresos, egresos y alertas por productos con bajo stock.

* **Registrar Productos:** Los usuarios podrán registrar nuevos productos con detalles como nombre, categoría, precio, y cantidad en stock.
* **Actualizar stock:** El sistema permitirá actualizar manualmente la cantidad de stock al recibir nuevos productos o vender existencias.
* **Alerta de stock bajo:** El sistema enviará notificaciones cuando un producto alcance un nivel de stock bajo predefinido.
* **Generación de Reportes:** Se podrán generar reportes detallados de movimientos de inventario, como ingresos y egresos de productos.

**3.2.3 Requerimientos Funcional 3: Facturación y Gestión de Ventas**

**Descripción**: Debe permitir la creación y gestión de facturas, así como el seguimiento de las ventas.

* **Generar facturas:** Los usuarios podrán generar facturas para las ventas realizadas, incluyendo detalle del cliente, productos vendidos, precios, y descuentos aplicables.
* **Historial de Ventas:** El sistema debe mantener un historial de todas las ventas realizadas, accesible para consultas posteriores.
* **Método de pago:** La aplicación soporta múltiples métodos de pagos, como efectivo, tarjeta de crédito, y transferencia bancaria.
* **Descuento y Promociones:** Los usuarios podrán aplicar descuentos o promociones a las ventas y éstas se reflejarán en la boleta generada

**3.2.4 Requisito Funcional 4: Reportes y Análisis**

**Descripción:** El sistema debe generar reportes detallados para facilitar el análisis del rendimiento de la empresa.

* **Reportes de Ventas:** Generación de reportes que incluyan detalles de las ventas por periodo, por producto, o por vendedor.

**3.2.5 Requisito Funcional 5: Gestión Financiera**

**Descripción:** Debe incluir funcionalidades para el seguimiento y gestión de las finanzas de la empresa.

* **Registro de ingresos y Egresos:** El sistema permitirá registrar todas las transacciones financieras, tanto ingresos como egresos.
* **Balance Financiero:** Generación de un balance financiero que muestre el estado económico de la empresa en un periodo determinado.
* **Integración con bancos:** Opcionalmente, el sistema podrá integrarse con bancos para conciliar transacciones.
* **Generación de Reportes Financieros:** Reportes como estado de cuenta, balances, y el flujo de caja serán generados para análisis y toma de decisiones.

## 3.3 Requisitos no funcionales (Organización) y de calidad (Producto)

### 3.3.1 Requisitos de Rendimiento o Eficiencia

**Vercel:**

* **Ancho de banda: 100GB**
* **Fast Data Transfer: 100GB**
* **Edge Requests: 1M**

**Firebase:**

* **Cloud Storage:**
* **GB Almacenados: 5GB**
* **GB Descargados: 1GB por día**
* **Operaciones de carga: 20,000 por día**

**Operaciones de descarga:**

* **50,000 por día**

**Firestore:**

* **Almacén de datos: 1 GB total**
* **Consumo de recursos: 10 GB por mes**
* **Escritura: 20.000 escrituras por día**
* **Lectura: 50.000 lecturas por día**
* **Eliminación: 20.000 eliminaciones por día**

**Github**:

* **Capa gratuita**

**PouchDB:**

* **Código abierto: grátis**

### 3.3.2 Requisitos de Seguridad

Para asegurar la seguridad de la aplicación ocuparemos el sistema de roles para controlar el accesos de los usuarios, esto para evitar que gente no pueda acceder a lugares que no debe

además de que se ocuparan patrones de diseño seguros, por otra parte nosotros nos adherimos a los patrones de diseño de nextUs

### 3.3.3 Requisitos de Usabilidad

La aplicación está diseñada para que sea simple de diseño para no saturar a los usuarios con una interfaz muy agresiva además de contar con un diseño intuitivo para la mayoría de las edades y cuenta con un tiempo de respuesta de 2 segundos para no arruinar la experiencia del usuario

### 3.3.4 Requisitos de Disponibilidad

La gracia de este proyecto es que tendrá una disponibilidad del 95% al mismo tiempo nosotros estaremos integrando POUCHDB para que al momento de no tener internet el sistema siga funcionando sin problema

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

### 3.3.5 Requisitos de Portabilidad

En término de portabilidad no lo haremos para otros dispositivos además de computador, no será compatible para celular o tablets además de que la página no será responsiva para que otros dispositivos ya que es un sistema pos de ventas exclusivo para computador de las pymes

### 3.3.6 Requisitos de Mantenibilidad

La mantenibilidad de la aplicación consistirá principalmente en actualizaciones que sean referentes a las actualizaciones de los frameworks utilizados o cambios en la base de datos

Identificación del tipo de mantenimiento necesario del sistema.

### 3.3.7 Requisitos de Funcionalidad

* Realizar funciones de punto de venta.
* Visualizar estadísticas sobre ventas en forma de gráficos.
* Control sobre las ventas, es decir, ver el valor total y el valor total con descuento
* Ver las ventas totales del día, cancelar ventas, como otras formas de tratar las ventas.
* Control de inventario, es decir, permitirá eliminar, agregar y modificar productos de su inventario.
* Conexión con una base de datos para almacenar la información y datos necesarios, considerando Firestore para este proyecto.
* Ocupar pouchDB para operar sin conexión.
* Generar reportes personalizables y que pueden ser exportados a pdf.
* Visualizar e interactuar con los proveedores, es decir, se podrá cambiar el estado de estado de sus productos, pagados, en espera, etc.
* Gestor de usuarios, asignando roles de usuario(administrador, trabajador, R.R.H.H).

## 3.4 Requisitos No funcionales Organizacionales

1. Para los requisitos no funcionales nos encargaron seguir la gamma de colores de la empresa en este caso seran los colores (verde,naranjo,blanco y negro) ademas de insertar su logo en las paginas junto al nombre de la empresa
2. los botones deben seguir un tipo de letra específico junto a sus colores correspondientes

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

La planificación del proyecto se ha distribuido en un total de 11 semanas. Se seguirán buenas prácticas de desarrollo ágil, como la implementación de sprints que se extenderán a lo largo de 9 semanas, permitiendo revisiones periódicas y ajustes según sea necesario.

El proyecto comenzará con la creación de MockUps en Figma durante la primera semana, seguido de la configuración del entorno de desarrollo en la segunda semana. La documentación del proyecto se realizará durante 2.5 semanas para asegurar una base sólida antes de comenzar el desarrollo.

Las etapas de desarrollo se dividirán en frontend y backend, cada una abarcando 6 semanas. Estas fases incluyen la implementación de las interfaces de usuario, el desarrollo de componentes clave de UI, la gestión de estado con Zustand, y la configuración de las rutas de API en Next.js.

Se asignan 2 semanas para la integración con Firebase Firestore y la sincronización offline, así como una semana adicional para debugging y ajustes finales en los servicios y la sincronización.

Se considerarán condiciones como la comunicación constante y una reunión por semana, y se analizará que todo lo establecido vaya bien encaminado, y la capacidad de resolver posibles imprevistos, asegurando una entrega buena entrega al término de la semana 11.

### 4.1.2 Definición del Equipo de Trabajo

El equipo de trabajo está compuesto por cuatro integrantes, desarrollando como parte de nuestro último examen final, estamos en proceso de adquirir las competencias necesarias para desempeñar de forma profesional el ámbito del desarrollo de software. A pesar de no ser aún profesionales, como grupo aplicaremos las mejores prácticas de gestión y desarrollo de proyectos a lo largo de este proceso.

| **ROL** | **Responsabilidades** |
| --- | --- |
| **Líder del proyecto** | Supervisión general del proyecto, coordinación de reuniones semanales, seguimiento de tareas y gestión de riesgos. |
| **Desarrollador Frontend** | Implementación de la interfaz de usuario según los MockUps, desarrollo de componentes clave de UI, y gestión del estado con Zustand. |
| **Desarrollador Backend** | Desarrollo de servicios, configuración de rutas API en Next.js, integración con Firebase Firestore y sincronización offline. |
| **Documentador** | Creación y mantenimiento de la documentación del proyecto, incluyendo SRS y manuales técnicos. |

### 4.1.3 Definición de Actividades principales del Proyecto

El proyecto se divide en las siguientes etapas clave:

* **Planificación**: Análisis de requisitos, priorización de tareas y creación del backlog.
* **Desarrollo**: Configuración del entorno, desarrollo de frontend con Next.js y Zustand, desarrollo de backend con rutas API, servicios, integración con Firebase y sincronización offline.
* **Implementación**: Despliegue en Vercel, debugging y corrección de errores.
* **Cierre**: Revisión de requisitos, ajustes finales y documentación para la entrega.

### 4.1.4 Resumen Costos del Desarrollo del Proyecto

**1-Mano de Obra**

La mano de obra será el mayor componente. Dado que son 4 personas trabajando desde el 16 de agosto al 6 de diciembre (aproximadamente 16 semanas), deberíamos estimar cuántas horas semanales trabajarán y cuál es la tarifa por hora o salario mensual.

**Duración del proyecto:** 16 semanas

**Número de personas:** 4

**Cálculo de salario:**

* **Tarifa por hora:** $10,000 CLP/hora
* **Horas semanales por persona:** 40 horas
* **Semanas de trabajo:** 16 semanas
* **Costo por persona:** 40 h \* 16 semanas \* $10,000 CLP = $6,400,000 CLP por persona.
* **Total para 4 personas:** 4 \* $6,400,000 = $25,600,000 CLP

2.1. **Firebase**

**Consideraciones:**

* **Firestore:** Considerando que puedes almacenar datos por encima del plan gratuito.
* **Almacenamiento:** Si subes imágenes y otros archivos multimedia.
* **Autenticación:** Gratis hasta cierto número de usuarios.

**Costos aproximados**:

* **Firestore:** 1 GB de almacenamiento adicional = $0.18 USD ($155 CLP) por mes.
* **Almacenamiento**: 1 GB en Cloud Storage = $0.026 USD ($23 CLP) por mes.
* **Autenticación**: Gratis hasta 10K verificaciones por mes.

**2.2. PouchDB (para almacenamiento offline)**

**Costo**: Generalmente es gratuito, pero puede haber costos de almacenamiento en caso de integrar con IndexedDB o almacenamiento local.

**2.3. Hosting en Vercel**

**Plan básico gratuito:** Cubriría sus necesidades básicas.

**Plan Pro (provisorio):** $20 USD/mes (~$17,000 CLP).

**2.4. Herramientas de Colaboración (Opcionales)**

**GitHub**: Plan gratuito.

**Trello/Notion**: Planes gratuitos disponibles.

**Hardware:** POS Mini varía entre $40,000 y $100,000 CLP.

### 4.1.5 Carta Gantt

El proyecto comienza el Lunes 12 de agosto y finaliza el Jueves 3 de octubre. A continuación se describen las fases principales con sus semanas de inicio y fin:

1. **Semana 1:** Creación de MockUps en Figma (Hito: Finalización de MockUps)
2. **Semana 2:** Configuración del entorno de desarrollo (Hito: Entorno preparado)
3. **Semana 3 a 5.5**: Documentación del proyecto (Hito: Finalización de la documentación)
4. **Semana 3 a 8:** Desarrollo Frontend (Hito: Interfaces de usuario listas)
5. **Semana 3 a 8:** Desarrollo Backend (Hito: Servicios y API funcionales)
6. **Semana 9:** Integración con Firebase y sincronización offline (Hito: Sincronización operativa)
7. **Semana 10**: Debugging y ajustes en servicios y sincronización (Hito: Corrección de errores)
8. **Semana 11:** Implementación y despliegue en Vercel (Hito: Proyecto desplegado y listo para uso)

Fecha de término del proyecto: **Final de la Semana 11.**

## 5. Anexos

### 5.1 Acta de Proyecto

**Propósito y justificación del proyecto**

| El propósito principal del proyecto es desarrollar un sistema web de tipo ERP (Enterprise Resource Planning) orientado a comercios. Este sistema está diseñado para ofrecer una solución que permita la gestión de operaciones comerciales, considerando las áreas de ventas, inventario y finanzas.  Este proyecto se realizará en el contexto de **CAPSTONE**, lo que significa que no tiene solo como objetivo la entrega de una solución funcional, sino que también la demostración de competencias clave adquiridas a lo largo de la carrera. A través de la implementación de este ERP, se pretende mostrar la capacidad de abordar un proyecto de software complejo, considerando metodologías ágiles, diseño de software y mejores prácticas en el ámbito de la gestión y la seguridad. |
| --- |

**Descripción del proyecto y entregables**

| El proyecto "Gestorio CL" busca desarrollar un sistema web ERP para comercios, facilitando la gestión de ventas, inventario y autenticación de usuarios. Este desarrollo forma parte del proyecto **CAPSTONE** del portafolio y tiene como objetivo demostrar las habilidades adquiridas a lo largo de la carrera.  Para los entregables por fase se considera lo siguiente:  **Planificación**   * **Análisis de Requisitos**: Documento de requisitos, Carta Gantt, Backlog de tareas.   **Desarrollo**   * **Configuración del Entorno**: Repositorio Git, entorno en Vercel, y bases de datos configurados. * **Frontend**: Estructura de Next.js, gestión de estados, autenticación, y componentes de UI desarrollados. * **Backend**: API Routes configuradas, servicios de ventas e inventario, y sincronización offline.   **Implementación**   * **Despliegue**: Sistema desplegado, ajustes post-despliegue, y monitoreo inicial.   **Cierre del Proyecto**   * **Revisión y Ajustes Finales**: Validación de requisitos, ajustes finales, y documentación preparada para la entrega. |
| --- |

**Objetivos**

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Alcance** | |
| **Autenticación de credenciales** | 99% de autenticaciones realizadas sin errores. |
| **Apartado de ventas** | 100% de las ventas registradas correctamente sin discrepancias. |
| **Gestión de inventario** | 98% de concordancia entre el inventario registrado y el físico. |
| **Apartado de reportes** | 95% de los reportes generados sin errores dentro del tiempo especificado. |
| **Gestión de proveedores** | 100% de los registros de proveedores sin errores en los datos. |
| **Cronograma** | |
| **Toma de requisitos (12/08 - 18/08)** | 100% de cumplimiento del plazo |
| **Creación de Mockups en Figma (19/08 - 01/09)** | 100% de cumplimiento del plazo |
| **Documentación (23/08 - 01/09)** | 100% de cumplimiento del plazo |
| **Configurar el entorno de desarrollo (26/08 - 01/09)** | 100% de cumplimiento del plazo |
| **Desarrollo Frontend (29/08 - 11/10)** | 90% de cumplimiento del plazo |
| **Desarrollo Backend (17/09 - 28/20)** | 95% de cumplimiento del plazo |
| **Implementación (28/10 - 03/11)** | 100% de cumplimiento del plazo |
| **Calidad** | |
| **Adecuación Funcional** | 90% de los requisitos funcionales implementados correctamente. |
| **Eficiencia de Desempeño** | Tiempo de respuesta inferior a 2 segundos bajo condiciones normales de uso (no aplica a workers asincrónicos). |
| **Compatibilidad** | 0% de errores de compatibilidad en los navegadores y dispositivos soportados. |
| **Capacidad de Interacción** | 99% de interacciones entre módulos realizadas sin fallos. |
| **Fiabilidad** | Menos de 1 fallo crítico por mes en producción. |
| **Seguridad** | 0 vulnerabilidades críticas sin resolver antes de la puesta en producción. |
| **Mantenibilidad** | Resolución de bugs en menos de 48 horas tras su reporte. |
| **Flexibilidad** | Nuevas funcionalidades implementadas en menos de 2 semanas tras su aprobación. |
| **Protección** | 0 incidentes de seguridad en el sistema tras la implementación de medidas de protección. |
| **Otros** | |
|  |  |
|  |  |

**Riesgos iniciales de alto nivel**

| Para el proyecto de se debe considerar los siguientes riesgos de alto nivel:   * **Fallas en la integración de tecnologías**   + **Impacto**: Problemas en la interoperabilidad entre Firebase, PouchDB, y otras tecnologías, lo que podría afectar la funcionalidad del sistema. * **Retrasos en el cronograma**   + **Impacto**: Posible incumplimiento de los plazos de entrega y aumento de costos. * **Problemas de seguridad de datos**   + **Impacto**: Exposición de datos sensibles, pérdida de confianza de los usuarios y posibles implicaciones legales. * **Falta de recursos humanos o técnicos**   + **Impacto**: Demoras en el desarrollo por falta de personal o herramientas necesarias. * **Dificultad en el despliegue**   + **Impacto**: Problemas al llevar el sistema a producción, afectando la disponibilidad para el cliente. |
| --- |

**Cronograma de hitos principales**

| **Hito** | **Fecha tope** |
| --- | --- |
| **Toma de requisitos** | **18/08** |
| **Creación de Mockups en Figma** | **01/09** |
| **Documentación** | **01/09** |
| **Configurar el entorno de desarrollo** | **01/09** |
| **Desarrollo Frontend** | **11/10** |
| **Desarrollo Backend** | **28/20** |
| **Implementación** | **03/11** |

**Lista de Interesados (stakeholders)**

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| **Francisco Julliet** | **Docente** | **Escuela de Telecomunicaciones e Informática** |
|  |  |  |

**Asignación del gerente de proyecto y nivel de autoridad**

**Gerente de proyecto**

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| **Bruno Pérez** | **Gerente de TI** | **Escuela de Telecomunicaciones e Informática** |

**Aprobaciones**

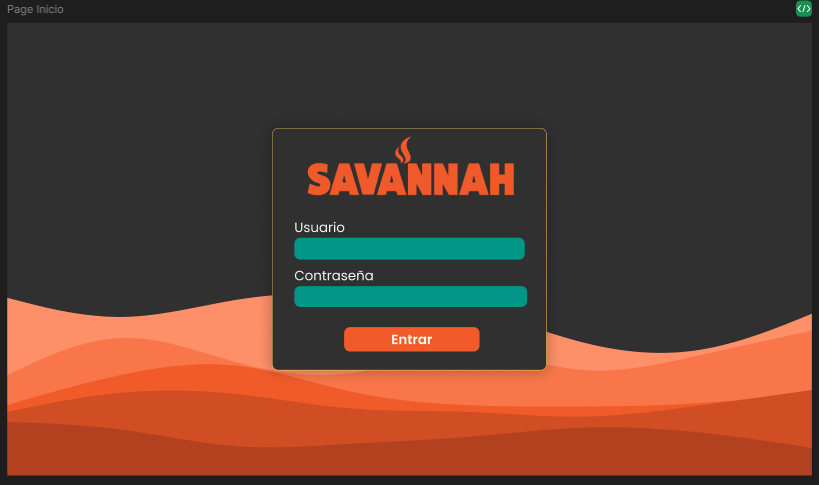
| **Patrocinador** | **Fecha** | **Firma** |
| --- | --- | --- |
| **Francisco Julliet** | **06 - Septiembre** |  |
| **Bruno Pérez** | **30 - Agosto** |  |
|  |  |  |

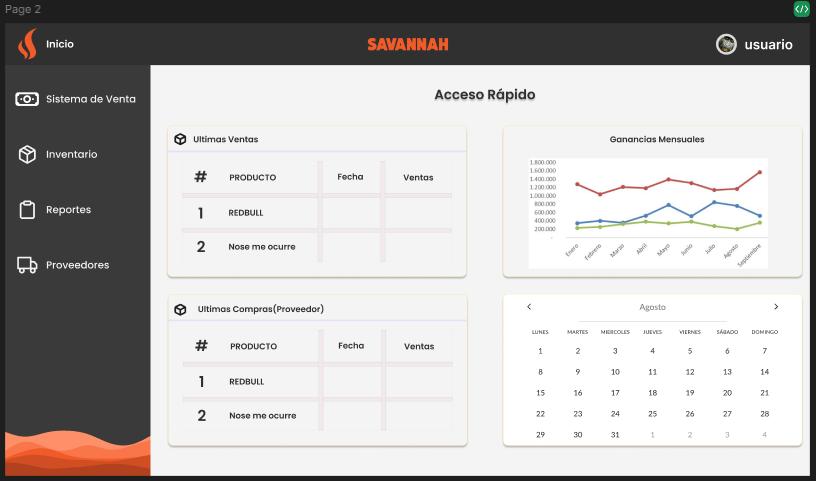
### 5.2 Matriz Especificación de Requerimientos

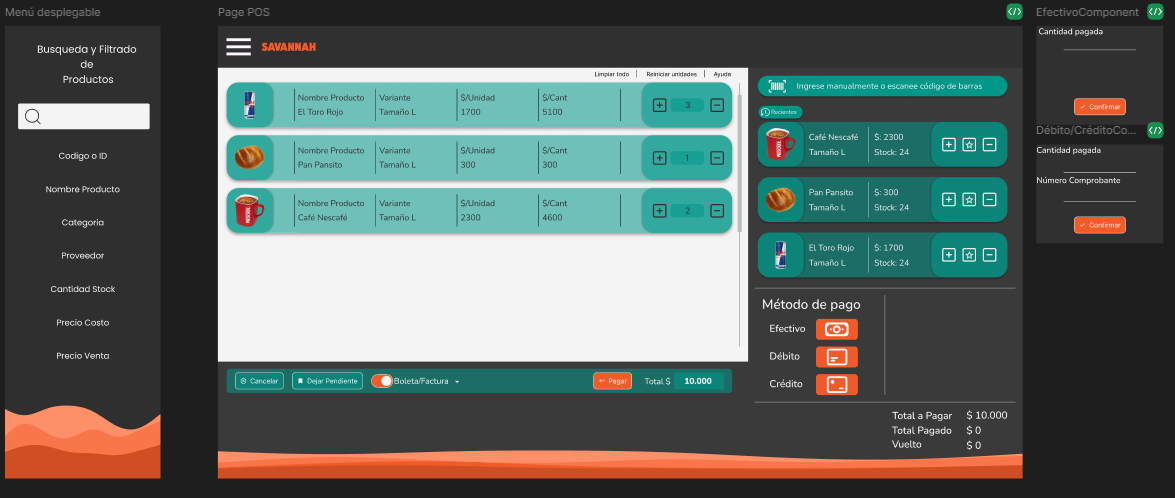
| **Requisitos Funcionales** | | |
| --- | --- | --- |
| **Gestión de Ventas:** | **Código** | **Requerimiento** |
| **f.1** | Procesamiento de diferentes métodos de pago (efectivo, tarjeta, pagos móviles) |
| **f.2** | Emisión de tickets y facturas personalizables |
| **f.5** | Mostrar el total del IVA |
| **f.6** | Mostrar el total Neto |
| **f.7** | Mostrar el total de ventas |
| **f.8** | Mostrar el total de ganancias |
| **f.9** | Cancelar venta |
| **f.10** | Modificar venta |
| **f.11** | Agregar artículo |
| **f.12** | Eliminar artículo |
| **f.13** | Modificar artículo |
| **f.14** | Reimprimir último ticket |
| **f.15** | Separar artículo afecto al excento de IVA |
| **f.16** | Dejar pendiente ticket |
| **f.17** | Agregar artículo |
| **f.18** | Eliminar artículo |
| **f.19** | Modificar artículo |
| **f.20** | Registro de hora de cada transacción |
| **f.21** | Generación de id para cada transacción |
| **f.22** | Eliminar ticket |
|  |  |  |
| **Gestión de Inventario:** | **Código** | **Requerimiento** |
| **f.25** | Seguimiento en tiempo real del inventario |
| **f.26** | Alerta de reposición automática |
| **f.27** | Gestión de variantes de productos (tamaño, color, etc.) |
| **f.28** | Integración con códigos de barras y SKU |
| **f.29** | Inventario contra factura o boleta |
| **f.33** | Agregar artículo |
| **f.34** | Eliminar artículo |
| **f.35** | Modificar artículo |
| **f.36** | Cuadre mensual |
| **f.37** | Cuadre anual |
| **f.38** | Comparación anual de ventas |
| **f.39** | Insertar varios productos |
| **f.40** | Buscar productos |
| **f.41** | Registro y visualización de inventario actual y mínimo |
|  |  |  |
| **Gestión de Clientes:** | **Código** | **Requerimiento** |
| **f.43** | Creación de perfiles de clientes con historial de compras |
| **f.44** | Programas de fidelización y descuentos |
|  |  |  |
| **Gestión de Empleados:** | **Código** | **Requerimiento** |
| **f.46** | Control de acceso basado en roles |
|  |  |  |
| **Informes y Análisis:** | **Código** | **Requerimiento** |
| **f.49** | Registro y visualización de ventas diarias |
| **f.50** | Generación de informes de ventas, inventario y rendimiento |
| **f.51** | Visualización de datos a través de gráficos y tablas |
| **f.52** | Análisis predictivo basado en datos históricos |
| **f.53** | Historial de ventas |
| **f.54** | Reporte de Inventario util-orden |
|  |  |  |

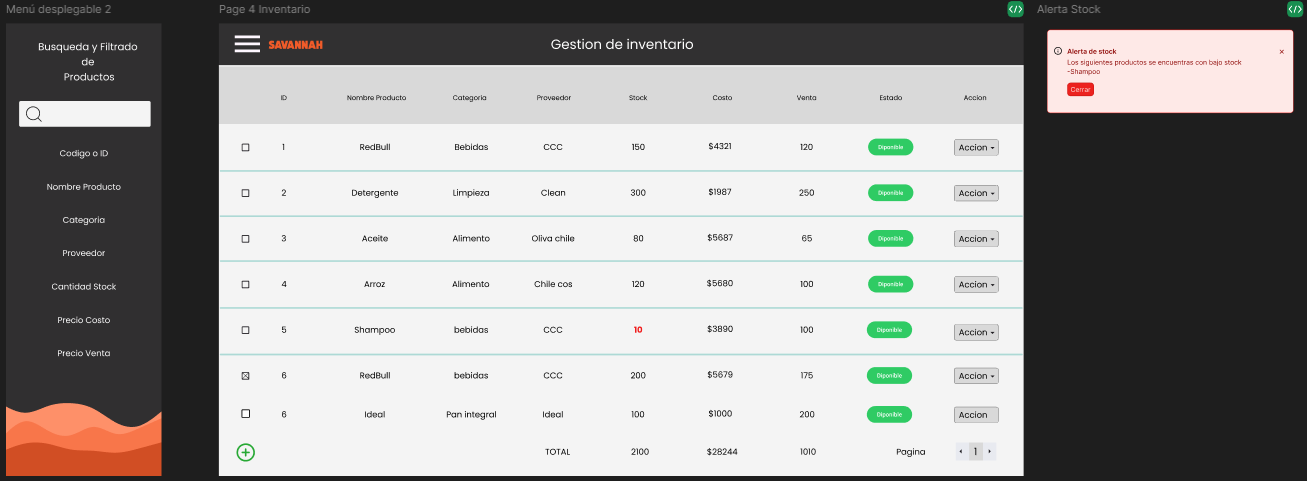
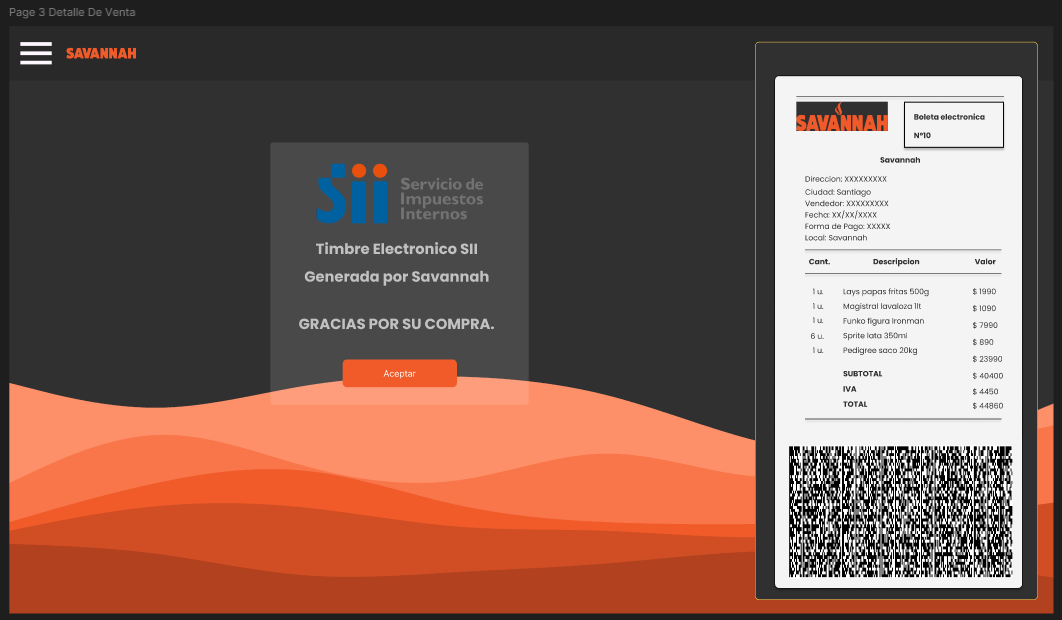
| **Requisitos No Funcionales** | | |
| --- | --- | --- |
| **Escalabilidad** | **Código** | **Requerimiento** |
| **nf.1** | Capacidad para manejar un aumento en la cantidad de datos y usuarios sin perder rendimiento |
|  |  |  |
| **Rendimiento** | **Código** | **Requerimiento** |
| **nf.3** | Respuesta rápida a las operaciones críticas (menos de 1 segundo para transacciones de ventas) |
| **nf.4** | Baja latencia en la actualización de inventarios y generación de informes |
|  |  |  |
| **Seguridad** | **Código** | **Requerimiento** |
| **nf.7** | Autenticación de dos factores (2FA) para acceso de usuarios |
| **Usabilidad** | **Código** | **Requerimiento** |
| **nf.12** | Interfaz de usuario intuitiva y fácil de aprender |

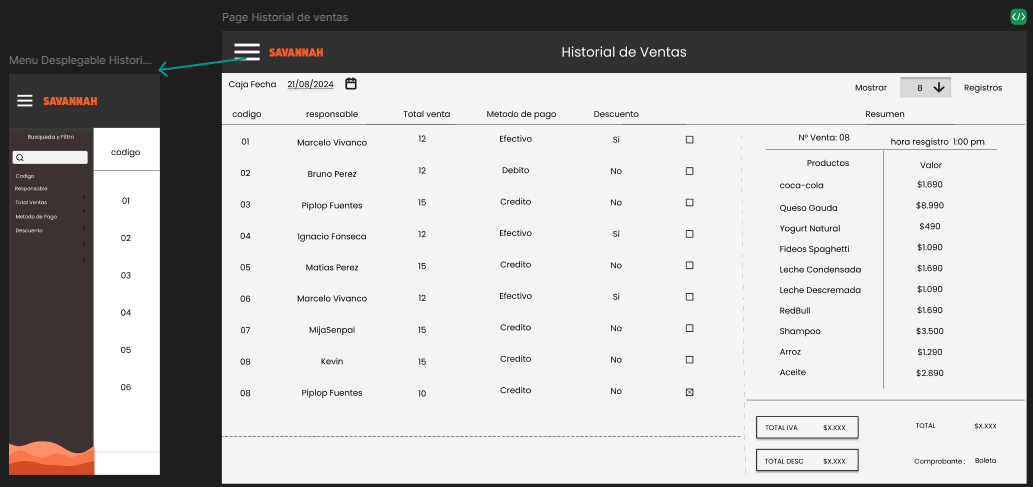
### 5.3 Prototipo de Software

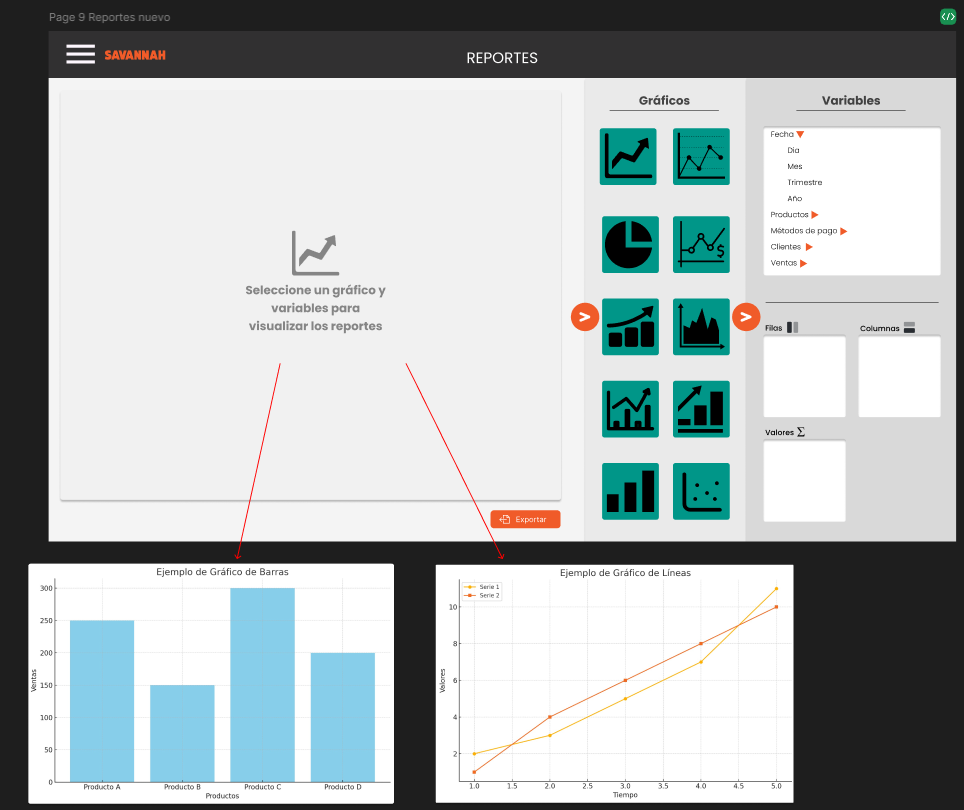




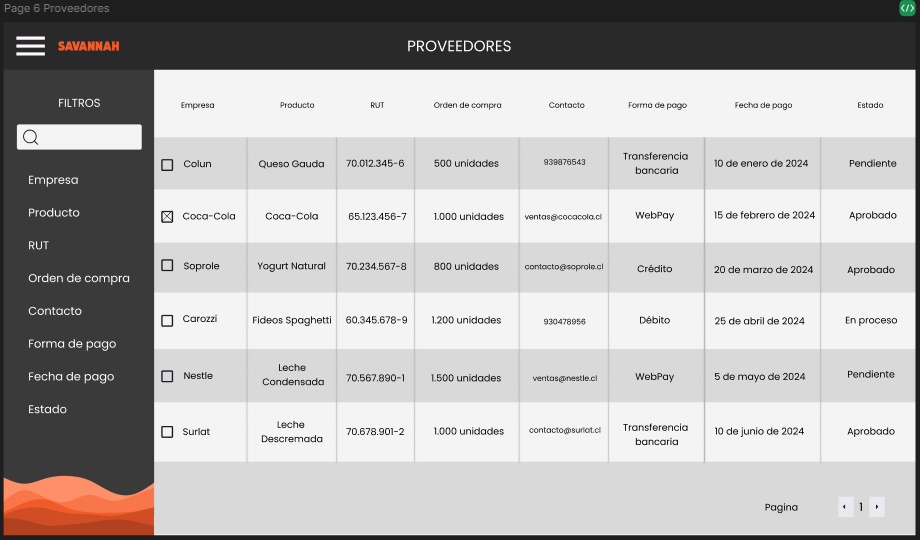




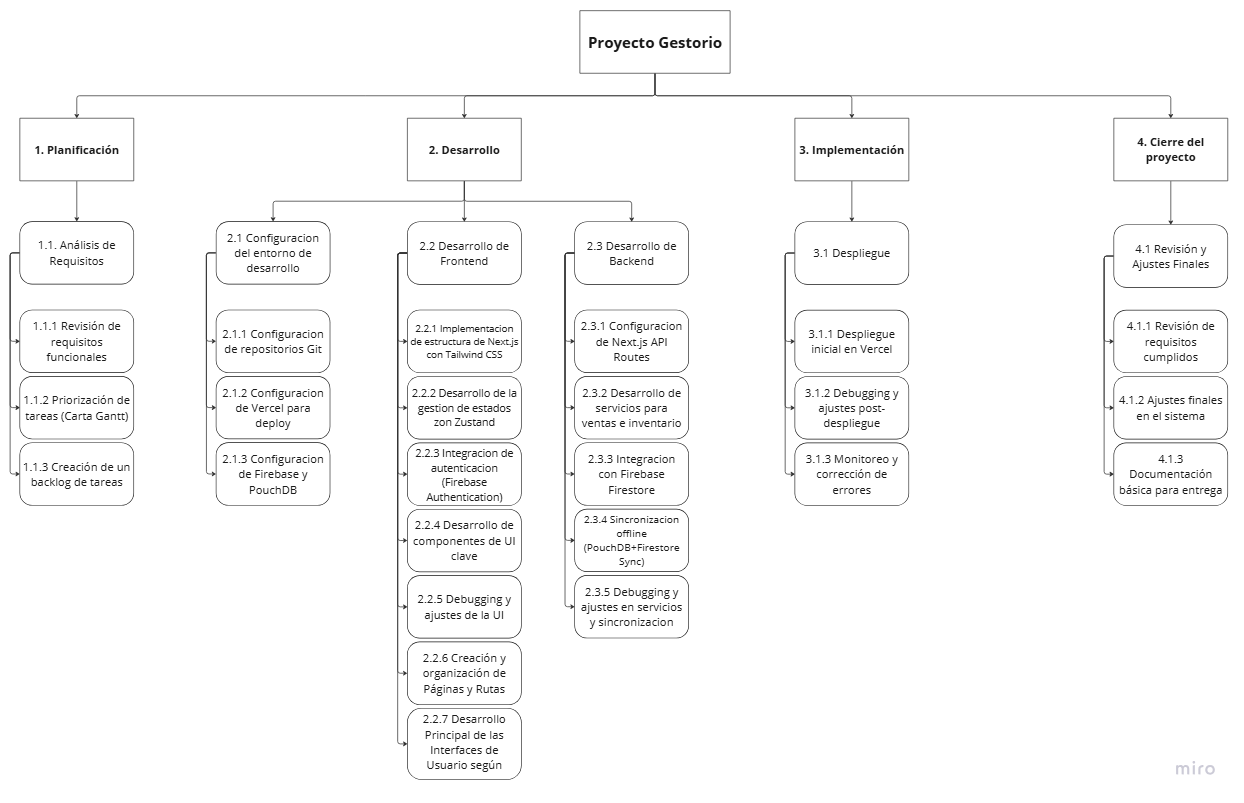








### 5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo



### 

### 5.5 Planilla Carta Gantt

