PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Delivery de Fármacos para Pacientes Crónicos (DIEFP-C)

Felipe Andrés Riquelme Hernández

Patricio Ignacio Torres Rojas

Carla Nicole Hidalgo Quintana

Enzo Franchescoli Aravena Quevedo

Cristian Francisco Monrroy Martínez

Profesora: Pamela Hermosilla Monckton

Ingeniería de Ejecución en Informática

2020

Índice

[Resumen iii](#_Toc45155763)

[Abstract iii](#_Toc45155764)

[Lista de Figuras iv](#_Toc45155765)

[Lista de Tablas v](#_Toc45155766)

[1. Introducción vi](#_Toc45155767)

[2. Descripción General de la situación en estudio vii](#_Toc45155768)

[2.1 Definición de Objetivos vii](#_Toc45155769)

[2.1.1 Objetivo General vii](#_Toc45155770)

[2.1.2 Objetivos Específicos vii](#_Toc45155771)

[3. Situación en estudio viii](#_Toc45155772)

[3.1 Contexto Organizacional viii](#_Toc45155773)

[3.2 Descripción de la situación actual viii](#_Toc45155774)

[3.3 Identificación de los problemas detectados ix](#_Toc45155775)

[4. Propuesta de solución x](#_Toc45155776)

[4.1 Descripción general de la solución x](#_Toc45155777)

[4.2 Modelo de proceso de desarrollo x](#_Toc45155778)

[4.3 Arquitectura lógica y física de solución x](#_Toc45155779)

[4.4 Herramientas de desarrollo xi](#_Toc45155780)

[4.4.1 Lenguajes de programación xi](#_Toc45155781)

[4.4.2 Motores de base de datos xi](#_Toc45155782)

[4.4.3 Modelado de sistemas xi](#_Toc45155783)

[5. Gestión del Proyecto xii](#_Toc45155784)

[5.1 Estudio de factibilidad xii](#_Toc45155785)

[5.1.1 Factibilidad técnica xii](#_Toc45155786)

[5.1.2 Factibilidad operacional xiii](#_Toc45155787)

[5.1.3 Factibilidad económica xiv](#_Toc45155788)

[5.1.4 Factibilidad legal xv](#_Toc45155789)

[5.2 Carta Gantt xvi](#_Toc45155790)

[5.3 Gestión de riesgos xvi](#_Toc45155791)

[5.3.1 Identificación del riesgo xvii](#_Toc45155792)

[5.3.2 Planes de Mitigación y Contingencia xviii](#_Toc45155793)

[6. Desarrollo del proyecto xxi](#_Toc45155794)

[6.1 Requerimientos funcionales y no funcionales xxi](#_Toc45155795)

[6.2 Análisis de requerimientos xxi](#_Toc45155796)

[6.3 Diseño xxi](#_Toc45155797)

[6.3.1 Componentes de SW xxi](#_Toc45155798)

[6.3.2 Base de datos xxi](#_Toc45155799)

[6.3.3 Interfaz xxi](#_Toc45155800)

[6.4 Implementación xxi](#_Toc45155801)

[6.5 Plan de pruebas xxii](#_Toc45155802)

[7. Conclusiones xxii](#_Toc45155803)

[7.1 Conclusions xxii](#_Toc45155804)

[8. Anexos xxii](#_Toc45155805)

# Resumen

Este documento presenta la problemática actual de la compra y retiro de medicamentos en las farmacias en tiempos de pandemia, ante esta problemática se propone y desarrolla la solución de crear una aplicación móvil de despacho exclusivamente de medicamentos, destinada principalmente a pacientes crónicos que son quienes más necesitan de medicación constante, que sea amigable con el usuario considerando que gran mayoría de estos pacientes forman parte del grupo etario de adulto mayor.

Palabras-claves: delivery, aplicación móvil, farmacias, pandemia, medicamentos.

# Abstract

This document presents the current problem of the purchase and withdrawal of medicines in drugstores in times of pandemic, faced with this problem is proposed and develops the solution of creating a delivery mobile app exclusively for medicines, aimed mainly at chronic patients who need constant medication, that be user-friendly considering that a large majority of these patients are part of the elderly age group.

Keywords: delivery, mobile app, drugstores, pandemic, medicines.

# Lista de Figuras

# Lista de Tablas

# Introducción

Viendo en retrospectiva la vida cotidiana del mundo apenas unos años atrás, llega a parecer descabellada la idea de que muchas empresas opten por el teletrabajo a comparación de como el mercado y como se ha laburado desde siempre. Para entrar en contexto, el año 2020 se ha visto en vuelto en una contingencia mundial debido a la expansión del virus COVID-19, lo que ha obligado a la población a permanecer en casa más tiempo de lo acostumbrado y esto ha abierto un nuevo nicho en el mundo de la tecnología; clases en línea a través de plataformas como Z*oom,* la expansión de Intranet’s para que empresas sigan produciendo sin sus trabajadores presentes o un aumento en las compras online de supermercados son algunas de las consecuencias que ha traído esta emergencia sanitaria. Pero ¿Qué hay de los medicamentos que personas crónicas necesitan con urgencia? ¿Qué puede hacer esta gente si no tienen acceso a farmacias dado el peligro que se está viviendo?

Para este problema hay aplicaciones que se han acondicionado de cierta manera para solucionar esto, siendo uno de los casos *PedidosYa* lo cual es algo positivo si se toma en cuenta que toda ayuda en estos momentos es algo bueno, pero originalmente siendo un servicio para comida a domicilio no lo hace muy amigable para el usuario destinatario que necesita comprar sus medicamentos sin tener que salir a la calle; a esto viene el objetivo de este informe, resolver este dilema a través del software formulado por el grupo “DIEFP-C” (nombre provisional) que busca aprovechar este sistema de despachos a domicilio pero con la única de razón de llevar los remedios que este cliente en especial requiere.

Se dará una explicación más detallada en un apartado posterior acerca del problema en sí y a lo largo del informe se visualiza como el grupo abarca la solución con la realización de la aplicación “DIEFP-C”, desde las herramientas que se optaron a usar y el modelo del sistema para clarificar todo al momento de programar.

# Descripción General de la situación en estudio

En esta época de la digitalización no es difícil encontrar una aplicación que se adecué a las necesidades que un usuario posea, ya sea desde comprar comida hasta herramientas de construcción en algunos casos; en cambio lo que es el acercamiento de medicamentos a las personas no es tan popular como los casos mencionados, siendo además un tema complicado por lo que implica despachar por ejemplo remedios con recetas retenidas a pacientes crónicos o con dificultades para movilizarse.

En esto último se enfoca DIEFP-C que, aunque este pensado para el uso del público en general, la necesidad actual lo es las personas que no tienen el libre acceso a sus farmacias más cercanas para poder abastecerse de esos medicamentos que urgen en sus vidas cotidianas. Habiendo ya aplicaciones que cumplen con una mínima ayuda a este tema, u algunos pocos que se han estrenado en Chile en este breve tiempo, se busca unificar esta función de despacho a domicilio enfocado únicamente a la ayuda de los pacientes que la requieran.

## Definición de Objetivos

Para llevar a cabo esta idea es necesario evocar los objetivos que serán el camino para llevar acabo el sistema propuesto. Por eso en los siguientes apartados se hará una explicación de cuál es el enfoque general del proyecto junto a sus propósitos específicos.

### Objetivo General

Implementar una aplicación móvil con preferencias a personas crónicas que les permita la compra y el despacho de sus medicamentos a partir de recetas entregadas.

### Objetivos Específicos

* Analizar y entender el funcionamiento del SDK *Flutter* para la realización de la interfaz en Android de la aplicación.
* Identificar los problemas actuales a causa de la escasez de sistemas con compra y despacho de medicamentos.
* Analizar las necesidades y expectativas de público objetivo de la aplicación.
* Analizar y entender el funcionamiento del entorno de desarrollo Android Studio para la integración de las herramientas de Flutter.
* Definir e implementar tareas de planificación, codificación, control y manejo de riesgos de la aplicación.
* Desarrollar aplicación que sea intuitiva y fácil de usar que acapare con las necesidades que el cliente requiera.
* Desarrollar un software capaz de arrancar en dispositivos móviles.

# Situación en estudio

En la siguiente sección se encontrarán diversos temas, como el contexto organizacional, la descripción de la situación actual y la identificación de los problemas detectados, todo esto con el fin de comprender tanto medidas tomadas por el gobierno como aplicaciones que compiten con la propuesta en la actual situación.

## Contexto Organizacional

Destinada al área de la salud, DIEFP-C es una aplicación creada con el objetivo de apoyar a los servicios de salud como hospitales, clínicas, farmacias y centros de salud familiar (CESFAMs) en el contexto mundial actual de la pandemia de COVID-19, facilitando la distribución de medicamentos a lo largo del país.

A partir de las recientes medidas del gobierno, se permite una distribución más libre de medicamentos a través de terceros, creando la oportunidad de surgimiento de nuevos servicios de distribución especializados para colmar la alta necesidad de productos de diversa índole dado el contexto nacional actual.

Si bien, el enfoque de DIEFP-C es principalmente en el área de la salud, sigue siendo considerado un servicio de distribución de productos, por lo que compite directamente con otros servicios delivery como son Rappi y Uber Eats, quienes han adaptado su sistema para justamente permitir la distribución de medicamentos como servicio adicional a su rol de delivery de comidas.

A pesar de que nuestro proyecto entra en esta área relativamente tarde, considerando la popularidad de otros servicios de delivery adaptados al presente nacional, DIEFP-C busca mostrar una visión más allá de la competencia dado el ese presente, buscando la colaboración de dichos servicios y de los servicios de salud como proveedores de productos para combatir las condiciones creadas por la pandemia, buscando mantenerse una vez terminada esta situación como una opción sólida para la distribución de medicina y fármacos, y de ser posible, expandirse a otros rubros tal como otros servicios delivery lo han hecho para este caso.

## Descripción de la situación actual

Si bien vivimos en una época en que la automatización ha sido uno de los objetivos más importantes de la tecnología, no es raro ver empresas que se niegan a este avance o mucho más personas que siguen prefiriendo realizar todo manualmente que tener que depender de un aparato o de un servicio tecnológico; especialmente este año 2020 las cosas han tomado un giro abrupto, el distanciamiento social y la imposibilidad en muchos casos de personas de salir de sus hogares ha hecho que las cosas tan sencillas como ir a abastecerse de mercadería para el mes u comprar algún aperitivo para disfrutar en la tarde se tornara más complicado.

Según lo mencionado anteriormente, han existido aplicaciones desde hace un tiempo ya que satisfacen algunas de estas necesidades; algunas de estas por ejemplo Cornershop o Rappi que funcionan de una forma parecida, donde el cliente solicita un producto para que posteriormente un usuario que este inscrito a la aplicación recibe dicha petición poniéndose en rumbo a buscar los artículos y luego despacharlos a la vivienda del que los solicito; aun así no existen muchas opciones para comparar y hacer la mejor elección a la hora de hablar de aplicaciones de despacho de remedios. Cabe destacar que Cornershop tiene algunos convenios con cadenas como Cruz verde, lo que significa que por parte de las farmacias si existe un interés en estos métodos de acercamientos de sus productos a los clientes necesitados.

## Identificación de los problemas detectados

El principal problema encontrado relacionado a la distribución de medicamentos, en el contexto nacional y mundial, es la justamente la falta de distribuidores dedicados al área de la salud. Si bien, actualmente, varios servicios de delivery como *Rappi, Cornershop, Uber Eats,* han ampliado su rubro de distribución para incluir el delivery de medicamentos dentro sus funcionalidades, el contexto de la pandemia ha evidenciado la dificultad de estos servicios para mantener todos sus servicios activos, puesto que deben atender demandas de alimentos aparte de medicamentos, que son un servicio adicional más que uno principal, lo que ha llevado a retrasos en las entregas de productos y colapsos por la alta demanda durante ocasiones especiales.

Por otro lado, la creciente de enfermedades durante esta pandemia ha generado una mayor demanda de medicamentos que se suma al gran número que se produce históricamente en este periodo del año (otoño-invierno). Todo esto bajo el contexto de cuarentena ha generado una inhabilidad de conseguir dichos medicamentos para muchas personas, ya sea por no poder salir al ser pacientes de riesgo de contagio del COVID-19, por no querer estar en riesgo de contagio en caso de tener que asistir a recintos médicos para obtener medicamentos, tal vez porque no hay recintos médicos o farmacias cerca de él como para poder obtener medicamentos, por que el paciente que necesita los medicamentos no puede salir de su casa por alguna condición física u otra que le impida el movimiento libre o por que el paciente no se maneja bien con las tecnologías o no confía en servicios que no sean dedicados a la salud.

A forma de entender mejor los problemas nombrados, a continuación, se presentan nuevamente estos problemas de manera más generalizada, con el fin de aclarar los objetivos que busca mejorar el proyecto:

* No hay distribuidores de productos dedicados principalmente a medicamentos.
* La dependencia de otros distribuidores no es 100% confiable, puesto que su rubro prioriza otros productos y su alta demanda puede causar colapsos durante momentos críticos para un paciente (crónico o no).
* El contexto de cuarentena impide a pacientes crónicos salir para buscar sus medicamentos de forma segura, al ser pacientes de riesgo.
* La dificultad de pacientes para salir de su casa para buscar sus medicamentos.
* La dificultad de algunos pacientes para pedir medicamentos a su casa ya sea por poco conocimiento de estos servicios o poca confianza en ellos.

# Propuesta de solución

En este capítulo se presenta la descripción general de la solución, el modelo de proceso de desarrollo, la arquitectura lógica y física de la solución y las herramientas de desarrollo.

## Descripción general de la solución

La solución a la situación de estudio planteada anteriormente consiste en la implementación de una aplicación móvil dedicada al delivery de productos farmacéuticos en donde se verán involucrados tres grandes entidades, las cuales son “usuario repartidor”, “empresa farmacéutica” y “usuario cliente” (quien solicitara una orden de compra). Este aplicativo se enfocará principalmente en aquellos clientes que presenten alguna enfermedad crónica y se les deba retener la receta. (Aquí explayarse un poco más ya que yo no manejo mucho el tema y tampoco se bien como manejaran ustedes el proceso, ya que aún no me queda claro este último)

## Modelo de proceso de desarrollo

El paradigma por utilizar será el paradigma orientado a objetos, en este se puede ver como una colección de objetos que interactúan o cooperan entre ellos para obtener la funcionalidad deseada. Por lo cual, para el desarrollo del programa se utilizará el kit de desarrollo de software Flutter y empleando el lenguaje de programación Dart, dado que Flutter hace uso de este paradigma utilizando objetos pequeños con alcances estrechos de comportamiento, compuestos juntos para obtener efectos más complicados. Constantemente durante la etapa de desarrollo se realizarán revisiones al código fuente que escribirán los distintos programadores involucrados a modo de evaluarlo, esta evaluación se realizara con el fin de buscar errores de diseño, programación y documentación. Al usar este paradigma buscamos garantizar la eficiencia del código fuente y también reducir eventuales problemas que se puedan presentar en la implementación continua de código.

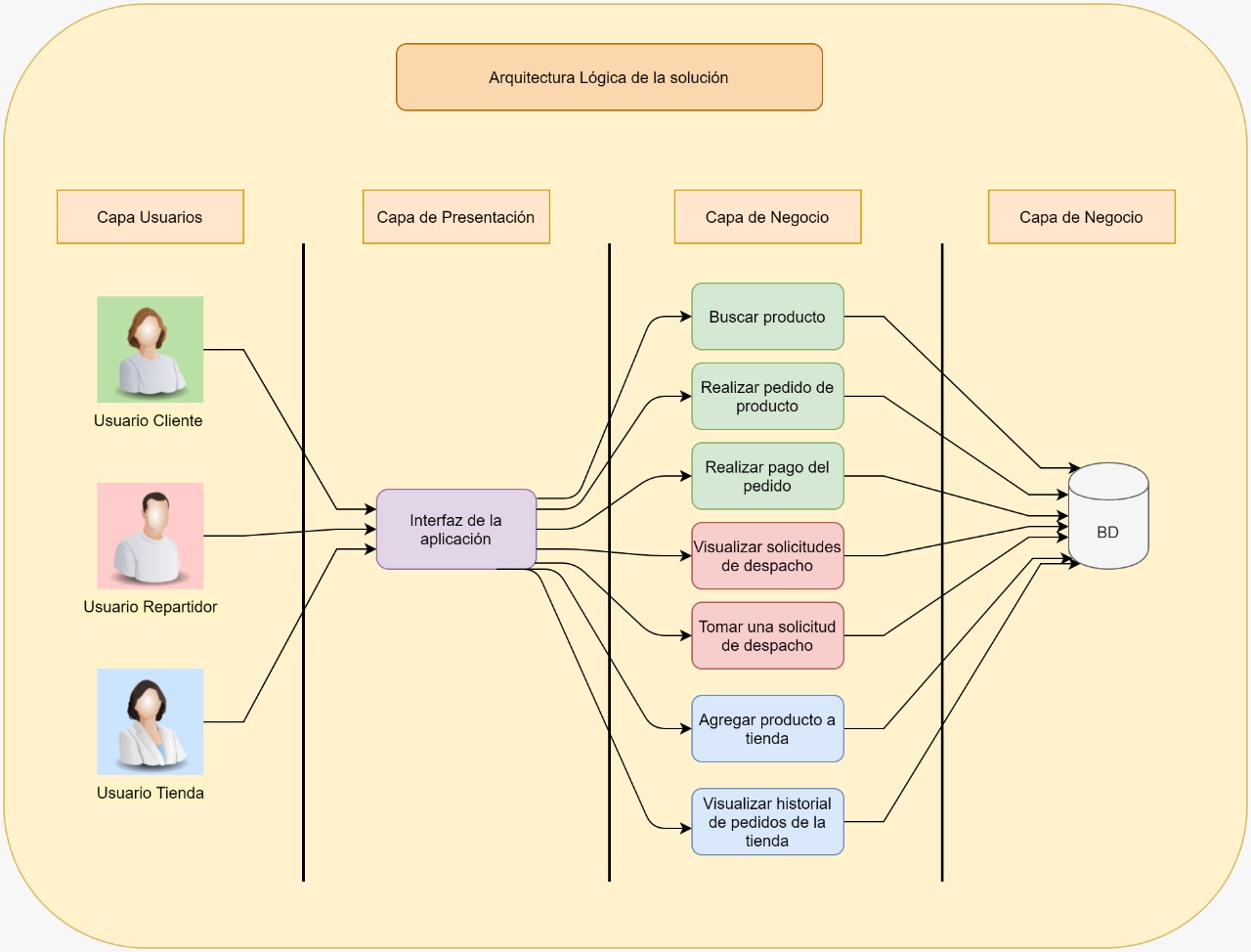
El modelo de desarrollo de software a utilizar es el iterativo e incremental, ya que al no saber exactamente el concepto final y especifico de la aplicación, es decir, se cuenta solo con requerimientos en un alto nivel, se necesitará ir realizando iteraciones pequeñas e incrementales a modo de ir refinando los requerimientos.

## Arquitectura lógica y física de solución

\*\*Este apartado en por ahora opcional quizá sea borrado dado que lo que se escribirá aquí es similar al siguiente que son las herramientas de desarrollo.

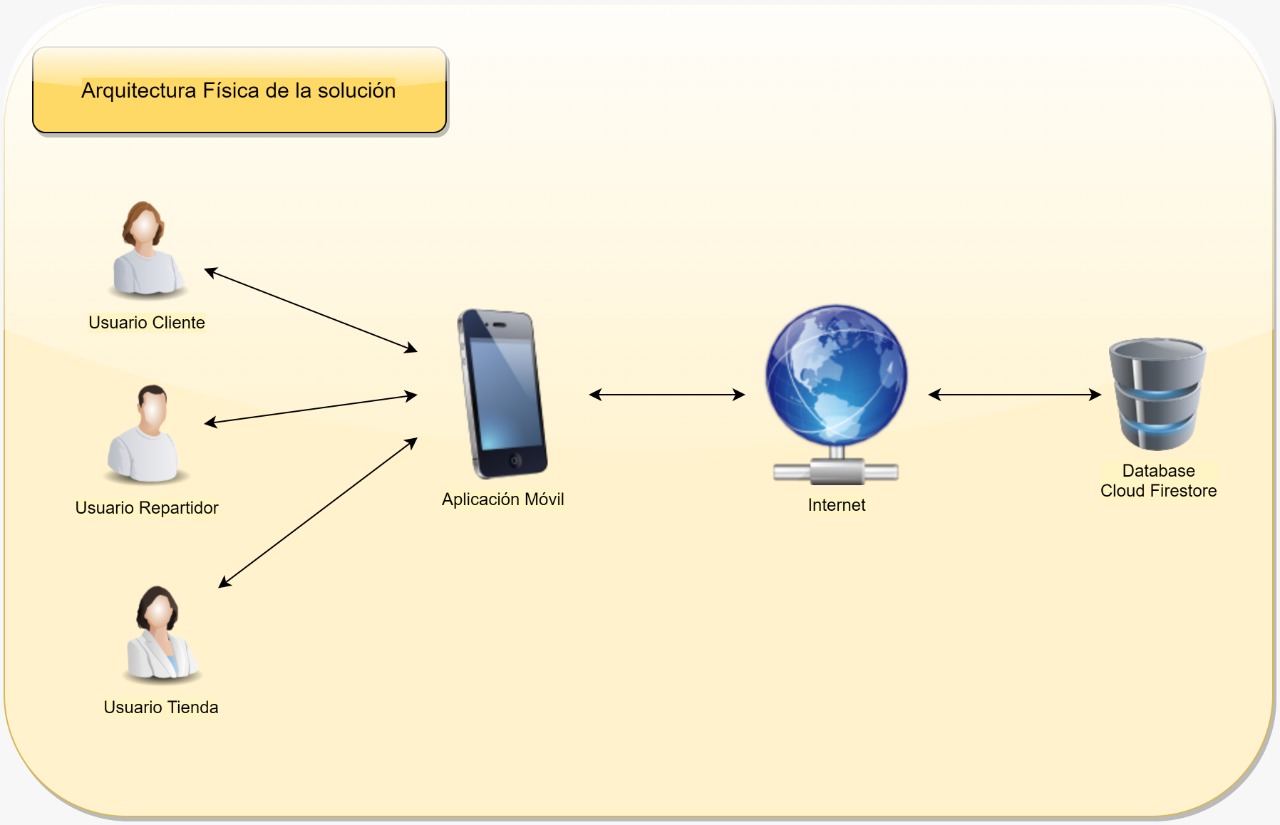
### Arquitectura lógica de solución

En la siguiente figura se darán a entender las diversas capas lógicas que existen en la aplicación y como son usadas por los distintos usuarios.



### Arquitectura Física de solución

En esta figura, se mostrará el hardware utilizado con el cual se conectan los usuarios a la aplicación y sus demás conexiones.



## Herramientas de desarrollo

En esta sección se describirán las herramientas escogidas para llevar a cabo el proyecto, ya sea en el ámbito de análisis y diseño como en la implementación de la aplicación.

### Lenguajes de programación

Para la programación se utilizará el kit de desarrollo de software Flutter junto con el lenguaje de programación Dart. Aparte de compilar en un código ensamblado directamente nativo, la principal característica de Flutter destaca en su “Hot Reload”, lo cual permite aplicar los cambios de forma dinámica, en menos de un segundo, sin perder el contexto de la aplicación y ahorrando bastante tiempo a la hora de generar cambios o realizar pruebas.

### Motores de base de datos

Como motor de base de datos, se optó por una NoSQL, la cual se basa en documentos, denominada Cloud Firestore. Al ser totalmente gestionada, rápida, sin servidor (no necesita montar un servidor) y nativa de la nube, simplifica las tareas de almacenamiento, sincronización y consulta de datos con respecto a las aplicaciones tipo móvil. Algunos de los principales motivos de su elección fueron su sincronización en tiempo real y su modo sin conexión, el cual almacena los datos en el caché, permitiendo el funcionamiento de la aplicación sin problemas.

### Modelado de sistemas

Para la etapa de análisis y diseño de este proyecto se usarán las herramientas de modelado de software Draw.io para el desarrollo de: - Diagramas de flujo de datos, - Lenguaje de Modelado Unificado, entre otros que se puedan necesitar para ejemplificar las funcionalidades del sistema. También se hará uso de las herramientas Balsamiq, para el modelado de prototipos, y Bizagi, para modelar el proceso de negocio a seguir por la aplicación.

# Gestión del Proyecto

La gestión del proyecto consiste en varios factores, como el estudio de la factibilidad, el uso de una carta Gantt con el fin de definir correctamente los tiempos de trabajos y por último la gestión de riesgo, el cual consiste en identificar las posibles amenazas y catalogarlas según su respectivo nivel de riesgo e impacto.

## Estudio de factibilidad

El estudio de factibilidad es un instrumento utilizado para orientar la decisión de continuar o abandonar un proyecto de software. Para este apartado se expondrán 4 categorías del estudio de factibilidad para la aplicación DIEFP-C. En primera instancia se listarán los recursos técnicos y humanos necesarios, luego se estimará la factibilidad económica a base del mercado actual. Finalmente se evidenciarán leyes a la sé que se tendrá que regir el software funcional.

### Factibilidad técnica

La factibilidad técnica nos entrega información acerca de si existe o la aplicación está al alcance de la tecnología necesario, esto se puede referir al hardware o las herramientas que se llegasen a utilizar (lenguaje de programación, kits de desarrollo); también hay que tener en cuenta las habilidades y conocimientos de los recursos humanos que son parte del desarrollo, si es posible contar con ellos para una correcta implementación del producto que requiere el cliente.

A continuación, se mencionarán dichos recursos listados según su correspondiente categoría, junto a una breve descripción de aquello:

* Recursos humanos:
* **Profesionales en el área de la informática:** Se habla del área de la informática en general porque dentro del equipo no solo se requieren ingenieros capaces de modelar y diseñar el sistema, sino que también debe incluir programadores que se encarguen de la etapa funcional del sistema. Esto da la posibilidad de distribuir mejor el equipo, y le da la ventaja a cada profesional desempeñarse en el área que mejor le acomoda.
* **Repartidores:** Son el foco de cómo funciona el sistema, ya que como la aplicación se centra en el despacho de productos al hogar del usuario, es vital que el intermediario exista y haga posible las entregas.
* Hardware:
* **Computadores:** Para el desarrollo de la aplicación se requieren de equipos capaces de realizar tareas de modelado, diseño y programación. En este caso cada integrante hará uso de computador personal para el desarrollo del sistema.
* **Smartphones:** Dado que el sistema está centrado en una aplicación móvil, se requieren de celulares inteligentes con los cuales poder hacer pruebas preliminares del sistema, con el fin de encontrar errores y solucionarlos. Para ello se optó por usar los celulares personales de los integrantes del equipo.
* Software:
* **Android Studio:** Se hará uso del entorno de desarrollo Android Studio como motor de compilación y base del proyecto propuesto. Además, se hace de la infraestructura para desarrollar todo el Backend del proyecto.
* **Flutter:** Se hará uso del kit de desarrollo de software de código abierto Flutter para el desarrollo de las interfaces, widgets y aspecto en general de la aplicación móvil, además se servir como medio integrador de las funcionalidades del sistema.
* **Firebase:** Se hará uso de la plataforma de desarrollo de aplicaciones web y móviles de Google Firebase. Al hacer uso de las infraestructuras de Google y de además estar ubicada en la nube, sirve como medio principal para integrar toda la aplicación con la base de datos, ya que la mayor ventaja que se nos ofrece es la no necesidad de montar un servidor ya que todas las herramientas vienen incluidas en la plataforma.
* **Cloud Firestore:** Se hará uso del servicio de almacenamiento de datos derivado de Google Cloud Plataform Firestore como motor de base de datos para la plataforma. Dado que es parte de todo el kit que ofrece Google, la fácil configuración que se otorga al estar todo integrado en la nube de Firebase.
* **Windows 10 Professional:** Se hará uso del sistema operativo de Microsoft para todo el desarrollo de la aplicación; desde el diseño y modelado del sistema hasta la etapa de codificación de este mismo.
* **Microsoft Office 365:** Se hará uso de este software para la confección de informes y posteriores presentaciones.
* **Lucidchart:** Se hará uso de esta herramienta de diagramación, para realizar los diseños relacionados con el modelo del sistema.
* **Moqups:** Se hará uso de esta aplicación de manufacturación y diseño de prototipos a escala para realizar las vistas de los requerimientos planteados para el sistema.
* **Google Cloud Services (Google Drive):** Se hará uso de esta nube para realizar respaldo de los informes, diagramas y documentación relacionada al proyecto.
* **Github:** Se hará uso de este controlador de versiones para mantener el control y orden del desarrollo y codificación de la plataforma

### Factibilidad operacional

Para el estudio operacional se busca de un personal apto para llevar a cabo las tareas del desarrollo de la aplicación; además de conocer si el usuario final es capaz de hacer uso del producto final. En cuanto a los participantes, el equipo va a estar constituido por los integrantes que se unieron en el curso de Taller de ingeniería de Software. El tamaño del equipo está conformado por 5 integrantes donde se distribuirá en las distintas labores para llevar a cabo el proyecto, donde en la siguiente tabla se mostrarán a más detalle cómo se constituyó esto.

|  |  |
| --- | --- |
| RR. HH | Cantidad |
| Jefe de proyecto | 1 |
| Desarrolladores | 2 |
| Analista | 1 |
| Asegurador de calidad | 1 |
| Documentadores | 3 |

Por el lado del usuario final hay que tener en cuenta a los dos agentes principales en el sistema, la persona que contrata los servicios de la aplicación y la que trabaja como repartidor para el sistema. Para el comprador, se espera otorgarle una sección de ayuda donde se le den instrucciones básicas sobre el funcionamiento de la aplicación e idealmente un módulo donde pueda recibir atención para dudas más específicas; por parte del repartidor al momento de registrarse habrá todo un proceso de documentación antes de poder empezar a trabajar para la aplicación, dicho proceso se le prepara para la clase de labores que debe realizar; no hablamos de capacitaciones para los usuarios ya que se espera que la aplicación posea de un grado de usabilidad que al momento de ejecutarse sea intuitivo y fácil de usar. Por el lado del equipo de desarrollo, se espera que se trabajen las habilidades de programación para las plataformas a usarse como Android Studio y todas las herramientas que lleva con sí.

### Factibilidad económica

La factibilidad económica nos entrega información respecto a los costos de realización del sistema, en donde se incluyen los costos de desarrollo del producto de software en el cual unos ejemplos serian la adquisición de nuevas tecnologías en caso de no contar con ellas y el gasto involucrado en la mantención del personal a cargo de la implementación. El cliente puede comparar los costos, indicados por este estudio, con los beneficios derivados del sistema desarrollado y dependiendo de los resultados de dicha comparación es que los clientes podrán determinar si desean continuar con el desarrollo del producto.

A continuación, se presenta una tabla con los costes estimados basados en proyectos anteriores:

* Recursos humanos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo | Cantidad | Valor en CLP/Hora aproximado | Duración del proyecto | Total del proyecto aproximado |
| Jefe de proyecto | 1 | $11.508 | 800 | $9.206.400 |
| Analista | 1 | $7.154 | 500 | $7.154.000 |
| Desarrolladores | 2 | $5.538 | 600 | $ 6.645.600 |
| Asegurador de calidad | 1 | $2.968 | 320 | $949.760 |
| Documentador | 3 | $8.000 | 300 | $7.200.000 |
|  |  |  |  | Total  $31.155.760 |

* Hardware

En relación con este apartado, como se determinó que se haría uso de los equipos

personales de cada integrante, no se puede estimar el valor del hardware.

* Software

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Cantidad | Valor en CLP (Aproximado) | Valor total |
| Android Studio | 5 | Gratuito | $0 |
| Flutter | 5 | Gratuito | $0 |
| Firebase | 5 | Gratuito | $0 |
| Api’s Google | 1 | $800/cada 200 usos | Se estima cuando la aplicación comience a correr. |
| Cloud Firestore | 5 | Gratuito | $0 |
| Windows 10 Professional | 5 | $ 179.999 | $899.995 |
| Microsoft Office 365 | 5 | $ 10.872/mes | $173.952 |
| Lucidchart | 5 | $ 15.374 | $46.122 |
| Moqups | 5 | Gratuito | $0 |
| Google Cloud Services (Google Drive) | 5 | Gratuito | $0 |
| Github | 5 | Gratuito | $0 |
|  |  |  | Total: $1.120.069 |

### Factibilidad legal

La factibilidad legal nos permite determinar los derechos que tiene los autores sobre todo el reporte y documentación acerca del proyecto. También nos señala los límites a los cuales se rige el software sobre las leyes según la Legislación Informática de la República de Chile. Si estos criterios no se cumplen, el producto final no podría ser entregado ni viable. Para tener en cuenta que dado el contexto por el COVID-19 se han hecho medidas por parte del gobierno que flexibilizan la venta de medicamentos con recetas retenidas, aunque aún no exista nada concreto las circunstancias avalan que las cosas podrían seguir así, pero con ciertas limitaciones dado que puede nacer nuevas redes de contrabando. Aun así hay cierto decreto del que si se puede sostener el proyecto, decreto n° 466 en la cual avala la venta de medicamentos en locales afiliados a farmacias; otra razón que le da más factibilidad a esta aplicación, es el hecho de que sirve solo como intermediario para las farmacias que tienen el permiso real, ya que como el sistema no es quien distribuye, no existen leyes que limiten los servicios que un emitente de esa clase pueda transportar. Para dejarlo en claro, se mostrará un listado con todas las leyes que hacen factible esta propuesta.

* Ley 21.096 de 5 de junio de 2018, que consagra el Derecho a Protección de los Datos Personales.
* Decreto n° 466 de 31 de diciembre de 1984, que aprueba Reglamento de Farmacias, droguerías, Nota Almacenes Farmacéuticos, Botiquines y Depósitos Autorizados.
* Ley 19.628, del 28 de agosto de 1999. Protección a la Vida Privada.
* Ley nº 20.453, promulgada el 18 de agosto de 2010. Consagra el principio de Neutralidad en la Red para los Consumidores y Usuarios de Internet.
* Ley nº 19.223, promulgada el 28 de mayo de 1993. Tipifica figuras penales relativas a la informática.
* Ley nº 17.336, promulgada el 28 de agosto de 1970. Propiedad Intelectual.

## Carta Gantt

Modificar la Gantt hecha

## Gestión de riesgos

Para el desarrollo exitoso de un proyecto de software, es necesario identificar las posibles amenazas que puedan afectar el trabajo, así como diseñar contramedidas en la forma de planes de mitigación y contingencia, para minimizar el impacto de éstas y mantener un ritmo de desarrollo estable.

A continuación, se muestra la metodología de evaluación de riesgos, para medir el impacto y frecuencia de ocurrencia de cada riesgo, además de listar los riesgos identificados junto a sus planes de mitigación y contingencia para cada uno. El número que acompaña cada categoría corresponde al puntaje asignado con el fin de cuantificar el riesgo total.

* Impacto del riesgo

Se categorizan los niveles de impacto de los riesgos en 4 niveles, presentados en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Nivel de Impacto | Valor de Impacto | Descripción |
| Catastrófico | 3 | La ocurrencia del riesgo puede provocar una falla crítica del sistema, lo que implica que éste pueda no funcionar correctamente. |
| Alto | 2 | La ocurrencia del riesgo puede provocar una falla crítica del sistema, pero no logra ser un impedimento para que el sistema siga funcionando hasta su recuperación. |
| Bajo | 1 | La ocurrencia del riesgo puede provocar una falla leve en el sistema, pero no impide el funcionamiento de éste, ya que puede recuperarse automáticamente de dicha falla. |
| Insignificante | 0 | La ocurrencia del riesgo no provoca fallas representativas, por lo que no son considerados problemas importantes, además el sistema se puede recuperar fácilmente de dicha falla. |

* Frecuencia del Riesgo

Se categorizan los niveles de ocurrencia de riesgos en 5 niveles, descritos a continuación:

|  |  |  |
| --- | --- | --- |
| **Nivel de Frecuencia** | **Valor de Frecuencia** | **Descripción** |
| Muy Alto | 4 | La probabilidad de que ocurra el riesgo es muy alta/muy probable. |
| Alto | 3 | La probabilidad de que ocurra el riesgo es alta/probable. |
| Moderado | 2 | La probabilidad de que ocurra el riesgo es intermedia. |
| Bajo | 1 | La probabilidad de que ocurra el riesgo es baja/ no es probable. |
| Muy Bajo | 0 | La probabilidad de que ocurra el riesgo es muy baja/ muy poco probable. |

### Identificación del riesgo

Los riesgos son eventos que pueden ocasionar daños al proyecto, ya sea a los servicios, sistemas o recursos que utilice. Existen varios tipos de riesgos asociados al proyecto, listados a continuación, donde se destacan riesgos relacionados a la parte informática del proyecto y riesgos asociados a la situación de los integrantes del equipo de trabajo que puedan dificultar la realización de este proyecto.

|  |  |  |
| --- | --- | --- |
| **ID** | **Riesgo** | **Descripción** |
| R1 | Cambio repentino en los requerimientos. | Se producen cambios significativos en los requerimientos definidos inicialmente. |
| R2 | Estimación incorrecta del tiempo de desarrollo de software. | El tiempo de desarrollo de software excede el tiempo estimado al iniciar el proyecto, es decir, se produce un atraso en el desarrollo y la planificación. |
| R3 | Tardía adaptación al entorno de desarrollo. | El equipo de desarrollo (más concretamente los desarrolladores/programadores) no logran adaptarse rápidamente al entorno de desarrollo escogido (en este caso: Android Studio) |
| R4 | Problemas de usabilidad en la plataforma web de la aplicación. | Los usuarios tienen problemas para acceder o manipular información en la plataforma web dedicada a la aplicación de forma rápida e intuitiva. |
| R5 | Vulnerabilidad de la plataforma web de la aplicación. | El sistema web de la aplicación puede presentar vulnerabilidades que pueden significar la pérdida de información sensible del usuario, además de ser más propenso a fallas de seguridad por ataques maliciosos. |
| R6 | Vulnerabilidad de la base de datos de la aplicación. | La base de datos puede presentar vulnerabilidades que pueden significar pérdida de información sensible de los usuarios de la aplicación. |
| R7 | Problemas de usabilidad en la Interfaz de usuario de la aplicación | La interfaz creada para la aplicación, enfocada a dispositivos móviles, no cumple con su función de ser fácil de usar para los usuarios de la app. |
| R8 | Inconsistencias de datos en la base de datos | Se produce una inconsistencia de datos en la base de datos al realizar operaciones sobre datos como actualizar correos y contraseñas, añadir productos, entre otros descritos en las funcionalidades. |

### Planes de Mitigación y Contingencia

A continuación, se listan los planes diseñados para combatir el impacto de los riesgos asociados, dando un plan de mitigación y uno de contingencia para cada riesgo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Impacto (I)** | **Frecuencia (F)** | **Criticidad (I + F)** | **Plan de Mitigación** | **Plan de Contingencia** |
| R1 | 2 | 2 | 4 | Definir detalladamente los requerimientos durante la fase de análisis del proyecto. | Reestructurar el proyecto de acuerdo con los nuevos requerimientos dados. |
| R2 | 2 | 1 | 3 | Determinar las capacidades del equipo de trabajo para crear una planificación acorde. | Re-asignar miembros del equipos a tareas que necesitan más ayuda para equilibrar el trabajo. |
| R3 | 3 | 2 | 5 | Investigar el entorno de desarrollo a utilizar antes de la realización del proyecto | Buscar herramientas e información para entender su uso rápidamente, en su defecto, cambiar el ambiente de desarrollo a uno más “fácil” o familiar a los desarrolladores. |
| R4 | 1 | 2 | 3 | Realizar testing constante de la plataforma, además de encuestas de usabilidad para diseñar la plataforma lo más “amigable” posible para los usuarios objetivo. | Rediseñar la plataforma web en base al feedback de usuarios, en su defecto, proporcionar instrucciones de uso a los usuarios como solución temporal. |
| R5 | 2 | 1 | 3 | Revisar y actualizar constantemente la aplicación, aplicar encriptación de datos para protección ante ataques. | Informar a usuarios inmediatamente de la vulnerabilidad, recomendar cambio de correos y contraseñas, respaldar la información. |
| R6 | 2 | 2 | 4 | Encriptar los datos de la base de datos, contratar servicios de Google Firebase para proteger la base de datos del acceso público. | Informar a usuarios inmediatamente de vulnerabilidad, recomendar cambio de correos y contraseña, respaldar la información, encriptar la información de ser necesario y contratar los servicios de Google Firebase para protección de la base de datos. |
| R7 | 1 | 3 | 4 | Realizar testing constante de la aplicación, además de realizar encuestas y pruebas de usabilidad para diseñar una interfaz más cómoda para los usuarios móviles. | Rediseñar la interfaz de acuerdo con el feedback de usuarios. |
| R8 | 3 | 2 | 5 | Diseñar la base de datos de manera efectiva y definir restricciones de datos antes de poner en marcha la base de datos con la aplicación. | Respaldar la información mantenida hasta el momento y rediseñar o arreglar la base de datos para corregir las inconsistencias. |

# Desarrollo del proyecto

Para realizar un correcto desarrollo del proyecto, se debe cumplir con cierta estructura en su diseño y respetar diversas reglas, las cuales consisten en identificar tanto requerimientos funcionales como no funcionales, realizar sus respectivos análisis y efectuar sus debidos planes de pruebas, como se presentarán a continuación.

## Requerimientos funcionales y no funcionales

En esta sección se presentan los requerimientos funcionales y no funcionales de la aplicación, los cuales se encuentran ubicados en el Anexo.

## Análisis de requerimientos

El análisis de requerimientos del sistema está orientado en determinar cómo se implementarán los requisitos identificados en la fase de captura de requerimientos, por lo tanto, en esta sección se describen, organizan y estructuran los componentes que interactúan con los actores del sistema, así como también el flujo de datos asociados a sus interacciones.

A continuación, se presenta el diagrama de casos de uso del principal enfoque del sistema, la compra y distribución de fármacos.

diagrama caso de uso

A continuación, se presentan los siguientes diagramas de secuencia, realizar pedido de producto y tomar pedido, estos diagramas representan la compra y distribución de los fármacos.

C:\Users\Enzo Aravena\Desktop\Inform final\hacer un pedido.emf

En seguida se describe secuencialmente el diagrama presentado en la figura \*Numero de figura\*, el cual muestra cómo se realiza un pedido por parte del cliente:

1. El cliente, autenticado por la aplicación, ingresa a la sección “Tiendas”.
2. El sistema muestra un listado de tiendas que están disponibles, es decir, que están operativas.
3. El cliente selecciona la tienda de la cual quiere encargar un producto.
4. El sistema muestra un listado de productos que son distribuidos por esa tienda.
5. El cliente selecciona el producto que desea comprar.
6. El sistema muestra la interfaz que posee los datos del producto y un botón que dice “Añadir producto al carrito de compras”.
7. El cliente presiona el botón para añadir el producto al carrito de compra.
8. El cliente se dirige a la interfaz de carrito de compras.
9. El sistema muestra todos los productos que el cliente haya añadido en el carrito, el precio total de la sumatoria de precios y el botón “Realizar pedido”.
10. El cliente presiona el botón “Realizar pedido” para efectuar el pedido.
11. Si el cliente realiza el pago mientras realiza un pedido, el sistema lo derivará al sistema de pago y una vez realizado el pago se registrará el pedido en la base de datos. Una vez registrado y cancelado el pedido el sistema le mostrara un resumen del pago y del pedido.
12. Si no, si el cliente no realizara el pago mientras realiza el pedido, si no que pagara cuando el pedido llegue a su domicilio, entonces el sistema no lo deriva al sistema de pago y registra inmediatamente el pedido. Una vez registrado el pedido el sistema mostrara un resumen del pedido.

tomar pedido

En seguida se describe secuencialmente el diagrama presentado en la figura \*Numero de figura\*, el cual muestra cómo un repartidor se adjudica un pedido:

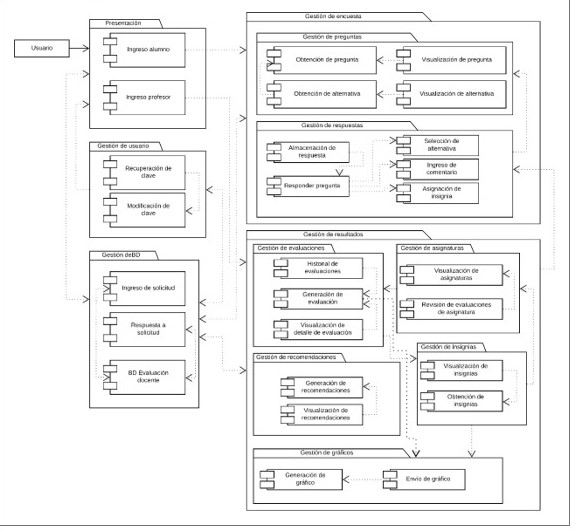
1. El repartidor, autenticado por la aplicación, ingresa a la interfaz “Mapa”.
2. El sistema muestra un mapa, en el cual se encuentran distribuidas las tiendas existentes, las tiendas que tengan pedidos saldrán con números en la parte superior del icono.
3. El cliente selecciona una tienda que tenga pedidos por entregar.
4. El sistema mostrara un listado con los pedidos que existen junto a un botón “Tomar pedido”.
5. El cliente selecciona la opción para tomar un pedido.
6. El sistema mostrara los datos del pedido y también los necesarios para llevar a cabo la entrega, como lo es la ubicación final de entrega.

## Diseño

Como parte del diseño, se presentarán los diversos componentes del software, su respectiva base de datos y la interfaz la aplicación en los siguientes puntos.

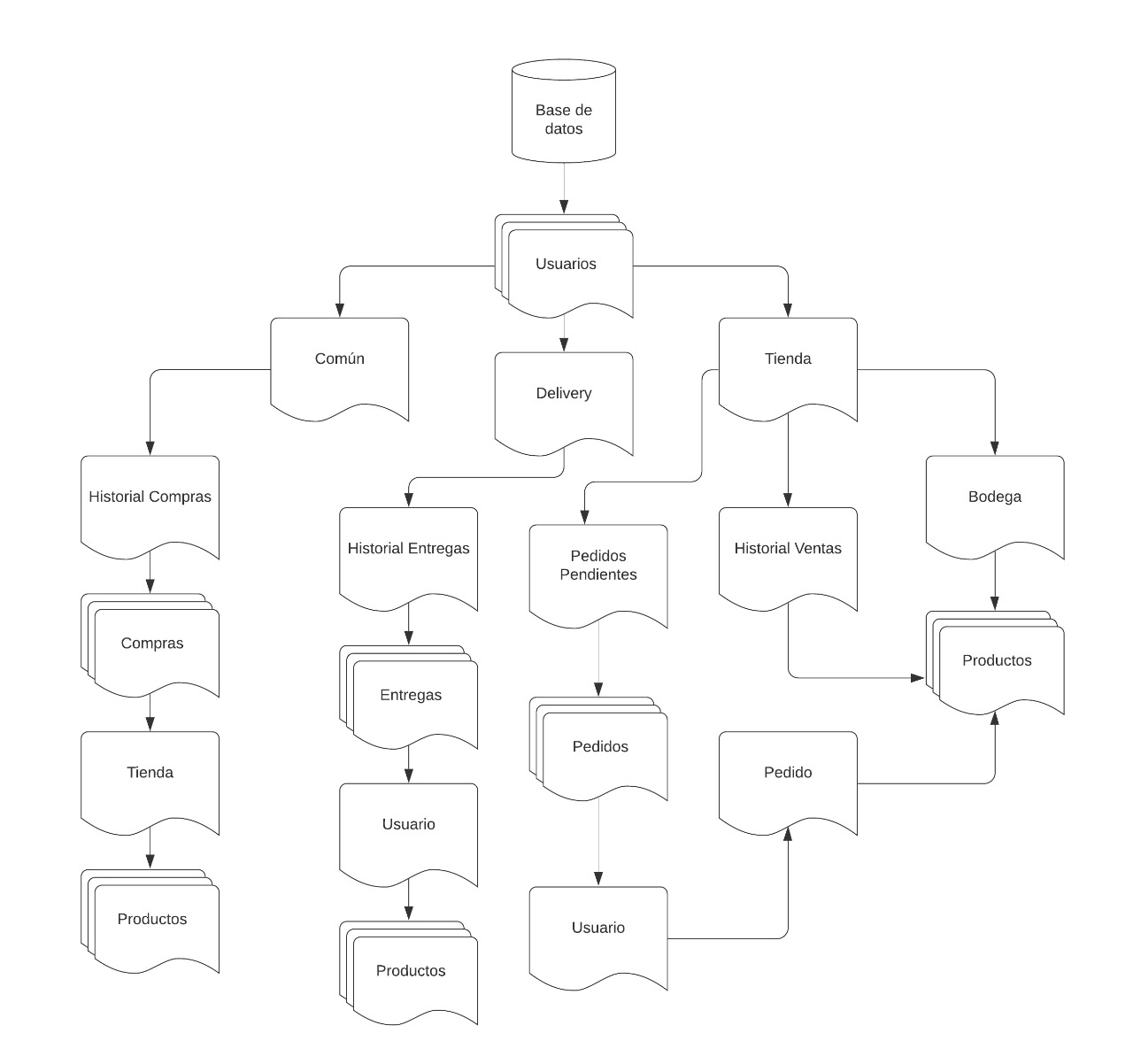
### Componentes de SW

Los componentes de software pertenecientes a la aplicación son los siguientes:



### Base de datos

La base de datos generada a partir un una NoSQL basada en documentos está compuesta de la siguiente forma:



### Interfaz

Para finalizar con el diseño de la aplicación, se expondrán los mockups realizados para ejemplificar la apariencia de esta, con el fin de generarle una idea al cliente. Anexo.

## Implementación

…

## Plan de pruebas

…

# Conclusiones

…

## Conclusiones

…

# Anexos

**RF01: Login**  
El sistema permite al usuario registrarse y/o iniciar sesión mediante 3 botones llamados: “Ingresar con Google”, “Ingresar con correo y contraseña”, y “Others”.

**RF01-01: Ingresar con Google**Al seleccionar el botón “Ingresar con Google” se redirecciona a una ventana externa para ingresar con la cuenta de Gmail deseada, una vez verificada la cuenta se muestra su foto y nombre, y el sistema proporciona al usuario completar sus datos a través de un botón llamado “Completar Datos”.

**RF01-01-01: Completar Datos Usuario**  
El sistema pide al usuario ingresar su rut y seleccionar el tipo de usuario de la aplicación, una vez ingresados los datos el usuario debe seleccionar el botón “Finalizar” para pasar a la siguiente ventana.

**RF01-01-02: Completar Datos Delivery**  
El sistema mostrará una pestaña donde el usuario deberá ingresar distintos datos para completar su información.

**RF01-01-02-01: Ingresar nombre**  
El usuario deberá ingresar su nombre y apellido en el recuadro “Nombre”.

**RF01-01-02-02: Ingresar número de celular**  
El usuario deberá ingresar su teléfono celular en el recuadro “Número”.

**RF01-01-02-03: Ingresar ciudad**  
El usuario deberá ingresar la ciudad en la que vive en el recuadro “Ciudad”.

**RF01-01-02-04: Ingresar código**  
El usuario podrá ingresar un código de invitación de manera opcional en el recuadro “Código”.

**RF01-01-02-05: Confirmación de los datos**  
Una vez el usuario ha ingresado todos los datos debe seleccionar el botón “Siguiente” para pasar a la siguiente pestaña.

**RF01-01-02-06: Seleccionar medio de transporte**  
El sistema le permite al usuario seleccionar el medio de transporte a utilizar, este dispone de 3 medios: automóvil, bicicleta y motocicleta. Una vez seleccionado el usuario selecciona el botón “Siguiente” para pasar a la siguiente pestaña.

**RF01-01-02-07: Envío de documentos**  
El sistema le muestra al usuario una pestaña que lo redirigirá a una nueva ventana donde el usuario debe enviar los documentos necesarios.

**RF01-01-02-08: Código de verificación**  
El usuario deberá ingresar el código de verificación proporcionado por la página en el recuadro “Código”.

**RF01-01-02-09: Verificación de datos**  
El usuario debe seleccionar el botón “Verificar Datos” y pasará a la siguiente pestaña.

**RF01-01-03: Completar Datos Farmacia**  
El usuario debe ingresar la patente de la farmacia en el recuadro “Patente”. Una vez ingresado los datos el usuario debe seleccionar el botón “Siguiente” para pasar a la siguiente pestaña.

**RF01-01-03-01: Envío de documentos**  
El sistema le muestra al usuario una pestaña que lo redirigirá a una nueva ventana donde el usuario debe enviar los documentos necesarios.

**RF01-01-03-02: Código de verificación**  
El usuario deberá ingresar el código de verificación proporcionado por la página en el recuadro “Código”.

**RF01-01-03-03: Verificación de datos**  
El usuario debe seleccionar el botón “Verificar Daros” y pasará a la siguiente pestaña.

**RF02: Página principal**  
El sistema mostrará al usuario una pantalla con su información donde podrá seleccionar el botón “Menú” que lo redirigirá a la pestaña del menú.

**RF03: Menú usuario**  
El sistema ofrece al usuario 3 botones, que lo redirigirán a distintas pestañas, llamados: “Carrito Pedidos”, “Menú Locales”, “Historial de Compras” y “Seguimiento Compras”.

**RF03-01: Carrito de Pedidos**  
El sistema muestra una lista de los medicamentos elegidos por el usuario, donde puede seleccionar uno o más productos si lo desea.

**RF03-01-01: Eliminar productos**  
El usuario puede eliminar de la lista de compra los productos seleccionados al momento de apretar el botón “Eliminar”.

**RF03-01-02: Comprar productos**  
  
Al usuario seleccionar el botón “Comprar”, el sistema lo redireccionará a una nueva pestaña.

**RF03-02: Menú Locales**  
El sistema muestra un mapa donde se puede ver la ubicación del usuario y las farmacias cercanas.

**RF03-02-01: Buscar locales**  
El sistema le permite al usuario ingresar un campo de búsqueda mostrando las farmacias más cercanas.

**RF03-03: Historial de Compras**  
El sistema muestra una lista donde el usuario puede ver las compras realizadas, mostrando la fecha y la tienda, y el usuario puede seleccionar el botón “Ver” para poder revisar el detalle.

**RF03-04: Seguimiento Pedido**  
El sistema muestra un mapa donde se señala la ubicación del delivery, mostrando el estado del pedido y el tiempo aproximado de entrega.

**RF04: Configuración**  
El sistema ofrece un menú de configuración básica que contiene los siguientes botones: “Cambiar Correo”, “Cambiar Contraseña” y “Cerrar Sesión”.

**RF04-01: Cambiar Correo**  
Se abre una pestaña nueva donde se le pide ingresar los datos al usuario y consta de dos botones, “Cambiar Correo” y “Atrás” para cancelar la acción.

**RF04-01-01: Ingresar Correo**Se le pide al usuario que ingrese el nuevo correo electrónico en los campos “Correo” y “Confirmar Correo”.

**RF04-01-02: Validación cambio correo**  
  
El sistema realiza el cambio de correo una vez el usuario selecciona el botón “Cambiar Correo”.

**RF04-02: Cambiar Contraseña**  
Se abre una nueva pestaña donde se le pide al usuario ingresar los datos y consta de dos botones “Cambiar Contraseña” y “Atrás” para cancelar la acción.

**RF04-02-01: Ingresar Contraseña**  
Se le pide al usuario que ingrese la nueva contraseña en los campos “Contraseña” y “Confirmar Contraseña”.

**RF04-02-02: Validación cambio contraseña**El sistema realiza el cambio de contraseña una vez el usuario selecciona el botón “Cambiar Contraseña”.

**RF04-03: Cerrar Sesión**Se abre una nueva pestaña preguntando al usuario si está seguro de cerrar sesión y consta de dos botones “SI” para cerrar la sesión y “NO” para cancelar la acción.

**RF05: Menú Delivery**  
El sistema ofrece al usuario 3 botones, que lo redirigirán a otras pestañas, “Pedidos Disponibles”, “Historial Entregas” y “Entrega de Producto”.

**RF05-01: Pedidos Disponibles**  
El sistema muestra un mapa y una lista de pedidos con el nombre de usuario y la distancia a la que se encuentra.

**RF05-01-01: Detalles**El sistema muestra una lista con el nombre del producto con sus características y el precio, cuenta con 2 botones “Aceptar” para realizar el pedido y “Cancelar” para volver a la página anterior.

**RF05-02: Historial Entregas**El sistema muestra una lista con las entregas realizadas por el usuario ordenadas por fecha.

**RF05-03: Entrega de Producto**  
El sistema muestra un mapa con la ubicación del usuario y la dirección donde hacer la entrega, además de un tiempo estimado de llegada al destino. Cuenta con 2 botones llamados “Aviso Llegada” y “Entregado”.

**RF06: Menú Farmacia**  
El sistema ofrece al usuario 3 botones, que lo redirigirán a otras pestañas, “Pedidos Pendientes”, “Bodega” e “Historial de Ventas”.

**RF06-01: Pedidos Pendientes**El sistema muestra una lista con los pedidos pendientes, esta contiene el rut y nombre del cliente, y el usuario puede seleccionar los pedidos, cuenta además con dos botones “Rechazar” y “Siguiente”.

**RF06-01-01: Tiempo de Espera**  
El sistema muestra un reloj donde el usuario puede seleccionar el tiempo aproximado de entrega del producto, y una lista donde muestra los clientes con un botón “Detalle” para mostrar el detalle de su pedido, y cuenta con 2 botones: “Atrás” para regresar a la página anterior y “Aceptar” para realizar el pedido.

**RF06-02: Bodega**El sistema muestra una lista con productos, sus características y la cantidad que contiene, el usuario puede seleccionar un producto para modificarlo o puede agregar un nuevo producto.

**RF06-02-01: Agregar Producto**  
El sistema muestra un formulario donde el usuario debe ingresar los datos: nombre del producto, código, peso y el stock disponible. Cancelando la acción al presionar el botón “Atrás” y confirmando la acción al presionar el botón “Guardar”.

**RF06-02-02: Modificar Producto**  
El sistema muestra un formulario donde el usuario puede modificar los datos: nombre, código, peso y stock disponible del producto. Cancelando la acción al presionar el botón “Atrás” y confirmando la acción al presionar el botón “Guardar”.

**RF06-03: Historial de Ventas**  
El sistema muestra un gráfico con las ventas realizadas por la farmacia y además una lista de los productos vendidos.