



**UNIVERSIDAD DON BOSCO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE COMPUTACIÓN**  
**DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA**  
**CICLO 02-2021**

**TEMA:**

Primera fase proyecto de cátedra

**INTEGRANTES:**

- |                                 |          |
|---------------------------------|----------|
| • Josué David Avelar Ponce      | AP173028 |
| • Ebony Gabriela Peña Gómez     | PG180491 |
| • Katherine Dayanna Díaz Zúniga | DZ180675 |
| • Nelson Orlando Lemus Cardoza  | LC111108 |

**GRUPO TEÓRICO:**

02T

**DOCENTE:**

Ing. Alexander Sigüenza

**FECHA DE ENTREGA:**

Lunes 6 de septiembre de 2021

## Contenido

1. Introducción .....	3
2. Diseño UX/UI .....	4
3. Lógica a utilizar para la resolución del problema .....	12
3.1. Diagramas UML .....	13
3.2. Metodología a utilizar .....	14
4. Herramientas a utilizar .....	18
5. Presupuesto de la aplicación .....	21
6. Referencias .....	22

## 1. Introducción

A través del presente trabajo se pretende mostrar los aspectos generales del proyecto de cátedra correspondiente a la materia Diseño y programación de software multiplataforma (DPS).

El proyecto a realizar se orientará a la creación de una aplicación móvil para una farmacia. Para una correcta ejecución del proyecto, es necesario plantear ciertos aspectos que nos ayudarán a llevar orden a la hora de su realización.

Para la etapa de planificación tomamos en cuenta aspectos generales tales como la definición del diseño UX/UI, la lógica que se utilizará para resolver el problema y crear la aplicación, los diagramas UML del proyecto, la definición de las herramientas a utilizar para la adecuada realización del proyecto.

Además, se define la metodología en la cual se basará el equipo de trabajo para desarrollar la aplicación, definiendo roles y otros aspectos importantes.

Cabe destacar que durante esta etapa de planificación se han creado tableros en Notion para tener un mejor panorama acerca de las actividades que hay que realizar en el futuro, y llevar un control de las actividades que se han completado y las que están en ejecución actualmente;

Para finalizar, en el documento se plantea un presupuesto para la realización de la aplicación móvil.

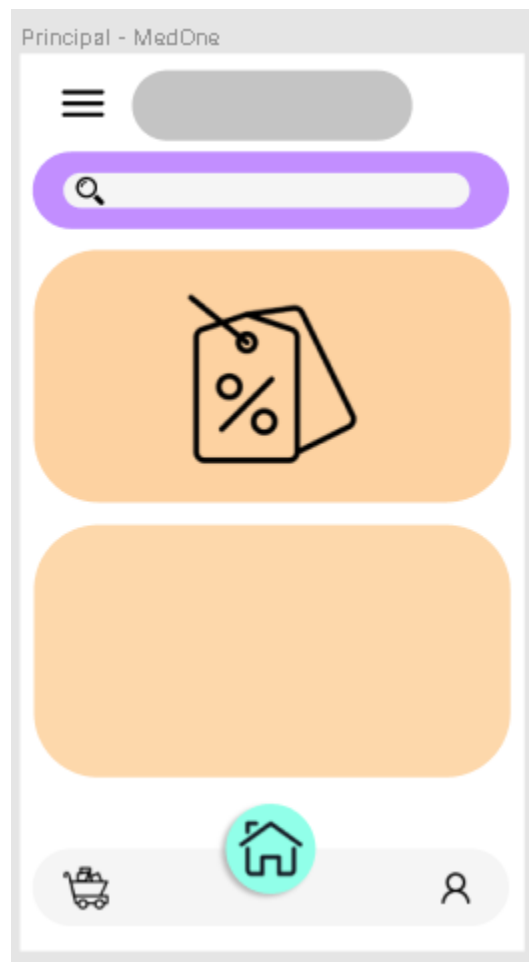
Se espera que por medio de este documento, el lector pueda comprender y tener un panorama más claro de lo que se pretende lograr con este proyecto, y que a partir de esto tenga una mejor idea de cómo se planea que la aplicación móvil funcione y cómo se pretende que luzca estéticamente, a través del uso de colores y estructura a utilizar en el frontend.

## 2. Diseño UX/UI

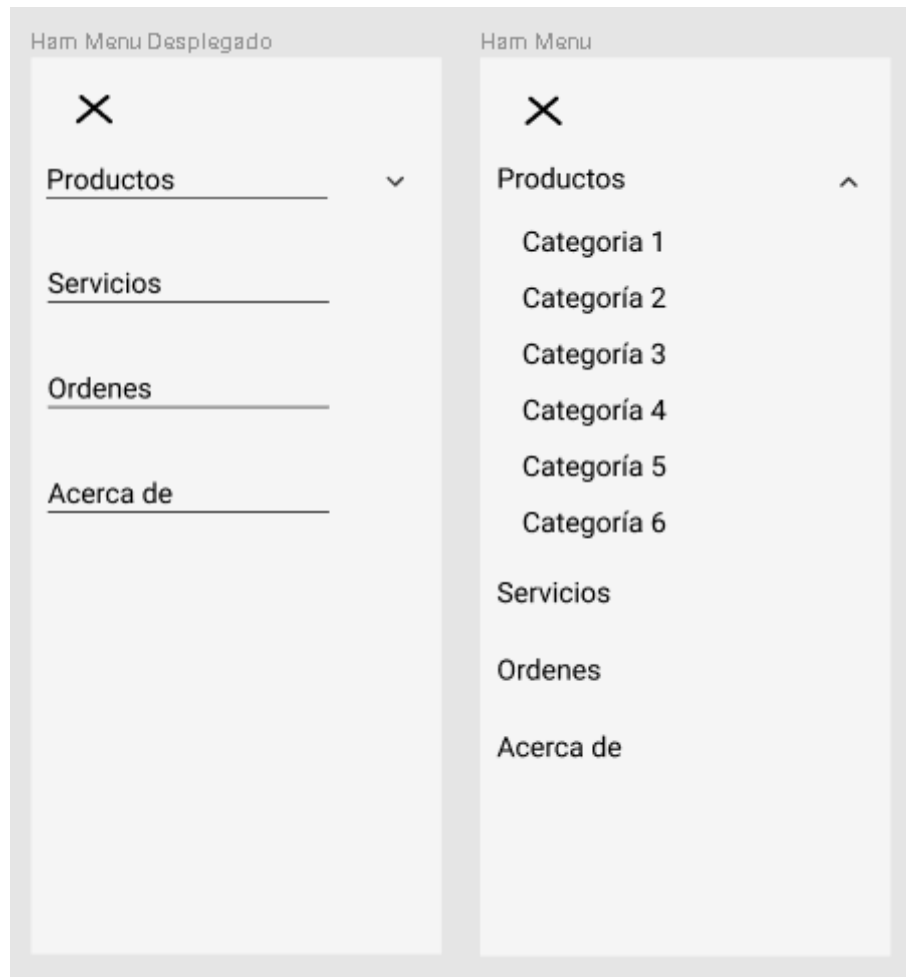
Los diseños MockUps cumplen una función importante dentro del avance de un proyecto, puesto que ilustran de mejor manera las ideas del equipo de trabajo para el desarrollo de todo tipo de aplicaciones, que estén sujetas a dar solución a un problema propuesto.

Para el desarrollo de la aplicación MedOne se han planteado distintas etapas que corresponderán al correcto funcionamiento de la misma. Entre ellas se tienen:

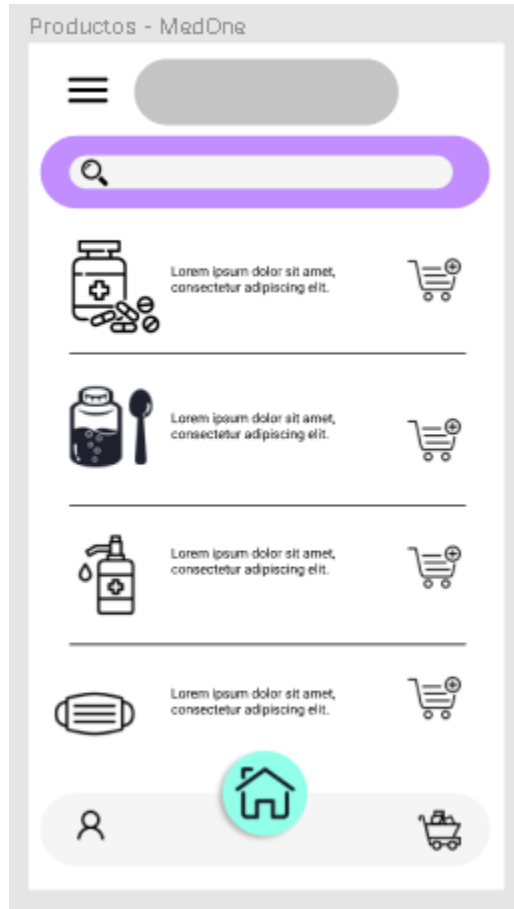
- ❖ **Home:** el cual poseerá información relevante como las promociones que la farmacia manejará y el logo de la misma, de igual forma, distintas opciones con las que el usuario podrá interactuar, por ejemplo, el carrito de compras, login, barra de búsqueda y un menú hamburguesa.



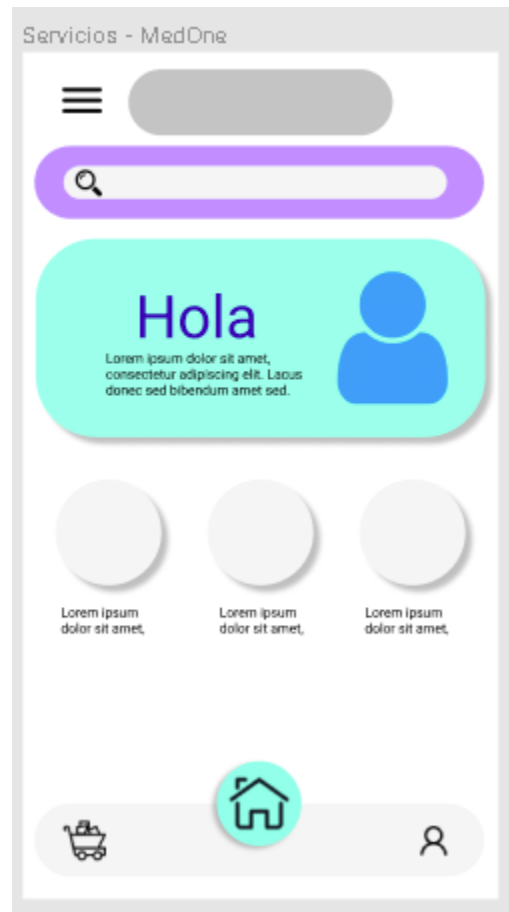
- ❖ **Menú Hamburguesa:** contendrá más información con la que el usuario podrá interactuar, como los productos, los servicios y más.



- ❖ **Productos:** contará con un menú desplegable el cual muestre las categorías a la que pertenezca cada producto. En ella, se mostrarán las descripciones de los productos y la opción de agregarlos al carrito.



- ❖ **Servicios:** mostrará información de los servicios que la farmacia maneje de forma estática sin interacción alguna.



- ❖ **Ordenes:** en este apartado, se visualizarán las compras que se han realizado con anterioridad.





- ❖ **Perfil:** en esta sección, se pedirán la información personal para poder ser registrada dentro de la aplicación.

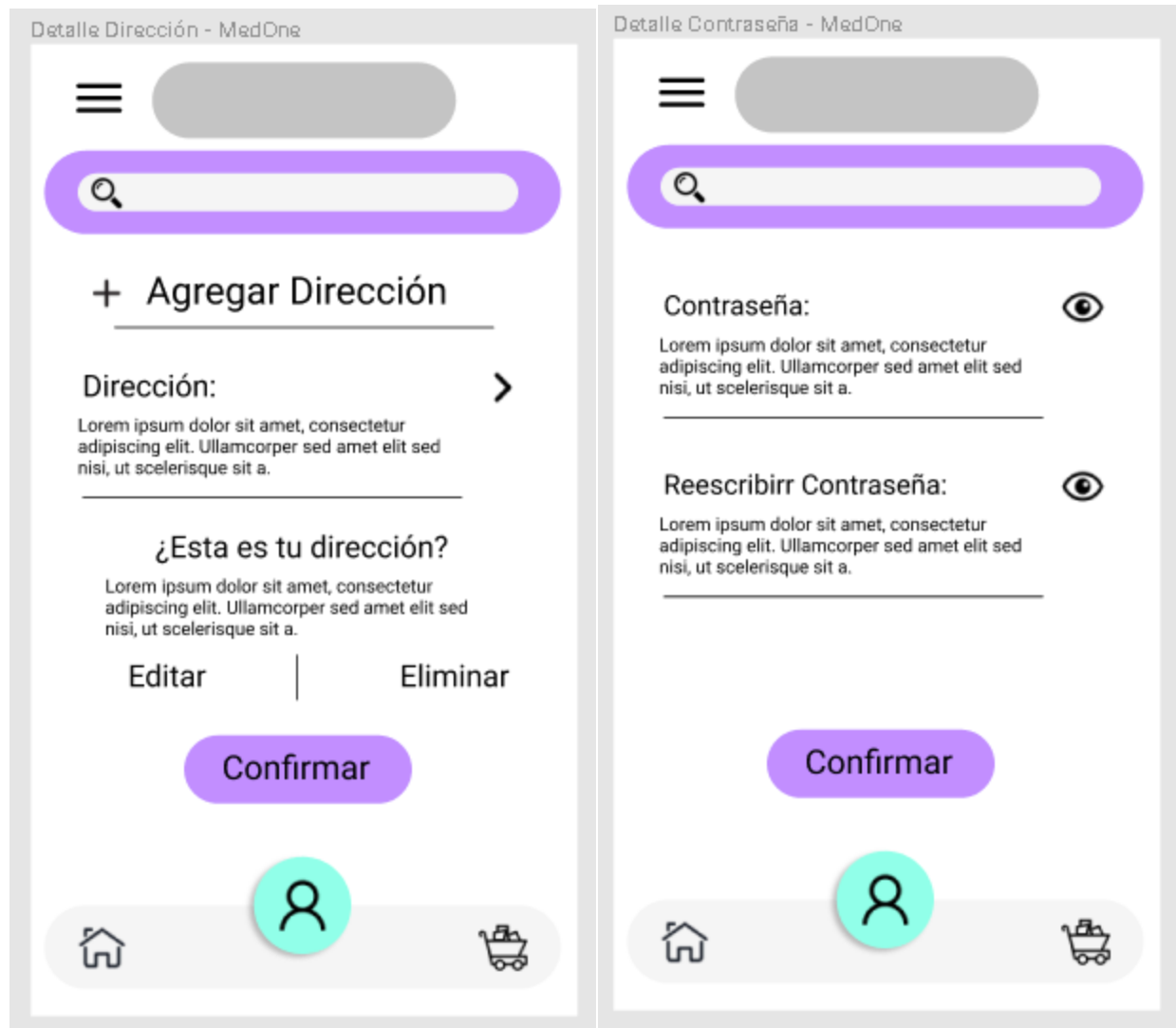
The image displays two side-by-side mobile application screens for user registration.

**Left Screen: Perfil - MedOne**

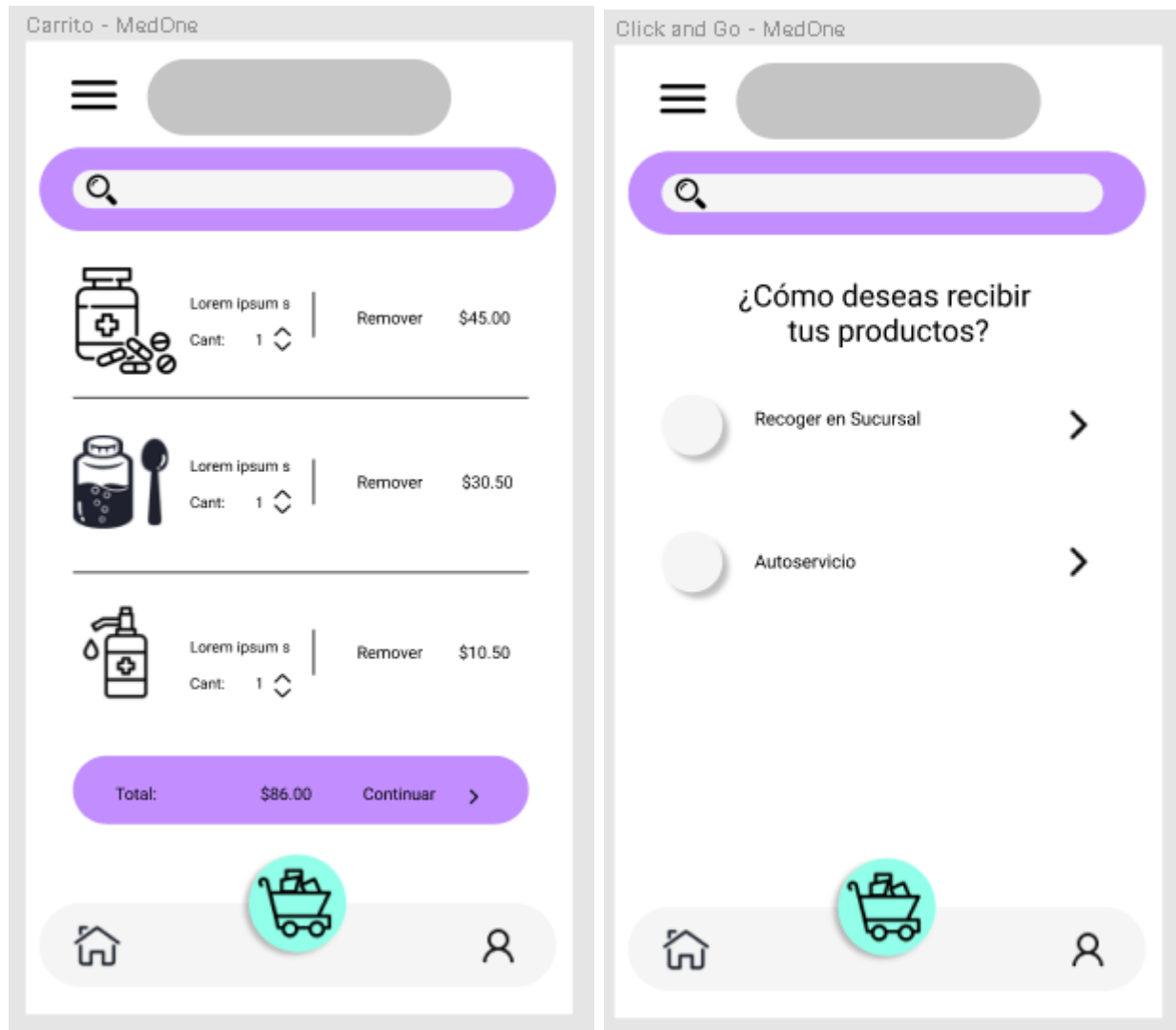
- Header: "Perfil - MedOne"
- Navigation: Hamburger menu icon, profile icon, and search icon.
- Search Bar: A rounded rectangular search bar with a magnifying glass icon.
- Menu Items:
  - Perfil >
  - Direcciones >
  - Contraseña >
- Registration Button: A rounded purple button labeled "Registrars".
- Bottom Bar: Home icon, profile icon, and shopping cart icon.

**Right Screen: Detalle Perfil - MedOne**

- Header: "Detalle Perfil - MedOne"
- Navigation: Hamburger menu icon, profile icon, and search icon.
- Search Bar: A rounded rectangular search bar with a magnifying glass icon.
- Form Fields:
  - Nombre: Input field with a label "Nombre:".
  - Apellido: Input field with a label "Apellido:".
  - Correo: Input field with a label "Correo:".
  - Teléfono: Input field with a label "Teléfono:".
- Confirmation Button: A rounded purple button labeled "Confirmar".
- Bottom Bar: Home icon, profile icon, and shopping cart icon.



- ❖ **Carrito y Método de Entrega:** en el carrito, se mostrarán información del producto seleccionado, así como la cantidad y su precio, de igual forma la opción de poder remover dicho elemento. Una vez realizada la compra el usuario podrá escoger el método de entrega que más le convenga.



### 3. Lógica a utilizar para la resolución del problema

La metodología que el usuario y el administrador podrán seguir para una buena interacción dentro de la aplicación será la siguiente, tomando en cuenta las ventanas anteriormente mostradas, así como sus etapas:

#### ❖ **Cliente.**

- **Home:** Es la ventana principal de toda la aplicación, el cliente podrá acceder a diferentes sitios, como los productos, carrito, ver promociones, entre otros. Sin la necesidad de haberse registrado.
- **Menú Hamburguesa:** Es un menú desplegable con el que tendrá interacción el usuario con las siguientes opciones.
  - **Productos:** La persona podrá visualizar los productos por categoría. Al momento de seleccionar la categoría, se mostrarán los diferentes productos, así como una descripción del mismo y la opción de poder agregarlo al carrito.
  - **Servicios:** Aquí únicamente podrá visualizar los diferentes servicios que maneja la farmacia.
  - **Ordenes:** El cliente tendrá acceso a ver las compras que ha realizado con anterioridad dentro de la farmacia.
  - **Acerca de:** Simplemente se podrá observar los derechos de autor y el copyright de la aplicación.
- **Perfil:** Para este apartado, si la persona no está registrada dentro de la aplicación, podrá ingresar todos sus datos para poder estarlo, una vez hecho eso, podrá visualizar la información que ha digitado.
- **Carrito:** La persona en esta ventana podrá aumentar la cantidad del producto que seleccionó con anterioridad, además de ver el precio del mismo y el total a pagar.
  - **Método de Entrega:** En este apartado, se podrá escoger la forma de entrega del producto que se manejará por parte de MedOne.
- **Búsqueda:** El usuario podrá realizar una búsqueda más específica de lo que desee.

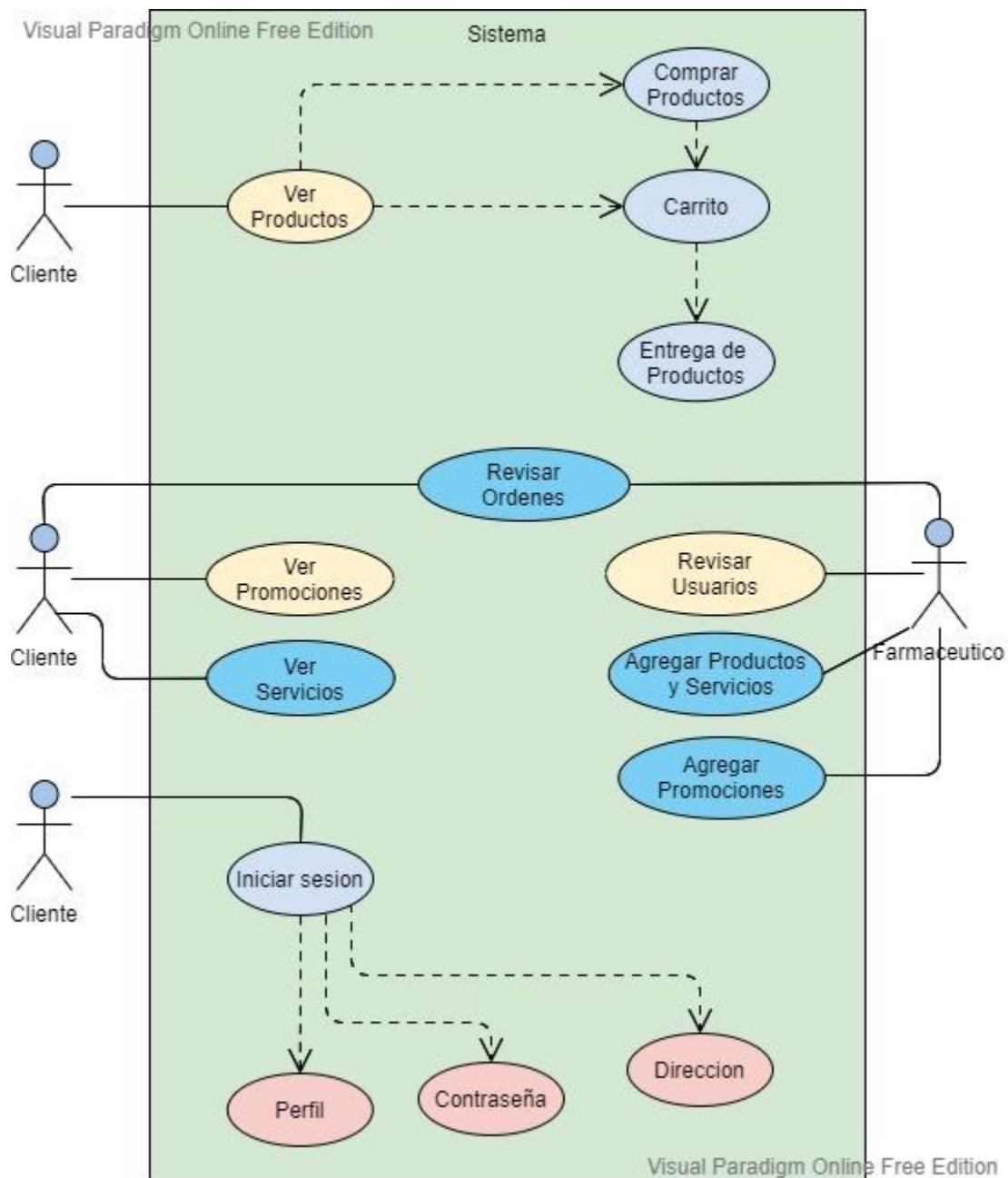
#### ❖ **Administrador.**

- **Home:** Para esta ventana, el administrador podrá agregar las promociones que considere pertinente.
- **Menú Hamburguesa:** En este menú podrá agregar o quitar las categorías manejadas por la farmacia.
  - **Productos:** Los productos son vital para las farmacias, por ende, este podrá, de igual forma, agregar o eliminar los productos según categoría.
  - **Servicios:** Aquí, como en anteriores apartados se maneja, se podrán agregar más servicios o incluso eliminarlos.

- **Ordenes:** El administrador podrá visualizar el historial de compras de los usuarios.
- **Perfil:** Para poder distinguirse de los demás, el administrador al momento de registrarse contará con un acceso especial, pero, similar al de los usuarios.

### 3.1. Diagramas UML

El diagrama UML seleccionado fue el: **Caso de Uso** esta herramienta ilustrará de mejor manera lo anteriormente planteado, así como los procesos que se llevan a cabo dentro de la aplicación.



### 3.2. Metodología a utilizar

Para la realización de este proyecto nos basaremos en la metodología SCRUM:

#### **¿Qué es SCRUM?**

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

#### **Roles en SCRUM**

En la metodología Scrum podemos identificar tres roles principales: Product Owner, Scrum Master y el Scrum Team. Estos tres roles obligatoriamente se requieren para crear el producto del proyecto, son los responsables del éxito de cada sprint y del proyecto en su totalidad. En conjunto se les conoce como el Scrum Core Team.

A continuación, describiremos con más detalle cada uno de los roles y es importante mencionar que ningún rol tiene autoridad sobre los otros:

#### **Product Owner**

El Product Owner o dueño del producto es conocido como “la voz del cliente”, ya que debe entender las necesidades y prioridades de los stakeholders, incluyendo los clientes y los usuarios. Este rol es responsable, entre otras cosas, de garantizar una comunicación clara sobre el producto, de definir los criterios de aceptación y asegurar que se cumplan. Dicho de otra forma, el Product Owner es responsable de asegurar que el equipo Scrum ofrezca valor.

A continuación, se enlistan algunas de las responsabilidades del Product Owner que consideramos más relevantes:

- Definir la visión del proyecto.
- Ayudar a finalizar la elección del Scrum Master para el proyecto.
- Identificar a los stakeholders.
- Ayudar a determinar a los miembros del equipo Scrum.
- Crear épicas y prototipos.
- Priorizar los elementos de la lista priorizada de pendientes del producto (Product Backlog).
- Definir los criterios de terminado.
- Ayudar a crear historias de usuario.
- Explicar las historias de usuario al equipo Scrum, al tiempo que crea la lista de tareas.
- Mantener la lista priorizada de pendientes del producto.
- Aceptar/Rechazar los entregables.
- Representar a los usuarios del producto o servicio con un profundo conocimiento de la comunidad de usuarios.
- Asegurar los recursos financieros del proyecto.

## **Scrum Master**

El Scrum Master debe tener una actitud de líder servicial (Servant Leader) para el equipo Scrum. Es quien modera y facilita las interacciones del equipo como facilitador y motivador. Este rol es responsable de asegurar un ambiente de trabajo productivo para el equipo, debe protegerlo de influencias externas, despejar los obstáculos y garantizar que se cumplan los principios, aspectos y procesos de la metodología Scrum.

A continuación, se enlistan algunas de las responsabilidades del Scrum Master que consideramos más relevantes:

- Ayudar a identificar a los stakeholders para el proyecto.
- Facilita la selección del equipo Scrum.
- Garantizar que los recursos de respaldo estén disponibles para el funcionamiento del proyecto sin problemas.

- Ayudar al Product Owner en la creación de la lista priorizada de pendientes del producto y en la definición de criterios de aceptación.
- Determina la duración del sprint.
- Apoya al equipo Scrum en la estimación del esfuerzo necesario para completar las tareas acordadas para el sprint
- Ayuda a actualizar el tablero Scrum y el registro de impedimentos.
- Facilita las reuniones de revisión de la lista priorizada de pendientes del producto.
- Se asegura que los problemas que afectan al equipo Scrum se discutan y se resuelvan.
- Garantizar que exista un ambiente ideal para el equipo Scrum.

## Scrum Team

El Scrum Team también es conocido como equipo de desarrollo, ya que es responsable del desarrollo, servicio o de cualquier otro resultado. Consiste en un grupo de personas que trabajan en las historias de usuario en la lista de pendientes del sprint para crear los entregables del proyecto. Nadie, ni siquiera el Scrum Master, indica al Scrum Team cómo cumplir los objetivos del sprint, es un equipo auto gestionado y multifuncional que cuenta con todas las habilidades necesarias. El tamaño óptimo de un equipo Scrum es de seis a diez miembros.

A continuación, se enlistan algunas de las responsabilidades del Scrum Team que consideramos más relevantes:

- Asegura una comprensión clara de las épicas y prototipos.
- Entender las historias de usuario en la lista priorizada de pendientes del producto.
- Estar de acuerdo con los demás miembros del Scrum Core Team sobre la duración del Sprint.
- Estimar las historias de usuario aprobadas por el Product Owner.
- Asignar las historias de usuario que se hacen en un Sprint.
- Desarrollar la lista de tareas en base a las historias de usuario ya convenidas y las dependencias.
- Crear entregables.
- Actualizar el registro de impedimentos y las dependencias.
- Actualizar la tabla del trabajo pendiente y el tablero Scrum.



- Realizar las reuniones diarias de pie (Daily Standup Meeting).
- Identificar oportunidades de mejora en la reunión de retrospectiva del sprint.
- Participar en la reunión de retrospectiva del proyecto.

Basado en lo anterior, consideraremos los siguientes roles:

<b>Product Owner</b>	<b>Josué Avelar</b>
<b>Scrum Master</b>	<b>Ebony Gómez</b>
<b>Scrum Team</b>	<b>Josué Avelar</b> <b>Ebony Gómez</b> <b>Dayanna Díaz</b> <b>Nelson Lemus</b> (por el hecho de ser un equipo pequeño y dado a que todos participaremos en las etapas de lógica, diseño y programación)

#### 4. Herramientas a utilizar

Para el desarrollo del proyecto se han estipulado 3 fases, las cuales se definen a continuación.

##### ❖ Fase 1: Organización – Prototipo de diseño.

- **Figma:** utilizado para la creación de los diseños UX/UI (mock ups),
- **Notion:** utilizado para distribuir las tareas basándonos en la metodología scrum y los roles de los participantes del proyecto.
- **Visual Paradigm:** utilizado en la creación de los diagramas UML para explicar el funcionamiento del diseño general de nuestro proyecto.

El detalle teórico de las herramientas a utilizar se define como:

- **Figma:** Es un software para editar gráficos vectoriales y que funciona como herramienta para la creación de prototipos que se basa principalmente en la web), las aplicaciones complementarias de Figma Mirror para Android y iOS permiten ver prototipos de Figma en tiempo real en dispositivos móviles, es decir que el conjunto de estas funciones se centra en crear una experiencia única al usuario.
- **Notion:** Es una macro herramienta multifuncional que permite crear: Notas, documentos, wikis, realizar proyectos y tareas de forma online y offline, una de las funciones más importantes es la sincronización multiplataforma (esta se basa en la nube), la cual es compatible para Windows y MAC, desde las respectivas IOS y Android, una vez creado tu usuario y cuenta lo que hagas en una plataforma estará disponible en todas las demás.
- **Visual Paradigm:** Ayuda a los equipos de desarrollo de software a capturar los requisitos correctos y transformarlos en un diseño preciso, este facilita familiarizarse con los diseños y diagramas de UML junto con ejemplos sobre cómo hacerlos.
- **Licencia Creative Commons:** se plantea desde un inicio el uso de licencias de derechos de autor para su uso progresivo durante todo el proyecto, con el objetivo de evitar cuestiones relacionadas con el plagio.

##### ❖ Fase 2: Repositorio – Diseño FronEnd.

- **GitHub:** se utilizará como repositorio para guardar las versiones de diseño y publicación de nuestro proyecto.
- **React Native:** utilizado para el diseño de la aplicación móvil, es decir; el FrontEnd de cara a cliente.
- **Licencia Creative Commons:** se utilizará para poder no caer en plagio de derechos de autor ya sea de nuestro diseño o de un futuro diseño sobre nuestro proyecto.
- **Android Studio:** se utilizará para el diseño y programación de la aplicación móvil.

El detalle teórico de las herramientas a utilizar se define como:

- **GitHub:** Es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.
- **React Native:** Es una plataforma muy reconocida como la mejor biblioteca de JavaScript de su clase para crear interfaces de usuario. Los componentes de React Native envuelven el código nativo existente e interactúan con las API nativas a través del paradigma de UI declarativo de React y JavaScript. Esto permite el desarrollo de aplicaciones nativas para equipos de desarrolladores completamente nuevos y puede permitir que los equipos nativos existentes trabajen mucho más rápidos.
- **Licencia Creative Commons:** Son licencias y herramientas de derechos autorales, la cual se enfoca en otorgar permisos de derechos autorales a diferentes obras creativas con el fin de respetar el origen y autor de las mismas, una fuente de contenidos que pueden ser copiados, distribuidos, editados, remezclados, y usados como base para crear, todo dentro de los límites de las leyes de derecho autoral.
- **Android Studio:** Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android

### ❖ Fase 3: Testeo – Desarrollo BackEnd.

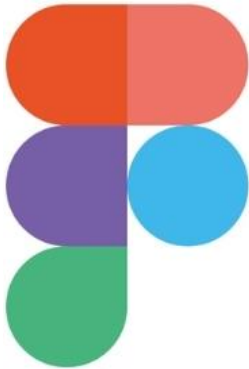
- **GitHub:** se utilizará como repositorio para guardar las versiones de diseño y publicación de nuestro proyecto.
- **React Native:** utilizado para el diseño y programación de la aplicación.
- **Licencia Creative Commons:** se utilizará para no caer en plagio de derechos de autor ya sea de nuestro diseño o de un futuro diseño sobre nuestro proyecto.
- **Node.js:** se utilizará como BackEnd junto con JavaScript que ya viene integrado para mejor manejo del proyecto en tiempo real. Agregaremos capacidades más interactivas con JavaScript usando React, React Router y Redux, además de darle estilo más estético con HTML y CSS.
- **Android Studio:** será utilizado para realizar el diseño y programación de la aplicación móvil, siguiendo la lógica planteada desde el inicio del proyecto.

El detalle teórico de la herramienta a utilizar se define como:

- **Node.js:** Es una de código abierto multiplataforma de back-end, JavaScript, entorno de ejecución que se ejecuta en el motor V8 y ejecuta el código JavaScript fuera de un navegador web. Node.js permite a los desarrolladores usar JavaScript para herramientas de línea de comando de escritura y de secuencias de

comandos del lado del servidor -Running secuencias de comandos del lado del servidor para producir la página web dinámica el contenido antes de que la página se envía al navegador web del usuario.

- **Android Studio:** Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android.



## 5. Presupuesto de la aplicación

La realización de un presupuesto resulta clave de cara a las futuras intenciones que se tengan con la aplicación, destacando que, esta se considera de tipo corporativa, es decir, una aplicación móvil que incorpora funcionalidades destinadas a la optimización de procesos y que están relacionadas directamente con la empresa y los usuarios de la misma.

Sabiendo el tipo de aplicación que se realizará, procederemos a plantear lo siguiente:

- ❖ **Precio/Hora por grupo:** Tendrá un valor de treinta dólares la hora (\$15/h para cada integrante del equipo, lo que resulta en \$60/h entre todo el equipo).
- ❖ **Pantallas que tendrá la aplicación:** Dispondrá de trece pantallas. Asignaremos un valor fijo de \$100 por pantalla.
- ❖ **Recursos que facilitará el cliente:** El proyecto iniciará desde cero, por lo que este parámetro tendrá un costo de \$0.
- ❖ **Costos asociados a servicios prestados por diseñador gráfico:** El diseñador gráfico se encargará de plantear una paleta de colores adecuada para toda la aplicación, así como imágenes para el uso en los sliders y creación de logos. Estimado de \$400, tomándolo como costo fijo.
- ❖ **Total, de horas trabajadas estimadas:** Se ha estimado trabajar un total de tres horas diarias durante cinco días a la semana por un periodo que comprende desde el siete de septiembre hasta el doce de noviembre, resultando un total de cuarenta y nueve días. Es el resultado de multiplicar los días de trabajo (49) por el precio/hora del grupo (\$60) por la cantidad de horas estimadas a trabajar por día.

Con lo anterior establecido, se procede a calcular el presupuesto del trabajo de la siguiente manera:

*Costo horas trabajadas*

$$= \text{Total de días estimados} \times \frac{\text{Precio}}{\text{Hora por grupo}} \times \text{Cantidad de horas al día}$$

$$\text{Días estimados: } 49; \frac{\text{Precio}}{\text{Hora}} \text{ por grupo: } \$60; \text{ Horas por día: } 3$$

$$\text{Costo horas trabajadas} = 49 \times \$60 \times 3 \text{ (horas diarias)} = \$8,820.00$$

$$\text{Costo diseñador gráfico} = \$400$$

$$\text{Costos fijos totales por cantidad de pantallas} = \$100 \times 13 = \$1,300$$

*Presupuesto total*

$$= \text{Costo diseñador gráfico} + \text{Costo horas trabajadas} + \text{Costo por pantallas}$$

$$\text{Presupuesto total} = \$400 + \$8,820 + \$1,300$$

$$\text{Presupuesto total} = \$10,520$$

## 6. Referencias

- ❖ 408Interactive. (31 de octubre de 2014). ¿Cómo hago el presupuesto de una App para mis clientes? <https://www.480interactive.com/blog/como-hago-el-presupuesto-de-una-app-para-mis-clientes/>
- ❖ Pradip Debnath. (2021, 06 de abril). Custom Bottom Tab Navigator in React Native | React Navigation v5 Tutorial. [Vídeo]. Youtube. <https://www.youtube.com/watch?v=gPaBicMaib4>
- ❖ Uxcristopher. (2021, 26 de julio). Figma tutorial - Cómo crear sombras. [Vídeo]. Youtube. <https://www.youtube.com/watch?v=B1eTYmiAf7w>
- ❖ Templune. (2020, 11 de octubre). Figma tutorial español | Como crear un menú lateral desplegable. [Vídeo]. Youtube. <https://www.youtube.com/watch?v=OHLMHzhQSJQ&t=7s>
- ❖ Garia, J. (20 de mayo de 2020). Notion: así funciona esta navaja suiza de productividad personal que acaba de lanzar un plan gratuito y casi ilimitado. Xataka. <https://www.xataka.com/aplicaciones/notion-asi-funciona-esta-navaja-suiza-productividad-personal-que-acaba-lanzar-plan-gratuito-casi-ilimitado>
- ❖ Capterra. (s.f.). Visual Paradigm. Recuperado el 03 de septiembre de 2021 de: <https://www.capterra.com/p/145716/Visual-Paradigm/>
- ❖ Gustavo, B. (08 de marzo de 2021). ¿Qué es GitHub y Cómo Usarlo? Hostinger Tutoriales. <https://www.hostinger.es/tutoriales/que-es-github>
- ❖ ReactNative. (s.f.). ReactNative – Learn once, write anywhere. Recuperado el 03 de septiembre de 2021 de: <https://reactnative.dev>
- ❖ Creative Commons. (s.f.). Sobre las licencias. Recuperado el 03 de septiembre de 2021 de: <https://creativecommons.org/licenses/?lang=es>
- ❖ Wikipedia. (s.f.). Node.js. Recuperado el 03 de septiembre de 2021 de: <https://en.wikipedia.org/wiki/Node.js>
- ❖ Wikipedia. (s.f.). Figma (software). Recuperado el 03 de septiembre de 2021 de: [https://en.wikipedia.org/wiki/Figma\\_\(software\)](https://en.wikipedia.org/wiki/Figma_(software))

### Otros enlaces de interés:

Enlace a Figma – MockUps de MedOne.

<https://www.figma.com/file/y0TBM6wnFz8PFrpJ4lCsWY/MedOne?node-id=17%3A9>

Enlace a Notion – MedOne. [https://phantom-koala-](https://phantom-koala-4d2.notion.site/2896796152f44b86a050a7ea3a13063f?v=a11d85884542404abce1a26ec466af2e)

[4d2.notion.site/2896796152f44b86a050a7ea3a13063f?v=a11d85884542404abce1a26ec466af2e](https://phantom-koala-4d2.notion.site/2896796152f44b86a050a7ea3a13063f?v=a11d85884542404abce1a26ec466af2e)

Enlace al repositorio de Github – MedOne.

[https://github.com/JosueAvelar/ProyectoDeCatedra\\_DPS\\_MedOne](https://github.com/JosueAvelar/ProyectoDeCatedra_DPS_MedOne)