



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

*Facultad de Ingeniería - Ingeniería en  
Computación*



# Mercado FI

## Segundo Parcial

Semestre 2023-1

Grupo: 03

**Cómputo Móvil**

**Equipo 5:**

**Alejandro Alpizar Lizbeth Viridiana**

**Calzada Maldonado Adrian Jonathan**

**Muñoz Garcia Arturo**

**Leyva Pérez José Luis**

**Profesor: Ing. Marduk Pérez de Lara Domínguez**

**Fecha: 6 de enero de 2023**

## ÍNDICE

<b>Objetivo</b>	<b>3</b>
<b>Introducción</b>	<b>3</b>
<b>MercadoFI</b>	<b>3</b>
Funcionalidades	3
Competencia	4
Distribución	5
Ventajas y desventajas	5
Tecnología implementada	6
Herramientas de desarrollo	7
Ganancias	9
Diseño	9
Prototipo de la aplicación	13
Características de la aplicación	13
Análisis de los datos	14
Detalles de almacenamiento	18
Involucrados (Perfiles)	20
Cotización	21
<b>Conclusiones</b>	<b>22</b>
<b>Referencias</b>	<b>23</b>

## **Objetivo**

Se realizará un análisis técnico de la aplicación MercadoFI, en el cual mencionaremos que tecnologías usamos, las pantallas, en que dispositivos se realizará, a que mercado va enfocado, sus ventajas y desventajas, etc; En el cual explicaremos el porque es una buena idea este tipo de aplicaciones que se creen.

## **Introducción**

En la universidad actualmente la dinámica que se maneja es la venta de artículos (comida, postres, boletos de conciertos, ropa etc) mediante grupos de Facebook, notamos que la deficiencia principal es el se pierden rápidamente las publicaciones, principalmente cuando no se obtienen reacciones o comentarios en ellas, la ventaja principal es que hay grupos que cuentan con muchos participantes. Con la intervención de una aplicación que actúe como intermediario entre los vendedores y compradores facilitamos la dinámica.

Al darnos cuenta de esto decidimos escoger este tipo de aplicación la cual es una aplicación retail, porque se ofrecen en venta diferentes artículos de diferentes proveedores a clientes. Proveedores y clientes son alumnos de la Facultad de Ingeniería, la cual va a facilitar la vida de los estudiantes en cuanto a búsqueda, compra y venta de artículos en Ciudad Universitaria.

## **MercadoFI**

### **Funcionalidades**

En la app se van a poder hacer varias cosas parecidas a diferentes app de compra, para que se les haga intuitivo o ya no salgan de lo que conocen. A continuación vamos a listar las funcionalidades las cuales va a implementar nuestra aplicación.

- Registro del usuario
- Inicio de sesión
- Búsqueda de artículos
  - Aplicación de filtros
- Manejo de inventarios

- Alta y baja de productos
- Características del producto
- Cantidad disponible
- Precio
- Pagos electrónicos y en efectivo
- Carrito de compras
- Usuario multifunción: vendedor y comprador
- Calificar vendedores y productos
- Autenticación en dos pasos
- Manejo de entregas
  - Pactar el punto de entrega
  - Pactar la hora de entrega
  - Habilitar chat entre vendedor y comprador

## Competencia

No hay aplicación parecida en Ciudad Universitaria, esto mejoraría la venta de productos y los puntos de entrega en Ciudad Universitaria, ya que al poder concentrar todos los artículos en una aplicación facilitará la búsqueda, compra y venta de productos mediante diferentes tipos de pagos como con tarjeta de crédito/débito y en efectivo.

Al estar enfocada a miembros de todas las facultades dentro de Ciudad Universitaria, no hay una app que directamente compiten con nosotros, sin embargo, sí hay aplicaciones o servicios que realizan actividades similares a la nuestra, como lo son:

- Mercado libre
- Marketplace en Facebook
- eBay

Las aplicaciones anteriores son las mejor posicionadas en el mercado actual, cada una ellas ya tiene establecida una firme infraestructura y hasta eficientes métodos de distribución. El principal reto para nuestra app es superar el atractivo que refleja una app con mayor renombre.

Afortunadamente para nuestra causa la mayoría de ellas no está especializada en un mercado tan informal como el que surge gracias a los alumnos de la UNAM, el potencial de agrupación de un mercado tan disperso es nuestra principal arma para hacernos de un número de usuarios y ganancias iniciales que vuelvan rentable a este proyecto.

## **Distribución**

La distribución de la app será mediante las principales tiendas de aplicaciones para ambos sistemas operativos y no tendrá ningún costo de adquisición. La cual esperamos que llegue a tiendas aproximadamente en 8 meses.

La publicidad también es un punto importante, inicialmente se dará a conocer a MercadoFI por los principales medios de distribución que ya utiliza la comunidad universitaria actualmente, específicamente nos enfocaremos en las redes sociales, grupos de Facebook, Telegram, WhatsApp, Instagram, etc.

## **Ventajas y desventajas**

Una de las principales ventajas es que no se necesitará de alguna patente o licencia que impida que otras apps.

En principio al no tratarse de un concepto especialmente único o innovador tecnológicamente hablando no es necesario adquirir o registrar una patente o licencia de forma inmediata.

Por precaución y pensando en un desarrollo a futuro, lo más idóneo es registrar adecuadamente el nombre, logotipos, arte, sonidos y lemas presentes en la aplicación.

También tenemos problemáticas es que se pueden derivar de la incursión en el mercado de una app dedicada a compraventa como la nuestra, hay que tener especial cuidado en la seguridad de todos los usuarios, es necesario monitorear que todos los productos ofertados por los vendedores sean legales y cuenten con cierto nivel de calidad. También es importante que cada vendedor y usuario sea debidamente registrado, en caso de una eventualidad o siniestro durante una transacción se necesita cierta información básica de ambas partes.

Al realizar un FODA podemos hacer sus características como se muestra

<b>Fortalezas</b> <ul style="list-style-type: none"> <li>• Es gratuita</li> <li>• Autenticación en dos pasos</li> <li>• Variedad en métodos de pago</li> <li>• Es multiplataformas</li> <li>• Puede usarse en cualquier facultad de la universidad.</li> </ul>	<b>Oportunidades</b> <ul style="list-style-type: none"> <li>• Usuarios multifunción</li> </ul>
<b>Debilidades</b> <ul style="list-style-type: none"> <li>• Los datos no se pueden cargar sin conexión y necesitan una conexión a Internet</li> <li>• No se cuenta con una campaña de marketing.</li> </ul>	<b>Amenazas</b> <ul style="list-style-type: none"> <li>• Riesgos de seguridad en contraseñas</li> <li>• Fraude al realizar los pagos</li> <li>• Autenticación de usuarios (sólo alumnos)</li> </ul>

## Tecnología implementada

Se planea que se encuentren en smartphones con sistema operativo IOS y Android, con un enfoque de desarrollo híbrido para abaratar costes de producción por medio de app. Se utilizará Flutter para poder compilar la aplicación en multiplataforma, por lo tanto la metodología de desarrollo es híbrida.

Esta dependerá totalmente de la nube ya que se utilizan administradores de bases de datos para poder gestionar los inventarios y las cuentas de usuario dentro de la app, se implementaron métodos para gestionar la seguridad de las cuentas y evitar accesos no autorizados dentro de la app. Gestión de pagos mediante tarjetas de crédito o débito.

Funcionará con datos, ya sea wifi o celulares y utiliza servicios de geolocalización la que será principalmente para poder mostrar el punto donde se hará la entrega del producto o donde se encuentra el usuario y el vendedor mediante.

Para poder obtener la geolocalización actual del usuario para android Usamos la librería de pubspec.yaml y los permisos de gps con ACCESS\_FINE\_LOCATION y ACCESS\_COARSE\_LOCATION. Add the following in AndroidManifest.xml.

Para iOS usamos para los permisos CLLocationWhenInUseUsageDescription y Info.plist

## Herramientas de desarrollo

### Flutter

Crear una aplicación y darle mantenimiento constante suele ser una tarea costosa y demandante, sobre todo si se considera la inmensa cantidad de dispositivos móviles disponibles en el mercado. Para crear las primeras versiones de Mercadofi se contempla el uso de un framework de desarrollo híbrido, con su ayuda se facilitará y reducirá los costes de uno de nuestros objetivos iniciales, garantizar de la mejor forma posible el funcionamiento y disponibilidad en dispositivos Android e IOS. Para esta tarea tan importante hemos optado por Flutter, un SDK desarrollado por Google para crear aplicaciones móviles.

Elegimos este framework ya que proporciona ciertas ventajas para tener en cuenta, brinda un rendimiento similar al nativo de cada plataforma, ofrece un entorno en el que el desarrollo se vuelve más rápido y la creación interfaces gráficas es muy flexible.

Flutter al ser creado por Google utiliza y aprovecha a “Dart” un lenguaje de programación creado por la misma empresa con el objetivo de convertirse en un sucesor de JavaScript, se diseñó con el objetivo de hacer el proceso de desarrollo lo más cómodo y rápido posible para los desarrolladores, brindando un conjunto bastante extenso de herramientas integrado, como su propio gestor de paquetes, varios compiladores/transpiladores, un analizador y formateador. Además, la máquina virtual de Dart y la compilación Just-in-Time hacen que los cambios realizados en el código se puedan ejecutar inmediatamente.

Una vez en producción, el código se puede compilar en lenguaje nativo, por lo que no es necesario un entorno especial para ejecutarlo.

## **GitHub**

Hoy en día es de vital importancia hacer uso de un repositorio durante el desarrollo de un proyecto, este facilita el trabajo en equipo, entendimiento del proyecto, resguardo de documentos importantes, avances y el estado actual de la última versión. Decidimos que GitHub será plataforma idónea para el correcto desarrollo de Mercadofi, además de las bondades anteriormente mencionadas particularmente GitHub es un servicio gratuito, cuenta con un sistema de control de versiones, es una de las plataformas de trabajo colaborativo en línea más grandes del mundo, es fácil de utilizar y se integra con Git.

## **Firebase**

Para desarrollar una aplicación no solo hay que enfocarse en la estética y ergonomía de uso, un aspecto vital, pero a veces dejado a un lado es diseño de un back-end eficiente, Firebase es una plataforma que ayudar en gran medida a que Mercadofi asegure la disponibilidad de su servicio, sea creada en un menor tiempo estimado y abarate distintos costos de producción.

Planeamos que gran parte de las actividades de la base de datos de la aplicación se aloje en el servicio de Realtime Database que proporciona la mencionada plataforma, se trata de una base de datos en tiempo real, back-end y organizada en forma de árbol JSON. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase.

## **Figma y Miro**

Antes de comenzar a programar y la creación del código duro es imperativo contar con un plan de acción o una imagen clara de los que sea que vamos a crear, el proyecto de Mercadofi no es una excepción a esto, para la creación de los wireframes principales decidimos utilizar Miro y para el desarrollo de un prototipo interactivo usamos Figma.

Miro es una aplicación para desarrollar flujos de trabajo en equipo de forma remota a través de una pizarra virtual infinita. En nuestro caso puede ser útil para crear rápidamente wireframes de la app o de sitios web. Idea en notas adhesivas y mapear en flujos de usuario.

Por otra parte, Figma se trata de una herramienta de prototipado web y editor de gráficos vectorial, que, a diferencia de las otras herramientas, se aloja en la web. Además, una de



las características que tiene Figma es que, al estar basada en el navegador y facilita compartir el proyecto todos los involucrados en su gestión y desarrollo.

## Ganancias

Los ingresos económicos que se pueden generar de la app pueden provenir de distintas fuentes, está planeado un espacio específico para publicidad, en donde diferentes empresas serán capaces de publicitarse un cierto número de veces a cambio de una remuneración monetaria. Otra ganancia contemplada es la posibilidad de destacar o darles relevancia prioritaria a productos ofertados por los vendedores a cambio de un precio fijo o un porcentaje de ventas. Por último y aunque no será obligatorio realizar transacciones dentro de la app será posible pagar por los productos dentro de ésta, a cambio de un porcentaje de venta, el comprador y vendedor se benefician con la diversidad de métodos de pago que puede recibir y realizar respectivamente.

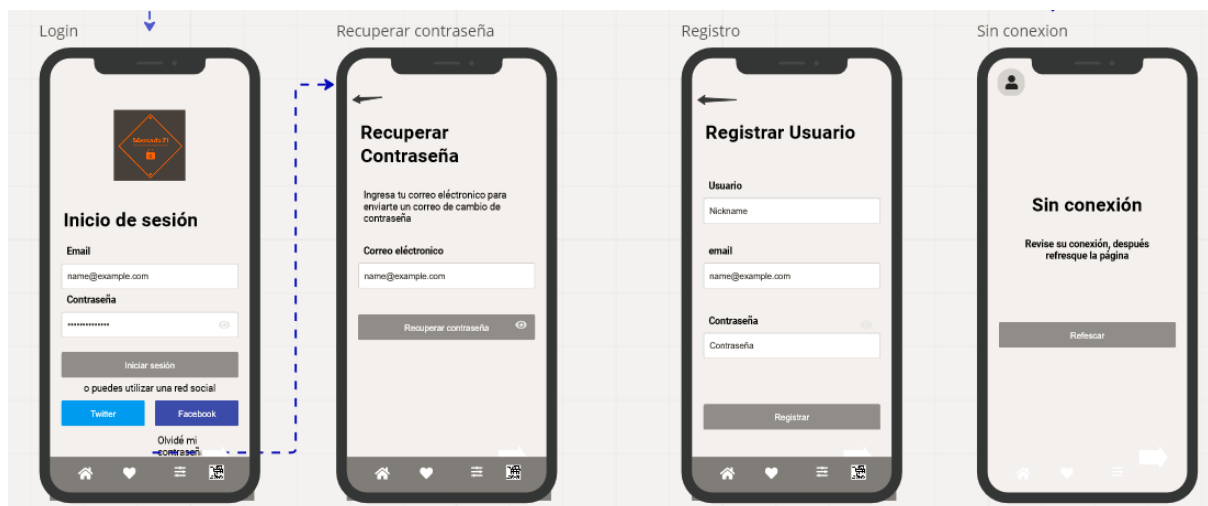
## Diseño

Diagrama de pantallas de la aplicación.

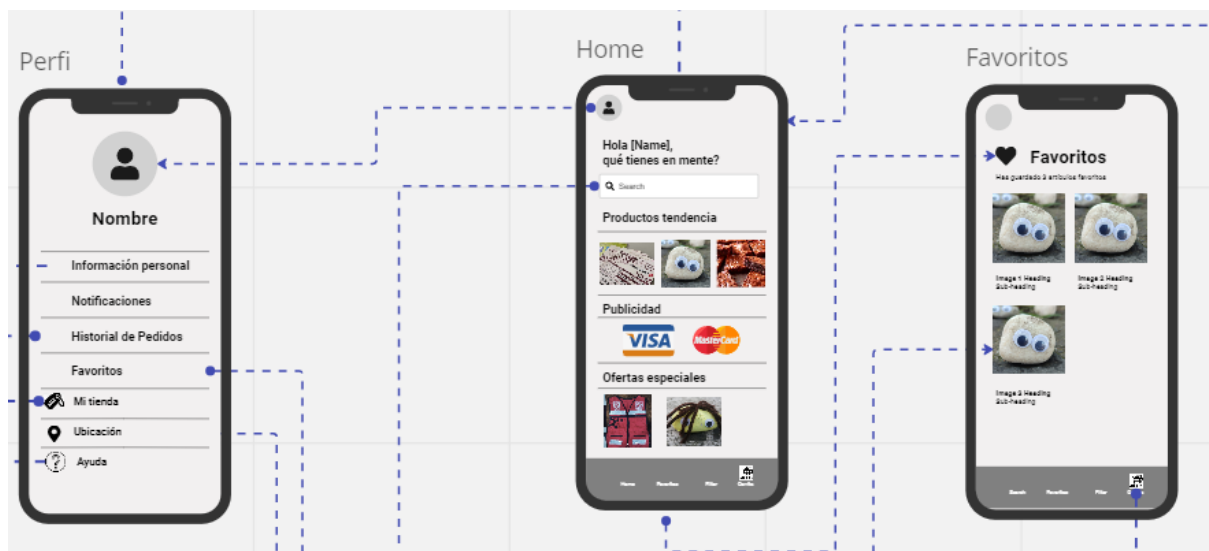
Para consultar el detalle con mejor calidad de imagen, visita el siguiente link:

[https://miro.com/app/board/uXjVPVHmbaE=?share\\_link\\_id=403076099059](https://miro.com/app/board/uXjVPVHmbaE=?share_link_id=403076099059)

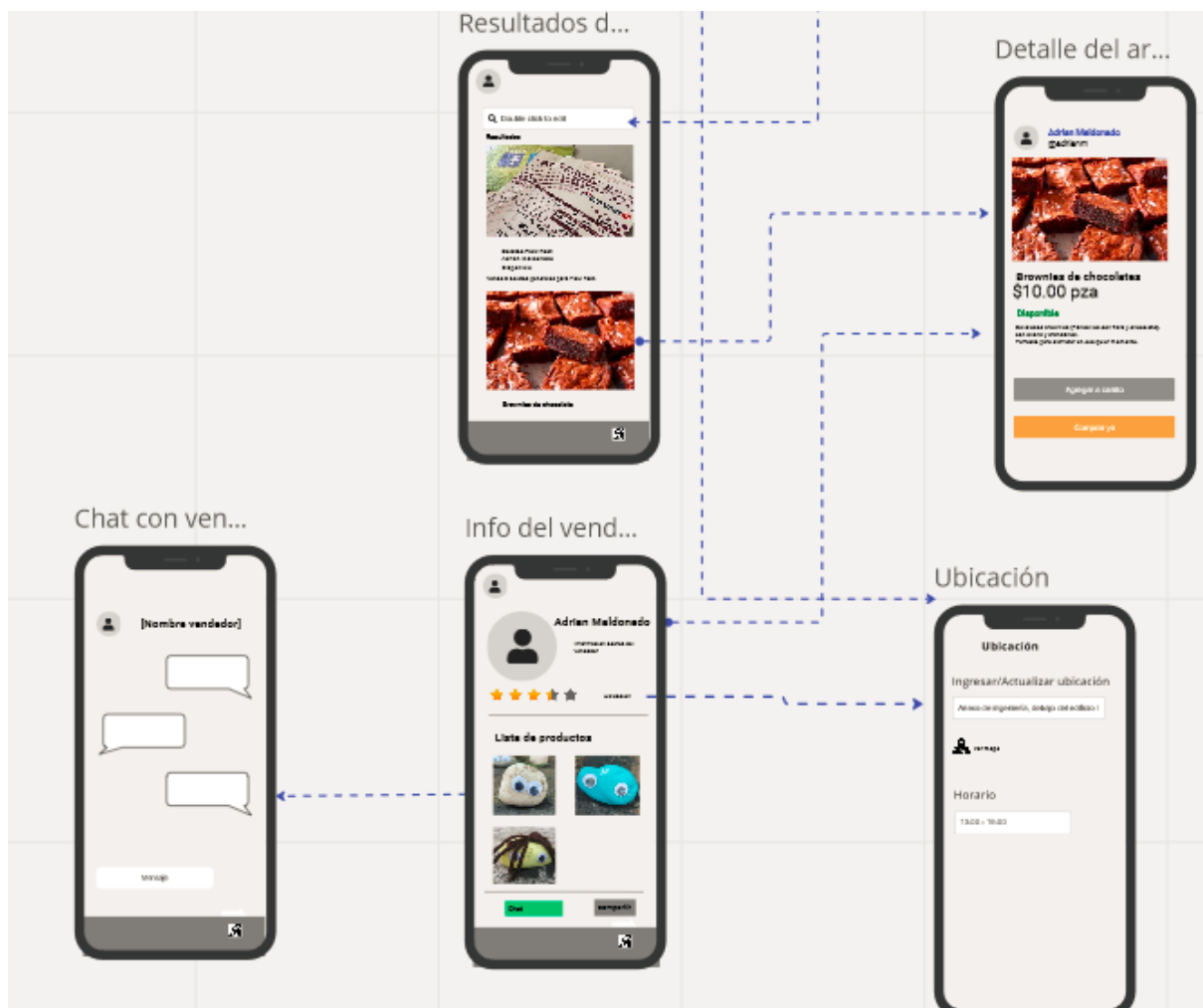
- **Login, Registro de Usuario y Sin conexión**



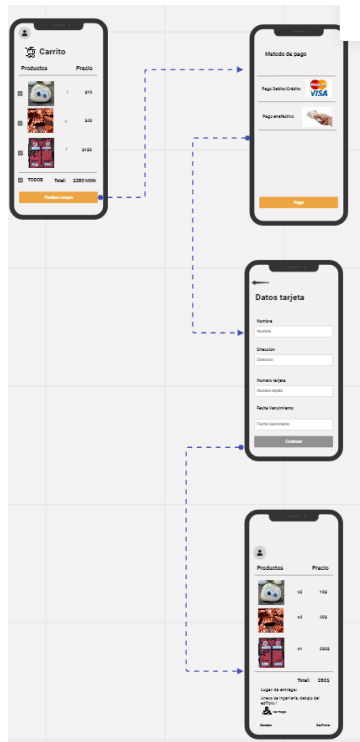
- **Home, Perfil, Favoritos**



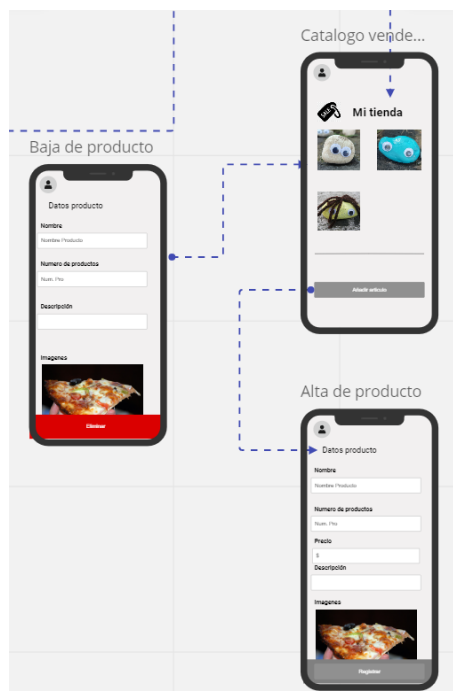
- **Resultado de búsqueda, Detalle del artículo, Perfil del vendedor, Chat con vendedor, Ubicación**



- **Carrito, Método de pago, Detalle de Pedido**



- **Catalogo vendedor, Baja de producto, Alta de producto**



- **Calificar vendedor, Calificar producto**



- **Ayuda, preguntas frecuentes y preguntas de seguridad**



## Prototipo de la aplicación

En el siguiente [link](#) se podrá observar la primera versión del prototipo de la aplicación, considerando las siguientes pantallas:

- Log out
- Login
- Registro
- Home
- Detalle del producto
- Búsqueda
- Perfil del vendedor
- Lista de conversaciones (chat)
- Chat individual

Video de demostración 1 [\(link\)](#)

Video de demostración 2 [\(link\)](#)

## Características de la aplicación

Considerando las siguientes características:

- Multiplataforma (iOS + Android)
- Dispositivos compatibles: Smartphones
- Diseño: Interfaz simple
- Aplicación gratuita con anuncios
- Sistema de login: Redes Sociales + email
- Integrada con servicio web, es decir que la aplicación debe obtener información mediante internet.
- Panel de administración: Un panel que permite dar de alta información para que pueda ser visualizada en tu aplicación (ej. alta de artículos del vendedor.)
- Idiomas disponibles: Español
- Notificaciones push: Esto permite enviar información a los dispositivos cuando no está abierta la aplicación (ej. códigos promocionales, mensajes del vendedor)
- Requiere guardar datos de información. Catálogo de productos
- Requiere el uso del hardware del dispositivo: GPS
- Consideramos que la app se encuentra en etapa de Diseño

## Análisis de los datos

Pantalla	Descripción (Funcionalidad)	Análisis de tipos de dato y atributos
<b>Logout</b>	Mostrará el logo de la aplicación mientras se inicia Se mostrarán los botones "iniciar sesión" y "registrarse"	img_splash: Image btn_Registrar: Button btn_Login: Button
<b>Registro de usuario</b>	Se encarga del registro de los usuarios nuevos, para esto solicita información básica como un nombre, e mail y contraseña	usuario: String email: String contraseña: String
<b>Login</b>	Solicita algún elemento para iniciar sesión: email y contraseña, facebook o twitter Olvidaste tu contraseña? Sirve para recuperar la contraseña Registrarse: muestra la pantalla para que el usuario se registre	email: String contraseña: String btn_Facebook: Button btn_Twitter: Button ForgotPwd: Button RegistroUsr: Button
<b>Recuperar contraseña</b>	Solicita el email con el que el usuario se registro para enviarle un correo y pueda cambiar contraseña Al presionar el botón "Recupera tu contraseña" se regresa a la pantalla "logout"	email: String btn_recoverPwd: Button
<b>Perfil</b>	Mostrará las diferentes opciones que se tienen para los usuarios dentro de la aplicación	Nombre: String Foto: Image Información personal: Button Notificaciones: Button Historial de Pedidos: Button Favoritos: Button Mi tienda: Button Ubicación: Button Ayuda: Button
<b>Información Personal</b>	Pantalla que permite ver y actualizar la información personal de los usuarios	FotoPerfil: Image Datos: String
<b>Notificaciones</b>	Mostrará cuando tengas algún mensaje de la aplicación Tache es para borrar las notificaciones	ListaNotificaciones: array notificacion: array msj_notificacion tache: Button

<b>Historial de pedidos</b>	Pantalla que muestra todos los productos comprados por el usuario, muestra el nombre, fotografía y precio del producto. También está la posibilidad de volver a comprar algún producto si este se encuentra disponible.	ListaDeProductos:array img_producto: Image precio: double volveraComprar: Button
<b>Calificación producto</b>	Pantalla muestra para poder ingresar una calificación del producto que va del 1 al 5 de estrellas y agregar comentarios respecto a este.	nombreProducto:String imagenProducto:image calificacionProducto:double comentarioProducto:string
<b>Calificación vendedor</b>	Pantalla muestra para poder ingresar una calificación del vendedor que va del 1 al 5 de estrellas y agregar comentarios respecto a este.	nombreVendedor:String imagenVendedor:image calificacionVendedor:double comentarioVendedor:string
<b>Mi tienda</b>	Pantalla que muestra los artículos publicados por los diferentes vendedores, nos da la opción de agregar los productos a un carrito de compra	ListaDeProductos:array img_producto: Image precio: double AgregarACarrito:Button CarritoDeCompra: Button
<b>Ubicación</b>	Pantalla que muestra la ubicación donde encontrarás a los vendedores	Ingresar/Actualizar ubicación: Text box Horario: Text box
<b>Soporte técnico</b>	Preguntas frecuentes, ayuda en tu cuenta	PreguntasFrecComp: Button ConfigCuenta: Button Seguridad: Button ComoPubl: Button ComoComp: Button Chat: Button
<b>Catálogo de Productos (Mi Tienda)</b>	Muestra los productos que se encuentran a la venta	listaProductos: array producto:array producto:array
<b>Baja de producto</b>	Muestra la lista de datos del producto que se quiere eliminar y el botón para eliminar	producto:array producto tiene los datos del producto botonEliminar:boolean
<b>Alta de producto</b>	Agrega o da de alta nuevo producto a el catálogo de un vendedor, para se solicita un nombre, número de existencias , descripción y una imagen representativa	producto:producto nombreProducto: String descripcion: String cantidad: int img_producto: Image precio: double

<b>Detalle de Producto</b>	Muestra el detalle del producto: nombre, imagen, nombre del vendedor, si esta disponible o no, una descripción, opción para agregar al carrito o comprarlo inmediatamente, comparaciones de diferentes productos	nombre: String img_comida: Image vendedor: String btn_addCart: Button btn_buyNow: Button calificacio:double nombreProductoComparacio1:String imagenProductoComparacio1:Image vendedorProductoComparacio1:String calificacionProductoComparacio1:doub le precioProductoComparacio1:double nombreProductoComparacio2:String imagenProductoComparacio2:Image vendedorProductoComparacio2:String calificacionProductoComparacio2:doub le precioProductoComparacio2:double nombreProductoComparacio3:String imagenProductoComparacio3:Image vendedorProductoComparacio3:String calificacionProductoComparacio3:doub le precioProductoComparacio3:double
<b>Info del vendedor</b>	Pantalla que muestra y permite modificar la información del vendedor así como la lista de los productos que vende	DatosVendedor: String ListaProductos:Array FotoPerfil:Image
<b>Chat</b>	Mostrará el chat con el vendedor o cliente, se podrá enviar y recibir mensajes. ListaMensaje se llenará con una lista de los mensajes Al apretar el botonEnviar enviará el mensaje y eliminará el recuadro donde esta el mensaje enviado	listaMensajes:array mensajeEnviar:string botonEnviar:Button



<b>Home</b>	Pantalla principal de la aplicación, se visualizará por defecto cada vez que un usuario entre a la app e inicie sesión con éxito, en ella se mostrará distinta información y herramientas importantes como lo es la barra de búsqueda, barra de navegación, un botón al perfil del usuario, botón de notificaciones, productos tendencia, espacio publicitario y productos con ofertas especiales	listaProductosTendencia: array listaProductosOferta: array busqueda: String numeroNotificaciones: int listaPublicidad : array botonNotifi:Button botonPerfil:Button
<b>Favoritos</b>	Pantalla que muestra los artículos agregados a la lista de favoritos	Imagen producto: Image Nombre producto: Text
<b>Resultados de búsqueda</b>	Esta pantalla se encarga de mostrar los resultados de búsqueda, se le dará prioridad de visualización a los vendedores que decidan pagar una cuota de promoción.	ListadeProductos: Array ListadeProductosPriori: Array
<b>Carrito</b>	Pantalla que muestra los productos agregados al carrito de compras	ListadeProductos: Array Todos: Checkbox Total: int Finalizar compra: Button
<b>Forma de pago</b>	Seleccionara el tipo de paga que quiere	pago:string
<b>Datos de tarjeta de crédito</b>	Ingresar los datos de la tarjeta de crédito	nombre:string titular:String númerno tarjeta:String Fecha:Date CV:string
<b>Resumen de pedido</b>	Muestra la información representativa de un pedido, tal como la lista de productos deseados, y para cada uno de estos se ve su nombre, una imagen representativa, cantidad y precio; se imprime un precio total y lugar de entrega deseado.	listaProductos: array producto:array total: double destino: String botonCancelar:Button botonContinuar:Button
<b>Sin conexión</b>	Mostrará mensaje que no se pudo cargar porque no hay conexión a internet Al apretar el botón de refrescar, actualiza la página	botonRefescar:Button

## Detalles de almacenamiento

Pantalla	Análisis CRUD	Almacenamiento
<b>Logout</b>	Read a la base de datos para cargar la imagen	Nube
<b>Registro de usuario</b>	Create para almacenar todos los parámetros solicitados en la creación de una nueva cuenta	Nube
<b>Login</b>	Create para almacenar los parámetros de email	Nube
<b>Recuperar contraseña</b>	Read a la base de datos para confirmar la existencia del correo	N/A
<b>Perfil</b>	Se utilizarán funciones para que los botones puedan direccionar a las demás pantallas	N/A
<b>Información Personal</b>	Update a la base de datos para poder modificar la información de los usuarios	Nube
<b>Notificaciones</b>	Se usará un read de la base de datos para cargar las notificaciones y un delete para borrar las notificaciones que quiera el usuario	N/A
<b>Historial de pedidos</b>	Se utilizará read para cargar todos los datos de los productos y estados de compra de los productos adquiridos, y un update en caso de querer volver agregarlos al carrito de compra	Nube
<b>Calificación producto</b>	Se usará un read para poder cargar el nombre del producto y imagen y un create para poder poner la calificación y comentario del producto	Nube
<b>Calificación vendedor</b>	Se usará un read para poder cargar el nombre del vendedor y imagen y un create para poder poner la calificación y comentario del vendedor	Nube
<b>Mi tienda</b>	Se usarán funciones para realizar la consulta de los productos almacenados en la base de datos, se utilizarán funciones para insertar los productos	Nube

	seleccionados al carrito de compra	
<b>Ubicación</b>	Se usarán funciones para actualizar la ubicación y horarios cada vez que un vendedor ofrezca sus productos.	Nube
<b>Soporte técnico</b>	Se utilizará un read a la base de datos para obtener la respuesta predeterminada dependiendo de que ayuda necesite el usuario.	N/A
<b>Catálogo de Productos (Mi Tienda)</b>	Se usará un read de la base de datos para la listaProductos y mostrarlos en pantalla	Nube
<b>Baja de producto</b>	Se usará un delet para borrar el producto de la base	Nube
<b>Alta de producto</b>	Para el alta de un producto se hará un create para crear un nuevo producto en la base de datos y un update para actualizar el número de productos ofertados por el usuario.	Nube
<b>Detalle de Producto</b>	Se hará una lectura de los datos del producto y la lectura de productos relacionados con este con su información correspondiente	Nube
<b>Info del vendedor</b>	Se utilizarán funciones para insertar y actualizar la información del vendedor dentro de la base de datos	Nube
<b>Chat</b>	Se hará un read para cargar los datos que le llegan al usuario y un update para que se actualice la conversación	Nube
<b>Home</b>	En la pantalla home se utilizará un read para leer toda la información que se mostrará en pantalla como las ofertas, publicidad, tendencias, etc. Por otra parte, se hará un update al historial de búsqueda del usuario en caso de utilizar la barra de búsqueda.	Nube
<b>Favoritos</b>	Se utilizarán funciones para leer la base de datos y cargar los productos seleccionados	Nube

<b>Resultados de búsqueda</b>	Se llamará a la función read a la base de datos para obtener los productos que coincidan o sean similares con el texto de búsqueda.	N/A
<b>Carrito</b>	Se utilizarán funciones para leer la base de datos y cargar los artículos agregados al carrito así como la información de estos	Nube
<b>Forma de pago</b>	N/A	Local
<b>Datos de tarjeta de crédito</b>	N/A	Nube
<b>Resumen de pedido</b>	El resumen del pedido hará un read para mostrar toda la información del pedido sin confirmar del usuario, en caso de ser confirmado se usará la función de update a la lista de ventas pendientes de todos los vendedores involucrados.	Nube
<b>Sin conexión</b>	N/A	N/A

## Involucrados (Perfiles)

Una aplicación de este tipo es muy amplia las necesidades, ya que se necesitan varios expertos con conocimientos y experiencia en varias áreas como diseñadores, programadores, tester, etc; también es importante resaltar la importancia de una buena base de datos para resguardar todos los datos de las personas y transacciones que son realizadas en esta. Por esta razón se consideran los siguientes roles como indispensables para el desarrollo de la aplicación:

- Un Project Manager
- Un programador (Flutter)
- Un tester
- Un desarrollador backend
- Un analista de ciberseguridad

## Cotización

Para el desarrollo de esta aplicación se tomaron en cuenta los siguientes perfiles, investigando en la aplicación de CompuTrabajo se encontró el costo de sus servicios por mes.

- Project Manager 32,000
- Desarrollador flutter 20,000
- Tester 35,000
- Desarrollador backend 15,000
- Analista de ciberseguridad 20,000

Se realizó la cotización de un equipo de cómputo para cada uno de los perfiles que serán contratados en este proyecto en la página oficial de la empresa Dell.

Laptop Dell Latitude 14" 3420; Costo: 22,992.63



Servicio/Producto	Cantidad/Meses	Costo	Total
Project Manager	8	32,000	256,000 MXN
Desarrollador Flutter	8	20,000	160,000 MXN
Tester	8	18,000	144,000 MXN

Desarrollador backend	8	15,000	120,000 MXN
Analista de ciberseguridad	8	12,000	96,000 MXN
Laptop	5	22,992,63	114,963,15 MXN
Total			890,963,15 MXN

El costo de la base del servidor y base de datos una vez liberada la aplicación será de 2 558 USD el cual será sumado al costo total de la aplicación.

2,558 USD = 49,560,87 MXN

Costo total al día de liberación = 890,963,15 + 49,560,87 = 940,524,02 MXN

Costo total de la aplicación 940,524,02 MXN.

## Conclusiones

Después de investigar y analizar a la comunidad de la UNAM desde la perspectiva más objetiva posible, podemos decir que aprendimos y aplicamos una perspectiva más amplia sobre el diseño y composición de una aplicación, revisar distintos ángulos como su historia, mercado, tecnología, diseño, etc. Permitió que tuviéramos una noción sobre los distintos componentes para crear una aplicación, y dada la naturaleza de la app estudiada de igual forma ampliamos nuestros conocimientos de cómo se plantea una app de tipo retail y buscando la facilidad para poder vender artículos de una mejor manera en .

Haciendo el análisis con poco y/o ninguna de la experiencia que se tiene dudamos en cuánto tiempo se puede desarrollar esta y también en el presupuesto pero pudimos hacer el análisis que para el desarrollo estimamos: Aproximadamente 940,524,02 MXN. Esperando que este llegue a la finalidad de sustituir las aplicaciones que se usan por la aplicación.

## Referencias

1. Pismennaya, K. (2022, 12 septiembre). Mobile App Development Team: Tips, Structure, and Roles - Mind Studios. Blog - Mind Studios. Recuperado 23 de septiembre de 2022, de <https://themindstudios.com/blog/mobile-app-development-team/>
2. Entiende a detalle qué es GitHub y su importancia para un negocio. (2020, 20 marzo). rockcontent. Recuperado 1 de enero de 2023, de <https://rockcontent.com/es/blog/github/>
3. colaboradores de Wikipedia. (2022, 26 julio). Firebase. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Firebase>
4. Creador de Wireframes online para cualquier diseño | Miro. (s. f.). <https://miro.com/es/wireframe-online/>
5. ¿Qué es figma?| La mejor herramienta de prototipado web. (s. f.). CEI: Escuela de Diseño y Marketing. <https://cei.es/que-es-figma/>
6. Cristancho, F. (2022, 1 agosto). ¿Qué es Flutter y para qué sirve? - Talently. Talently Blog. <https://talently.tech/blog/que-es-flutter/>
7. Perfil Project Manager Computrabajo. <https://mx.computrabajo.com/trabajo-de-project-manager#CA6D5176C8E4F0B961373E686DCF3405>
8. Perfil Desarrollador Flutter Computrabajo. <https://mx.computrabajo.com/trabajo-de-desarrollador-flutter#0C3A1D4CA0ED7F7A61373E686DCF3405>
9. Perfil Tester Computrabajo. <https://mx.computrabajo.com/trabajo-de-tester-software#B83A3B9D8311667061373E686DCF3405>
10. Perfil Desarrollador Backend Computrabajo. <https://mx.computrabajo.com/trabajo-de-backend-developer?sal=3#4260DCC4958F37FF61373E686DCF3405>
11. Perfil Analista Ciberseguridad Computrabajo. <https://mx.computrabajo.com/trabajo-de-analista-de-ciberseguridad#2C07913E076B047A61373E686DCF3405>
12. Laptop Dell Latitude 3420. <https://www.dell.com/mx/empresas/p/latitude-14-3420-laptop/pd?oc=xctol342014mmcla>