UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

EASY NEGOCIOS: PLATAFORMA MÓVIL PARA ADMINISTRACIÓN DE EMPRENDIMIENTOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

DIEGO ALBERTO HURTADO GONZALEZ

PROFESOR GUÍA:
JÉRÉMY BARBAY

MIEMBROS DE LA COMISIÓN:
HUGO ANDRÉS MORA RIQUELME
FELIPE BRAVO MARQUEZ

SANTIAGO DE CHILE
2022

Resumen

Día a día mucha gente decide emprender para aumentar sus ingresos mensuales, ya sea por necesidad o por gusto. Al tener un negocio, muchas veces se tiene una falta de conocimientos sobre lo que generan las ventas y cuál es la utilidad total del emprendimiento. Tener un flujo de caja al día, separar cuentas, y manejar bien las redes sociales, podría facilitar la administración del negocio. Por esto, hemos diseñado, implementado y validado *Easy - Negocios*, un software que facilita la gestión del emprendimiento y entrega tanto indicadores como herramientas útiles para el crecimiento de este, como administración de pedidos e información del flujo de caja, entre otros. Se evaluó mediante encuestas a emprendedores para revisar si con esta herramienta tienen claros sus productos estrellas, ordenan distintas inversiones, y pueden revisar cuánto se tiene de ganancias al final de mes, junto con reducir el tiempo utilizado en la gestión obteniendo mejores resultados. De las respuestas de la validación se obtuvo calificaciones mayoritariamente positivas y opiniones acerca del proyecto que se acercan bastante a los que se tenía contemplado para el trabajo futuro.

Siempre hay gente que dice que no te la vas a poder.

Agradecimientos

A lo largo del semestre y de mi vida, distintos personajes han sido claves tanto para el desarrollo del trabajo como para la vida personal. Quería agradecer a mi familia, que me apoyaron tanto en la salud como en lo económico durante la carrera. A la Ina y a Kraken, mis mascotas que me miran mientras trabajo. A mi amigo Enzo por ayudarme en la corrección y la redacción, al profesor Jeremy por haber estado durante el semestre para las dudas y conversaciones que surgían, y por último a la Josefina por darme regalos y aguantarme las semanas que colapsaba. Hay hartas personas que no están aquí escritas, pero que merecen agradecimientos.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

Cada día la gente se ve motivada a comenzar proyectos propios con un objetivo a cumplir. Este puede ir desde ayudar a las personas, reducir la contaminación, o tan solo contar con un ingreso mensual adicional. Esto se ve reflejado en la aparición de pymes de distinta índole dentro de las redes sociales, un fenómeno que aumentó durante la pandemia [14], aprovechando distintos problemas o necesidades de la comunidad para surgir con una solución.

Independiente de los objetivos que se tenga, estos proyectos pueden fallar, entre otras cosas, por no contar con los conocimientos necesarios que requiere la administración de un negocio. Lo anterior podría convertirse en una barrera insalvable para los emprendedores y emprendedoras, o bien, puede generar un negocio menos rentable, transformando el emprendimiento en una carga que afecte la eficiencia. Hay varios factores que afectan el desarrollo de un negocio a medida que va creciendo, y que no necesariamente se sabían desde un principio, como manejar ciertos indicadores, fechas, cantidades, entre otros. Dos de las razones por las que más fallan los emprendimientos a su segundo año de vida es un mal flujo de caja y un mal manejo de precios [11].

El flujo de caja de un proyecto son los ingresos y egresos que este tiene en el tiempo. Mantenerlo al día y saber utilizarlo trae al administrador una serie de beneficios tanto tributarios como ejecutivos. Esto es muy importante, ya que ayudaría tanto a la retención de impuestos, pago de deudas, gestión de inversiones y hasta se podría mostrar como prueba a un tercero al momento de pedir un préstamo. Para esto solo es necesario mantenerlo ordenado, con la información al día y saber interpretarlo.

Otro punto importante a considerar, especialmente con el contexto actual, es el uso de las redes sociales y su papel en la difusión de información en forma masiva. Estas son, sin duda, un instrumento muy poderoso para el usuario, ya que darse a conocer y llegar a un público objetivo es clave para un crecimiento rápido y enfocado. Pero muchas veces se requiere de mucho tiempo para la gestión de estas, siendo que la información difundida es la misma en la mayoría de los casos.

Por último, el tema del stock y los pedidos es un factor importante, pues el emprendedor necesita saber en todo momento con qué materia prima cuenta para que no sobre ni falte ningún producto al momento de estar en el mercado. Esto requiere una capacidad de

orden muy grande y muchas veces no está todo enlazado, por lo que se termina perdiendo información y oportunidades de venta.

Teniendo todo esto en cuenta, se cree que una aplicación bien hecha para abordar todos estos puntos mejoraría la cantidad de información que se tiene del negocio, disminuiría el tiempo de administración utilizado y serviría como un indicador en tiempo real acerca de las utilidades y gastos, entregándole al usuario lo necesario para tomar decisiones con respecto a lo económico. Como cada caso es diferente, los resultados de esta aplicación se evaluarían mirando la tasa de fracaso en general de los negocios o viendo cuanto tiempo se ha utilizado en su gestión desde que se empezó empleó la aplicación.

En este trabajo se desarrolló una herramienta que ayuda a los emprendedores a mantener toda la información del negocio ordenada. Tal aplicación le entrega al usuario lo necesario para saber dónde está situado, cuáles son los productos que más venden y cuáles los que más entregan ingresos (muchas veces confundido), cuáles son los indicadores económicos del negocio y la rentabilidad de este, quiénes le pidieron qué productos y en qué fechas, contar con un stock digital para que al momento de comprar materia, se tenga más información y también un punto donde las redes estén unificadas y se disminuya el tiempo de gestión y de difusión entre una y otra.

Se validó que esto ayudó a al menos 16 emprendimientos y personas, desde ventas de frutas al por mayor hasta ventas de objetos más grandes y caros, algunos negocios de compra y venta y otros de producción, tratando de abarcar la mayor variedad. De las respuestas se obtuvo feedback y opiniones para poder tener clara la dirección a seguir. Los potenciales usuarios que se contactaron para validar la aplicación fueron obtenidos mediante contactos, difusión en redes sociales, foros de la universidad, de la facultad, y por último, grupos de emprendedores y ferias.

De la validación y análisis se considera que el producto es exitoso, ya que unifica toda la información del negocio y disminuye el tiempo utilizado en administración y gestión de este, entregando los mismos o mejores resultados y es suficiente como herramienta para gestionar la totalidad del emprendimiento.

Todo lo que se desarrolló en este proyecto tiene la licencia *Open Source* 'GNU GPLv3', permitiendo a la comunidad aportar en este y utilizarlo con total libertad, distribuirlo, y hasta modificarlo. Sin embargo, esta licencia tiene condiciones como que se mantenga la licencia en las modificaciones y el código abierto este libre. Esto se hizo con la intención de que la herramienta siga abierta y gratis, pues no se busca que esta corresponda a un gasto a considerar dentro de la misma.

1.1. Objetivos Iniciales

Al comenzar la planificación del proyecto se establecieron tanto objetivos generales como específicos con tal de poder, al final del desarrollo, validar y estudiar el funcionamiento de la solución implementada para cubrir los problemas planteados.

1.1.1. Objetivo General

Desarrollar una herramienta que ayude a emprendedores y emprendedoras a nivel nacional de manera simple, cumpliendo como función ser el núcleo del emprendimiento, entregando información fundamental del negocio. Evaluar con una encuesta de uso a emprendedores y emprendedoras que se ofrecieron para probar el producto.

1.1.2. Objetivos Específicos

Dentro del proyecto se cuenta con múltiples objetivos específicos, todos considerados iguales de importantes y relacionados con el manejo y administración de un negocio. Los objetivos específicos son:

1. Diseñar una interfaz amigable e inclusiva, utilizable por todo público independiente de sus conocimientos y condiciones, con vistas convenientes y específicas para cada indicador a mostrar.
2. Implementar todo el procesamiento de la información que agregue el usuario, con tal de lograr entregar a éste los datos e indicadores que facilitarían la gestión del negocio, ofreciendo ayuda para todas las aristas señaladas en la sección 2.1.
3. Validar las distintas funcionalidades y sus formas con el usuario final, comprobando que la solución propuesta cubre de buena manera los problemas observados en un principio.

1.2. Estructura del informe

En este reporte se analizarán las opciones y alternativas con las que se cuenta actualmente para cumplir con los objetivos de la herramienta a desarrollar, mostrando punto a punto las ventajas y desventajas que estas tienen y por qué un software nuevo sería útil (Capítulo 2). Se especificarán los detalles de la solución propuesta con sus distintos *features* y se detallará el proceso de desarrollo de esta (Capítulo 3). Se continuará por explicar la implementación y su proceso, señalando distintas decisiones que se tomaron para llegar al producto final (Capítulo 4), seguido de lo que fue la validación a los usuarios finales con el análisis correspondiente (Capítulo 5). Por último se muestran las conclusiones del trabajo realizado, con un pequeño compilado de los aprendizajes y una mirada a futuro (Capítulo 6)

Capítulo 2

Estado del Arte

Desde que existe el comercio se ha presentado el problema de su gestión, agregando cada vez nuevas dificultades y al mismo tiempo nuevas herramientas, como son, por ejemplo, la necesidad de difusión masiva y las redes sociales. En todo lo que es administración de negocio, aquí describiremos sus bases y ramas (Sección 2.1), las soluciones básicas que se utilizan actualmente para estas aristas (Sección 2.2) y los productos un poco más avanzados que existen (Sección 2.3). Se concluirá con un arreglo recapitulando los *features* de cada solución para demostrar la necesidad de *Easy - Negocios* (Sección 2.4).

2.1. Aristas del manejo de un negocio

Existen ciertos pilares dentro del manejo de cualquier emprendimiento, cierta información que es necesario manejar de forma ordenada para poder saber y tener claro cómo y hacia dónde va el negocio. Si bien cada negocio podría tener algunas diferencias, estos que se presentan a continuación son lo más básico y común que tiene la mayoría de los emprendimientos.

- **Flujo de caja:** Consiste en mantener al día los ingresos y egresos del negocio. La importancia de mantener este flujo al día es fundamental en cualquier tipo de negocio, sea del área que sea, pues le dice al emprendedor como está su rentabilidad con respecto al tiempo para ver si el proyecto es o no rentable. Esta información es clave al momento de invertir grandes cantidades o tener que tomar la decisión de mantener o abandonar el negocio.
- **Manejo de Stock:** En todo momento el administrador debe saber con qué productos o materia prima se cuenta, ya que esa es la única forma realmente de saber cuando comprar más o cuando dejar de aceptar pedidos. Para lograr esto es clave manejar bien el punto anterior, ya que se puede sacar información de cuantos productos de cuáles se venden más frecuente y poder prevenir el déficit de este.
- **Administración de pedidos:** Si bien es muy importante el punto de manejo de stock para una correcta gestión de los pedidos, no es todo, pues administrar de buena forma

los pedidos incluye además de tener el producto, entregarlo a la fecha, las cantidades correctas y mantener la información recibida en un principio en caso de cualquier malentendido.

- **Diferenciación de clientes:** Un punto que muchas veces queda separado del negocio, pero que le entrega un valor agregado enorme, es conocer a cada cliente, darle un trato único a cada persona con la que se trata. Esto hace que el cliente se sienta más cómodo al momento de comprar y prefiera ese lugar frente a otros en caso de tener competencia.
- **Comunicación y difusión en redes sociales:** Hoy en día las redes sociales son punto muy importante a considerar dentro de cualquier negocio, ya sea utilizarlas como canal de venta, como forma de comunicarse o como difusión masiva a través, por ejemplo, de anuncios pagados. Es la forma que hay hoy en día de darse a conocer.
- **Manejo de Pagos:** No hay necesidad de explicar la importancia que tienen los pagos dentro de un emprendimiento, pero si podría ser necesario mencionar el problema que se tiene con este punto. Cada pago puede venir de una fuente distinta (efectivo, transferencia desde distintos bancos, tarjeta, etc...) y el tema de administrar esto se vuelve engorroso, pues no se sabe bien dónde revisar para verificar que todas las deudas estén pagadas. Lo mas lógico es revisar directamente la cuenta bancaria o el correo, pero muchas veces la información del pago no es suficiente para saber detalles de la transacción. Ayudaría mucho a un buen manejo de pagos tener una central que revisar para poder ir marcando cuando se pagó una venta y así mantenerlo al día.

2.2. Soluciones existentes básicas para el manejo de negocios

Existen ciertas soluciones y herramientas simples que se utilizan actualmente para la administración de los negocios. Cada una de estas soluciones actuales cuenta con ventajas y desventajas considerando cada ámbito de la administración de un negocio. Estas opciones que se describen a continuación se seleccionaron a través de la observación y mediante una consulta a un grupo seleccionado de emprendimientos.

- **Lápiz y Papel:** Por años fue la más usada o casi la única forma de registrar cada evento que ocurría dentro de un negocio, pero a medida que los números se vuelven más grandes, los cálculos más complicados y con más variables, y aparecen herramientas que permiten almacenar la información de manera más segura sin tener riesgos como que se pierdan o que se rompan, se ha avanzado por reemplazarla. Con esta herramienta se puede manejar ventas, registrar compras, manejar stock, administrar pedidos y todo lo más básico, excepto manejar las redes sociales, pero es muy limitada y lenta al momento de calcular o querer obtener indicadores a fin de mes.
- **Excel:** La herramienta que grandes empresas usan por excelencia. Cuenta con una cantidad de funciones suficientes como para hacer todo lo mencionado menos el manejo de redes (y es poco común diferenciar clientes aquí). El gran problema que tiene esta herramienta es que es demasiado grande y poderosa, con una interfaz poco amigable

para aquel que no domina la tecnología. Para obtener indicadores o relacionar información hay que utilizar funciones y referencias, tener hartas páginas para ordenar la información. La curva de aprendizaje de Excel podría no ser la mejor para todo público, ya que al ser una herramienta pensada para abarcar mucho, resulta compleja para muchos usuarios.

- **Redes Sociales:** Cada red social sirve para administrarse a sí misma, pero cada una cuenta con distintas interfaces y formas de funcionar. Para poder manejar *WhatsApp*, *Instagram*, *Telegram* y otras como *Signal* o *TikTok* se debe conocer sus distintas herramientas, como sistemas de anuncios, cobros, formas de difundir, etc. Por lo que para el usuario termina siendo confuso y complicado.

2.3. Soluciones existentes avanzadas para el manejo de negocios

Otra gama de soluciones a considerar son las más avanzadas y exclusivas que también se utilizan para la administración de un negocio. Estas herramientas son más específicas para ciertos puntos en particular, pero muchas veces son de costo o limitadas según precio y tipo de usuario. Vale la pena mencionar que estas, incluyendo las básicas, no son exclusivas y pueden llegar a ser hasta complementarias.

- **Simple Stock Manager:** Desarrollada por *Learn24bd*, es la herramienta de manejo de stock más descargada en *Google Play Store* [8]. Esta herramienta tiene como función única el manejo de stock, pero no se mete en lógica de pedidos ni de indicadores del negocio. Lo otro que carece es que al momento de vender o comprar, el stock no se actualiza solo. Esta herramienta, además, está pensada para empresas con sistema de bodega y dimensiones más grandes.
- **Facebook Business Suite:** Permite unificar las redes de *Meta*, como *Facebook* o *Instagram*, también permite ver un resumen de la empresa y su actividad, crear anuncios y otras cosas [2]. El problema de esta aplicación es que se dejan de lado las otras redes y la versión gratis es muy limitada.
- **WhatsApp Business:** Sirve únicamente para poder ordenar de mejor manera las conversaciones de *WhatsApp*, dando la opción de responder con mensajes automáticos y facilitando la comunicación por ese medio. Carece de todos los otros puntos.
- **Mercado Libre / Amazon:** En estas plataformas se puede administrar un negocio y cuentan con una gran cantidad de herramientas como comunicación con el cliente, manejo de stock, manejo de pedidos, entre otras. También se cuenta con una central de manejo de pagos para la plataforma, problema que no se cubre con las otras soluciones. El problema de estas soluciones es que son plataformas de venta, y la administración a través de estas solo sirven para aquellas ventas producidas en la plataforma, lo que limita al vendedor a un canal especifico. Si bien estas aplicaciones son buenísimas, no atacan el mismo problema que nuestra herramienta *Easy - Negocios*.

- **Importante:** No confundir el proyecto presentado con una potenciadora de emprendimientos, como sería *Balloon Latam* [7], o con una de vinculación de personas como es *Kellun* [6], que sirve, en parte, para relacionar al cliente con el emprendedor en caso de negocios emergentes, pero que no ataca ninguna de las aristas mencionadas.

2.4. Arreglo Recapitulativo

De todas las tecnologías utilizadas en el rubro, no se conoce ninguna que cuente, al mismo tiempo, con un manejo histórico de las cuentas (ingresos y egresos), mostrando indicadores básicos del negocio de manera automática, manejo del stock, de las ventas por producto, anotación de pedidos y por último un manejo centralizado de las redes y los depósitos bancarios desde distintas fuentes.

En la tabla 2.1 se resumirán distintos factores importantes a considerar y se discriminará cada herramienta estudiada, comparando con *Easy - Negocios*.

Tabla 2.1: Soluciones Actuales Para el Manejo de un Negocio

	Flujo de Caja	**Stock**	**Pedidos**	**Clientes**	**RR.SS**	**Precio**	**Central de Pagos**	**Registro Histórico**
Lápiz y Papel		✓	✓			Gratis		✓
Excel	✓	✓	*Inc.			Con Licencia		*Posible
RR.SS			✓	✓	✓	Gratis		
F.B.S					✓	Gratis Limitado		
S.S.M		✓				Gratis Limitado		
Mer. L. / Amazon		✓	✓	✓		Comision x Venta	Interna	✓
Easy - Neg.	✓	✓	✓	✓	*	**Gratis**		✓

- Inc. es de Incompleto. Si bien uno puede anotar en las celdas cada pedido, carece de funciones especializadas
- Es 'Posible' porque depende de la forma que uno use Excel si cuenta o no con el registro.
- Central de Pagos 'Interna' es porque solo se cuenta con un manejo de pagos a compras a través de la misma herramienta.
- El punto de RR.SS en *Easy - Negocios* es porque se diseñó y vio la factibilidad, mas no se implementó.

Al lograr el desarrollo de una herramienta que facilita y unifica alguno de estos puntos, las

personas emprendedoras lograrían disminuir su tiempo en gestión, aumentando su eficiencia u otorgando más espacio para el ocio. La idea es que esta herramienta sirva tanto para personas capacitadas como no capacitadas, ya que cuenta con una curva de aprendizaje más amigable que la de otras herramientas muy poderosas, mediante una interfaz fácil de interactuar y hasta intuitiva, ofreciendo funcionalidades con pocos 'clics' de cada pantalla.

Capítulo 3

Diseño

Para que la aplicación sea fácil de usar y fácil de extender, es muy importante cumplir con ciertos puntos, como un esquema de datos simple y extensible, un flujo de información claro y una interfaz gráfica simple e intuitiva. En este capítulo se describirá, a grandes rasgos, el modelo de la Base de Datos (Sección 3.1), explicando su estructura, luego se continuará con lo que es el Flujo de la Información (Sección 3.2), haciendo énfasis en las relaciones interna, para continuar con la interfaz gráfica y de las tecnologías que se utilizaron (Sección 3.3), y por último se hablará de las funcionalidades y *features* que se pensaron en un principio para la aplicación (Sección 3.4). Sobre este texto se basará luego la Implementación (capítulo 4).

3.1. Modelo Base de Datos

Se tenía que pensar desde un inicio en un modelo de base de datos relacional que sirva para obtener la información de una forma en que cada tabla se pueda ordenar de manera fácil e intuitiva en pantalla. La idea es que toda la información esté vinculada a lo que es un proyecto para que, con las facilidades de SQL, se puedan poner referencias y llaves foráneas. El modelo de Base de Datos inicial y el definitivo tienen ciertos cambios que se fueron evaluando a conveniencia mientras se trabajaba.

La decisión de optar por una base de datos relacional se explica por diversas razones. Primero que nada, se tenía claro que la información que se iba a guardar en esta base de datos tenía una estructura clara, se podía organizar por tablas y columnas de manera muy conveniente. La segunda razón era que las conexiones que se iban a tener interiormente iban a ser de la mayor importancia para la seguridad de los datos, y esto en las relacionales se puede hacer de forma mas cómoda [1]. La tercera razón que vale la pena mencionar es que se tiene más experiencia personalmente trabajando con bases de datos relacionales, por lo que la adaptación a esta iba a ser más rápida.

Al ser relacional se optó por *MySQL* debido a la experiencia personal que se tiene al momento de realizar consultas y a la comodidad al trabajar con *XAMPP*, contando con un administrador y visualizador de bases de datos como *phpMyAdmin* al trabajar en local.

El modelo de datos diseñado cuenta con las siguientes tablas:

- **Projects** (***id***, project_name, project_description, update_date, creation_date): Es la tabla principal de cada proyecto, desde esta se puede obtener toda la información necesaria. Sirve para unificar y relacionar las distintas ramas de cada proyecto.
- **Products** (***id***, *project_id*, *category_id*, product_name, product_description, net_price, gross_price, stock, measure_unit, stock_update_date, creation_date, update_date): Sirve para contener cada producto y sus campos, la mayoría son opcionales y las fechas son automáticas.
- **Clients** (***id***, *project_id*, client_name, client_decription, creation_date, update_date): Para almacenar los clientes, cuenta con la mínima información necesaria para que sea más simple agregarlos.
- **Investments** (***id***, *project_id*, total_net_price, total_gross_price, ticket, investment_description, investment_date, owned_product, cuantity, creation_date, update_date): Tiene información suficiente de las inversiones que hace el usuario, permite elegir si es boleta o factura, o si el objeto es de compra y venta. No todos los campos son obligatorios para simplificar la interacción con el usuario.
- **Sales** (***id***, *project_id*, total_net_price, total_gross_price, ticket, sale_description, sale_date, creation_date, update_date): Corresponde a las ventas que hace cada proyecto. La idea es que las ventas no tienen un cliente asignado, pues está pensado para ventas en local.
- **Orders** (***id***, *project_id*, *client_id*, delivery_date, order_description, address, creation_date, update_date): Se refiere a los pedidos que le llegan al negocio. Estos tienen un cliente, una dirección y una fecha de entrega.
- **Categories** (***id***, *project_id*, category_name, category_description): Para poder catalogar los productos en distintas categorías y organizar de mejor manera la interfaz, dando la opción de filtrar.
- **Order_Product_Relation** (***id***, *order_id*, *product_id*, cuantity): Sirve como tabla relacional entre las órdenes y los productos, de esta forma una misma orden puede tener varios pedidos.
- **Sales_Product_Relation** (***id***, *sales_id*, *product_id*, cuantity): Sirve como tabla relacional entre las ventas y los productos, de esta forma una misma venta puede tener varios productos sin necesidad de agregar más de un objeto en la aplicación.

Para facilitar la visualización del modelo se puede observar la figura del anexo 6.1, este tiene bastantes conexiones y referencias internas que facilitan el trabajo de organizar la información. Se pensó en una tabla central, que tenga los proyectos con su *id* y campos de información básica, y desde esta tabla agrandar el modelo para los distintos campos que se pensaron para un proyecto. Todas las tablas tienen alguna forma 'simple' de obtener su proyecto referenciado, y cada proyecto es totalmente independiente uno del otro.

El resto de tablas mostradas en la figura del anexo 6.1 o en la definición corresponden cada una a uno de los problemas que se quiere solucionar con la herramienta, a excepción

de la tabla *Categories* y las de relación, que serían la *Sales_ y Order_ Product_Relation*, estas tres sirven para ordenar la información y facilitar el funcionamiento de ciertas *features* en la aplicación. Cada tabla tiene su *id* único, lo que siempre es recomendable al momento de trabajar con bases de datos relacionales, aparte de alguna referencia que la vincule a su proyecto.

Una de las grandes ventajas de mantener llaves foráneas entre las tablas es la consistencia y orden de la información. En este sentido, se sacó provecho de ciertas declaraciones con las que cuenta *MySQL*, como es el ejemplo de *'ON DELETE CASCADE'*, que ayuda a mantener consistencia al momento de eliminar datos, ya que automáticamente elimina todos sus 'datos hijos', eliminando la información que ya no es válida en la base de datos. Al momento de la implementación, esto trajo un problema del que se hablará, junto a su solución, más adelante 4.1.2.

3.2. Flujo de la información

El camino que sigue la información dentro de una aplicación es muy importante, tanto para la manipulación de esta (cuando es necesaria) como para la seguridad de la herramienta desarrollada. Este camino tenía que ser simple desde el input del usuario, pensando en el público objetivo, como en la llegada a la base de datos, pensando en el desarrollo del proyecto a futuro.

Todas las relaciones internas que tienen los datos se diseñaron para que sean lo más independientes posibles dentro de un proyecto. Esto se pensó para evitar al máximo las interacciones involuntarias dentro de la base de datos, como se puede ver también en la descripción del modelo en la Sección 3.1, casi todas las tablas están vinculadas a *projects* pero no entre ellas, y cuando se vinculan, es a través de tablas intermediarias para proteger esa relación.

Los inputs se diseñaron para tener una lógica que siga la idea de la independencia o 'encapsulación' de la información. La idea es que al momento que el usuario ingrese datos en los inputs, esta se valide primero en frontend para avisar en el momento en caso de haber un error en esta, para luego enviarla mediante los endpoints al backend y tenga una segunda validación. Por último, se ejecuta la *query* de SQL, cuidando las llaves foráneas y referencias que estos datos podrían tener y manteniendo siempre el orden de las tablas, ya que este será muy importante al momento de obtener la información para el flujo de caja.

En cuanto a los inputs y su validación, siempre se pensó en una aplicación muy cómoda para el usuario, que no presente mayores problemas, por lo que los campos que no sean estrictamente necesarios al momento de agregar un dato, se pueden dejar como *nulls* en la base de datos. Esto no afecta al correcto funcionamiento de la herramienta, pues siempre se tiene en consideración al momento de diseñar el flujo que campos podrían estar vacíos.

3.3. Interfaz y tecnologías

Lo que fue el diseño del frontend de la herramienta es de las partes más importantes de esta fase en general, ya que el real potencial de esta aplicación está en la simpleza de utilizarla. Se va a revisar la interfaz y su lógica por detrás. También se analizará la decisión de tecnología a tomar, comparando las opciones que se consideraron fuertemente tanto para backend como para frontend, las tecnologías más especificas como librerías y emuladores se verán en el Capítulo 4, correspondiente a la implementación.

Para hacer el prototipo de la interfaz se utilizó la herramienta *Marvel Wireframing* y se optimizó la cantidad de clics necesarios para las acciones fundamentales. El prototipo realizado se puede observar en el anexo 6.3.3, en este conjunto de figuras se puede ver como cada una de las tablas en el modelo de datos tiene su propio recuadro de manera directa y con un solo clic. Se consideró que estos eran los pilares de cada negocio y era muy importante que fueran de fácil acceso.

En cuanto a las tecnologías que se utilizaron, como lenguaje se pensó desde un principio en utilizar *javascript*, contra *Dart*, por su popularidad para hacer tanto herramientas web como móviles, y considerando que se contaba con algo de experiencia previa. Como herramientas extras se utilizaron *Node.js* con *npm* y *React Native*. El primero es un entorno de ejecución y se eligió por, dentro de varias razones, ser de código abierto y con una curva de aprendizaje bastante amigable para el desarrollador, es bastante escalable y cumple con el requisito de ejecutar código *javascript*. En cuanto a *React Native*, la decisión de optar por este framework, en lugar de su competidor Flutter, fue de mayor complejidad. Finalmente, el primero presentó una serie de ventajas con relación al desarrollo multiplataforma [3] [4], que lo pusieron por delante del segundo. Asimismo, fue relevante en la determinación el hecho de ser desarrollado por *Facebook* con filosofía de código abierto.

Una de las grandes características de las tecnologías que se utilizaron es que son multiplataforma. Al desarrollar la herramienta, si se tiene en cuenta que funciones son exclusivas para algún sistema operativo y cuáles son generales, se facilita mucho el trabajo de generar una solución que sirva a más gente. Pensando en una aplicación móvil, con abarcar *Android* y *iOS* ya se está incluyendo a más del 90 % de Chile, como se ve en la imagen 3.1.

Figura 3.1: Distribución Teléfonos Móviles obtenidos de La Tercera [13]

3.4. Funcionalidades de la Herramienta

Para concluir con lo que fue la fase de diseño de la herramienta, se nombrara cada *feature* que se pensó para esta con su importancia. Cada uno de estos se consideró como importante al momento de gestionar un emprendimiento o negocio y tienen un plan de implementación con un acceso simple mediante la interfaz.

- **Productos:** Tener en cuenta cuáles son los productos que maneja un negocio es fundamental, pues con base en esto se pueden ordenar las ventas, pedidos, ingresos y se pueden obtener estadísticas acerca de donde proviene la mayor cantidad de ingresos y cuáles marginan más que otros. Al mismo tiempo, tener todos los productos (con posibilidad de ordenar por categoría) permite la creación de un catálogo de manera rápida para entregar a posibles clientes.
- **Clientes:** La personalización de estos ayuda mucho para hacer sentir especial al cliente. Poder guardar los detalles de que es lo que ha comprado para poder recomendarle, describirlo a gusto para poder identificarlo si se vuelve un cliente frecuente, y distintas estrategias que se pueden utilizar para mejorar la experiencia del cliente y fidelizarlo al negocio.
- **Compras:** Guardar las inversiones que se hacen dentro de un negocio ayuda al administrador a decidir cuanto gastar en este, mantener los gastos e ingresos del negocio independientes de los personales para ver la real rentabilidad es un factor que mucha

gente olvida al momento de emprender. Tener acceso rápido a las compras, ya sean de maquinaria, materia prima, objetos para el negocio, entre otros, agilizaría las decisiones y ayudaría a tener una real idea del balance del negocio. También hay que considerar que este punto es obligatorio para poder generar un flujo de caja correcto. Dentro de este ítem se tiene la opción de seleccionar si fue boleta o factura en caso de que al negocio le interese seguir sus impuestos.

- **Ventas:** El historial de ventas ayuda en varios ámbitos, desde la generación del flujo de caja en el cual este es un punto fundamental hasta el detalle de cuáles son los productos preferidos por los clientes. Esto último puede ayudar a decidir, por ejemplo, que cosas promocionar más que otras. Las ventas son el pilar fundamental de cada negocio, ya que es la forma que tienen para generar ganancias y sobrevivir en el tiempo. Al igual que en el ítem de compras, aquí se tiene la opción de seleccionar si fue boleta o factura en caso de que al negocio le interese seguir sus impuestos.

- **Pedidos:** No confundir con el punto anterior, una venta se hace en el momento y un pedido se hace con tiempo. Tener los pedidos anotados y con capacidad de ordenar por criterios como fecha para no olvidar ninguno o por cantidad de productos para calcular bien el stock es algo que al usuario le ayudaría a mantener una buena reputación de no fallar en sus entregas. Para un negocio, fallar una entrega por cualquier razón se puede significar perder un cliente, ganar mala reputación, entre otras, porque siempre los malos comentarios se propagan mas rápido que los buenos. También tener las direcciones a mano ayuda al usuario a optimizar su recorrido de entrega si es ese el caso. Por último, y al igual que los últimos dos ítems, se puede seleccionar si fue boleta o factura para el seguimiento y la devolución de impuestos.

- **Manejo de Stock:** Siempre tener claro lo que se tiene para poder tener claro que se puede vender y que no. Esta información es muy útil al momento de publicitarse, interactuar con un cliente, ofrecer productos extra, etc. Un buen manejo de stock permite adelantar el trabajo en el caso de negocios que requieren preparación, o comprar materia por adelantado en fechas importantes. Este punto podría ser poco utilizado por negocios que ofrecen un servicio (por ejemplo, clases particulares) pero puede ser clave para aquellos que utilizan grandes cantidades de materiales, como una pastelería.

- **Flujo de Caja:** Toda la información de ingresos y egresos se encuentra en este lugar. Poder mirar el balance, el producto estrella, los márgenes de ganancias, y todo con rangos de fecha para poder diferenciar según meses, semanas importantes, festivos, entre otros, puede aportar mucho a un negocio en cuanto a decisiones de compras, estrategias de publicidad, adelanto de fechas *peak*, y hasta analizar por donde vale la pena seguir ampliando el negocio en caso de querer hacerlo.

- **Central de Pagos:** Cuando el volumen de ventas es alto, se vuelve difícil tener que cobrarle a cada cliente en caso de ser pedidos, ya que muchas veces existen distintos medios de pago, tipos de notificaciones según banco, etc. Este punto surgió de una necesidad con un potencial usuario de *Easy - Negocios*, a la que le complicaba el tema de ordenar los pagos. Tener una central de pagos adonde lleguen notificaciones y entregue información como nombre del cliente, monto que pago, fecha de pago, podrían ayudar mucho a solucionar este problema.

- **Manejo de Redes:** Las redes sociales son muy importantes hoy en día tanto para lo que es publicidad y difusión como para comunicación con clientes, el problema es que son cada vez más. *WhatsApp, Instagram, Telegram* son ejemplos de las más comunes y cada una tiene distinta interfaz, contactos y funcionalidades. Poder unificar todas en un punto para simplificar tareas masivas como mandar una imagen o un texto (o en caso de *Instagram*, subir una historia) ahorraría mucho tiempo de uso en estas y maximizaría el público alcanzado. Al momento de incluir este punto se estudió acerca de las *APIs* que cada una brinda para ver la posibilidad y esto también afecto en la decisión de utilizar *React Native* para el desarrollo.

Capítulo 4

Implementación

La implementación y el trabajo de un proyecto como *Easy - Negocios* se puede separar en lo que es el *Backend*, descrito en la Sección 4.1, con sus ramas para poder mostrar, y el *Frontend*, redactado en la Sección 4.2, con su estilo y justificación. Por último, y teniendo toda esta información, se hablará de las funcionalidades y features con las que cuenta la aplicación en la Sección 4.3.

4.1. Backend

Esta sección del escrito explicará la implementación de todo lo relacionado a la comunicación con la base de datos y sus endpoints. Esto va desde las tecnologías que se utilizaron (Subsección 4.1.1), sus ventajas y formas de uso, luego la conexión a la Base de Datos y su forma final (Subsección 4.1.2), explicando el uso de variables de entorno y las diferencias entre lo que es una conexión local o una externa. Se explicará el uso de las librerías utilizadas que se consideran más importantes (Subsección 4.1.3), justificando su necesidad dentro del proyecto. Luego se revisarán ciertos detalles de seguridad (Subsección 4.1.4), como son los endpoints, lo que entregan y sus parámetros. Por último se explicará a detalles el paso a producción de la aplicación (Subsección 4.1.5).

4.1.1. Tecnologías

El entorno que se utilizó fue *Node.js* debido a las ventajas ofrecidas. Desde el administrador de paquetes *npm* hasta la amplia selección de librerías que trae. Este entorno ha tomado mucha importancia en el mundo del desarrollo últimamente, por lo que aparte de haber sido una gran opción a elegir, fue muy bueno aprenderlo y practicarlo.

Las razones más fuertes detrás del uso de *Node.js* son las facilidades de escalar el proyecto, punto importante al dejarlo con licencia GNU GPLv3, como es el caso, ya que el apoyo de la comunidad está muy activo a día de hoy, y este es un entorno que se está utilizando bastante a nivel mundial y, por último, funciona muy bien al momento de hacer aplicaciones para

hartas plataformas sin tener que hacer cambios significativos en el código, lo que ayuda a hacer soluciones generales y con mayor alcance. Las mayores desventajas que se estudiaron, pero que no afectan de gran manera, es la ineficiencia de *Node.js* al ejecutar tareas pesadas, o la dificultad de mantener el código con su modelo de programación asíncrona, el cual se utilizó bastante en la implementación, pero se aprendió a obtener su potencial [12].

Node.js trae consigo un administrador de paquetes llamado *npm* [5], cuya existencia ya significa una ventaja para la elección del entorno de ejecución. Este administrador, *npm*, dentro de su gran cantidad de ventajas y características que tiene, hay dos que resaltan. El manejo de múltiples versiones de distintas librerías para que sean compatibles de manera automática (o avisando en caso de no poder y tener que hacerlo manual) y la existencia del archivo *package-lock.json*, el cual muestra exactamente las dependencias del proyecto y sus versiones. Este archivo ayuda a poder replicar de manera exacta el estado de un proyecto en el tiempo. Muchas veces durante la implementación se tuvo que volver atrás en *Git* y gracias a este archivo uno podía volver al punto exacto en el que se estaba.

El proyecto *Easy - Negocios* cuenta con una estructura similar a la clásica *Model - View - Controller*. Una carpeta *src*, con el código de la aplicación y que se separa en *src/routes*, *src/controllers* y archivos de configuración, ya sea de la aplicación, de librerías o de la base de datos en *src*. En *src/routes* se encuentra definido cada *endpoint* de la aplicación, con su respectiva documentación y función asociada. Este archivo se vincula a aquellos guardados en *src/controllers*, los cuales tienen cada función asociada previamente y su consulta *SQL* definida.

La utilización de estas tecnologías fue una buena decisión al momento de implementar, ya que al ser apoyadas por una gran comunidad y contar con buena documentación y librerías, se pudo lograr cada tarea como se esperaba y en el proceso se pudo aprender más acerca de estas, que hoy en día agregan un gran valor a la persona.

4.1.2. Modelo y Conexión Base de Datos

Al modelo de base de datos obtenido en la fase de diseño en el capítulo 3.1 se le hicieron algunos cambios funcionales durante la implementación. Estos cambios en su mayoría surgieron por pequeñas funcionalidades que se iban a simplificar mucho por algún pequeño cambio. Como se puede ver en la descripción de más adelante, o en la figura del anexo 6.11, ahora todas las tablas tienen el *project_id*, ya que eso facilita la consulta de obtener todo lo que tiene un proyecto sin necesidad de hacerlo a través de la tabla *orders* o *sales*.

Por otro lado, se agregó en casi todas las tablas el atributo 'active'. Este atributo es un booleano que representa si el valor existe o fue eliminado, pero sin necesidad de eliminar efectivamente el dato. El problema por el cual se agregó esta columna, es que, por ejemplo, si el flujo de caja se genera históricamente con un producto que durante un periodo de tiempo valía x, pero que se actualizó y luego valía y, entonces tomaría el valor de y como el único. Pero si al momento de actualizar el producto lo que se hace realmente es desactivarlo y hacer otro exactamente igual, pero con el valor cambiado, entonces sigue en la base de datos la información necesaria para generar un flujo de caja correcto. En resumen, la columna 'active' sirve para el versionamiento de objetos cuando necesitan ser versionados para hacer un flujo

de caja correcto.

Los otros cambios que se pueden ver son pequeños detalles que simplificaban algunas consultas o *features*, pero que no significan un cambio significativo en como funciona la base de datos. El modelo de base de datos quedó de esta forma:

- **Projects** (***id***, project_name, project_description, update_date, creation_date): Es la tabla principal de cada proyecto, no sufrió cambios desde el diseño.
- **Products** (***id***, *project_id*, *category_id*, product_name, product_description, net_price, gross_price, stock, measure_unit, stock_update_date, creation_date, update_date, active): Sirve para contener cada producto y sus campos, la mayoría son opcionales y las fechas son automáticas. Se le agregó la columna mencionada *active*.
- **Clients** (***id***, *project_id*, client_name, client_decription, creation_date, update_date, active): Para almacenar los clientes, cuenta con la mínima información necesaria para que sea más simple agregarlos. Sigue igual a excepción de la columna *active*.
- **Investments** (***id***, *project_id*, total_net_price, total_gross_price, ticket, investment_description, investment_date, owned_product, cuantity, creation_date, update_date, active): Tiene información suficiente de las inversiones que hace el usuario, permite elegir si es boleta o factura, o si el objeto es de compra y venta. No todos los campos son obligatorios para simplificar la interacción con el usuario. Se le agregó la misma columna mencionada anteriormente.
- **Sales** (***id***, *project_id*, total_net_price, total_gross_price, ticket, sale_description, sale_date, creation_date, update_date, active): Corresponde a las ventas que hace cada proyecto. La idea es que las ventas no tienen un cliente asignado, pues está pensado para ventas en local. También cuenta con la columna *active*.
- **Orders** (***id***, *project_id*, *client_id*, delivery_date, order_description, address, creation_date, update_date, active): Se refiere a los pedidos que le llegan al negocio. Estos tienen un cliente, una dirección y una fecha de entrega. De igual manera se le agregó *active*.
- **Categories** (***id***, *project_id*, category_name, category_description, active): Para poder catalogar los productos en distintas categorías y organizar de mejor manera la interfaz, dando la opción de filtrar. Solo se le cambió la columna *active*.
- **Order_Product_Relation** (***id***, *order_id*, *product_id*, *project_id*, cuantity, active): Sirve como tabla relacional entre las órdenes y los productos, de esta forma una misma orden puede tener varios pedidos. Aparte de la columna *active*, se le agregó el *project_id* para simplificar las consultas.
- **Sales_Product_Relation** (***id***, *sales_id*, *product_id*, *project_id*, cuantity, active): Sirve como tabla relacional entre las ventas y los productos, de esta forma una misma venta puede tener varios productos sin necesidad de agregar más de un objeto en la aplicación. Además de la columna que se agregó a casi todas, también se le agregó el *project_id* para simplificar algunas consultas.

Para lo que fue la conexión a la base de datos, tanto la local para el momento de desarrollo como a la remota en la fase de migración y validación, se hicieron con la librería MySQL2,

para esto se tiene un archivo *src/database.js* con la forma que se ve en el *listing* 4.1.2, el cual importa dos cosas fundamentales: Primero el componente de la librería con el cual hace la conexión y luego la variable desde *src/config.js*, un archivo que se ve como el *listing* 4.1.2. Este archivo, a su vez, utiliza la librería *dotenv* para utilizar las variables como variables de entorno y no escribirlas directamente en el código. Esto último por temas de seguridad al momento de tener el directorio en *GIT* o la nube.

Listing 4.1: src/database.js

```
import mysql from 'mysql2/promise'
import {config} from './config'

export const connect = async() => {
    return await mysql.createConnection(config);
}
```

Listing 4.2: src/config.js

```
import {config as dotenv} from "dotenv"
dotenv();

export const config = {
    host: process.env.DB_HOST,
    user: process.env.DB_USER,
    password: process.env.DB_PASSWORD,
    database: process.env.DB_DATABASE,
    multipleStatements: true
}
```

Con estas librerías y archivos se puede establecer una conexión a una base de datos de manera segura y asegurando que si se tiene que cambiar la dirección de esta se puede hacer de manera directa.

4.1.3. Librerías de terceros

Una de las grandes ventajas de *Node.js*, como ya se comentó, es su administrador de paquetes *npm*, permitiéndonos administrar, actualizar y agregar librerías de una forma fácil y directa con sus versiones. Se van a mencionar, explicar y justificar las librerías más importantes que se utilizaron para el Backend. La librería para la conexión a la base de datos se explicó en su respectiva sección (Sección 4.1.2).

- **Express**: Este módulo nos permite crear los *urls* o rutas del servidor junto con distintos detalles adicionales que hacen que esto sea más cómodo al momento de interactuar con la base de datos, como analizar las *requests* con *JSON* y ponerlas en un formato cómodo.
- **Morgan**: Nos permite ver por consola las peticiones que van llegando con su código de éxito o fracaso. Esto es de mucha utilidad al momento de desarrollar y testear, ya

que en caso de haber errores permite descartar que sean de conexión o de sintaxis si es que la petición se hace correctamente (código 200). El código que entrega también nos puede dar información acerca si se está consultando a la base de datos directamente o a algún caché.

- **Cors**: Nos sirve para especificar que orígenes pueden conectarse al servidor. Esta librería nos evita uno de los clásicos problemas de conexión con *http* por certificados, lo que es muy útil al momento de desarrollar y probar en local o en ambientes controlados.
- **Swagger (jsdoc y ui-express)**: Estas librerías son de documentación. Los distintos módulos sirven para su configuración y para poder contar con una interfaz gráfica como se ve en el anexo [6.12]. En esta documentación podemos ver ruta, tipo de consulta y parámetros a entregar para que la consulta sea exitosa, también se puede hacer tests directamente desde la interfaz.

4.1.4. Seguridad

Al desarrollar cualquier aplicación se tiene que tener muy en cuenta la seguridad, especialmente si es web o móvil y va a tener harta gente interactuando y manejando información personal. Distintas formas básicas de controlar esto es siguiendo buenas prácticas como la validación de los datos y las variables de entorno para la información privada.

La primera forma fundamental de seguridad en el backend es la validación de datos. Si bien una validación en frontend es bien recibida y buena práctica, es modificable por un usuario malicioso, por lo que una segunda validación en backend es estrictamente necesaria. Al validar los datos en esta parte evita al usuario la manipulación del flujo y avisa en caso de error. Los datos se validan antes de hacer la consulta *SQL* y esta se ejecuta si y solo si la validación es correcta.

Una segunda forma de seguridad en el backend es el uso de variables de entorno. Estas variables pueden contener credenciales, información sensible, llaves u otros campos que el desarrollador o desarrolladora no quiera que sean públicos. En el caso de *Easy - Negocios* se utiliza como variable de entorno las credenciales de la base de datos, y la librería que se utiliza para esto es dotenv. Esta librería permite administrar este tipo de variables y utilizarlas dentro del proceso de una forma segura.

Y la tercera buena práctica con respecto a la seguridad en el backend es el correcto uso de los endpoints. Definir que parámetros se pasan por la ruta o cuáles se entregan por *body*. Condicionar ciertas rutas a algunos usuarios y asegurarse de que un usuario x no pueda acceder a la información del proyecto de un usuario y. Son cosas básicas esperables de un buen proyecto en cuanto a la privacidad de los datos.

Estas tres prácticas aportan tanto a la validación de que los datos correctos terminen en la base de datos como a la privacidad que un usuario espera de una herramienta para administrar proyectos. Vale la pena mencionar que para esta iteración de *Easy - Negocios* no se tomó en cuenta la encriptación de los datos de cada proyecto, esto porque no se consideró como prioritaria y se optó por desarrollar cosas más urgentes.

4.1.5. Deploy Backend

Una vez que todo se desarrolló y funcionó de manera local, se tiene que migrar tanto la base de datos como el código para que estén en algún servidor accesible a los usuarios y la aplicación se pueda comunicar con esta mediante la API. Para hacer esto existen bastantes herramientas como *Docker*, para encapsularlas, y las ofrecidas por Google para hospedarlas.

Para migrar la base de datos basta con crear una instancia en algún servidor y poder contar con alguna IP y credenciales de acceso. En el caso de *Easy - Negocios*, se utilizó una base de datos SQL almacenada y hospedada en Google con la herramienta *Cloud SQL*. Esta tiene bastantes opciones de configuración, pero una vez creada entrega una dirección IP y credenciales de acceso. Se analizó la opción de utilizar una tabla *BigQuery*, que tiene una interfaz más cómoda, pero se optó por la *SQL* para tener las reglas y la rigidez que traen consigo (IDs únicas, consultas flexibles, entre otras)

Para lo que es el código, lo primero que se hace es utilizar *Docker* como herramienta para contener la aplicación entera en un entorno propio. Esto ayuda mucho al momento de manejarla porque permite independizar la aplicación de cualquier dependencia local y hacerla funcionar con lo mínimo. Una vez que funciona en un contenedor, se puede estar seguro de que su comportamiento no depende de la máquina en la que se corre.

Una vez se tiene el contenedor, hay que elegir donde hospedarlo. Como se había utilizado *Cloud SQL* para la base de datos, y normalmente los servicios de las mismas compañías funcionan bien juntos, se decidió utilizar algún *host* de Google. Dos de las opciones que permiten correr contenedores son *Cloud Run* y *App Engine*. La diferencia entre estas dos es que la primera se prende con las consultas y cuando hay inactividad se apaga, haciéndola más lenta, pero más barata. La segunda está disponible todo el tiempo, asegurando tiempo de respuesta al instante, pero consumiendo más recursos. La comparación de latencia y precio de estas dos se pueden ver en las figuras 4.1 y 4.2

	Cloud Run	**App Engine**
Run 1	632 ms	471 ms
Run 2	485 ms	568 ms
Run 3	562 ms	470 ms
Average	559 ms	503 ms

Figura 4.1: La primera consulta es bastante más lenta, pues se tiene que activar el servicio, pero después se igualan y en promedio es un poco más lento *Cloud Run*

	Cloud Run	**App Engine**	**Heroku Hobby Plan**
Monthly cost	$0.09	$11.29	$7.00

Figura 4.2: *Cloud run* tiene un valor 99 % más barato

Dentro de estas dos opciones, se utilizó *Cloud Run*, priorizando el precio de mantenimiento.

Una vez hecho el contenedor de la API y puestas las credenciales de la base de datos, se hizo el *deploy* y quedo recibiendo respuestas en un URL. Vale decir que se utilizaron todas las prácticas para mantener las credenciales e información sensible seguras con el sistema de secretos de Google.

4.2. Frontend

En esta sección se explicará el proceso de implementación del frontend, esto es, todo lo que el usuario puede ver, modificar, interactuar, entre otras. Se van a explicar nuevamente las tecnologías que se utilizaron, pero ahora para esta parte en específico (Subsección 4.2.1), luego las librerías externas y las ventajas e importancia de cada una (Subsección 4.2.2), para seguir con otro punto repetido e importante que es la seguridad (Subsección 4.2.3). Luego se hablará de las distintas rutas que tiene la aplicación por dentro (Subsección 4.2.4), para seguir con la interfaz definitiva y su objetivo (Subsección 4.2.5) y se hablará de cómo funciona lo desarrollado en Android, junto con la forma de emularlo (Subsección 4.2.6). Por último, se explicará la migración realizada para tener la aplicación disponible (Subsección 4.2.7).

4.2.1. Tecnologías

React - Native es un framework que, aparte de ser código abierto, sirve para desarrollar todo tipo de aplicaciones, por lo que es sumamente utilizado en la industria. Viene con bastantes componentes funcionales para cada caso, y, si bien tiene orientación a objetos, las buenas prácticas actualmente llevan a la programación funcional. Otra de las tecnologías que se utilizó fue *Expo*, pero esta va a ser explicada más adelante cuando se explique la interfaz 4.2.5 y Android 4.2.6.

Por convención, *React - Native* funciona a base de componentes y pantallas. Dentro del directorio principal se encuentra *./components* y *./screens*. Los componentes son todo aquello que se tiene que renderizar en la vista del usuario y las pantallas son aquellas que utilizan los componentes desarrollados.

La curva de aprendizaje de *React - Native* es bastante amigable y su filosofía es *'Learn one, Write everywhere'*, lo que habla de la gran cantidad de usos que tiene este framework.

Con respecto a la implementación, escribiendo con *React - Native*, se utilizan mucho los llamados *Hooks*. Como ya se mencionó, si bien soporta programación orientada a objetos, predomina la programación funcional, y es por esto que surgió la necesidad de implementar estos llamados *Hooks*, que React Native mismo los define como *'Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class.'* [10].

El uso de *Hooks* es fundamental, y en *Easy - Negocios* se utilizan dos tipos de manera frecuente, los *State Hooks* y los *Effect Hooks*. El primero guarda, como lo dice su nombre,

el estado o valor de una variable, es similar a tener un objeto con un valor asociado e ir cambiando este valor. El segundo permite realizar efectos secundarios en los componentes. Un *Effect Hook* se ve de esta manera:

Listing 4.3: Esta función realiza efectos con cada carga de la pantalla

```
useEffect(() => {
    if (route.params && route.params.object_id) {
      navigation.setOptions({ headerTitle: "Actualizando Compra" });

      setEditing(true);

      (async () => {
        const object = \
            await getInvestment(...project_id, ...objectt_id)
        setInvestment({
          total_net_price: object.total_net_price.toString(),
          total_gross_price: object.total_gross_price.toString(),
          ticket: object.ticket,
          investment_description: object.investment_description,
          owned_product: object.owned_product,
          cuantity: object.cuantity.toString(),
          investment_date: object.investment_date ? \
              new Date(object.investment_date.slice(0, 19)) : null
        })
      })();
    }
  }, [isFocused]);
```

Lo que dice este fragmento de código es que al momento de cargar la pantalla que lo contiene se revisará si ya existe un objeto. En caso de existir, se va a cargar la información correspondiente a este en una variable. En la última línea podemos notar el *isFocused*, que se refiere a que esta función se va a ejecutar cada vez que la pantalla sea cargada.

Vale la pena mencionar, también, que dentro de este *Effect Hook* también hay *State Hooks*. El *setEditing(true)* y el *setInvestment(...)* lo que hacen es cambiarle el estado a variables para que mantengan un valor. La definición de estas variables se ve así:

Listing 4.4: Uno de los valores es un booleano y el otro es una estructura JSON

```
  const [editing, setEditing] = useState(false);

  const [investment, setInvestment] = useState({
    total_net_price: null,
    total_gross_price: null,
    ticket: null,
    investment_description: null,
    owned_product: null,
```

```
    cuantity: null,
    investment_date: null
  });
```

Gracias a estos componentes incluidos en *React - Native* es que se puede programar de manera directa 'simulando' una programación orientada a objetos y aprovechando también los beneficios de la programación funcional.

4.2.2. Librerías de terceros

Si bien *React - Native* es un framework, este es muy grande, por lo que tiene distintas partes y extensiones que se pueden utilizar por separado. Quizá no es correcto hablar de librerías, sino que de paquetes, pero se explicarán los más importantes que fueron utilizados dentro de *React* y se explicará de manera breve e introductoria lo que es *Expo* para más adelante poder comentarlo con más detalle.

La primera librería, y fundamental, que se utilizó para trabajar con *React - Native* es *React - Navigation*. Esta librería permite gozar del clásico estilo de navegación de las aplicaciones móviles pero usando *React - Native*. Con esta librería uno puede navegar a través de las pantallas, entregando parámetros entre una y otra para poder traspasar información importante. Un ejemplo de como se navega con esta librería es:

Listing 4.5: Por la navegación se le entrega a 'ProductListScreen' la variable 'project_id'

```
<TouchableOpacity
    style={styles.ProjectHomeButton}
    onPress={() => navigation.navigate('ProductListScreen',
            {project_id: props.route.params.project_id})}>
<Text style={styles.ProjectHomeText}>Productos</Text>
</TouchableOpacity>
```

Se puede ver en la función *onPress()* que se va a navegar hacia la pantalla 'ProductListScreen', pasándole por la navegación la variable definida 'project_id', la cual es necesaria para hacer la consulta de los productos del proyecto a la base de datos.

Otro módulo de *React - Native* que se utilizó fue *React Native Picker & React Native Date Time Picker*, los cuales facilitan la entrega de inputs de parte del usuario. El uso de estos módulos fue muy útil, ya que facilitaron mucho el trabajo de los inputs, y considerando que la herramienta desarrollada tiene como característica que la interfaz sea intuitiva y fácil, el hecho de tener estos módulos ayuda a que la estética sea conveniente. Como se puede ver en la imagen 4.3, a la izquierda está el seleccionador de fecha, que es bastante clásico, y a la derecha es como se ve el seleccionador de opciones con un menú desplegable.

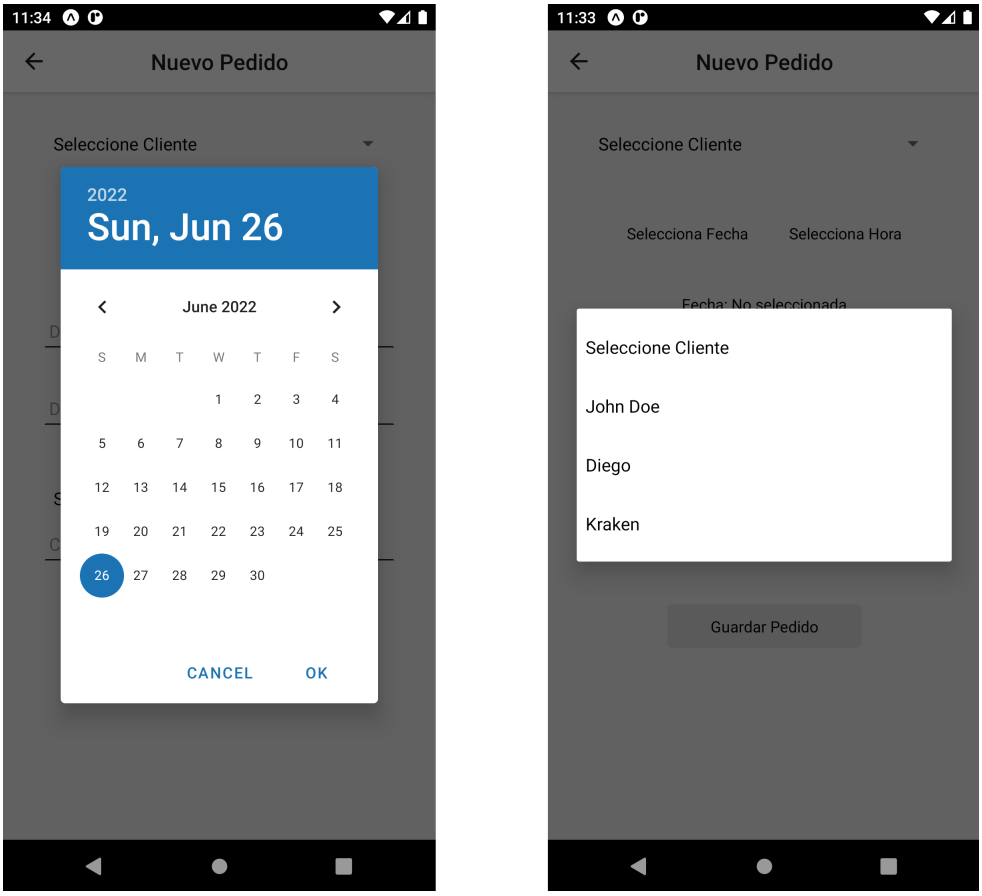

Figura 4.3: Native Pickers para fecha y para selección

Y la tercera y última librería importante que se utilizó es *Expo*, la cual también puede llamarse plataforma por sus características. Esta es la conexión que tiene el frontend con un emulador de Android o iOS y es la que permite ir utilizando la aplicación a medida que se desarrolla sin necesidad de un dispositivo físico. Va a ser explicada a más detalle en la sección de la interfaz 4.2.5.

Firebase

Un módulo completo que se utilizó dentro del Frontend es *Firebase*. Este módulo cuenta con funcionalidades para autenticación, *hosting*, almacenamiento, testeo y una cantidad enorme de otras. Solo se utilizó la parte de autenticación.

Lo bueno de este módulo es que funciona con *Expo* y es confiable. Es de *Google* y cuenta con opciones para ingresar con inicio de sesión, con cuentas de *Google*, entre otras. Toda la lógica de ingreso, almacenamiento de usuarios, IDs, información sensible, permisos, se manejan de forma automática con el uso de este módulo.

Este módulo entrega funcionalidades dentro del código como información de la autenticación, del estado del usuario, de las características que este tiene (en caso de haber distintos tipos de usuario), de tokens de acceso, entre otras útiles. Y fuera del código tiene herra-

mientas para facilitar la recuperación de contraseña, la confirmación por email al momento de registrarse, y otro tipo de funcionalidades usadas frecuentemente cuando se trata con usuarios.

4.2.3. Seguridad

Con respecto al frontend, al ser el medio que tiene el usuario para interactuar con nuestro servicio, hay que ser muy cuidadoso con los usuarios maliciosos. La mejor forma de desarrollar una aplicación segura es siguiendo las buenas prácticas con respecto a la programación y a la manipulación de información.

La medida más básica de seguridad para el frontend, que a su vez es de las mas vulneradas, es el escape de caracteres. Cerca de la mitad de los ciberataques son por inyección de código [9], el cual se puede evitar asegurándose de lo que procese el input no se ejecute directamente y llegue como texto. Para eso sirve en gran parte el escape de caracteres, para que si alguien, por ejemplo, escribe $' < h1 > Hola < /h1 >'$ como input, llegue exactamente eso y no un **Hola** con formato de *header* HTML.

La otra medida de seguridad que se puede hacer a nivel de interfaz es una validación instantánea de los datos. Si un usuario es malicioso entonces podrá evitar esta validación, pero si se está utilizando la herramienta de buena manera entonces es una mejor experiencia de usuario que en caso de estar escribiendo algo mal como input, este te avise de manera instantánea y no esperar a enviar un mensaje al backend para darse cuenta de que había un error. Para esto sirve una validación de datos a nivel frontend.

4.2.4. Rutas y Paso de Argumentos

Al momento de desarrollar una aplicación web, muchas veces hay que elegir entre pasar argumentos mediante un *body* o por URL. Al ser desarrollada como aplicación nativa, por más que sea móvil, no es diferente. Se explicará en que casos se decidió pasar argumentos por ruta y en qué situaciones el argumento se entrega por un *body*, justificando la decisión.

Cada endpoint tiene que recibir distintos tipos de argumentos y retorna la información de distinta forma. Los endpoints y su documentación realizada por *Swagger*, librería mencionada previamente, se pueden ver en el anexo a partir de la imagen 6.13.

En resumen, cada uno de las tablas de la base de datos tiene un *Get All*, un *Get Unique*, un *POST*, un *DELETE* y un *PUT*. Para los *GETs* y el *DELETE*, los parámetros son entregados por la ruta, ya que solo son IDs. Para el *POST* se requiere más información, pues necesita cada campo en la base de datos, por lo que se decidió enviar estos a través de un *body* en JSON. Al ser tanta información, no se entregan por URL por comodidad y para mantener el orden. Además, se considera mas segura esta forma. Por último, para el *PUT* se requiere un ID por ruta que corresponde al objeto que se está modificando y a un body con la información para sobreescribirlo.

Cada una de estas consultas tiene también un valor de retorno. Los *GETs*, lógicamente,

retornan un JSON o un arreglo de JSONs con la información que se consultó. El *DELETE* y el *POST* retornan un estado HTTP, indicando si fue exitosa o no la consulta. Y por último, el *PUT* retorna un estado HTTP e información acerca del cambio, es decir, cuantas columnas cambiaron y cuantas fueron exitosas.

Al ser una aplicación desarrollada en *React - Native* y con funciones generales, esta está enfocada en ser multiplataforma. Si bien se descartó la posibilidad directa de ser una app web (habría que cambiar ciertas funciones) sirve para todo sistema operativo móvil, y funciona de igual manera que una web en cuanto a las rutas, navegación, direcciones, etc. Esto es un gran punto a considerar para poder migrarla a una herramienta global.

4.2.5. Interfaz

Uno de los pilares de la herramienta desarrollada es que con pocos 'clicks' el usuario puede lograr lo que quiera, que se pueda utilizar de manera simple e intuitiva. Para esto se llegó a una interfaz que muestra lo fundamental de manera muy directa, que permite agregar, eliminar y modificar la información clave. Sin colores muy llamativos ni elementos que distraen del objetivo principal que es administrar el negocio.

Para ir emulando esta interfaz se utilizó la plataforma *Expo*, ya mencionada como librería anteriormente. Esta es una herramienta que, junto a *Android Studio* en el caso de *Easy - Negocios*, permite emular el código desarrollado. Una de las grandes ventajas de *Expo* es su buena documentación, que es *open source* y que permite mostrar de manera casi instantánea los cambios, siendo compatible con hartas librerías de *React*, facilitando el trabajo.

La interfaz terminada se acerca bastante a lo que fue la interfaz diseñada, cumpliendo el objetivo de resaltar los puntos importantes para el negocio, la pantalla principal de cada proyecto cuenta con cada punto accesible de manera directa como se puede ver en la figura 4.4.

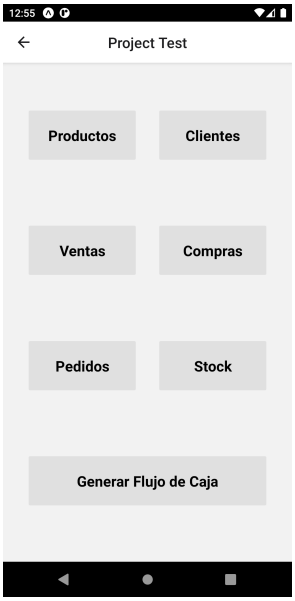

Figura 4.4: Vista principal de un proyecto

Si bien la diseñada tenía colores más vivos, se decidió optar por los más sobrios por un tema de estética e iluminación. Aparte al ser implementado con ese tipo de colores, la opción de eliminar resalta más, como se ve en la imagen 4.5, causando el efecto que un usuario esperaría al ver una 'X' para borrar un elemento. Con muchos colores se iba a perder el enfoque y podía hacer más difícil la lectura para la gente con problemas a la vista, considerando que dentro del público objetivo esta gente poco cercana a la tecnología.

Por último, y para tener una idea de cuantos clics se tienen que hacer para lograr una acción, se puede ver que desde la pantalla principal de un proyecto, figura 4.4, a la lista de clientes (figura 4.5) hay un solo paso, y para llegar al formulario de agregar cliente (figura 4.6) es un solo paso más, en el que hay que escribir los inputs, donde solo es obligatorio el que es necesario, y se agrega un cliente.

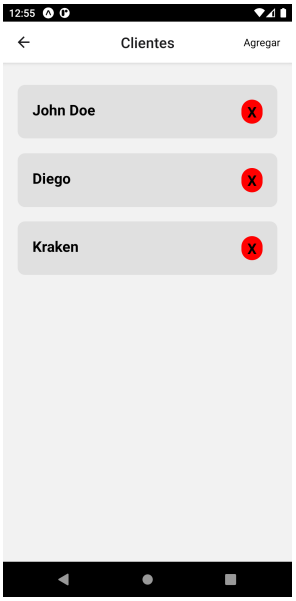

Figura 4.5: Lista de Clientes en la Interfaz Final

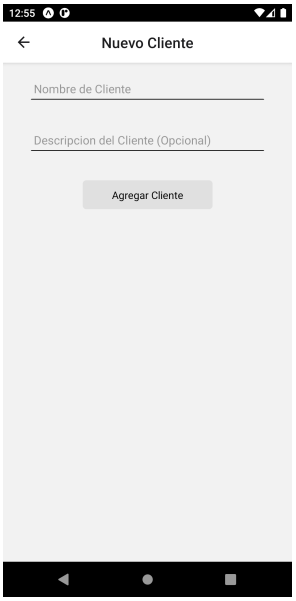

Figura 4.6: Formulario para agregar Cliente en la Interfaz Final

Se puede ver entonces que la implementación de la interfaz es bien simple, logrando, al menos en teoría, el objetivo de que sea intuitiva y fácil de navegar. Todo esto se logró gracias a las ventajas y módulos disponibles en *React - Native* y a la emulación apoyada por *Expo*. La interfaz para los otros puntos de un proyecto no tiene grandes cambios, ya que mantener una misma forma de uso para cada campo simplifica el uso de la herramienta.

La totalidad de la interfaz final se puede ver en el anexo 6.41, si bien tiene detalles estéticos, estos se arreglarán cuando las necesidades más urgentes sean cubiertas.

4.2.6. Android

Al ser una aplicación móvil, era necesario ir observando en cada momento cuando se realizaba un cambio. Para hacer esto se puede conectar un celular y verlo reflejado ahí o se puede usar algún tipo de emulador. En este caso se utilizó la ya mencionada herramienta *Expo* con un emulador de Android, aunque en ningún momento se utilizó alguna función exclusiva de *React* para este sistema operativo, por lo que en iOS cuenta con las mismas funcionalidades y acciones.

La aplicación desarrollada apunta a ser multiplataforma en todo momento y en caso de utilizar alguna función exclusiva a un sistema operativo se piensa hacer alternativas para mantener su globalidad.

4.2.7. Deploy Frontend

El *deploy* realizado en esta sección es necesario para tener la herramienta disponible a descargas. Con lograr estar en *Google Play Store* para los dispositivos con sistema operativo Android, y en *AppStore* para los dispositivos de *Apple*, ya se incluye al 90 % de Chile, como se explicó en 3.3. En *Easy - Negocios* solo se logró estar en la primera, pues la segunda requería la utilización de un computador de la marca y un valor de aproximadamente 100 dólares anuales.

El primer paso fue crearse una cuenta de desarrollador de *Google*, pagando una cuota única para tener acceso a *Google Play Console*, página web para administrar las aplicaciones que se suben a *Google Play Store*. El segundo paso es configurar la cuenta, preparar el lanzamiento y tener un archivo tipo *.abb* para subir a la tienda. La utilización de *Expo* fue clave para este punto, pues cuenta con bastantes facilidades y atajos que permiten hacer este paso de manera más directa.

Una vez se tiene el archivo, *Google Play Store* pide explicar de que se trata la aplicación, pantallazos con las funcionalidades, textos descriptivos, y especificaciones técnicas para cumplir con los términos y condiciones de la empresa. Una vez se entrega todo y su sube el archivo, entra a fase de verificación, donde puede estar hasta siete días antes de que la aplicación sea verificada y este disponible en *Play Store*.

Ya teniendo disponible la primera versión, subir parches y actualizaciones es más fácil y directo, sumándole que la fase de verificación demora hasta menos de una hora. *Expo* cuenta con funcionalidades específicas para automatizar el versionamiento de la aplicación y actualizarla directamente en la tienda.

La forma final de la estructura de la herramienta se ve como en la figura 4.7, especificando comunicaciones desde los distintos módulos.

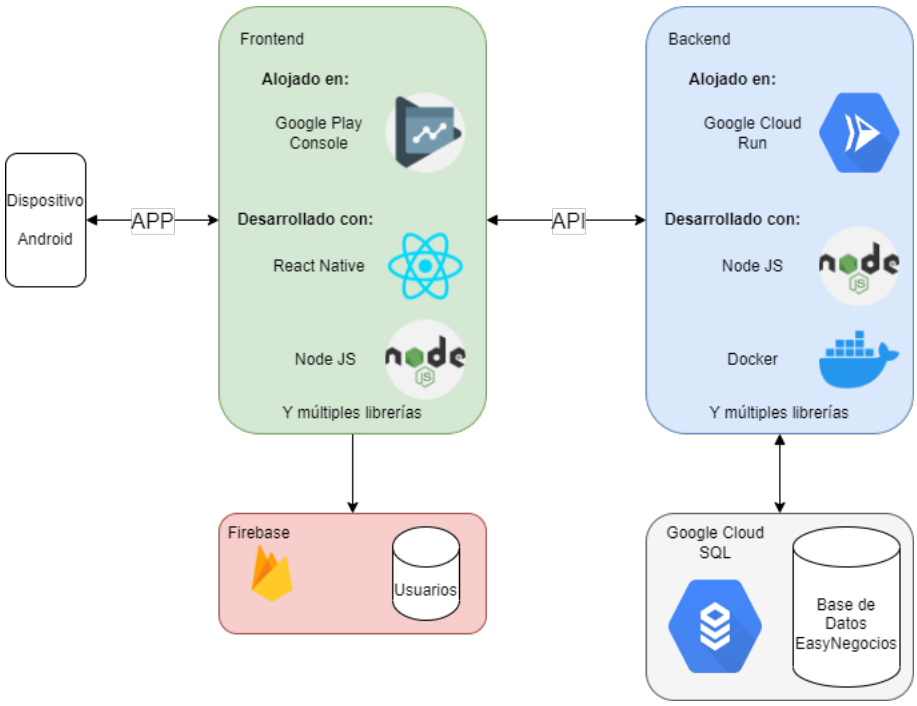

Figura 4.7: Estructura del proyecto: En esta figura se pueden ver los distintos módulos con las tecnologías principales que se utilizaron en cada uno. Para el caso de iOS, lo único que cambiaría es el lugar de alojamiento del Frontend.

4.3. Funcionalidades de la Herramienta

Si bien se diseñó la aplicación para que cuente con una cantidad de *features*, no se lograron implementar todas por falta de conocimiento y de tiempo para aprendizaje y desarrollo. Sin embargo, se priorizaron las necesidades del usuario y de la administración para cubrir las que se consideraron más necesarias. No se lograron implementar el uso de redes sociales y la central de pagos.

- **Productos:** Para este punto lo que se puede hacer es listar los productos existentes, agregar nuevos, modificar los antiguos y eliminar. Los productos tienen una característica especial que fue la razón principal por la que se hizo el cambio de la columna 'active' en la base de datos, y es que si en una venta hay una cantidad de productos x, y luego, meses después, se actualizan los precios y se venden productos x', entonces no es la idea que el flujo de caja tome en cuenta todo el historial con el precio actualizado. Para solucionar esto se hizo que el método 'PUT' no actualice productos, sino que desactive el antiguo y cree uno nuevo. Igual que el 'DELETE', este no elimina, solo desactiva, ya que en otro caso al momento de generar el flujo de caja iba a buscar el producto y al no encontrarlo iba a causar un error. Eliminar los elimina de la lista mas no de la base de datos.
- **Clientes:** Este punto es bien simple, solo requiere dos campos y uno es opcional. La idea de este es que sea fácil de usar y que la descripción, en caso de ponerla, sirva

para identificar a clientes frecuentes y poder darles una atención más personalizada. Se contempla como idea a futuro agregarles el campo dirección como opcional, para que si hacen un pedido esta se escriba de forma automática, obviamente siendo modificable en caso de que, por ejemplo, este comprando un regalo hacia otra dirección.

- **Ventas:** Esta sección corresponde a las ventas que se hacen en el momento o local, que no necesariamente son pedidos. Cada negocio tiene su sistema de venta y la idea es que este punto sea lo suficientemente como para abarcarlos todos. Al momento de agregar una venta, como se ve en el anexo [6.50], se elige producto, cantidad, información de la fecha, si es boleta o factura (importante para el tema de los impuestos en el flujo de caja) y una descripción. Campos como la fecha de venta o si es boleta o factura pueden no ser ingresados y nulos o tener un valor por defecto, decidido por la frecuencia de uso en comparación a su contraparte (boleta). Para obtener y ordenar esta información se tuvo que hacer un *JOIN* en la consulta *SQL*, este traía la información referida en la tabla *Sale_Product_Relation*, pero sin traer campos extra por temas de seguridad y privacidad en caso de haber una vulnerabilidad.
- **Compras:** Este punto también está pensado para la generación del flujo de caja, la idea es que sea muy fácil agregar una nueva compra (se pueda poner poca información) pero si se utiliza un poco más de tiempo se puedan completar más campos. Como se puede ver en el anexo [6.52], uno puede elegir si es boleta o factura (boleta por default) o, por ejemplo, si es compra y venta (para aquellos negocios que funcionan así). La idea de este último punto es que actualice el stock de forma automática.
- **Pedidos:** Esta pestaña sirve para ordenar los pedidos que pueda tener un negocio, teniendo a simple vista la persona, fecha, dirección y detalles del pedido. Si bien no está implementado todavía, se espera a futuro poder filtrar y ordenar la lista por distintos criterios para tener un mayor orden. Se podría también ocupar un sistema de notificaciones para asegurar que no se olvide ningún pedido. Para implementar los pedidos se tuvo que hacer un *JOIN* de las tablas incluidas en *Order_Product_Relation* y ordenar la información obtenida. Siempre que se hacen estas consultas se trata de obtener solo la información justa y necesaria, por temas de seguridad en caso de haber una vulnerabilidad.
- **Manejo de Stock:** Una lista con cada producto que se tiene y su stock, no es más complicado que eso, y puede ser de mucha utilidad al emprendedor o emprendedora para saber qué productos ofrecer a la venta, hacer ofertas, saber qué comprar, etc. También se cuenta con planes para poder ordenar por nombre y por cantidad de productos, así sería más directo saber de qué cosas va quedando poco y es necesario comprar. Se actualiza automáticamente en caso de comprar y seleccionar 'Compra y Venta'
- **Flujo de Caja:** Ha pasado por muchas versiones de vista antes de la versión definitiva. Este punto es sin duda uno de las fuertes fuentes de información para el emprendedor o emprendedora acerca de su negocio. Saber cuánto ha ganado o perdido en un periodo de tiempo para tener la información de qué épocas son más fuertes que las otras en cuanto a movimiento. Implementar esto fue complicado debido al enfoque permanente de mantener las vistas simples.

Vale la pena mencionar que el esfuerzo por mantener la simpleza y limpieza de las vistas e interfaces ha sido permanente y seguirá siéndolo a futuro en el desarrollo de *Easy - Negocios*,

ya que ese sello es muy importante y sigue la línea de la globalidad de la herramienta. Quedan como *features* diseñadas, pero sin implementar, las dos ya mencionadas que son el manejo de redes sociales y la central de pagos.

Capítulo 5

Validación

Para poder obtener opiniones del trabajo desarrollado y confirmar que *Easy - Negocios* tiene un impacto en sus usuarios, se hicieron una serie de validaciones. Estas incluyen a terceros en lo que es la validación externa (Sección 5.1), observando prácticamente y mediante preguntas cómo es la interacción con el usuario. Luego se analizará la validación interna, mostrando teóricamente como se logró el resultado (Sección 5.2) y se terminará comparando los objetivos principales con los resultados obtenidos (Sección 5.3).

5.1. Validación Externa

La validación es de las partes más importantes en el desarrollo de un producto, pues este nos da una idea de cuánto realmente sirve la herramienta desarrollada y le da un uso real. Hay bastantes formas de validar una aplicación utilizando participación de externos. Desde encuestas, pruebas en vivo, videos, explicaciones hasta mostrar casos de uso y escuchar opiniones.

En este caso se optó por contactar a varios potenciales usuarios y usuarias para que opinen acerca de la herramienta. La idea era observar todo el proceso en el que la persona de prueba empieza a interactuar con la aplicación, sin interrumpirle ni señalarle nada. En particular, se contactó a gente que tiene el sistema operativo *Android* y algún emprendimiento de cualquier tipo, se descargaron la aplicación para que la utilicen y exploren por el tiempo que estimen conveniente y luego se les entregó una encuesta enfocada a validar.

La encuesta contaba con 9 preguntas, 3 con respuestas de texto y 6 de selección, con un enfoque hacia la evaluación de los objetivos tanto generales como específicos, preguntando por las distintas aristas del negocio mencionadas en la Sección 2.1. La encuesta tuvo un total de 16 participaciones, los cuales son una parte de los que se contactaron por redes y foros. Haremos un análisis de cada pregunta con sus respuestas y, en caso de las respuestas con texto, se resumirá la idea general captada. Se podrá ver la totalidad de respuestas anonimizadas en el Anexo 6.3.3.

1. "**(Opcional) Escribe tu nombre y/o el de tu negocio, me ayudaría para escribir en el informe y demostrar que son casos reales":** Por temas de privacidad solo se mostrarán los nombres de los negocios, no de las personas.

    - Thisfruta
    - Outdoor Equipment
    - Molko Outdoor
    - monpetitpast
    - Yaroomba
    - Hurtragos
    - De.Vela
    - Psíncronia Terapias

2. "**Según tu primera impresión. ¿Para qué sirve Easy - Negocios?":** Palabras clave que aparecieron harto en estas respuestas son 'administrar', 'manejar', 'gestionar'. Todas las respuestas tienen relación con el control de los distintos puntos del negocio, por lo que la idea de la herramienta se entendió bien dentro de los usuarios de prueba. Esto da una idea de que el plano general de la aplicación está bien definido y un usuario que la ve a través de *Play Store* puede tener una idea de la funcionalidad en su primera impresión.

3. "**(Del 1 al 10) ¿Consideras que la interfaz es intuitiva, simple y fácil de utilizar?":** El promedio de las respuestas fue un 7,56. Se considera que es un buen número y cumple con el objetivo de ser una interfaz simple, al menos para los sujetos de prueba. La menor de las respuestas fue un número 5, por lo que se podría rescatar que ninguno de los y las sujetos de prueba estuvo totalmente perdido al momento de abrir la interfaz.

4. "**(Del 1 al 10) ¿Consideras que esta herramienta le sirve a un negocio para ordenar y almacenar su información importante?":** El promedio de las respuestas fue un 8,06. Siendo el mínimo un 4, pero estando la mayoría centralizadas entre el 8 y el 9. Esta distribución muestra que la mayoría de los usuarios de prueba está de acuerdo con que la herramienta cumple el objetivo general que se definió en la sección [1.1.1], considerándolo como logrado.

5. "**¿Consideras que el correcto uso de esta herramienta podría ahorrar tiempo en la administración de un negocio?":**

    - 11 personas contestaron 'Sí, creo que utilizarla podría reducir el tiempo que le dedico al negocio sin reducir las ventas.'
    - 2 personas contestaron 'No, utilizarla me tomaría más tiempo del que me tomo actualmente.'
    - 1 persona contestó 'Depende de los productos. Un negocio de barrio no sirve. Productos a pedido sí.'
    - 1 persona contestó 'Creo que sí, pero está destinada a negocios de productos'

- 1 persona contestó 'Sí, pero sería bacan que tuviera Integración a la página web, cosa que la información se le pase automáticamente y no estar uno rellenándolo'

6. "**¿Consideras que el correcto uso de esta herramienta podría significar un aumento en los ingresos del negocio?":**
   - 6 personas contestaron 'Sí, con esta herramienta mejoro el balance. Me ahorro dinero destinado en otras cosas o el orden me permite aumentar las ventas.'
   - 6 personas contestaron 'No, creo que la forma de administrar que ofrece la herramienta y los ingresos no tienen relación.'
   - 1 persona contestó 'No necesariamente, pero si tener todo más ordenado, y a la mano'.
   - 1 persona contestó 'Quizás, tendría que probar'
   - 1 persona contestó 'No a subir ventas, pero sí a mejorar y optimizar tiempo... que si eventualmente puede llevar a más ventas, pero no de forma directa'
   - 1 persona contestó 'No lo sé'
7. "**¿Te es útil poder generar un flujo de caja por fechas y al instante?":**
   - 6 personas contestaron 'Sí, pero de la forma que funciono hasta ahora también lo tengo disponible.'
   - 6 personas contestaron 'Sí, y es algo que no hacía hasta ahora.'
   - 2 personas contestaron 'Sí, pero el de Easy - Negocios carece de información y opciones suficientes.'
   - 1 persona contestó 'Eventualmente xd'
   - 1 persona contestó 'No.'
8. "**¿Crees que Easy - Negocios podría ayudar a la gente a manejar de mejor manera sus negocios?":**
   - 10 personas contestaron 'Sí, pero solo si mejoran ciertos puntos.'
   - 6 personas contestaron 'Sí.'
   - Nadie contestó 'No.'
9. "**Comentarios (que es lo que más o menos te gusta de la herramienta), Feedback, Opiniones, Críticas. (No son necesarias, pero son demasiado bienvenidas)":** Hartos de los comentarios tienen que ver con consejos para la interfaz para que quede más amigable, otros hacen alusión a la confusión entre 'ventas' y 'pedidos'. Otros comentarios hablan acerca de la actualización de stock, que debería ser automática. Lo otro que se repite es la confusión entre categorías y productos. Hartas respuestas mencionan el potencial que le ven a la aplicación de aplicarse los comentarios. Estos comentarios quedarán también plasmados en la sección 6.3.2 de mejoras.

En general, los objetivos de *Easy - Negocios* se consideran cumplidos. Si bien hay harto espacio para mejoras, se logró cubrir con una nota relativamente para la iteración los puntos clave y las aristas de un negocio que se lograron implementar, pues 'Redes Sociales' y 'Manejo

de Pagos' no llegaron al producto final. Si bien se pudo haber hecho una validación más dedicada, se optó por utilizar poco tiempo del usuario para lograr llegar a más gente.

La comparación más detallada de cada arista de negocio implementada y lo que se considera logrado se puede ver en la sección 5.3, donde se analiza más profundamente cada punto de la herramienta.

5.2. Validación Interna

La otra validación que se realizó es interna, que corresponde a la que uno mismo le da a la herramienta con datos recopilados, para ver si se logran ciertos resultados de eficiencia, dimensión, flujo de información y cualquier paso dentro del proceso que nos permita llegar a los resultados esperados. Primero veremos la cantidad de clics para lograr acciones (Subsección 5.2.1) y luego pasaremos a datos como tiempo de respuesta, accesibilidad, seguridad y estabilidad, obtenidos a través de *Google Play Store* (Subsección 5.2.2).

5.2.1. Distancias en Grafo de acciones

La cantidad de acciones o eventos que se deben cumplir para realizar una acción en una herramienta habla mucho de la eficiencia que esta tiene. Comúnmente las acciones más realizadas dentro de un tema deben ser las que requieren menos clics para lograr, pues probablemente el usuario las realizara con más frecuencia.

En el caso de *Easy - Negocios*, al tener un menú con los recuadros principales como pantalla principal, para hacer cualquier acción de agregar, ya sean productos, clientes, ventas u otro de las posibilidades, hace falta el clic en el recuadro, clic en agregar, llenar los campos y clic en 'agregar'. Esto es un total de tres clics para la acción más importante. Vale la pena comentar que en la mayoría de los casos los campos son opcionales, con tal de ahorrar aún más tiempo si es que no se quiere colocar información.

Para lo que es actualizar y eliminar objetos, hace falta clic en el recuadro del tema desde el menú principal, luego en el objeto o en la 'X' y al final apretar 'Actualizar' o 'Confirmar' en el caso de la eliminación. Esto también se hace en tres clics.

Dicho esto, se tiene que la mayoría de acciones se hace en tres clics, a excepción de dos. La generación del flujo de caja se obtiene solamente con un clic, mientras que agregar una categoría (de productos) es posible con cuatro.

Todo este análisis supone que la persona sabe utilizar la herramienta, sin perder tiempo recorriéndola previamente. También supone que el usuario se encuentra en la pantalla de inicio de un proyecto, o sea que ya inició sesión y seleccionó un proyecto. Todo lo relacionado con manipular información se puede ver gráficamente en la figura 5.1.

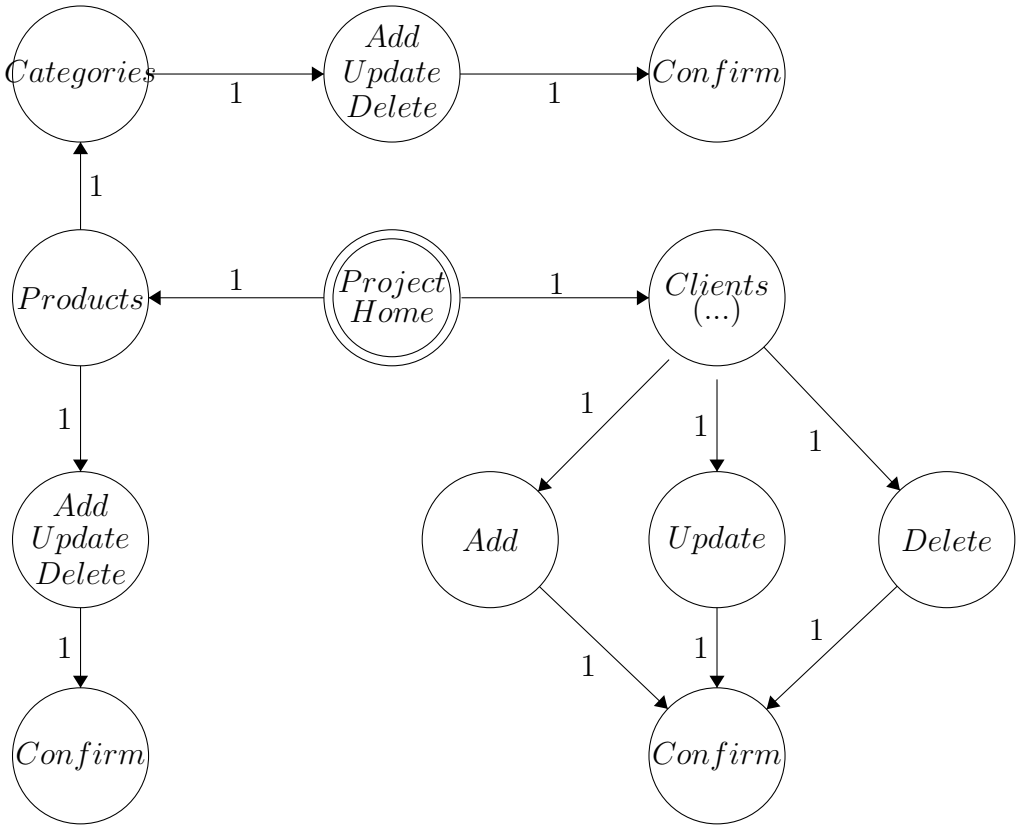

Figura 5.1: El grafo de acciones de *Easy - Negocios* para manipular la información. La generación del flujo de caja estaría a un clic desde *Project Home*. El nodo central es la pantalla inicial de un proyecto.

5.2.2. Indicadores de eficiencia

Los indicadores más técnicos de una herramienta nos hablan de la eficiencia. Esto incluye estabilidad, rendimiento, accesibilidad y seguridad. Estos cuatro puntos son analizados y testeados en profundidad por *Google Play Console* al momento de subir una aplicación a *Google Play Store*. Se mostrarán los resultados de estos test en la tabla 5.1 y luego se comentará acerca de las razones.

Tabla 5.1: Indicadores de Validación Interna

Stability	No issues found		
Performance	3 unique issues	Cold Start Time	Your app took 5,475ms to launch for the first time, after being installed on this device model
		Slow Rendering	1 % of frames took more than 700ms to render
		Slow Rendering	1 % of frames took more than 700ms to render
Accessibility	8 unique issues	Low contrast	Consider increasing this item's text foreground to background contrast ratio
		Touch target size	Consider making this clickable item larger
		Content labeling	The item may not have a label readable by screen readers
		Implementation	This editable TextView has an android:contentDescription. A screen reader may read this attribute instead of the editable content.
Security and trust	No issues found		

Primero que nada se puede observar que en estabilidad y seguridad, la herramienta *Easy - Negocios* no tiene problemas según el test de *Google Play Console*. En cuanto al rendimiento y a la accesibilidad, se encuentran unas cuantas advertencias.

Con respecto al rendimiento, tenemos un problema que tiene que ver con el tiempo de inicio de la aplicación y otros dos que hacen alusión al tiempo de renderizado de la pantalla. Estos problemas se deben a la tecnología utilizada y mencionada en 4.1.5, el *Cloud Run*. La razón principal de la elección fue el precio, y dentro de la comparación hecha no se consideraba que las opciones de configuración a elegir iban a ser, también, las más baratas.

El hecho de haber minimizado costos en este punto hace que el frontend se demore más en recibir la información del backend, manteniendo a la aplicación en un estado de espera en el que no puede renderizar totalmente la pantalla por falta de datos.

Por último, con respecto a la accesibilidad se tienen ciertos comentarios de contraste, de tamaño, de recuadros, de etiquetar contenidos y de implementación. Cada uno de estos

puntos, *Google Play Console* los entrega con pantallazos para especificar.

El punto de bajo contraste se refiere principalmente a los *placeholders* de la pantalla de inicio de sesión. El tamaño de los recuadros se refiere a los botones de inicio de sesión y registro. La falta de etiquetas para lectura está específicamente en una pantalla de carga al abrir la aplicación. Y, por último, los de implementación dicen que hay ciertos *inputs* con etiquetas para lectura de pantalla y que podrían ser leídas estas en vez de los *placeholders*, pero las etiquetas se pusieron para entregar más detalles y ser más claras, por lo que se considera que no son importantes.

En resumidas cuentas, se puede ver que la mayoría de los problemas señalados internamente por *Google Play Console* son solucionable de manera fácil (hablando de los de accesibilidad) o pagando más por el servicio en *Google Cloud Platform*, pero se decidió dejarlos así por decisiones de diseño y de comodidad para el usuario. Todo este análisis *Google* lo entrega como información que uno podría tomar en cuenta, pero que no es urgente, en caso de haber algún error, la aplicación no se puede subir a la tienda.

5.3. Comparación con Objetivos Iniciales

Toda validación debe tener como meta que los resultados sean similares o mejores que los objetivos planteados inicialmente, por lo que se comparará lo obtenido con lo esperado en cuanto a que esta aplicación desarrollada sea simple, que sirva para el público especificado, y que cumpla con ordenar la información y darle más datos al usuario de los que tenía previo al uso de esta.

Desde un principio se quería una aplicación simple, una aplicación que mantenga un orden tanto para el usuario como para el negocio, poder llevar las cuentas y los números de manera cómoda e intuitiva. Se quería una aplicación que genere indicadores útiles para el negocio y funcione de manera eficiente, ahorrándole tiempo al emprendedor o emprendedora sin afectar negativamente su balance.

Partiendo por la simpleza, *Easy - Negocios* demostró ser lo suficientemente simple como para que los usuarios de prueba logren navegar por ella y lograr las tareas básicas, pero no lo suficientemente simple como para que sea una característica a resaltar, es más, fue comentado en la encuesta de validación que había que seguir trabajando en la interfaz si la idea era llegar a un público menos cercano a la tecnología. Agrandar los botones para las tareas básicas o incluir un tutorial inicial podrían ayudar en esto.

En cuanto al orden, se logró una aplicación que permite administrar productos, clientes y otros datos fundamentales de un negocio de manera directa y limpia. El orden logrado para mostrar y almacenar la información es un punto bien alcanzado, pues con pocos clics se puede observar la información en tiempo real y con capacidad de agregar, modificar y eliminar cualquier tipo de campo.

Siguiendo por los indicadores entregados, hasta el momento el flujo de caja solo entrega gastos e ingresos por pedidos y ventas con distintos filtros y por un rango seleccionado de

fechas. Si bien fue bien recibido en la validación, con usuarios que entienden la importancia de un flujo de caja actualizado, se cree que para lograr la totalidad de este punto se debe trabajar más en la gráfica y en las opciones de este ítem.

Continuando con la eficiencia de *Easy - Negocios*, según los tests de *Google Play Console* mencionados anteriormente y el criterio personal, este punto estaría logrado, pues la mayoría de los problemas se solucionarían pagando mejores planes o cambiando puntos de la interfaz que se considera que están correctos y de los emprendedores y emprendedoras que validaron, nadie comento lo contrario.

Por último, el objetivo de abarcar un público objetivo amplio no se considera logrado, pues si bien no se puede asegurar que *Easy - Negocios* no funcionaría para gente mayor o con discapacidad, tampoco se puede asegurar que sí lo haría por falta de sujetos de este espectro para la validación. Se tiene seguridad que para gente que tiene cierto nivel de cercanía a la tecnología, esta herramienta es utilizable y navegable, pero no se tiene la misma seguridad con el espectro de edad no validado. Los puntos referidos en el diseño, pero no en la implementación, como el manejo de pagos o de redes sociales, se consideran como no logrados por su ausencia en la herramienta final.

Capítulo 6

Conclusión

Este trabajo surgió por un problema que tiene la comunidad y una necesidad que se vio especialmente en gente que no tiene conocimientos de comercio. Sin duda en el camino se descubrieron nuevos problemas y se tomaron decisiones que parecían correctas. Describiremos lo logrado durante el proyecto (Sección 6.1), luego lo aprendido durante todo el semestre (Sección 6.2), y por último, lo que se espera a futuro de la herramienta desarrollada (Sección 6.3).

6.1. Logros

En el transcurso del proyecto se experimentó lo que es el trabajo pensando en la ayuda a la sociedad, lo que personalmente encuentro muy atractivo y positivo, pues se está acostumbrado a siempre trabajar por beneficio personal. En lo que son los logros del proyecto, se pudo solucionar parte de lo planteado en un principio, pues esta aplicación si brinda una nueva forma de organización para los emprendedores y emprendedoras, pero no para el público objetivo.

Los objetivos de *Easy - Negocios* eran satisfacer las aristas básicas de un negocio con una sola herramienta, para disminuir el tiempo de gestión de los negocios. La validación realizada sobre la aplicación obtenida demuestra que los emprendedores y emprendedoras agradecen y valoran el trabajo realizado. Si bien no se logró implementar todas las *features* diseñadas, si se terminó una gran parte y se considera como logrado el objetivo principal luego de ver los resultados de la validación.

Explicado todo esto, si se tenía como idea y pilar que la aplicación sea intuitiva e inclusiva, pero los sujetos de prueba fueron, en su mayoría, gente joven que tiene un acercamiento 'natural' con la tecnología. Se hubiese agradecido obtener validación de gente de la tercera edad, con nula educación financiera, y de gente con discapacidad visual (la herramienta cuenta con desarrollo de lectura de pantalla con definiciones y pistas, mas no se validó).

Por más que se podría fortalecer este punto, los usuarios que validaron se tomaron el tiempo de reconocer y destacar el real potencial que tiene *Easy - Negocios*, viendo en esta

herramienta una solución para muchas personas que actualmente no tienen el tiempo o el conocimiento de administrar un negocio. Si bien no se implementó todo lo diseñado, se puede decir que lo que si se logró implementar, tiene una buena dirección y una utilidad real.

6.2. Aprendizajes

En cuanto a lo que se aprendió durante el proceso, se separará por categorías y explicará cada una por separado. Primero está lo que se aprendió de Software y tecnologías, que es muy valorado a nivel personal y a nivel laboral, luego está lo aprendido de economía y finanzas, especialmente con el tema de boletas, facturas y esos tópicos más administrativos y por último se aprendió de las personas que ayudaron a validar el proyecto, entendiendo que cada necesidad es diferente.

En lo que fue el desarrollo y trabajo de código, se logró desarrollar una herramienta entera y funcional con tecnologías altamente utilizadas hoy en día en la industria y sin conocimientos previos. Todo aprendizaje técnico siempre es bienvenido y valorado, haber logrado este proyecto no solo demuestra la flexibilidad fomentada en el departamento de computación, sino que entrega motivación para seguir desarrollando y aprendiendo.

Fuera de lo técnico, se logró trabajar en algo enfocado a la comunidad y obteniendo validaciones de personas y potenciales usuarios reales. El trabajo con usuarios, reconocimiento de requisitos, cumplir plazos y obtener feedback son puntos claves en todo proyecto que tenga una interfaz pensada para que cualquier usuario pueda utilizarla de manera intuitiva.

Lo otro que se logró aprender y es muy valorado y necesario es todo lo que incluye el proceso de migración de la aplicación. Herramientas *DevOps* como *Docker*, *GCP*, subir aplicaciones a las *Stores* de los sistemas operativos más utilizados a nivel mundial y poder conectar todo esto para que funcione en cualquier dispositivo y no solamente de manera local es algo que vale la pena resaltar y nombrar como aprendizaje valioso.

Al ser una herramienta catalogada como de finanzas o negocios, también se tuvo que aprender de estos temas. Para poder incluir las opciones de boleta y factura, entender como funciona la retención y devolución de impuestos a nivel nacional, la encapsulación de los negocios y la importancia de poder distinguir clientes, manejar stock, y que cada arista mencionada cumple un papel fundamental en la administración y gestión de un negocio.

En cuanto al trabajo con usuarios, se aprendió que cada emprendimiento es único y sigue sus necesidades. No se puede cumplir los requisitos de todo, pues algunos piden cosas contrarias entre ellas, pero si se puede llegar a un punto común que sirva de núcleo para cualquier emprendedor o emprendedora. También hay que tener en cuenta que cuando se trabaja pensando en un público objetivo o amplio, es necesario reunir sujetos de prueba que abarquen la totalidad del rango, pues en el caso de *Easy - Negocios* faltaron opiniones de gente mayor o con discapacidades.

Por último, se aprendió de los errores. Desarrollar una aplicación que tiene como parte de su público objetivo la inclusión debe tener si o si validaciones de este sector. Un mal cálculo

de los plazos al momento de distribuir tareas que parecían muy simples, pero que significaron bastantes problemas, atrasando un poco todo el desarrollo y generando un efecto de bola de nieve.

El desarrollo de un *PMV* muy extenso, con todas las funcionalidades, no permitió obtener feedback a tiempo e ir validando el desarrollo en paralelo a este mismo. Este error, sin duda, fue del que más se aprendió, pues se cree que si uno va avanzando en la herramienta al mismo tiempo que potenciales usuarios dan su opinión, estos sirven de guía para crear una herramienta que realmente les acomoda y satisface sus necesidades.

6.3. Proyección a Futuro

De ahora en adelante esta herramienta quedará a disposición de quién le vea una utilidad y quiera aportar a la comunidad, se hablará de la licencia elegida (Subsección 6.3.1), se darán ideas de como ampliar la herramienta (Subsección 6.3.2), algunas de para las cuales se diseñó desde un principio y otras que surgieron en el camino y, por último, se hablará de posibles formas para continuar el desarrollo de manera recreativa (Subsección 6.3.3).

6.3.1. Elección de Licencia

Al haber elegido una licencia GNU GPLv3, se le permite a cualquier miembro de la comunidad utilizar el código para lo que estime conveniente, dando la posibilidad hasta de monetizar a costa de este. Sin embargo, se debe compartir siempre las modificaciones y tenerlas abiertas a la comunidad. Al ser una herramienta que puede ayudar a la gente, se espera que el crecimiento de esta siempre teniendo en cuenta la buena voluntad del desarrollador o desarrolladora.

6.3.2. Mejoras

Tanto en el diseño como en el proceso de implementación de la herramienta y la validación, llegaron ciertas ideas para expandir la herramienta de manera útil, que cubren necesidades de los usuarios y tienen distintos niveles de dificultad. Los niveles de dificultad son con respecto a lo que se logró aprender desde cero durante un semestre.

1. Manejo de Redes Sociales: Una de las aristas mencionadas en un comienzo. Sin duda esta tarea es extensa en términos de desarrollo, pero muy útil para los usuarios. Poder unificar las redes disminuiría bastante el tiempo de gestión y ayudaría a una difusión más eficiente y consistente, asegurándose de enviar a cada parte la misma información. Vale decir que se estudió la factibilidad general de este punto al momento de diseñar. **Dificultad técnica: Alta**.
2. Centro de Notificaciones de Pago: Otra de las postuladas en un principio y esta surgió de una potencial usuaria. Poder tener un lugar en el que llegan notificaciones cuando

un cliente paga para poder llevar un correcto manejo de cuentas y saber cuando cobrar. Lo que se podría hacer, pensando de manera ambiciosa, es utilizar la misma aplicación como central de pago (como funciona Mercado Libre o Amazon), dando la opción de pagar en algún link con, por ejemplo, *WebPay*, y así recibir instantáneamente la información de pago de los compradores. **Dificultad técnica: Alta**.

3. Envío de notificaciones para pedidos: Trabajo menor en dificultad técnica. Poder configurar un tiempo de aviso para los pedidos con tal de que no pasen desapercibidos por el negocio y no se comprometa la reputación de este. **Dificultad técnica: Baja**.

4. Vinculación de negocios: Idea ambiciosa, pero se podría hacer una red de negocios registrados en *Easy - Negocios* y dar la posibilidad de contactarse con otro negocio de interés. Por ejemplo, sin algún negocio se trata de pastelería y alguno cerca tiene como pilar los frutos secos, se podrían comunicar para potenciarse mutuamente. **Dificultad técnica: Media - Alta**.

5. Negocios con múltiples administradores: Muchos negocios y emprendimientos no nacen de una persona, sino que de un grupo. Dar la posibilidad de que a un mismo proyecto registrado en *Easy - Negocios* se le asigne más de un administrador podría ayudar a estos casos a repartir los trabajos asignados de cada uno. También ayudaría en la transparencia entre socios y en la comunicación y coordinación interna. **Dificultad técnica: Media - Baja**.

6. Modo offline: Parte de la inclusividad es poder funcionar en zonas con poca conectividad, por lo que una modalidad offline aportaría mucho en este punto. Si bien sería imposible respaldar a tiempo real con la nube, si se podrían utilizar momentos en que el usuario tiene acceso a internet para sincronizar los datos y minimizar la perdida de información en caso de, por ejemplo, perder el dispositivo. **Dificultad técnica: Alta**.

7. Materia prima: La pantalla de manejo de stock se podría incluir a la de Productos, y reemplazarla por una pantalla de *Materia Prima*. Esta funcionaría muy parecida a la de productos, pero ayudaría mucho al usuario a ordenarse internamente y separar esas categorías. **Dificultad técnica: Baja**.

8. Actualización automática de stock: Un comentario repetido dentro de las encuestas de validación. Esta acción es clave para poder cubrir el punto de manejo de stock en su totalidad. Si bien al momento de diseñarlo esta era la idea (por lo que la base de datos lo tiene pensado), no se logró por orden de prioridades. La idea es que al hacer una venta, un pedido o una compra, el stock de los productos se actualice de manera automática. **Dificultad técnica: Baja**.

9. Agregar múltiples ventas o pedidos a un mismo objeto: También era la idea desde un comienzo, fue la razón por la que las tablas de pedidos y ventas están unidas mediante una tabla intermedia a los proyectos, pero tampoco se logró por priorizar otras cosas. Esta funcionalidad está diseñada y no debiese requerir un cambio mayor. **Dificultad técnica: Baja**.

10. Aplicar otros comentarios de la validación: Dentro de la encuesta vemos distintos requisitos. Modificar la interfaz para agrandar botones, hacer un tutorial, generar gráficos para el flujo de caja, distintos filtros y buscadores, exportar a Excel, facilitar la creación

de facturas, comparar meses, entre otros. En resumen, las posibilidades y caminos son muchos. **Dificultad técnica: Variable**.

6.3.3. Opciones de desarrollo futuro

Para poder desarrollar estas mejoras, junto con la elección de licencia que ayuda a fomentar un trabajo comunitario, y el desarrollo que personalmente pueda seguir por cuenta personal en el mismo repositorio público, se puede contar con la ayuda de algún estudiante cursando el trabajo de título.

Poder coguiar en este proceso sería muy bueno tanto para aprender como para transmitir parte del conocimiento obtenido durante el desarrollo y ver como crece la herramienta, esperando que algún día logre llegar a cumplir el rol en la sociedad que se esperó desde un principio. La persona que quiera continuar con el desarrollo de este proyecto debe ser alguien comprometido con la causa social, cuyo beneficio a obtener del desarrollo sea la satisfacción de ayudar.

Ojalá una persona emprendedora, que conozca de primera mano los problemas con los que cuenta un negocio. Técnicamente no es necesario que tenga experiencia, pero si ganas de aprender y de trabajar sobre algo que ya está desarrollado, pues los desafíos que se le vendrían no son menores. Podría tener la posibilidad de implementar las posibles mejoras mencionadas o navegar por su cuenta la herramienta y estudiar, ya sea por su cuenta o apoyado y validado, que cosas podrían mejorar dentro de *Easy - Negocios*

Con una mirada en la comunidad, el trabajo colaborativo, quizá hasta conexiones internacionales para trabajar en la traducción de la herramienta y la ampliación del público objetivo, se pueden lograr grandes cosas. *Easy - Negocios* podría ser la herramienta social que ayuda a miles de personas a surgir y salir adelante. Personalidades inquietas en busca de más, pero sin los conocimientos para escalar de manera correcta.

Bibliografía

[1] MongoDB Developer David Pawlan. Relational vs. non-relational database: Pros & cons. MongoDB relationa vs Non Relational Webpage. Accessed: 2022-06-22.

[2] Facebook. Information about facebook business suite. Facebook Business Help. Accessed: 2022-04-12.

[3] Facebook. React native. React Native Webpage. Accessed: 2022-06-22.

[4] Fireart. React native vs flutter: Which one is better for 2022? Fireart discussion Webpage. Accessed: 2022-06-26.

[5] Manoj Kalla. Overview and functionality of npm (node package manager). NPM functionalities Webpage. Accessed: 2022-06-25.

[6] Empresa Kellun. Kellun. Kellun About Us Webpage. Accessed: 2022-04-12.

[7] Empresa Balloon Latam. Balloon latam. Balloon Latam About Us Webpage. Accessed: 2022-04-12.

[8] Learn24bd. Simple stock manager. SimpleStockManager in Play Store. Accessed: 2022-04-12.

[9] manager of the Threat Intelligence Production Team within IBM Security X-Force Michelle Alvarez. Injection attacks: The least glamorous attack is one of the most threatening. SecurityIntelligence Webpage. Accessed: 2022-06-26.

[10] React Native. Introducing hooks. React Native hooks Webpage. Accessed: 2022-06-26.

[11] Joshua Stowers. 5 major pitfalls that stop entrepreneurs by year 2. BusinessnewsDaily Webpage. Accessed: 2022-04-27.

[12] Senior Tech Consultant at Simform Tejas Kaneriya. Advantages & disadvantages of node.js : Why to use node.js? Simform discussion Webpage. Accessed: 2022-06-25.

[13] La Tercera. El mercado chileno de los celulares: Samsung lidera; apple domina en aparatos premium y huawei en los más baratos. La tercera Webpage. Accessed: 2022-06-22.

[14] Empresario Tomas Pablo Roa. 2020 record de creación de empresas. Eleconomistaamerica news Webpage. Accessed: 2022-04-27.

Anexos

Anexo A. Diseño de la aplicación

En este anexo se puede encontrar el modelo diseñado de la base de datos y la interfaz tentativa de la aplicación. A continuación se encuentra el primer bosquejo de modelo de base de datos.

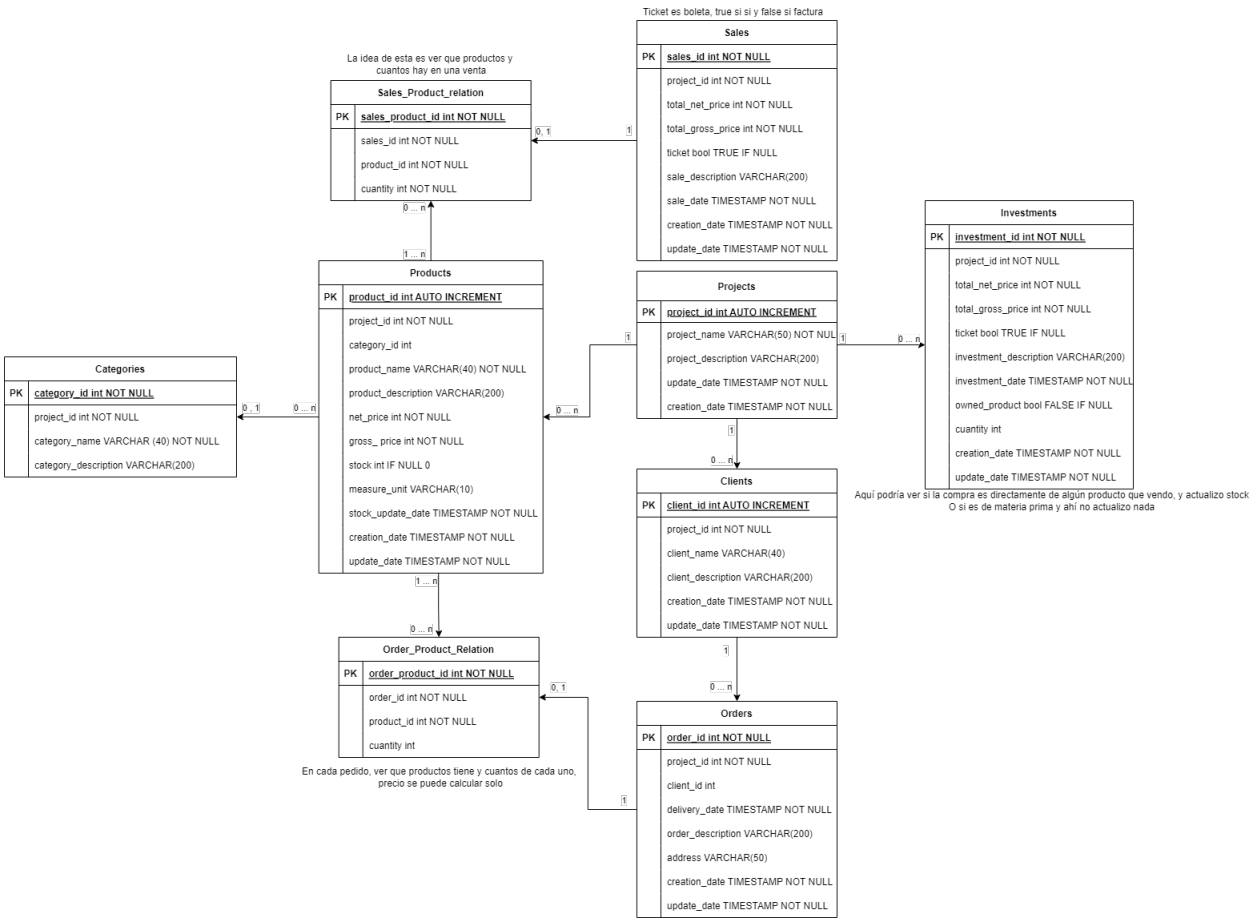

Figura 6.1: Diseño preliminar del Modelo de Datos, se pueden ver las relaciones internas y que todo es obtenible a partir de un ID de proyecto

A continuación se puede encontrar el diseño completo de la aplicación. Este es un wire-

frame hecho en Marvel. Al momento de implementar se fue fiel al diseño inicial, pero este no incluye la pantalla de ingreso.

Figura 6.2: Diseño Vista Inicial

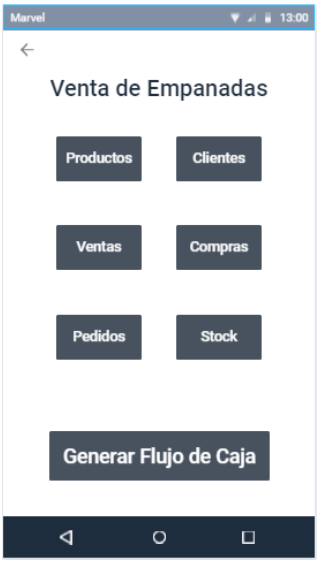

Figura 6.3: Diseño Vista principal

Figura 6.4: Diseño Vista Productos

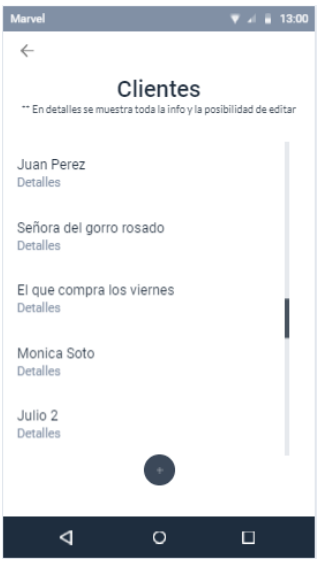

Figura 6.5: Diseño Vista Clientes

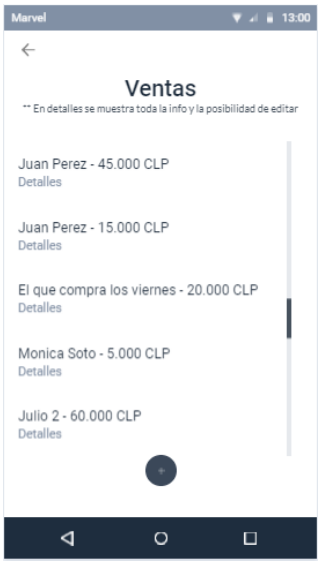

Figura 6.6: Diseño Vista Ventas

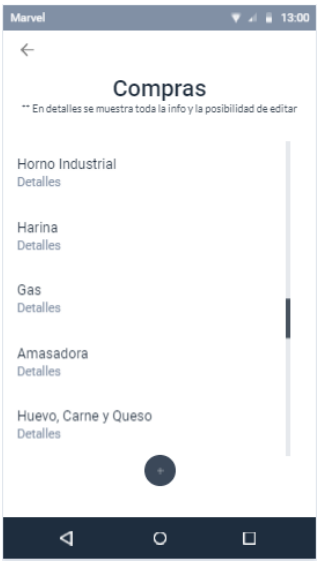

Figura 6.7: Diseño Vista Compras

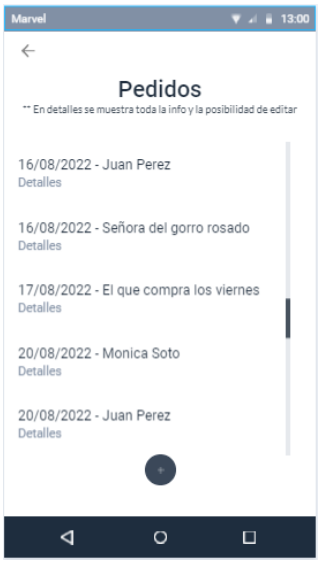

Figura 6.8: Diseño Vista Pedidos

Figura 6.9: Diseño Vista Stock

Figura 6.10: Diseño Vista Flujo de Caja

Anexo B. Documentación de la Herramienta

En este anexo se encuentra el modelo final de la base de datos y la documentación realizada por la librería Swagger mencionada en 4.1.3.

El modelo definitivo e implementado se ve de la siguiente manera:

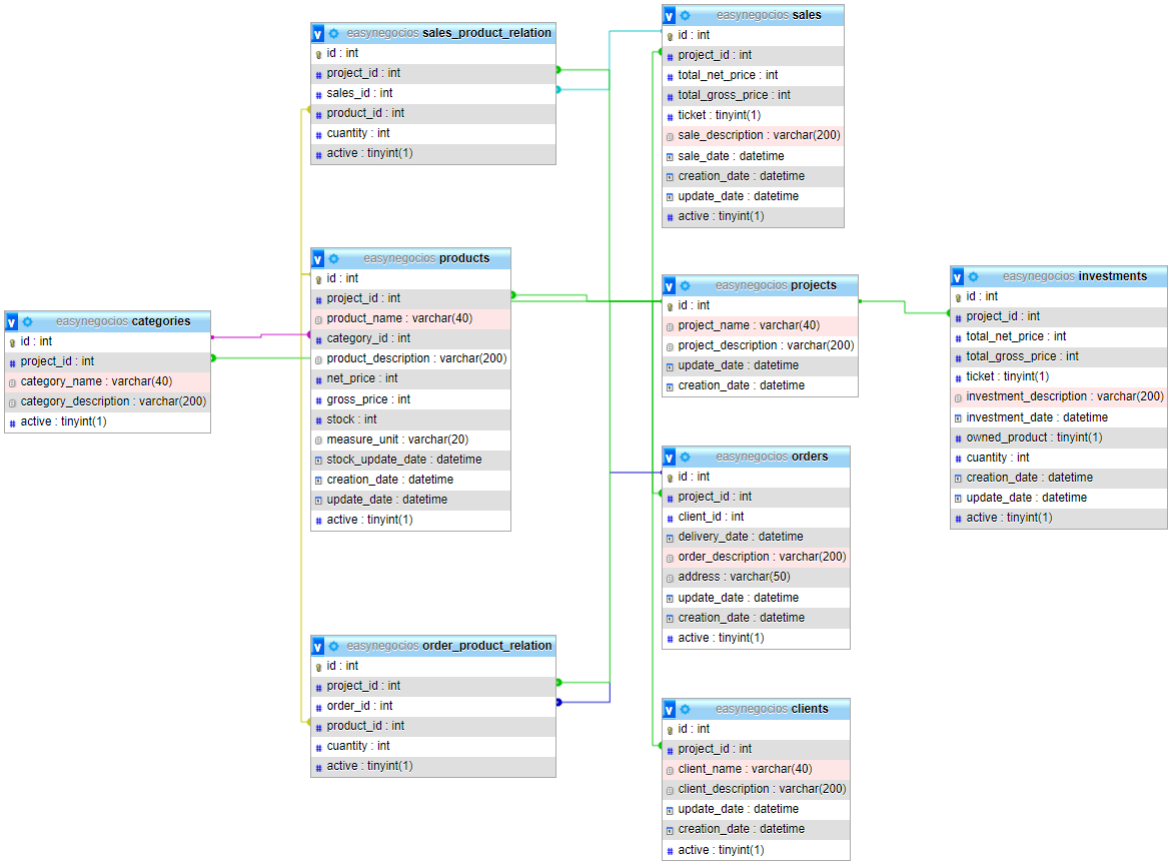

Figura 6.11: Forma final del Modelo de Datos

Esta documentación contiene todos los endpoints con los que cuenta la API de *Easy - Negocios*, incluyendo los parámetros por URL, body, método de consulta y una descripción de cada uno. En 6.12 se puede ver parte de la vista general, y en todas las siguientes se ve en detalle como se define cada endpoint.

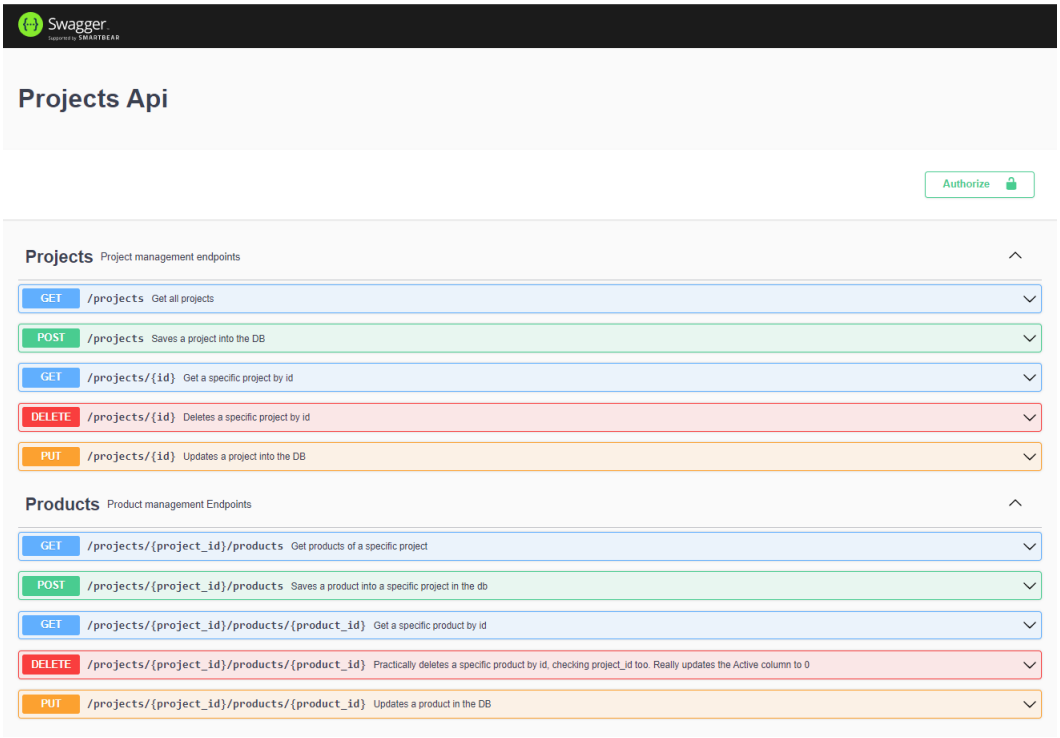

Figura 6.12: Fragmento documentación realizada con el módulo Swagger.

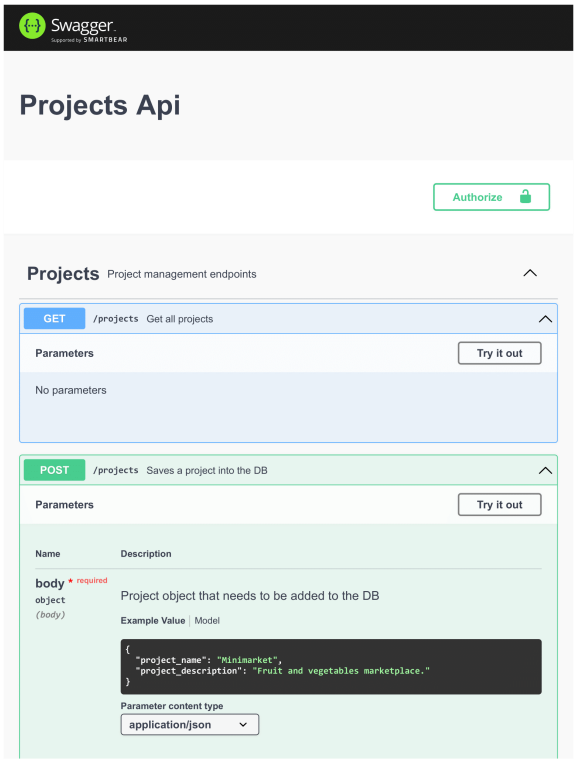

Figura 6.13: Detalles de la consulta 'GET' y 'POST' de proyectos, especial énfasis en lo que pide el *request body*.

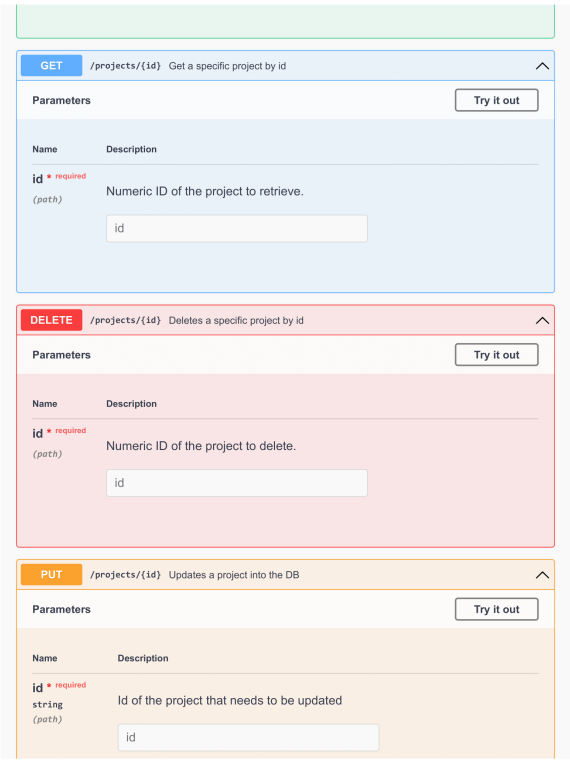

Figura 6.14: Detalles de 'GET' individual, 'DELETE' y 'PUT' de proyectos.

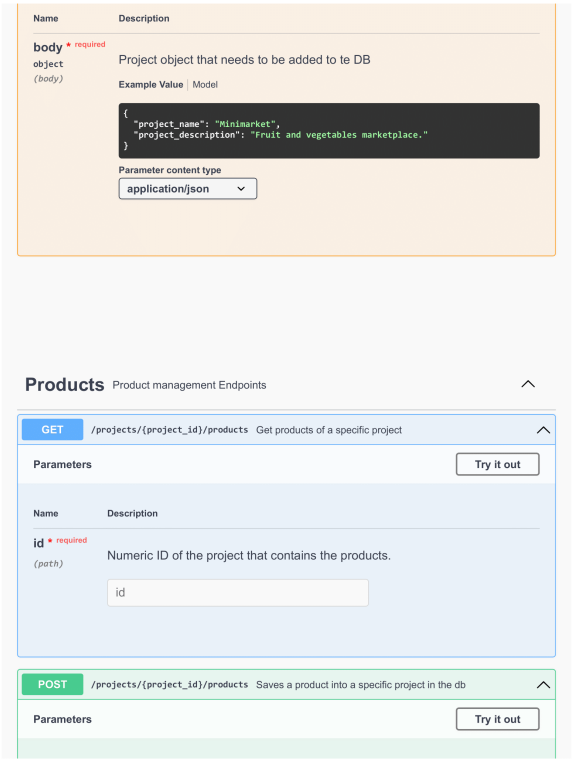

Figura 6.15: *Request body* para el 'PUT' de proyectos y detalles del 'GET' de productos.

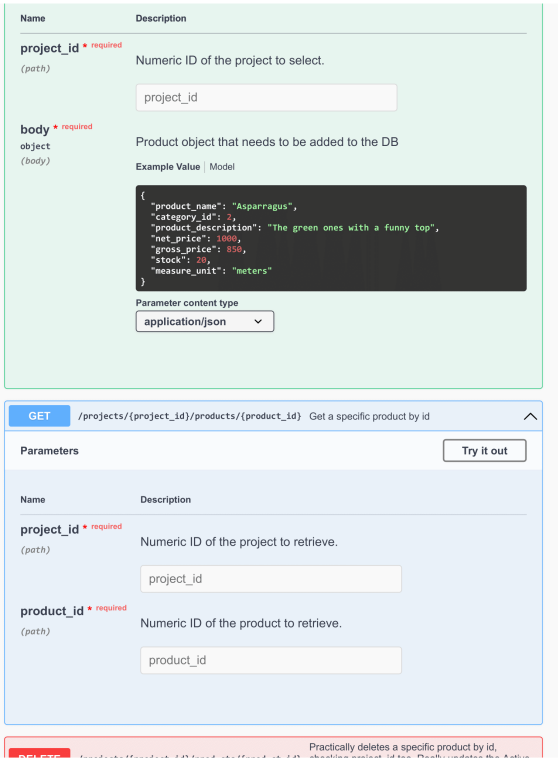

Figura 6.16: Énfasis en el *request body* para agregar productos y detalles del 'GET' de productos individuales.

Figura 6.17: Consultas 'GET' y 'PUT' de productos.

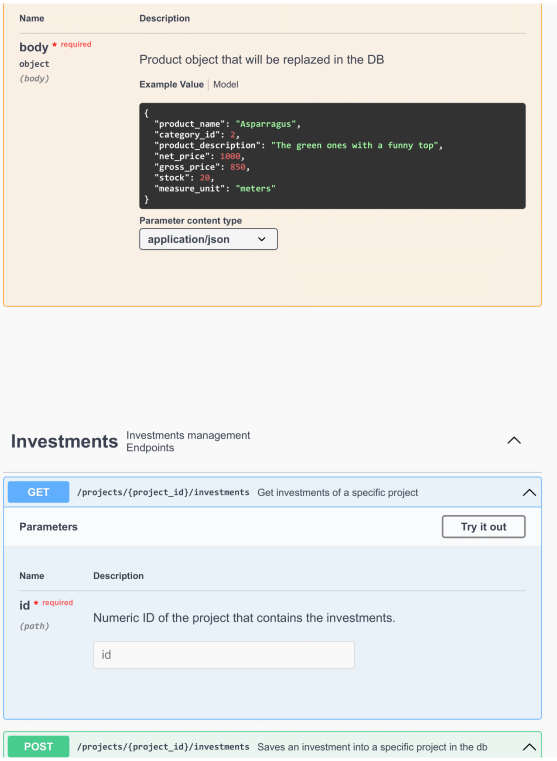

Figura 6.18: Detalles del *request body* para el 'PUT' de productos y 'GET' de compras.

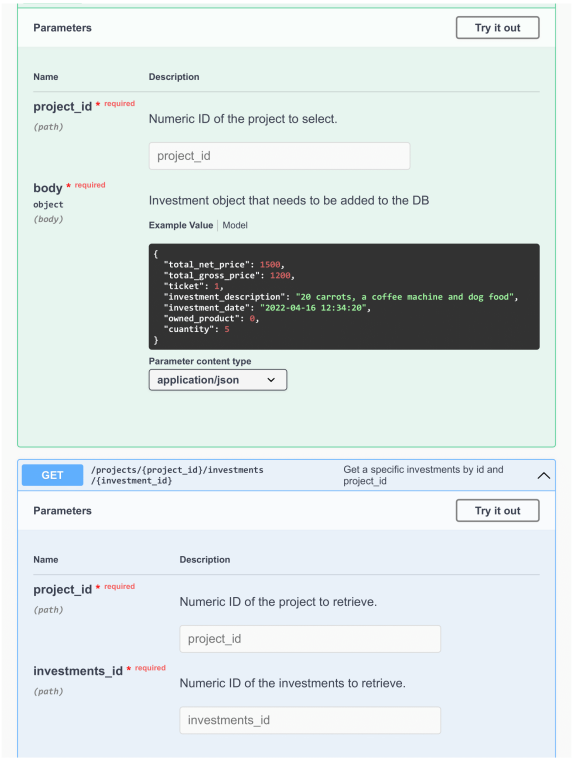

Figura 6.19: Consulta 'POST' de compras con su *request body* y 'GET' individual de la misma tabla.

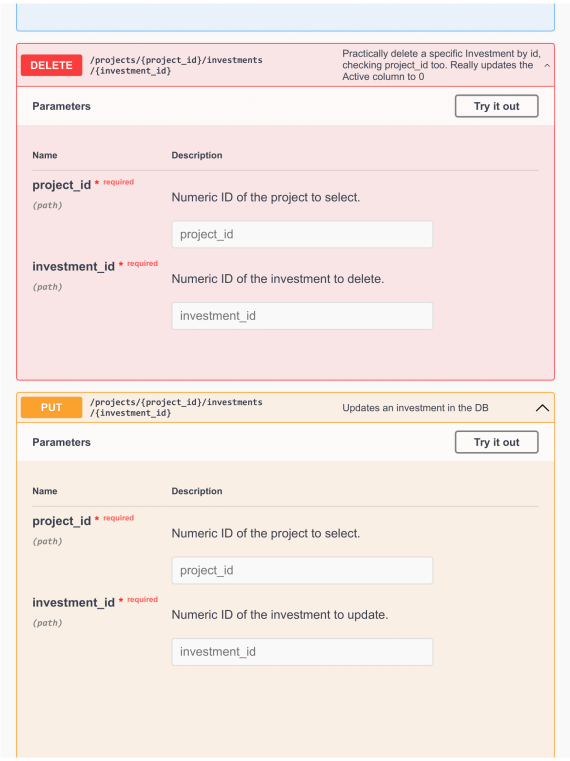

Figura 6.20: Detalles del 'DELETE' y del 'PUT' de compras.

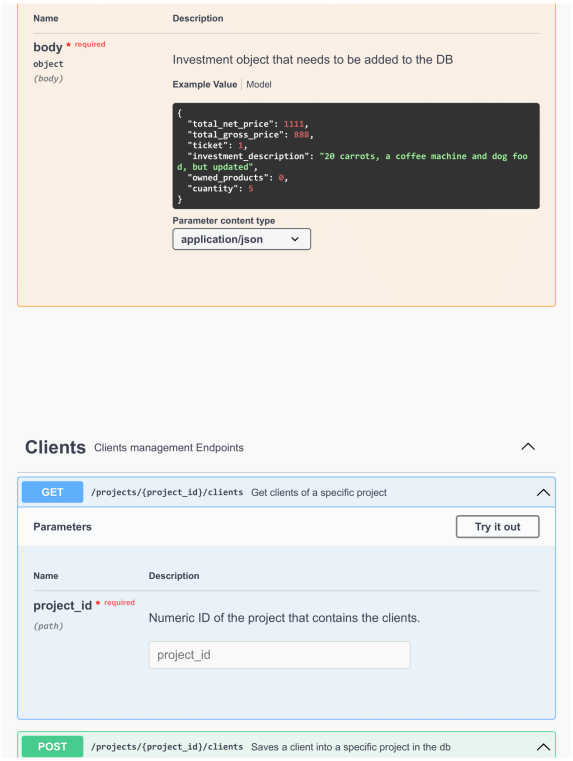

Figura 6.21: Detalles del *request body* de la consulta 'PUT' de compras y el 'GET' de clientes.

Figura 6.22: Consulta 'POST' de clientes con su *request body* y el 'GET' individual de clientes.

Figura 6.23: 'DELETE' y 'PUT' de clientes.

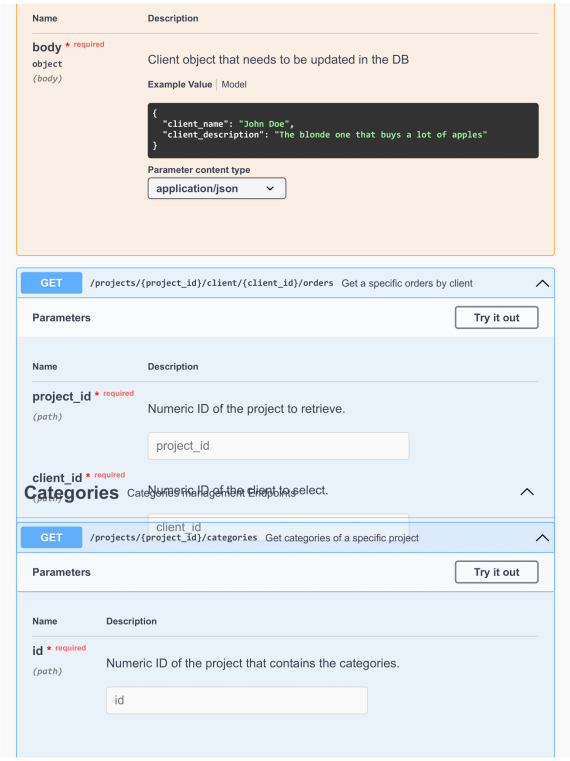

Figura 6.24: Detalles del *request body* del 'PUT' de clientes. Se ve también el 'GET' individual de estos y el 'GET' general de categorías.

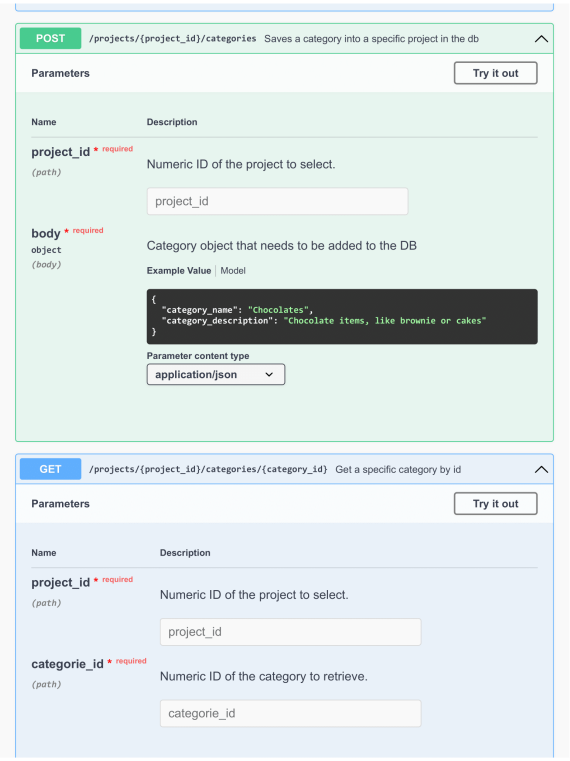

Figura 6.25: 'POST' de categorías con su *request body* y 'GET' individual.

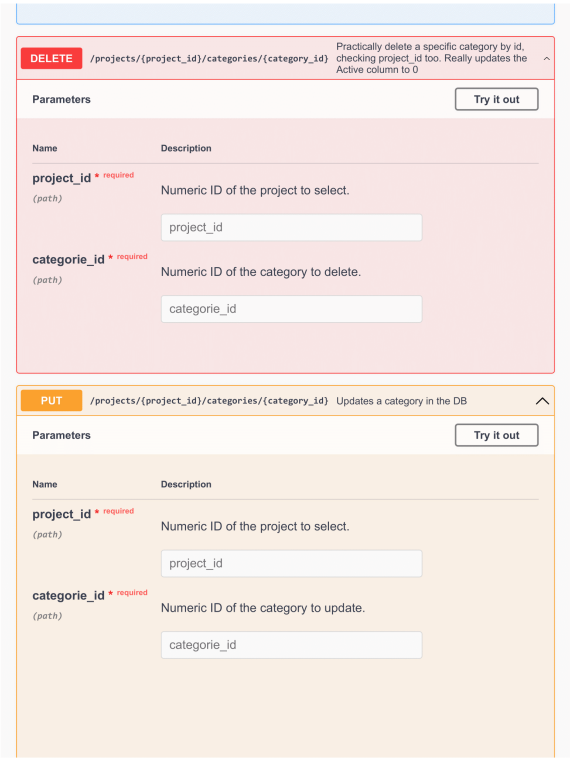

Figura 6.26: 'DELETE' y 'PUT' de la tabla categorías.

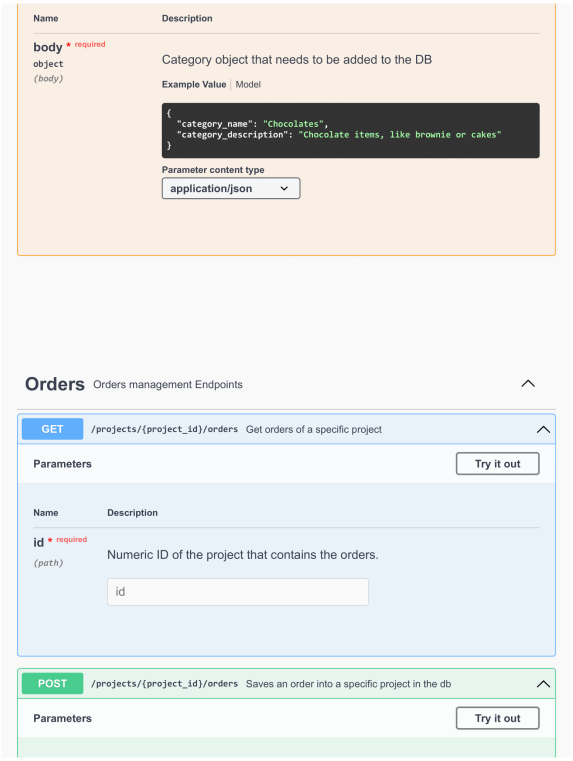

Figura 6.27: Énfasis en el *request body* del 'PUT' de categorías y el 'GET' general de pedidos.

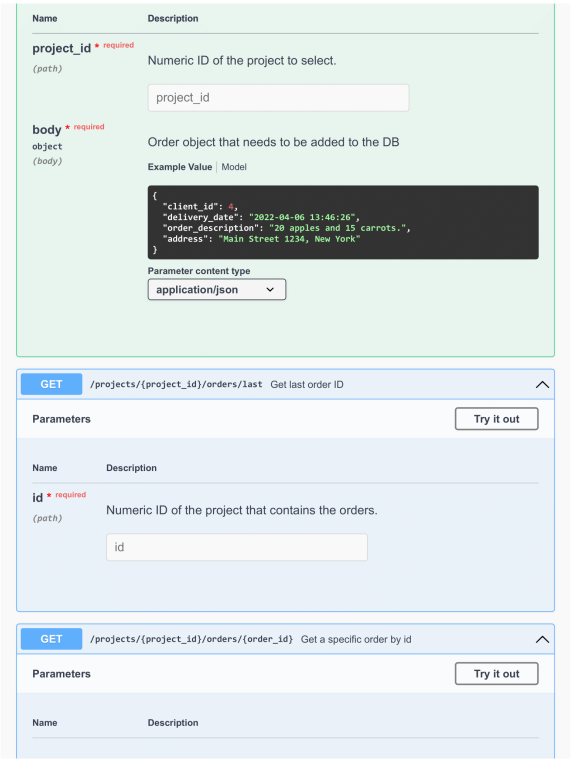

Figura 6.28: *Request body* para agregar pedidos y distintos 'GET' que tiene esta tabla.

Figura 6.29: Detalles de otro 'GET', 'DELETE' y 'PUT' de la tabla de pedidos.

Figura 6.30: Detalles del *request body* para actualizar pedidos.

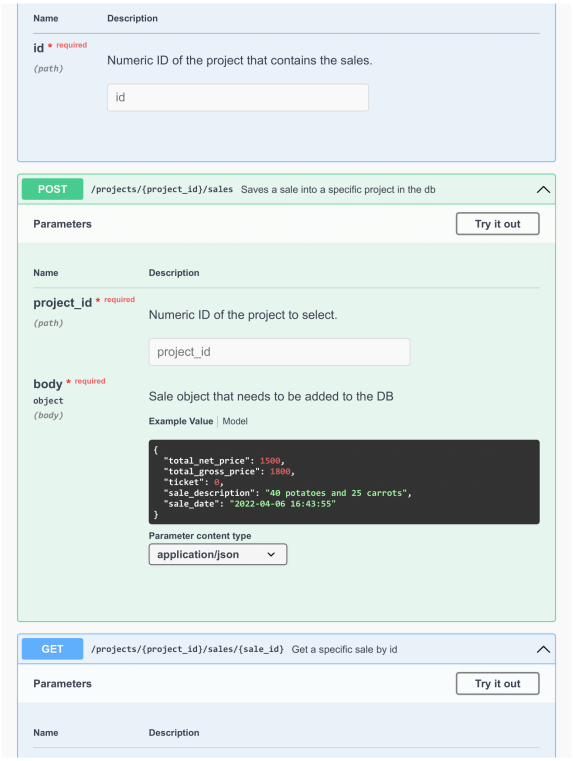

Figura 6.31: Distintos 'GET' y el 'POST' con su *request body* para ventas.

Figura 6.32: Consultas para obtener, eliminar y actualizar ventas.

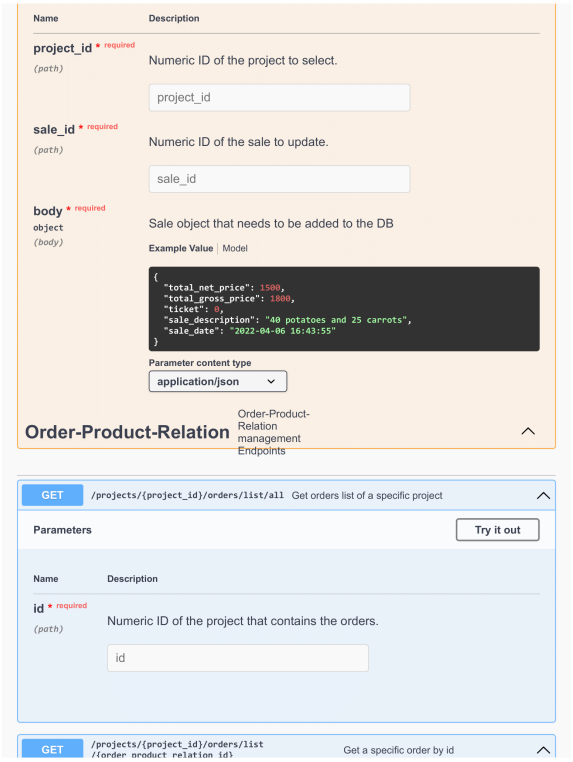

Figura 6.33: Detalles del *request body* para actualizar las ventas.

Figura 6.34: Consultas para administrar las relaciones pedido-producto de los proyectos.

Figura 6.35: Consultas 'POST' y 'PUT' con sus *request body* para las relaciones pedido-producto.

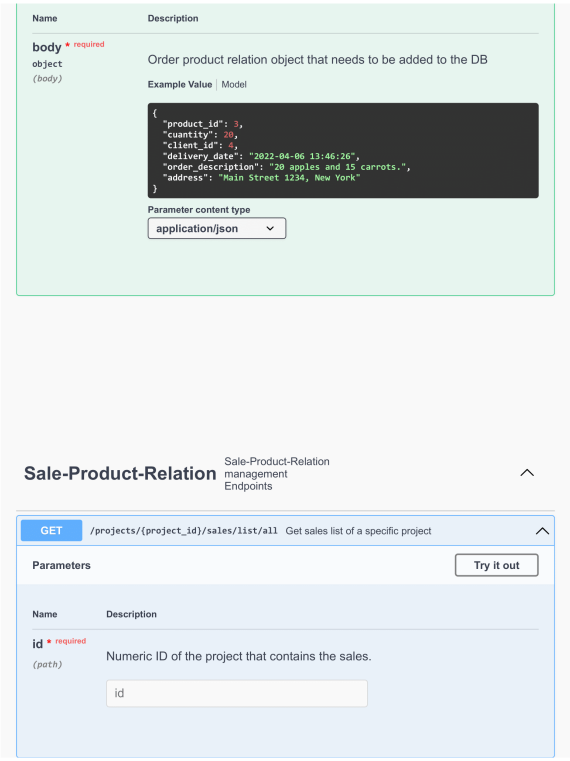

Figura 6.36: Detalles del *request body* de 'POST' para las relaciones pedido-producto. También se ve el 'GET' de las relaciones venta-producto.

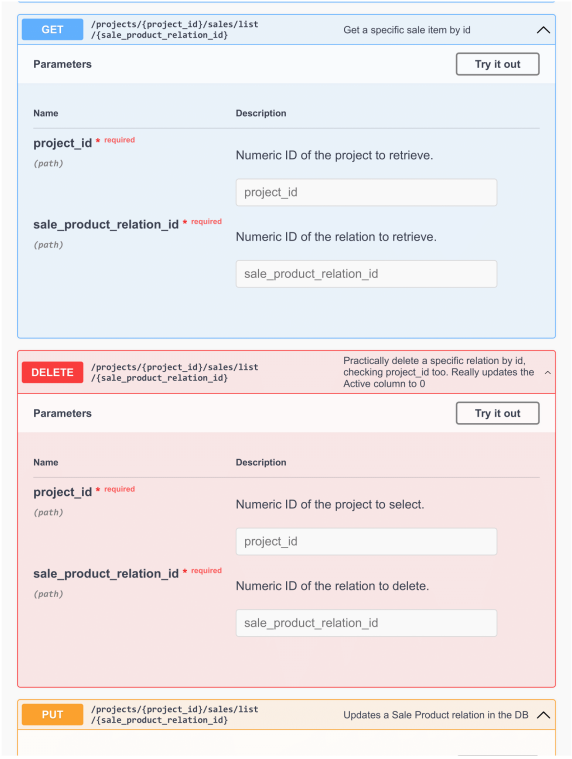

Figura 6.37: Consultas 'GET' y 'DELETE' de veta-producto para el proyecto.

Figura 6.38: Detalles del 'PUT' para venta-productos.

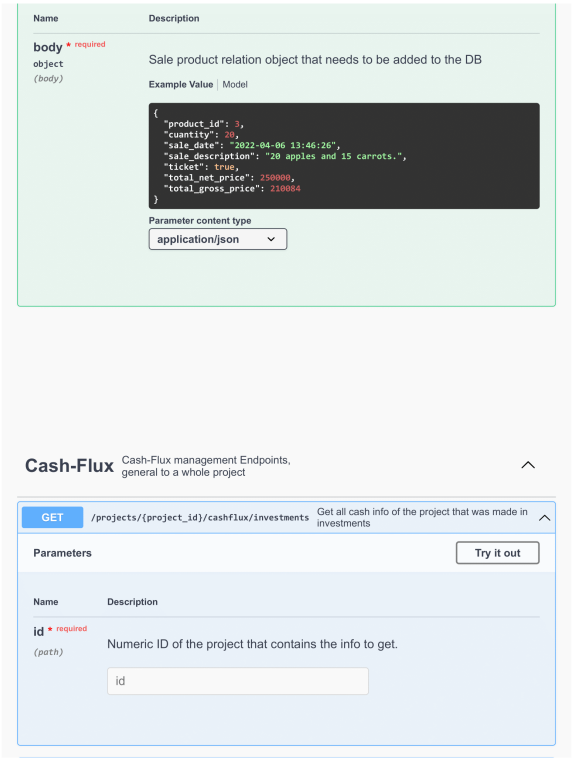

Figura 6.39: Detalles del *request body* del 'POST' para venta-productos.

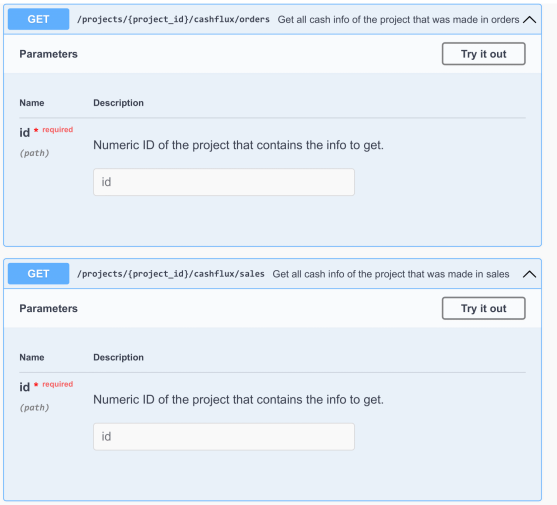

Figura 6.40: Distintos 'GET' que tiene la relación venta-producto.

Anexo C. Implementación de *Easy - Negocios*

En este anexo se puede ver la forma final de la aplicación. Esta es la vista de la implementación y la definitiva. Se puede ver que se siguió bastante el modelo diseñado, manteniendo su estructura y teniendo como foco la facilidad de uso. Se puede ver cada pantalla con su formulario de agregar información cuando es necesario.

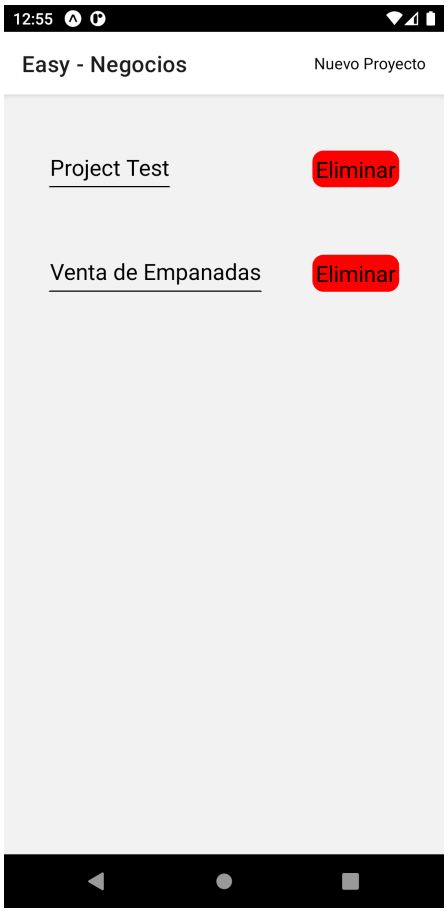

Figura 6.41: Vista Inicial *Easy - Negocios*

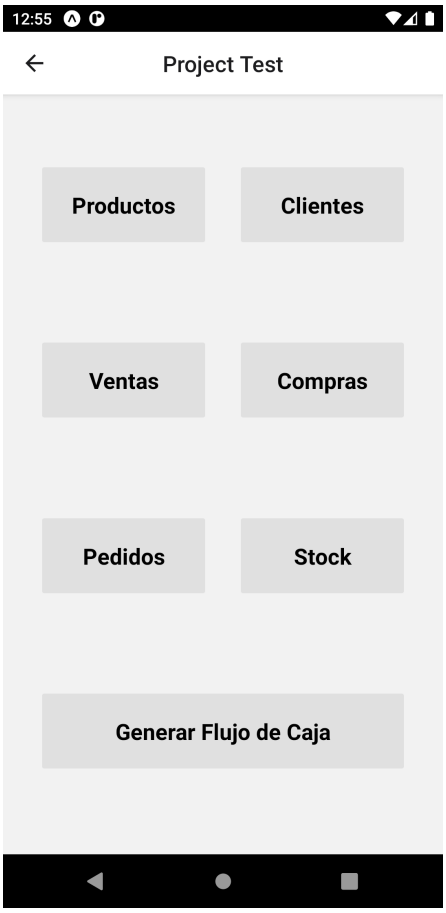

Figura 6.42: Vista Proyecto *Easy - Negocios*

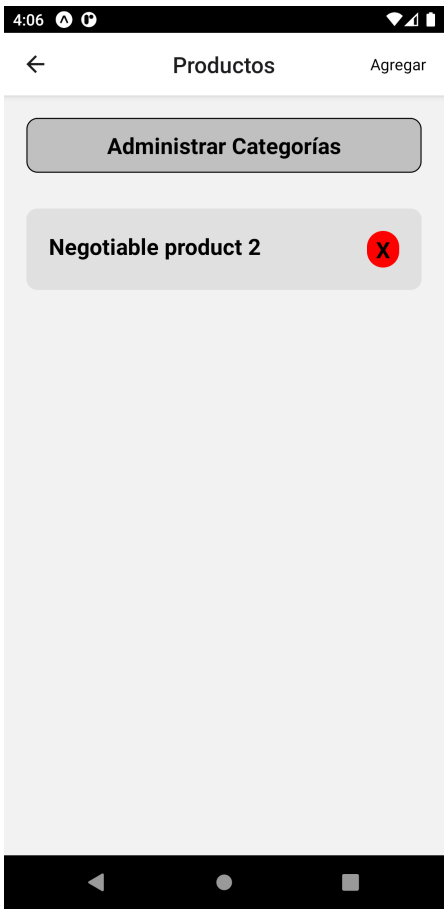

Figura 6.43: Vista Productos

Figura 6.44: Formulario Productos

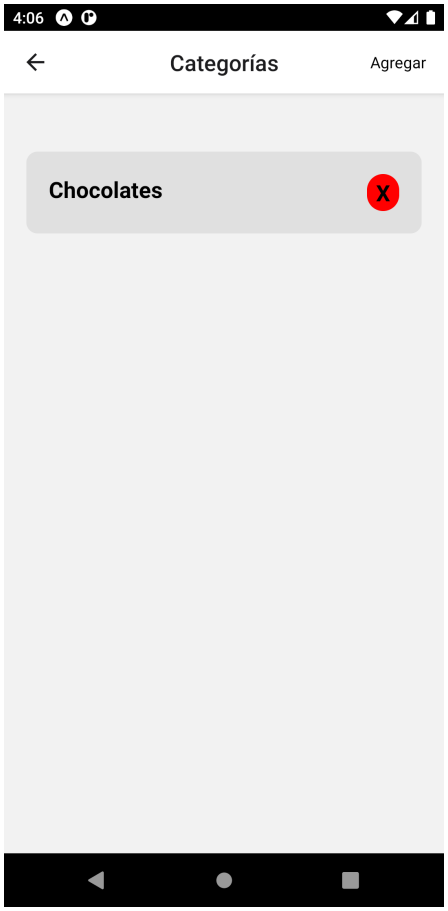

Figura 6.45: Lista de Categorías

Figura 6.46: Formulario Categorías

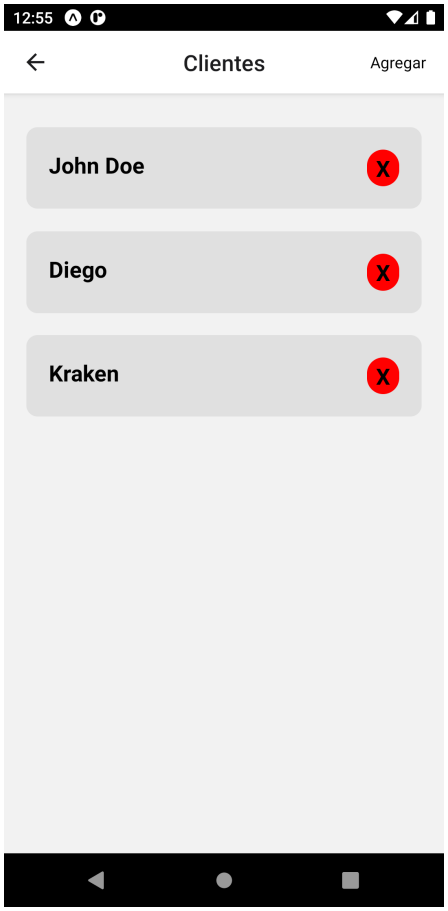

Figura 6.47: Vista Clientes

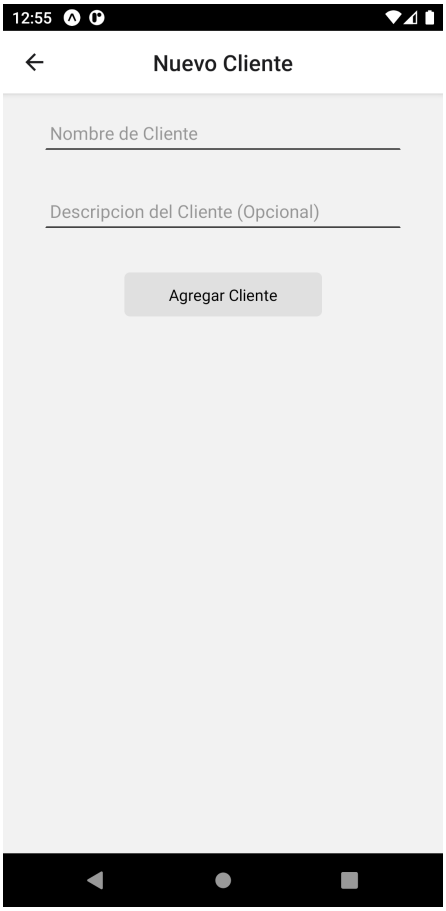

Figura 6.48: Formulario Clientes

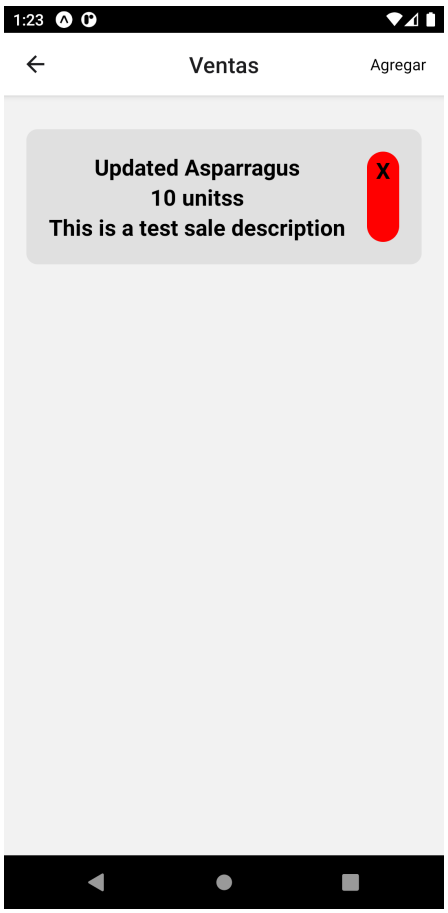

Figura 6.49: Vista Ventas

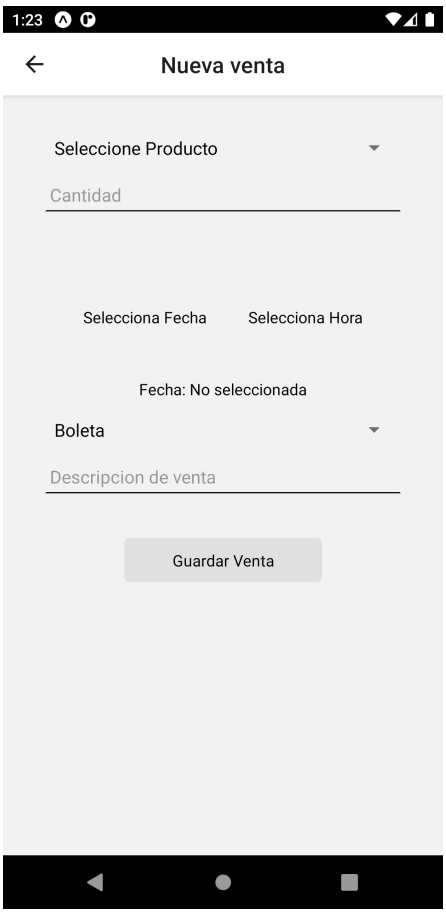

Figura 6.50: Formulario Ventas

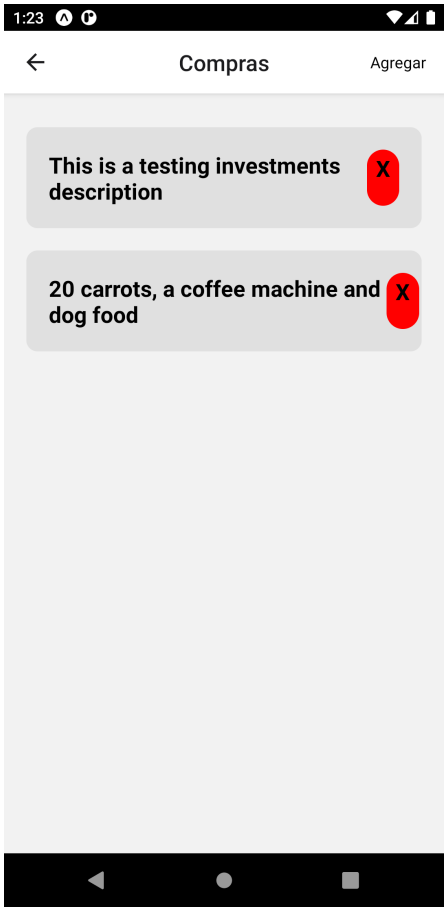

Figura 6.51: Vista Compras

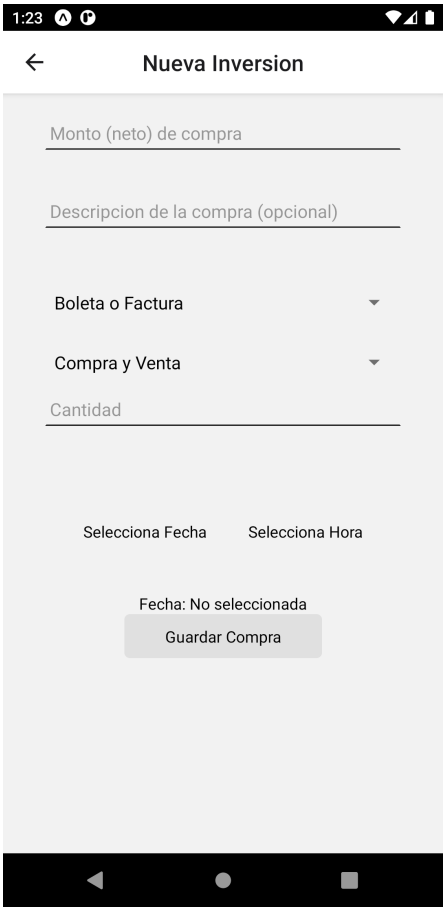

Figura 6.52: Formulario Compras

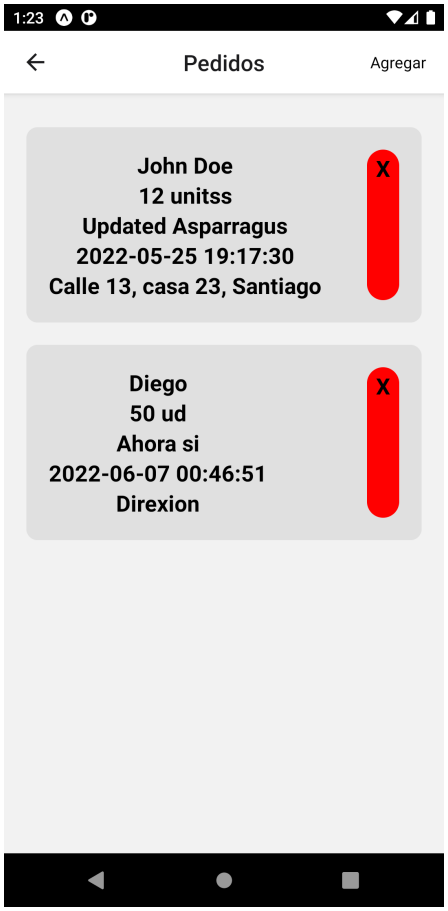

Figura 6.53: Vista Pedidos

Figura 6.54: Formulario Pedidos

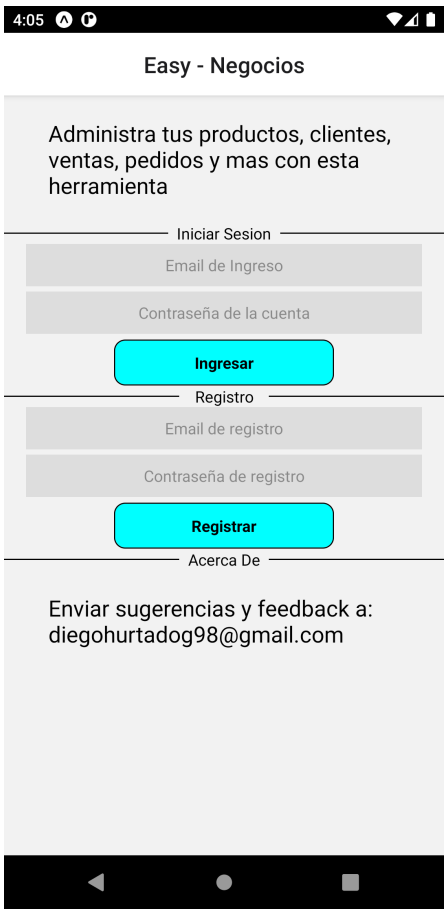

Figura 6.55: Pantalla de Inicio

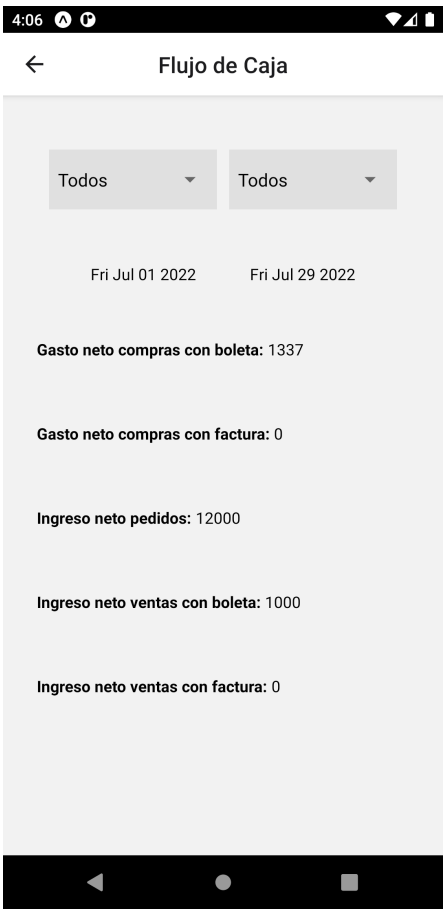

Figura 6.56: Generación flujo de caja

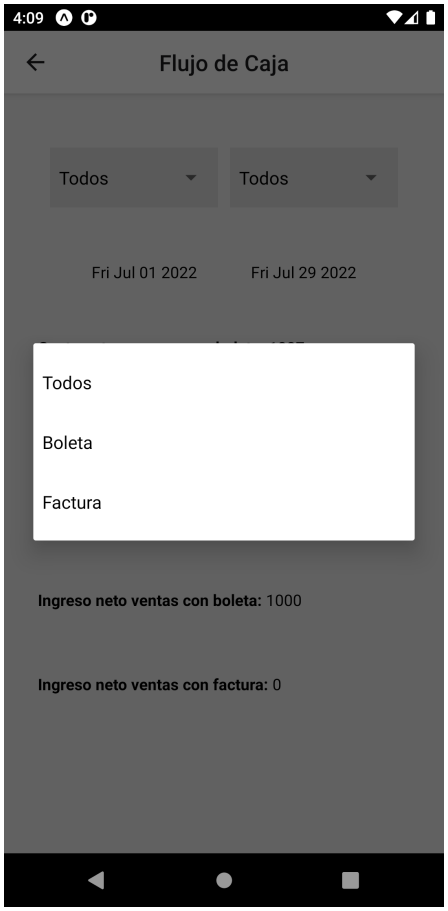

Figura 6.57: Filtro de Generación del Flujo de Caja

Figura 6.58: Filtro de Generación del Flujo de Caja

Figura 6.59: Formulario de actualización de Producto

Figura 6.60: Selección Categoría del Formulario de actualización de Producto

Figura 6.61: Pestaña de confirmar eliminación

Anexo D. Respuestas de la Validación

En este anexo se podrá ver la totalidad de respuestas de la encuesta, para poder ver las distribuciones en las que son con escala y todos los párrafos que se escribieron en las respuestas que tenían este tipo de pregunta. Se omite la primera pregunta por privacidad de quienes colocaron sus nombres personales.

'Según tu primera impresión. ¿Para qué sirve Easy - Negocios?':

- 'Administrar tu negocio / llevar registro del inventario'
- 'Aplicación que centraliza y simplifica la gestión de negocios, principalmente de venta.'
- 'Para llevar las cuentas de mi negocio'
- 'Una suerte de base de datos para el negocio. Manteniendo el stock, pedidos, ventas, etc. Finanzas en general del negocio'
- 'Diría que para tener control de ventas y todo lo relacionado'
- 'Para gestionar datos de proyectos de negocios'
- 'Para poder tener un claro registro del movimiento de mercancía de tanto en unidades como en dinero.'
- 'Administrar stock de productos y ventas'
- 'Para administrar negocios'
- 'Para manejar un negocio'
- 'Para gestionar las diferentes áreas de un negocio'
- 'Gestionar tu negocio'
- 'Para ver los ingresos y egreso de un negocio'
- 'Para administrar el negocio'
- 'Para administrar mi negocio de forma más organizada.'

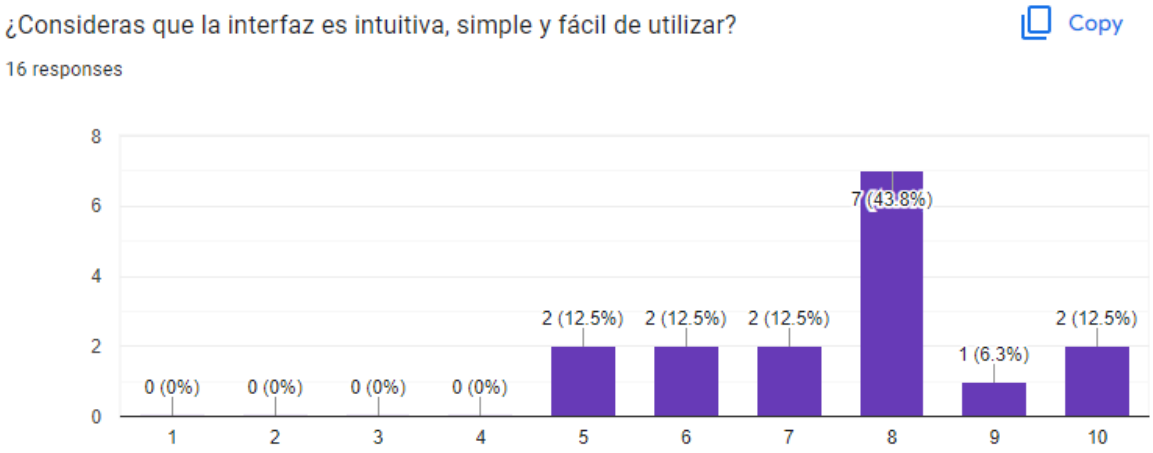

Figura 6.62: Gráfico de barras de la respuesta a la pregunta ¿Consideras que la interfaz es intuitiva, simple y fácil de utilizar?

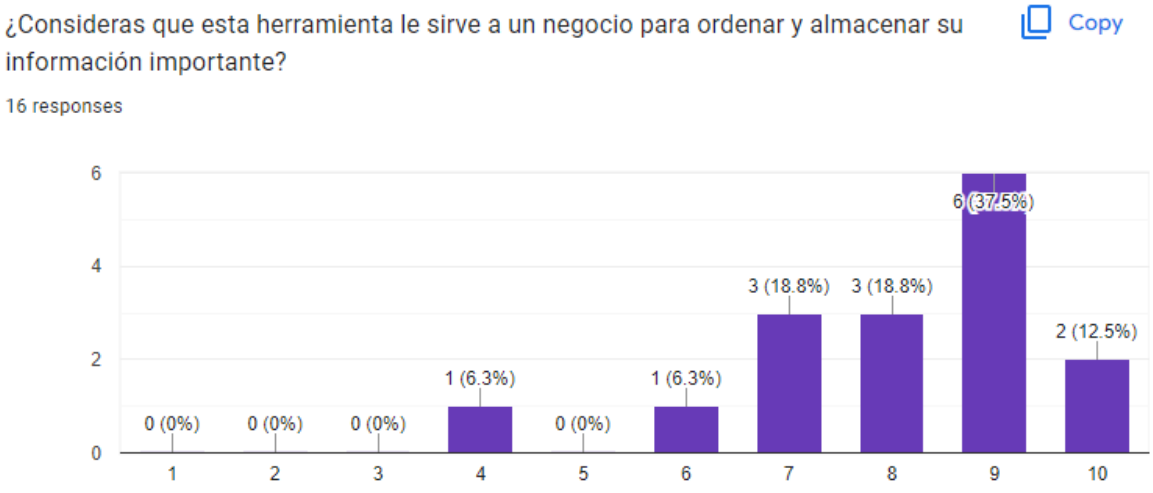

Figura 6.63: Gráfico de barras de la respuesta a la pregunta ¿Consideras que esta herramienta le sirve a un negocio para ordenar y almacenar su información importante?

Figura 6.64: Gráfico de torta de las respuestas a la pregunta ¿Consideras que el correcto uso de esta herramienta podría ahorrar tiempo en la administración de un negocio?

Figura 6.65: Gráfico de torta de las respuestas a la pregunta ¿Consideras que el correcto uso de esta herramienta podría significar un aumento en los ingresos del negocio?

Figura 6.66: Gráfico de torta de las respuestas a la pregunta ¿Te es útil poder generar un flujo de caja por fechas y al instante?

Figura 6.67: Gráfico de torta de las respuestas a la pregunta ¿Crees que Easy - Negocios podría ayudar a la gente a manejar de mejor manera sus negocios?

'Comentarios (que es lo que más o menos te gusta de la herramienta), Feedback, Opiniones, Críticas. (No son necesarias, pero son demasiado bienvenidas)':

- 'El botón de agregar queda muy escondido y teniendo en cuenta que las principales acciones que se deben hacer es agregar cosas, queda poco amigable al flujo de uso'
- 'Me confundí entre ventas y pedidos, pensando que eran para lo mismo. Seguro hay alguna manera de unificar ambos y que sea más explícita su función. Me costó guardar los productos, confundiendo categoría con producto. Luego de jugar un poco lo resolví. Que las ventas/pedidos puedan ser llenadas por los mismos clientes, por ejemplo linkeando a una página web o un formulario, permitiría escalabilidad tanto a la aplicación como a los negocios que la utilicen.

  En términos generales, un gran trabajo y con un potencial de facilitar fuertemente la gestión de diversos negocios. Felicitaciones a los programadores.'
- 'La elección de los colores y diseño están muy simples. Lo otro que no entendí es si es que cuando agrego una venta debería descontarse de lo que hay en inventario, porque me seguía apareciendo lo mismo.'
- 'Quizás trabajar en hacer más amena la interfaz, dado que parte del público objetivo de la app podría no saber cómo utilizar muy bien un (pensando en un grupo etareo de 45-65 años). Siento que la idea es buena, pero faltan detalles que podrían ser muy importantes para el impacto efectivo en el público.'
- 'Seria bueno que al generar una venta se actualice el stock de forma automática, esto simplificaría mucho mas para alguien que quiere saber cuanto debe reabastecer mas que otros puntos, tambien al generar una venta seria util añadir mas elementos distintos en una misma para ahorrar tiempo, asi mismo añadiría un mini tutorial ya que los primeros minutos son algo confusos, como ultimo podria ser seleccionar las funciones a utilizar para optimizar la pantalla (Aunque sean un poco muchas criticas, la verdad la idea de app esta bastante buena y tiene muchos puntos fuertes, como lo es la sencillez de aprender a usarla, su bajo peso y rapidez de funcionamiento)'
- 'Se ve a lo que va y creo que la interfaz no está mal hecha. Añadiría cosas como un buscador (proyectos, categoría, clientess, etc), sincronización entre dispositivos, exportar a excel y proyectos recientes.'
- 'Podría tener un mini tutorial, o algo que explique todo lo que se puede hacer, pq al principio como que me costó ver que había que hacer. Y lo otro, mejorar la gráfica'
- 'Añadir visualización de imagenes de productos, gráficos en ventas, compras y flujos de caja segun períodos.'
- 'El registro primero no me funcionó, al aparecer por poner una contraseña muy corta(?) Porque una más larga funcionó, peor no había feedback. Quizás sería bueno alguna sección que indique para que son las secciones siento que no es muy intuitivo'
- 'Las mayus, mucho botón. En general muy intuitiva y facil de usar.'
- 'Creo q es necesaria una interfaz mas bonita a la vista para disfrutar un poco mas al usar la aplicación. Me gustaría poder ver gráficos de ventas y utilidades para poder

representar visualmente el desempeño de mi emprendimiento. Me gustaría quizás que exista una función que permita generar Invoice o Cotizaciones. Creo q es más util para un emprendimiento que tiene una gran cantidad de productos diferentes más que para uno que cuente con muy pocos (uno en mi caso) Se podrían agregar servicios extras pagados como un Contador que ayude con las cuentas o para pagar impuestos o asesorías para mejorar el rendimiento del negocio'

- 'Hay cosas que no entiendo y mejorario eventualmente el diseño'
- 'Es interesante, sirve para negocios de productos, no tanto para negocios de servicios, donde lo que cuesta no es tanto llevar el flujo de caja.'
- 'Creo que debería ser más rápido el transpaso de información, creo que yo no encontraría el tiempo de rellenar paso por paso'
- 'Me gusta el color que tiene la interfaz. Había un botón como de salir al entrar a la app y creo que estuviera menos a la vista se vería más lindo. También no pude ingresar pedidos porque los productos que había agregado no aparecían, entonces al parecer no se podía completar la acción, pero bueno nada. Está bueno que se guarde el "progreso.[en] la app. Y eso, para finalizar, me recordó a Roblox kskfks onda lit en la forma en que se ven las cosas'