

Desarrollo de la aplicación móvil “ACOLITAPP” -Una app para solicitar favores.

Barreto Siavichay Jorge

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles
Desarrollo de aplicaciones móviles multiplataforma

Eduard Martín Lineros
Carles Garrigues Olivella

02/06/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de la aplicación móvil “ACOLITAPP” - Una app para solicitar favores.</i>
Nombre del autor:	<i>Jorge Barreto Siavichay</i>
Nombre del consultor/a:	Eduard Martín Lineros
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	<i>Máster universitario de Desarrollo de aplicaciones para dispositivos móviles.</i>
Área del Trabajo Final:	<i>Desarrollo de aplicaciones móviles multiplataforma</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Acolitapp, favores.</i>

Resumen del Trabajo

La aplicación “Acolitapp” permitirá que las personas puedan solicitar favores a las demás mediante el dispositivo móvil. El usuario escribirá el favor o ayuda que necesita así como el valor monetario que pagará o la recompensa que dará a la persona que realice el favor solicitado. Cuando se solicite el favor, los demás usuarios podrán revisar y aceptar la solicitud para proceder a realizar un trato con el solicitante. La app también permitirá calificar el nivel de satisfacción tanto de la persona que solicitó el favor como de la persona que lo realizó.

La aplicación a desarrollarse será multiplataforma (tanto para Android como iOS) y podrá ser usada por cualquier persona, además se aplicará técnicas de usabilidad y accesibilidad. La App se desarrollará desde cero y para lograrlo se seguirán las etapas del ciclo de vida de software (análisis y diseño, desarrollo, pruebas, ejecución), mientras que para el diseño de las pantallas se utilizarán herramientas de prototipado, para luego plasmarlo en el desarrollo de la App, utilizando las últimas tecnologías web (Ionic, HTML5, BootStrap, TypeScript, etc.).

Como resultado se obtendrá una versión Beta de la aplicación móvil generada para la plataforma Android, en donde se podrá observar el cumplimiento de los requisitos funcionales y la finalidad de la misma; sin embargo, no se centrará tanto en los requisitos no funcionales como sincronización, seguridad, rendimiento, autenticación, encriptación, entre otros.

Abstract:

The “Acolitapp” application allow people to request favors from others through their mobile advice. The user will write the favor or help they need as well as the monetary value they will pay or the reward they will give to the person who performs the requested favor. When the favor is requested, the other users will be able to review and accept the request to proceed to make a deal with the applicant. The App will also allow to rate the level of satisfaction of both the person who requested the favor and the person who did it.

The application to be developed will be a multi-platform (both for Android and iOS) and can be used by anyone. Further, usability and accessibility techniques will be applied. The app will be developed from scratch and to achieve it the stages of the software life cycle will be followed (analysis and design, development, testing, execution), while for the design of the screens, prototyping tools will be used to later translate it into the App development using the latest web technologies (Ionic, HTML5, BootStrap, TypeScript, etc.)

As a result, a Beta version of the mobile application generated for the Android platform will be obtained where will be able to observe the accomplishment of the functional requirements and its purpose; however, it will not focus so much on non-functional requirements such as synchronization, security, performance, authentication, encryption, and more.

Índice

1. Introducción	4
1.1 Contexto y justificación del Trabajo	4
1.1.1 Problemática	4
1.1.2 Marco teórico	4
1.1.3 Solución	5
1.1.4 ¿Existen otras Apps orientadas al mismo problema?	6
1.1.5 Puntos fuertes y débiles con respecto a otras aplicaciones	6
1.2 Objetivos del Trabajo	7
1.3 Enfoque y método seguido	9
1.4 Planificación del Trabajo	10
1.4.1 Recursos necesarios	10
1.4.2 Tareas, hitos y planificación temporal	11
1.5 Breve resumen de productos obtenidos	13
1.6 Breve descripción de los otros capítulos de la memoria	13
1.6.1 Diseño	13
1.6.2 Implementación	14
1.6.3 Conclusiones	14
2. Análisis, Diseño y Arquitectura	15
2.1 Diseño Centrado en el Usuario	15
2.1.1 Usuarios y Contexto de Uso	15
2.1.1.1 Encuesta realizada	15
2.1.1.2 Análisis de resultados de las encuestas	18
2.1.1.3 Conclusiones de los resultados obtenidos	20
2.1.1.4 Perfiles de usuario	21
2.1.2 Diseño conceptual	23
2.1.2.1 Personajes	23
2.1.2.2 Escenarios de uso	26
2.1.2.3 Flujo de interacción	30
2.1.3 Prototipado	31
2.1.3.1 Sketches	31
2.1.3.2 Prototipos horizontales de alta fidelidad	34
2.1.4 Evaluación	39
2.1.4.1 Test inicial	39
2.1.4.2 Planificación de la evaluación	41
2.2 Diseño técnico	43
2.2.1 Definición de los casos de uso	43
2.2.1.1 Actores del sistema	43
2.2.1.2 Diagrama UML de los casos de uso	43
2.2.1.3 Detalle de los casos de uso	44
2.2.2 Diseño de la arquitectura	49
2.2.2.1 Diagrama de la base de datos	49
2.2.2.2 Diagrama de clases	50
2.2.2.3 Arquitectura general del sistema	51
2.2.2.4 Arquitectura de la aplicación	52
3. Implementación de la Aplicación	53

3.1 Desarrollo	53
3.1.1 Frameworks, librerías y plugins	53
3.1.2 Estructura de la Aplicación.....	55
3.1.2.1 Compartido	56
3.1.2.2 Guards.....	56
3.1.2.3 Pipes.....	56
3.1.2.4 Modelado	56
3.1.2.5 Servicios	57
3.1.2.6 Componentes	57
3.1.2.7 Páginas.....	58
3.1.3 Pantallas Implementadas.....	59
3.1.4 Estados y actores del Favor.....	61
3.1.5 Aspectos Relevantes del Desarrollo.	62
3.2 Pruebas	64
3.2.1 Entorno de pruebas.....	64
3.2.2 Pruebas Funcionales	65
3.2.2.1 Definición de casos de pruebas y resultados.....	65
3.2.2.2 Pruebas del flujo y los estados del favor.....	70
3.2.3 Evaluación de la Aplicación	71
4. Conclusiones	73
4.1 Conclusiones del trabajo	73
4.2 Logro de los objetivos planteados	74
4.3 Seguimiento de la planificación y metodología propuesta	74
4.4 Líneas de trabajo futuro	75
5. Glosario	77
6. Referencias	79
7. Anexos	81

Lista de Figuras

Figura 1.1: Lista de tareas e hitos de la planificación	11
Figura 1.2: Diagrama de las tareas e hitos de la planificación.	12
Figura 2.1: Extracto de la encuesta - sección “conociendo al usuario”.	16
Figura 2.2: Extracto de la encuesta - sección “Información de uso, descarga y compra de Aplicaciones móviles”	17
Figura 2.3: Extracto de la encuesta - sección “Información relacionada a la App a crear”	18
Figura 2.4: Extracto de resultados de la primera sección de la encuesta.....	19
Figura 2.5: Extracto de resultados de la segunda sección de la encuesta.	19
Figura 2.5: Extracto de resultados de la tercera sección de la encuesta.....	20
Figura 2.6: Actor 1 (Jorge), con el perfil de usuario que solicita favores.	24
Figura 2.9: Actor 2 (Jessica), con el perfil de usuario que ayuda con favores solicitados.....	25
Figura 2.10: Flujo de interacción de la aplicación AcolitApp.....	30
Figura 2.11: Diagrama UML de los casos de uso de la aplicación.	44
Figura 2.12: Diagrama de la base de datos de la aplicación	49
Figura 2.13: Diagrama de clases de la aplicación	50
Figura 2.14: Diagrama de paquetes de la aplicación	50
Figura 2.15: Arquitectura cliente-servidor que usará el sistema.....	51
Figura 2.16: Arquitectura MVC (Modelo-Vista-Controlador) de la aplicación. ..	52

1. Introducción

1.1 Contexto y justificación del Trabajo

1.1.1 Problemática

El problema surge a partir del poco tiempo que posee una persona para poder realizar muchas actividades al mismo tiempo en ciertos días, pues existen ocasiones en las que una persona no puede realizar varias cosas en un mismo día por falta de tiempo o porque no puede estar en dos o más lugares en ese instante, por lo que debe decidir entre hacer las cosas incompletas o dejar de hacer una tarea para poder realizar la otra. También existen personas que tienen un empleo, empresa o negocio propio y por lo tanto pasan gran parte del tiempo en su lugar de trabajo, lo que les impide realizar otro tipo de actividades o acudir a otro lugar dentro de ese horario, obligándolos así a pedir permiso o buscar en ese momento a otra persona para que le ayude con la otra actividad que debe realizar.

Creo que a todas las personas, alguna vez en la vida nos ha sucedido que necesitamos realizar muchas tareas a la vez, pero no es posible realizarlas porque nos encontramos realmente muy ocupados en algo específico y por lo tanto se debe recurrir a pedir un favor a otras personas para que nos ayuden con lo que necesitamos.

1.1.2 Marco teórico

Este problema se ha dado desde el origen de los tiempos y las personas lo han resuelto según las condiciones y posibilidades que tienen en ese momento [5]. Pues, la gente ha optado por llamar a algún familiar para que le ayude a realizar la actividad pendiente, otras se ranclan del trabajo, cierran su negocio por un momento o piden permiso a su jefe para realizar otras actividades urgentes. También existen ciertas personas que prefieren solicitar un taxi para que les hagan algún mandado y otras simplemente usan aplicaciones que satisfacen una necesidad específica de manera cómoda; por ejemplo, usar una app para pedir comida a una dirección en particular, comprar entradas al cine, pedir un vehículo, etc. En ciertas ocasiones una persona decide dejar pendiente una actividad para realizarla en otro momento y cuando dispone del tiempo necesario.

En los últimos años se han creado gran cantidad de aplicaciones móviles para poder resolver problemas específicos y que así la persona pueda realizar más de una actividad a la vez o pueda organizar de mejor manera el tiempo y las actividades a realizar [6]. Existen aplicaciones en las que se pueden dar prioridad a las actividades y hacernos recordar cada cosa que tenemos que hacer [7]. Otras aplicaciones específicas por ejemplo nos permiten saber la ubicación de nuestros hijos sin estar cerca de ellos mientras ellos van a la escuela, separar

un turno en un hospital, seguro o clínica, sin acudir personalmente, etc., lo cual evita que pidamos favores a otras personas.

Las aplicaciones móviles también han permitido interactuar con otras personas que no conocemos personalmente pero que mediante la App nos podrían ayudar a cumplir y satisfacer una necesidad específica; por ejemplo, cuando necesitamos la recomendación de algún lugar para comer o comprar algún artículo en particular o simplemente cuando necesitamos transportarnos de un lugar a otro como lo hacemos utilizando aplicaciones como Uber o Cabify [8, 9].

En todos estos casos existe tanto la persona que tiene la necesidad y la persona que es capaz de satisfacerla, es decir, pedimos un favor y alguien puede estar dispuesto a ayudarnos a cambio de nada o a cambio de un valor monetario, un intercambio de cosas o una recompensa.

1.1.3 Solución

La solución que se propone para este problema es desarrollar e implementar la App “ACOLITAPP”, una aplicación móvil multiplataforma (para iOS y Android), mediante la cual las personas podrán solicitar favores a los demás, y que a su vez las personas que tengan la oportunidad y capacidad de realizar tal favor, lo hagan a cambio de recibir una recompensa por la ayuda prestada.

La app funcionará de la siguiente manera:

1. La persona escribirá el favor o ayuda que necesita, así como el valor monetario que pagará o la recompensa que entregará a la persona que realice el favor solicitado. La recompensa puede ser algún obsequio en especial, como por ejemplo: entradas al cine, una cena en un restaurant, un artículo para vehículo, etc.
2. La persona publicará el favor solicitado y las demás personas podrán verlo utilizando la misma aplicación.
3. Las personas podrán aceptar o no las diferentes propuestas y llegar a un acuerdo con la persona que lo solicitó. El solicitante seleccionará a la persona que considere más adecuada para el favor que necesita ayuda y de esta manera se establecerá el trato entre las dos personas.
4. Al final, tanto el solicitante como la persona que aceptó realizar el favor podrán calificarse mutuamente según el nivel de satisfacción con respecto al trato realizado y si se realizó correctamente o no el favor solicitado.

La aplicación podrá ser utilizada por cualquier persona y en cualquier dispositivo móvil con plataforma Android en su versión Beta.

Como resultado final se quiere obtener una versión Beta de la aplicación móvil “ACOLITAPP”, en donde se pueda observar la finalidad de la App, es decir que cumpla con los requisitos funcionales descritos anteriormente. En este caso no

se centrará mucho en el desarrollo del backend o servidor al que se conectará la App, tampoco se tratará de implementar o abordar a fondo los requisitos no funcionales como la sincronización, seguridad, rendimiento, autenticación, encriptación, etc., debido a que son temas muy extensos y demasiado grandes y complejos para poder implementarlos en un tiempo muy corto.

1.1.4 ¿Existen otras Apps orientadas al mismo problema?

Existe una gran variedad de aplicaciones en las que una persona puede hacer trato con otra para cumplir una necesidad específica, entre una de ellas está la App móvil “Uber”, mediante la cual se puede pedir viajes a personas que poseen vehículos y así realizar viajes compartidos sin tener que esperar un taxi o autobús [8].

Por otra parte, existe una aplicación denominada FavForMe que nació en el 2014 y cuyo objetivo es conectar personas para que entre ellas puedan intercambiar favores. FavForMe se basa en un sistema de monedas o créditos virtuales que no son específicamente bitcoins, y que al momento de registrarse en la aplicación le otorga 5 créditos, y a medida que se van realizando favores se irán acumulando monedas para que también ese usuario pueda solicitar favores [4]. Por otra parte, la App tiene un sistema de ranking, por lo que, mientras más monedas posea el usuario, más posibilidades tendrá de ubicarse en una mejor posición dentro del ranking [3].

También existen aplicaciones como Helpin’, que se basa en el intercambio de favores y en la cual la persona pública qué es lo que necesita y qué es lo que ofrece a cambio [1]. En este caso se tiene la idea de que a la gente le puede interesar más colaborar que pagar por un Servicio; pues así, muchas personas intercambian sus habilidades (ofrecen clases de matemáticas, hacer un tatuaje, podar el césped), cosas (una silla para ordenador, un libro), empleo (contratar un árbitro, payaso, un biólogo, etc.), etc con el fin de obtener también algo a cambio [2].

1.1.5 Puntos fuertes y débiles con respecto a otras aplicaciones.




Uno de los puntos fuertes con respecto a las aplicaciones que se usan para un objetivo específico como por ejemplo Uber [8], es que la App no está centrada a cumplir y satisfacer una sola necesidad en particular, pues abre las puertas para que mediante la ayuda de otras personas se pueda satisfacer más de una necesidad con la misma App. Por ejemplo, separar un turno en una clínica, comprar artículos, acudir a un lugar específico, etc.

Otro punto fuerte con respecto a otras Apps usadas para intercambiar favores es que no necesariamente necesitará acumular monedas o créditos para poder solicitar un favor, pues podría ver algún vídeo publicitario antes de pedir ayuda con dicho favor, o simplemente pedirlo sin hacer nada a cambio. Además en esta aplicación no solo podrá calificar la persona que solicita el favor, sino que la persona que lo realizó también podrá calificar a la persona que lo solicitó; esto

con el fin de que mutuamente puedan valorar el nivel de satisfacción del trato realizado y el cumplimiento del favor como tal. Como un punto fuerte adicional está la libertad de decidir si la recompensa a dar será algún valor monetario o obsequio en particular.

Como un punto débil que se tiene, es el no poder brindar facilidades de pago mediante la aplicación, por ejemplo, poder pagar con una tarjeta de Crédito. Otra debilidad es no manejar categorías para los diferentes tipos de favores que se puedan solicitar, por lo que será muy difícil filtrar el tipo de solicitudes que se desea ver o recibir.

A continuación se presenta una comparativa de la App con Uber y FavForMe:

	 Uber	 FavForMe	 AcolitApp
Cumplen una sola necesidad específica (ejemplo: solicitar transporte).	X		
Cumplen más de una necesidad (ejemplo: solicitar transporte, pedir comida, enviar productos, cortar césped).		X	X
Pago mediante tarjeta de crédito, Android Pay, PayPal.	X		
Pago mediante Créditos (monedas ficticias).		X	
Pago mediante efectivo, obsequio u otro favor.			X
Posee un sistema de ranking y reputación.		X	
Permite calificar a la persona que brinda el servicio.	X	X	X
Permite calificar a la persona que solicitó el servicio.			X
No necesita tener puntos o buena reputación para solicitar un servicio.	X		X
Maneja categorías para separar los tipos de servicios que se pueden solicitar en la App.	No necesita categorías.		

1.2 Objetivos del Trabajo

En este apartado se describen los objetivos del Trabajo Fin de Máster, indicando a alto nivel los requisitos funcionales y no funcionales de la aplicación móvil.

Objetivo Principal

- **Desarrollar la App “ACOLITAPP” - aplicación móvil multiplataforma (iOS y Android) para solicitar favores.**

Objetivos Secundarios

- Demostrar los conocimientos adquiridos durante el máster, plasmándolos en este trabajo final.
- Adquirir experiencia en definir una planificación realista y sacar adelante un proyecto completo.
- Investigar los usuarios de la aplicación y recoger requisitos que ayudarán a conocer los usuarios y perfiles que usarán la app.
- Conocer y aprender herramientas online para diseñar y construir un prototipo de alto nivel de la aplicación.
- Profundizar mi conocimiento en el framework ionic para el desarrollo de aplicaciones multiplataforma.
- Ser capaz de documentar y justificar las decisiones tomadas en el desarrollo del proyecto y los resultados logrados.
- Obtener una versión beta que refleje los requisitos funcionales y no funcionales definidos.

Requerimientos Funcionales

- La app permitirá registrar nuevos usuarios o iniciar sesión a los usuarios previamente registrados.
- El usuario podrá listar los favores que ha solicitado.
- El usuario podrá listar los favores en los que ha brindado ayuda (favores realizados).
- La app permitirá al usuario solicitar nuevos favores.
- El usuario podrá elegir la forma en la que pagará por un favor realizado (un valor monetario, una recompensa o devolver con otro favor).
- El usuario podrá ver los favores que han solicitado los demás usuarios.
- Un usuario podrá aceptar o no las diferentes propuestas de favores y así llegar a un acuerdo con la persona que lo solicitó.

- El usuario que solicita un favor, podrá ver la información de las personas que pretenden ayudarlo, y así seleccionar a la persona que considere más adecuada.
- Tanto el solicitante como la persona que aceptó realizar el favor podrán calificarse mutuamente según el nivel de satisfacción del favor realizado.

Requerimientos No Funcionales

- Construir pantallas interactivas y fáciles de usar (usabilidad).
- Implementar una App multiplataforma (tener un solo código capaz de generar código nativo para iOS y Android).
- Generar una versión Beta de la App (solo para la plataforma Android).
- Los datos de la App se almacenarán y consumirán a través de un servidor backend.
- Contar con el archivo apk para que se pueda instalar en cualquier dispositivo Android, sin la necesidad de descargarla de una tienda de apps.

1.3 Enfoque y método seguido

El enfoque que se seguirá es el comúnmente se sigue al momento de desarrollar cualquier sistema. De esta manera se seguirá las etapas del ciclo de vida de un software (análisis y diseño, desarrollo, pruebas, ejecución), mientras que las pantallas serán diseñadas e implementadas usando las últimas tecnologías de desarrollo.

En la etapa de análisis se comparan diversas tecnologías, librerías, frameworks, etc., para escoger las herramientas más adecuadas para esta aplicación. Además se aplicará el diseño centrado en el usuario, en el cual se investigará a los usuarios de la aplicación y se definirán los perfiles, se elaborarán escenarios de uso, y se diseñará y construirá un prototipo de alto nivel de la aplicación.

El diseño de las pantallas serán realizadas en una herramienta de prototipado, para luego plasmarlo en el desarrollo. En cuanto a la implementación, se tiene una propuesta tentativa de desarrollarla con Ionic y de esta manera mostrar una versión demo para la plataforma Android porque por el momento no se dispone de dispositivos para compilar y probar en iOS. Se piensa utilizar Ionic porque tengo mucha más experiencia en desarrollo web y en el uso de HTML, Javascript, Angular y CSS, por lo que el tiempo de desarrollo sería menor; además porque a futuro se piensa reutilizar el código y generar la App para las demás plataformas sin la necesidad de volver a codificarla, para posteriormente distribuirla y así pueda ser utilizada lo más pronto posible en el País donde resido.

La app pasará por diversas pruebas, primero utilizando emuladores, para posteriormente probar en dispositivos reales con sistemas operativos Android.

Por otra parte, la aplicación se desarrollará desde cero, es decir, se creará un nuevo producto porque realmente no hay recursos existentes ni componentes que se puedan reutilizar. De esta manera, se crearán diseños propios de las pantallas.

Al desarrollar desde cero la App, se tomará en cuenta los requisitos funcionales para que el producto final cumpla con el objetivo principal; pues se tendrá una versión Beta debido a que el desarrollo no se centra en el backend, ni en que sobresalgan otros requisitos no funcionales como la sincronización, seguridad, rendimiento, autenticación, encriptación, portabilidad, etc., pues todas estas mejoras se podrán realizar en versiones posteriores debido a que son temas muy extensos, complejos y que necesitan un mayor análisis.

Para poder controlar los tiempos propuestos en la planificación se pretende simular y seguir de manera parcial el método Kanban Personal [10, 11], porque es un método muy simple para gestionar un proyecto de estas dimensiones. Además, de esta manera se podrá tener un control y organización de las tareas que se deben realizar, las que se están realizando y las que están culminadas. Pues este método se reforzará estableciendo prioridades y tiempos específicos en las tareas, con el fin de cumplir con la planificación propuesta. Por otra parte, cada semana se obtendrán algunos entregables pequeños dependiendo de la fase en la que se encuentre la App móvil.

1.4 Planificación del Trabajo

1.4.1 Recursos necesarios

Los recursos de tipo software, como librerías, frameworks y componentes utilizados para el desarrollo serán básicamente software libre, mientras que los recursos de tipo hardware a utilizar serán: una computadora, un celular con sistema operativo Android, emuladores y si es posible una tablet.

En cuanto a la implementación de la App móvil, se usarán tecnologías actuales, librerías genéricas y entornos de desarrollo potentes y de fácil uso; mientras que para obtener una aplicación híbrida y multiplataforma, se analizarán los frameworks más utilizados hoy en día (PhoneGap, Xamarin, Ionic, Titanium, Flutter) y se seleccionará uno en particular (preferentemente Ionic).

Por otra parte, se utilizarán tecnologías web tales como HTML, CSS y Javascript, y sobretodo Angular. También se hará uso de un editor de código fuente (Visual Studio Code, Sublime Text o Atom) y se optará por el entorno de ejecución Node js, el cual permite instalar las librerías necesarias muy fácilmente.

Con respecto al almacenamiento de datos, se elegirá un sistema de gestión de base de datos (PostgreSql, MySql, etc.) y una herramienta gráfica para poder

gestionar dicha base de datos. Los datos se almacenarán o recuperarán mediante servicios REST implementados en un servidor backend.

1.4.2 Tareas, hitos y planificación temporal

A continuación se presenta la descripción de los recursos necesarios para realizar el proyecto, las tareas a realizar y la planificación temporal de cada tarea, utilizando un diagrama de Gantt.

En este caso me he propuesto trabajar en el proyecto unas 2 horas diarias de lunes a viernes y 5 horas los días sábados.

	Objetivo	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de recursos
1	Aplicar el diseño centrado en el usuario	Investigar los usuarios de la aplicación y definir perfiles.	6 horas?	lu. 16/03/20	mi. 18/03/20		computadora
2	Aplicar el diseño centrado en el usuario	Elaboración de escenarios de uso y definición de los flujos de interacción en el sistema	6 horas?	ju. 19/03/20	sá. 21/03/20	1	computadora,uml
3	Aplicar el diseño centrado en el usuario	Diseñar y construir un prototipo horizontal de alto nivel de la aplicación.	12 horas?	sá. 21/03/20	vi. 27/03/20	2	herramientas ibéricas y entornos descargados
4	Aplicar el diseño centrado en el usuario	Evaluar los diseños y corregir los errores.	6 horas?	vi. 27/03/20	sá. 28/03/20	3	herramientas ibéricas y entornos descargados
5	HITO 1 - PEC 2 - DISEÑO	Diagramas UML y listado de casos de uso definidos.	0 horas	mi. 01/04/20	mi. 01/04/20		herramientas ibéricas y entornos descargados
6	HITO 2 - PEC 2 - DISEÑO	Diseño de la arquitectura (diseño de la BD, modelo de clases, arquitectura del sistema con paradigma MVC)	0 horas	mi. 01/04/20	mi. 01/04/20		herramientas ibéricas y entornos descargados
7	HITO 3 - PEC 2 - DISEÑO	Prototipos de las pantallas, estableciendo la jerarquía y estructura de navegación	0 horas	mi. 01/04/20	mi. 01/04/20		herramientas ibéricas y entornos descargados
8	Diseñar e implementar la aplicación móvil	Investigar, comparar y elegir las librerías, herramientas y entorno a utilizar para el desarrollo de la aplicación	6 horas?	lu. 30/03/20	mi. 01/04/20	4	computadora
9	Diseñar e implementar la aplicación móvil	Investigar e implementar técnicas para mejorar la usabilidad y accesibilidad de la App.	2 horas	vi. 03/04/20	vi. 03/04/20	8	herramienta de prototipado
10	Diseñar e implementar la aplicación móvil	Codificar y crear las pantallas según los prototipos	10 horas	sá. 04/04/20	mi. 08/04/20	9	framework de desarrollo
11	Diseñar e implementar la aplicación móvil	Permitir que las pantallas se adapten a diferentes tamaños de dispositivos (pantallas responsive)	6 horas	mi. 08/04/20	sá. 11/04/20	10	herramientas ibéricas y entornos descargados
12	Diseñar e implementar la aplicación móvil	Implementar la interacción de pantallas con el usuario y con datos quemados en el código.	3 horas	lu. 13/04/20	ma. 14/04/20	11	framework de desarrollo
13	Diseñar e implementar la aplicación móvil	Realizar pruebas de funcionalidad en dispositivos móviles físicos.	2 horas	ma. 14/04/20	mi. 15/04/20	12	dispositivo móvil,emuladores
14	Diseñar e implementar la aplicación móvil	Generar una primera versión con el diseño de pantallas y con datos quemados en código.	2 horas	mi. 15/04/20	ju. 16/04/20	13	computadora,emulador,dispositivo móvil
15	Diseñar e implementar la aplicación móvil	Crear modelo de datos de la información a almacenar y/o mostrar en el dispositivo	3 horas	ju. 16/04/20	vi. 17/04/20	14	Herramienta de modelado
16	Diseñar e implementar la aplicación móvil	Implementar un servidor que almacene los datos generados por la App	4 horas?	sá. 18/04/20	sá. 18/04/20	15	herramientas ibéricas y entornos descargados
17	Diseñar e implementar la aplicación móvil	Creación de los servicios REST necesarios para manipular la información	8 horas	sá. 18/04/20	ju. 23/04/20	16	herramientas ibéricas y entornos descargados
18	Diseñar e implementar la aplicación móvil	Consumir servicios necesarios del servidor (Back-end)	3 horas	ju. 23/04/20	vi. 24/04/20	17	credenciales de acceso al servidor,servicios web
19	Diseñar e implementar la aplicación móvil	Implementar controladores, menus, adaptadores para mostrar información, eventos en botones, etc	15 horas	sá. 25/04/20	vi. 01/05/20	18	framework de desarrollo
20	Diseñar e implementar la aplicación móvil	Desplegar, generar y los ejecutables para las plataformas de Android e iOS	5 horas	sá. 02/05/20	sá. 02/05/20	19	computadora,framework de desarrollo
21	Diseñar e implementar la aplicación móvil	Correcciones y modificaciones a realizar en base a la revisión y validación de la App.	10 horas	lu. 04/05/20	vi. 08/05/20	20	Documentación de los cambios a modificar.
22	Pruebas y autoevaluación del producto desarrollado	Definir las pruebas de funcionalidad que se realizarán	4 horas	sá. 09/05/20	sá. 09/05/20	21	datasets
23	Pruebas y autoevaluación del producto desarrollado	Realizar pruebas en dispositivos reales y evaluar el producto	3 horas	sá. 09/05/20	lu. 11/05/20	22	editor de código
24	Pruebas y autoevaluación del producto desarrollado	Establecer los aspectos a mejorar para posteriores trabajos.	2 horas	ma. 12/05/20	ma. 12/05/20	23	computadora,word
25	HITO 1 - PEC 3 - IMPLEMENTACIÓN	Fichero Zip con la App empaquetada	0 horas	mi. 13/05/20	mi. 13/05/20		computadora
26	HITO 2 - PEC 3 - IMPLEMENTACIÓN	Código fuente de la aplicación	0 horas	mi. 13/05/20	mi. 13/05/20		word
27	HITO 3 - PEC 3 - IMPLEMENTACIÓN	Instrucciones de compilación	0 horas	mi. 13/05/20	mi. 13/05/20		word
28	Publicar, documentar y presentar el trabajo realizado	realizar los cambios y correcciones finales a la aplicación	14 horas	lu. 18/05/20	sá. 23/05/20	24	computadora,word
29	Publicar, documentar y presentar el trabajo realizado	Empaquetar y documentar el código fuente de la aplicación	8 horas	sá. 23/05/20	ju. 28/05/20	28	computadora
30	Publicar, documentar y presentar el trabajo realizado	Subir el código a un repositorio público de aplicaciones	4 horas	ju. 28/05/20	sá. 30/05/20	29	repositorios
31	Publicar, documentar y presentar el trabajo realizado	Completar la memoria del trabajo final.	10 horas	sá. 30/05/20	mi. 03/06/20	30	computadora,word
32	HITO 1 - PEC 4 - ENTREGA FINAL	Código fuente con la documentación respectiva	0 horas?	vi. 05/06/20	vi. 05/06/20		word
33	HITO 2 - PEC 4 - ENTREGA FINAL	URL del repositorio donde se encuentra la App	0 horas?	vi. 05/06/20	vi. 05/06/20		repositorios
34	HITO 3 - PEC 4 - ENTREGA FINAL	Memoria del trabajo final	0 horas?	vi. 05/06/20	vi. 05/06/20		computadora,word

Figura 1.1: Lista de tareas e hitos de la planificación



Figura 1.2: Diagrama de las tareas e hitos de la planificación.

1.5 Breve resumen de productos obtenidos

Después de todo el proceso seguido y una vez implementada la aplicación móvil se obtendrá los siguientes productos:

- Planificación y metodología de trabajo.
- Diagramas UML de los casos de uso.
- Listado de casos de uso definidos, incluyendo actores, precondiciones, flujos y postcondiciones.
- Modelo de clases de la App.
- Diseño de la BD (modelo entidad-relación).
- Arquitectura del sistema siguiendo el paradigma Modelo-Vista-Controlador (MVC).
- Prototipos de las pantallas implementadas.
- Código fuente de la aplicación.
- Fichero Zip con la App empaquetada.
- Instrucciones de Compilación de la App móvil.
- URL del repositorio donde se encuentra la App.
- Memoria del trabajo final.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria de trabajo final contendrá 3 capítulos que describirán los aspectos considerados en el diseño, desarrollo y valoración económica de la aplicación móvil.

1.6.1 Diseño

Este capítulo contendrá el diseño centrado en el usuario para la aplicación móvil, pues para cumplir con este fin se seguirán las fases del Diseño Centrado en el Usuario (usuarios y contexto de uso, diseño conceptual, prototipado y evaluación). Al seguir estas fases se obtendrá como resultado la definición de los casos de uso y el diseño de la arquitectura de la aplicación.

Con la definición de los casos de uso se elaborarán escenarios de uso y se detectarán las funcionalidades que deberá tener la aplicación móvil para satisfacer a sus usuarios. Además, como productos se obtendrán los diagramas UML que sirven para identificar los actores y flujo, y el listado de los casos de

uso donde se especifican los actores, precondiciones, flujo y postcondiciones. Con respecto a la arquitectura de la aplicación, se identificarán las entidades que se presentarán en la base de datos, las clases y la estructura de la API, siguiendo el paradigma Modelo-Vista-Controlador (MVC).

1.6.2 Implementación

En este capítulo se describe la forma en que fue desarrollada la aplicación según el diseño realizado en la etapa anterior, pues el diseño se transforma en una aplicación móvil utilizando las herramientas seleccionadas. El proceso de implementación se divide en 2 etapas:

Desarrollo: en esta etapa se codificará la aplicación utilizando los frameworks, librerías, APIs y demás herramientas que se seleccionen.

Pruebas: en esta etapa se establecerá un conjunto de pruebas para verificar y validar el correcto funcionamiento de la App.

1.6.3 Conclusiones

Este último capítulo incluirá las conclusiones del trabajo final, donde se relacionarán los objetivos conseguidos y los no conseguidos, así como las posibles ampliaciones del trabajo, etc.

2. Análisis, Diseño y Arquitectura

2.1 Diseño Centrado en el Usuario

Para el diseño de la aplicación “AcolitApp” se aplicará el Diseño Centrado en el Usuario (DCU), el mismo que consta de 4 fases:

1. Usuarios y contexto de Uso [Análisis]
2. Diseño conceptual [Diseño]
3. Prototipado [Diseño]
4. Evaluación [Evaluación]

Cada una de estas fases se iterarán entre ellas para obtener un buen diseño de acuerdo a lo que el usuario necesita y satisfaga sus necesidades de manera satisfactoria; pues con el DCU se definirán los servicios y productos tratando de que sean atractivos, comprensibles, intuitivos, y que el usuario obtenga la mejor experiencia sin realizar el mayor esfuerzo al utilizar la App diseñada.

Se comenzará con la búsqueda de usuarios para cuestionarlos sobre el uso de Apps móviles y el uso que darían a la aplicación a implementar. Luego de extraer la información se analizará para saber qué tipos de usuarios utilizarían la Aplicación móvil y según esto definir los requisitos, condiciones de uso y las funcionalidades que tendrá la misma para satisfacer las necesidades de estos usuarios.

2.1.1 Usuarios y Contexto de Uso

Como primera parte del DCU, la aplicación móvil “AcolitApp” pasará por la fase de análisis de usuarios y contextos de uso, cuyo objetivo es conocer las características de los usuarios, sus necesidades y el contexto de uso. Al concluir esta fase, se podrá detectar las funcionalidades que deberá tener la aplicación para satisfacer a los usuarios.

En esta fase se investigó los diferentes métodos de indagación y se tuvo que decidir entre realizar entrevistas o encuestas; sin embargo, se decidió utilizar encuestas debido a que la entrevista es más un método cualitativo y no cuantitativo. Por otra parte, por el poco tiempo disponible se optó realizar encuestas con la herramienta de google(Google Forms), debido a su facilidad de uso y sus sencillos gráficos estadísticos.

2.1.1.1 Encuesta realizada

Una vez que se determinaron las preguntas y posibles respuestas de la encuesta, se hizo una prueba con un familiar y 2 compañeros de trabajo para que revisen y analicen las preguntas, con el objetivo de saber si eran entendibles dichas preguntas y si las opciones a elegir eran adecuadas para obtener información relevante para las siguientes fases. Mediante esta prueba se pudo identificar que faltaban preguntas puntuales con respecto a la aplicación,


mientras que otras no se entendían al leerlas una sola vez. Además, se eliminaron preguntas que no eran relevantes para el análisis requerido y que a su vez hacían que la encuesta sea demasiado larga y tediosa para las personas.

Una vez que se creó la encuesta, se generó un enlace y se envió a diferentes tipos de personas, entre los cuales están: familiares, amigos, compañeros de trabajo y algunos grupos de WhatsApp de mi barrio. La encuesta elaborada se encuentra en el anexo 1 y está disponible en el siguiente enlace: <https://forms.gle/r6cySatRYjDTki476>

La encuesta consta de 22 preguntas divididas en 3 secciones, que han sido consideradas importantes para recolectar las características y perfiles de los diferentes usuarios, así como el uso que le dan a las aplicaciones móviles. Además existe una sección para identificar las necesidades y preferencias que tienen los usuarios con respecto a la nueva aplicación a desarrollar. Estas 3 secciones se estructuran de la siguiente manera:

Conociendo al usuario

Aquí se realizan preguntas relacionadas a información personal y básica del usuario, tal como la edad, sexo, estado civil, nivel de estudios, ocupación o dedicación, etc. Dicha información será útil para definir perfiles de usuario y realizar las siguientes preguntas con respecto a la App a desarrollar.



The image shows a screenshot of a survey form. At the top, there is a blue header with the text "Conociendo al usuario". Below the header, there are three questions, each with a red asterisk indicating it is required. The first question is "Edad *", followed by a text input field with the placeholder "Tu respuesta". The second question is "Estado Civil *", with three radio button options: "Soltero", "Casado", and "Otro". The third question is "Sexo", with two radio button options: "Mujer" and "Hombre".

Figura 2.1: Extracto de la encuesta - sección “conociendo al usuario”.

Información de uso, descarga y compra de Aplicaciones móviles

En esta sección se realizan preguntas relacionadas al uso que dan al Smartphone o tableta y a las aplicaciones móviles. Además, con estas preguntas

se estimará la cantidad de personas que podrían comprar una App y el valor que estarían dispuestos a pagar por la misma, sobre todo si descargan ciertas aplicaciones sin importar los comentarios o valoración que dan los usuarios.

Información de uso, descarga y compra de Aplicaciones móviles

¿Cuál es el nivel de uso de aplicaciones en su Smartphone?

Alto. Utiliza varias aplicaciones móviles y más de 5 vez al día.

Medio. Usa alguna aplicación móvil ocasionalmente.

Bajo. Utiliza el móvil, únicamente para hablar por teléfono o escribir mensajes.

¿Ha comprado o compraría alguna Aplicación móvil en las tiendas de aplicaciones?

Sí

No

Tal vez

¿Cuánto pagaría por una Aplicación móvil?

Nada

Menos de 3 dólares

Figura 2.2: Extracto de la encuesta - sección “Información de uso, descarga y compra de Aplicaciones móviles”.

Información relacionada a la aplicación a crear.

En esta sección se realizan preguntas relacionadas a la aplicación que se va a desarrollar, así como el conocimiento de alguna aplicación similar o si ha utilizado aplicaciones para realizar cosas que también se podrían realizar en la nueva App. También se pregunta si utilizaría ciertas funcionalidades a implementar, se recibe sugerencias de las características e información que debería tener la misma.

Información relacionada a la aplicación a crear

¿Ha usado o conoce alguna aplicación para solicitar favores?

Sí

No

Otro: _____

Si su respuesta anterior fue Si, Especifique son esas aplicaciones que ha usado o conoce.

Tu respuesta

Seleccione los casos en los que ha utilizado una aplicación móvil

Pedir comida a domicilio

Solicitar transporte

Realizar pedidos de farmacias, licorerías, supermercados, florería, bisutería, etc. a domicilio

Figura 2.3: Extracto de la encuesta - sección “Información relacionada a la App a crear”.

2.1.1.2 Análisis de resultados de las encuestas

Según los resultados obtenidos a través de las encuestas, se puede identificar que el número de personas que han respondido de manera voluntaria son 26.

A continuación se resume sobre los resultados de cada sección (los resultados de la encuesta están en el anexo 2, y el análisis completo de los resultados se encuentra en el *anexo 3*):

Conociendo al usuario

La encuesta la han realizado personas de diferentes edades, en donde la mayoría de personas tienen entre los 23 y 30 años, poseen un título de tercer nivel, no tienen hijos y poseen un Smartphone. Además existe el mismo número de hombres y mujeres encuestadas.

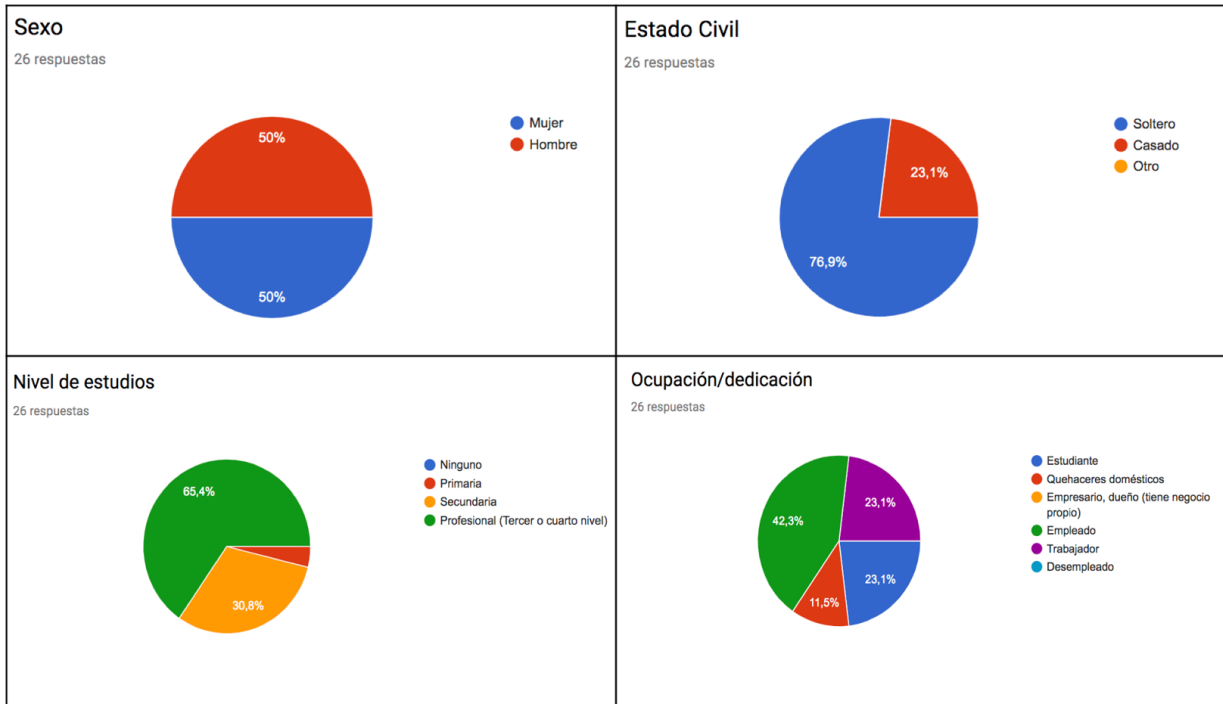


Figura 2.4: Extracto de resultados de la primera sección de la encuesta.

Uso, descarga y compra de aplicaciones móviles

En cuanto al uso, descarga y compra de aplicaciones móviles se verifica que la mayoría de personas utilizan varias aplicaciones móviles por más de 5 veces al día y prefieren no pagar nada o pagar menos de 3 dólares por una App. Además, a veces califican o comentan las aplicaciones utilizadas, sin embargo, si toman en cuenta las valoraciones y comentarios de las Apps antes de descargarlas.

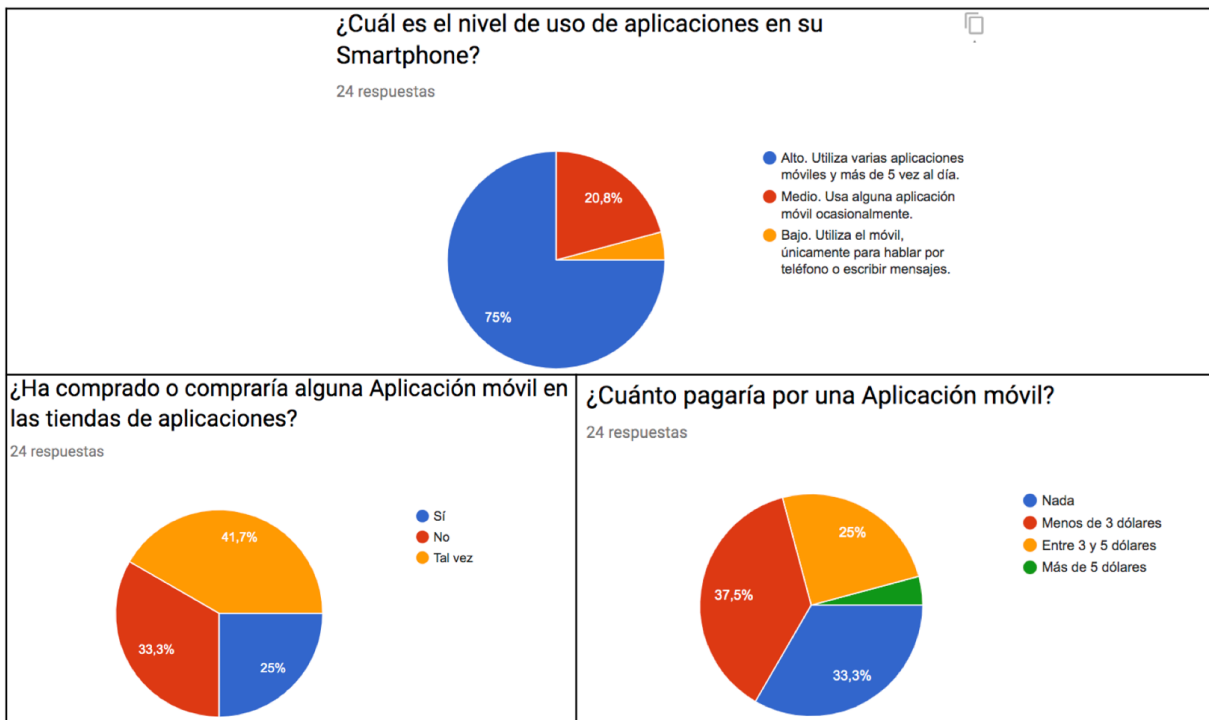


Figura 2.5: Extracto de resultados de la segunda sección de la encuesta.

Información relacionada a la aplicación a crear

Los resultados muestran que la mayoría de las personas han utilizado aplicaciones móviles para pedir comida a domicilio o solicitar transporte y que estarían dispuestos a solicitar o realizar favores mediante una aplicación.

Entre la información que consideran importante conocer para tener seguridad y confianza en la App están la identificación, nombres y apellidos, nacionalidad y ocupación, mientras que las características que más desean tener son: solicitar favores a cambio de algún tipo de recompensa, obsequio u otro favor, recibir notificaciones solo de favores que estaría dispuesto a ayudar y calificar el nivel de satisfacción del favor.

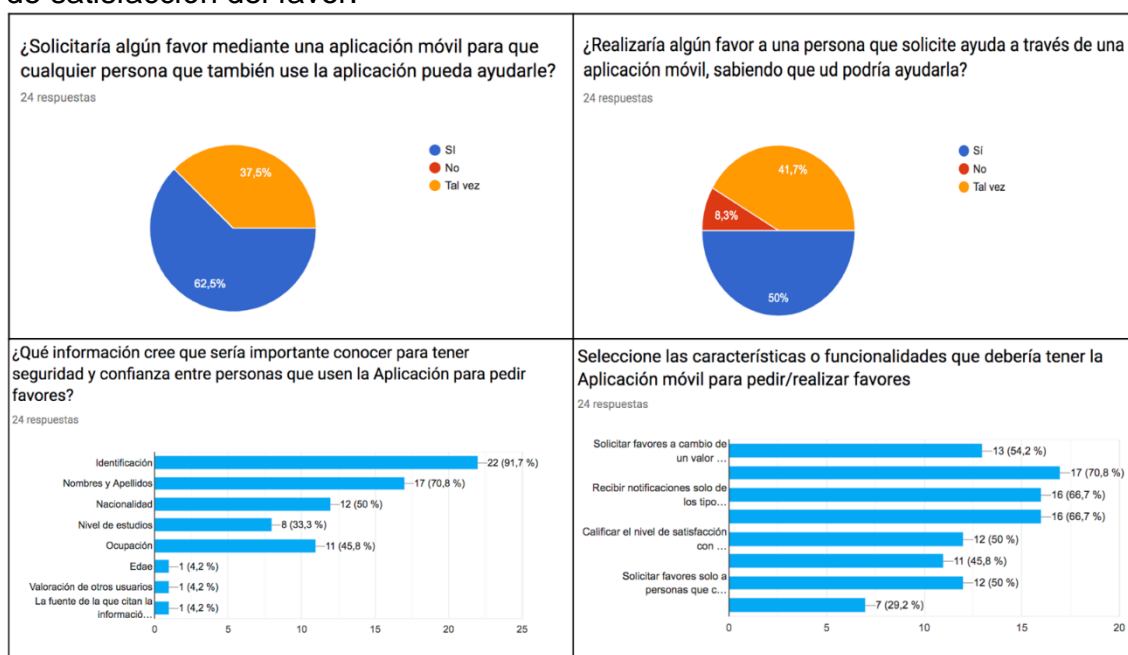


Figura 2.5: Extracto de resultados de la tercera sección de la encuesta.

2.1.1.3 Conclusiones de los resultados obtenidos

Según los resultados obtenidos a partir de las encuestas realizadas se puede concluir que es factible desarrollar la Aplicación móvil para solicitar y realizar favores entre personas debido a que la mayor parte de encuestados desean una aplicación con estas características y además gran porcentaje estaría dispuesto a usarla.

También se pudo verificar que las personas más interesadas en la aplicación son aquellas que tienen una edad entre 23 y 30 años, que han terminado sus estudios de nivel secundario o tienen un título de tercer o cuarto nivel, que son empleados, trabajadores o estudiantes, sin importar su estado civil, sexo y número de hijos.

Se pudo concluir también que los estudiantes, trabajadores y amas de casa usarían la aplicación especialmente para realizar favores a las demás personas, pero no pagarían por la aplicación o pagarían en ciertos casos menos de 3

dólares; mientras que los empleados y trabajadores con profesión la usarían para solicitar favores y estarían dispuestos a pagar hasta 5 dólares por la App.

Finalmente, las personas que tienen una profesión y son empleados, son las que más han utilizado aplicaciones para pedir comida a domicilio, solicitar transporte y realizar pedidos en diferentes tiendas; sin embargo, casi todas las personas que tienen un Smartphone o Tablet quisieran tener una App para solicitar o realizar el favor, prefiriendo que el favor se lo haga a cambio de alguna recompensa, obsequio u otro favor en lugar de un valor monetario.

2.1.1.4 Perfiles de usuario

Una vez realizado las encuestas a diferentes tipos de personas y gracias al análisis realizado se pudieron identificar dos perfiles de usuario, agrupados según sus características y aspectos socio-demográficos.

Estos dos perfiles agrupan por un lado a personas que solicitan favores y por otro lado a personas que ayudarían a realizar los favores solicitados. A continuación se detallan los dos perfiles de usuario para la aplicación “AcolitApp”:

Perfil 1	Usuario que solicita favores
Características del perfil	<p>Este perfil está compuesto por personas que utilizarían la app para solicitar favores y encontrar a personas que lo puedan ayudar.</p> <p>Este tipo de usuario por lo general son personas con un título de tercer o cuarto nivel y unos cuantos con título secundario. Además son empleados o trabajadores con una carga laboral muy grande y muy poco tiempo disponible para realizar otras actividades.</p> <p>Por otra parte, este perfil lo comprende personas entre los 23 y 50 años, con mucha experiencia en el uso de aplicaciones móviles, utilizándolo la mayor parte del tiempo para el trabajo, pedir comida a domicilio o solicitar transporte cuando no tienen tiempo de tomar un autobús.</p> <p>Uno de los objetivos de estos usuarios, es tener aplicaciones que ayuden en su vida diaria, por ejemplo Apps para recordatorios, chatear con grupos de trabajo, brindar asistencia, redactar o leer documentos, reservar lugares, manejar el dinero, entre otras. Además, antes de descargar una aplicación toman en cuenta la valoración y comentarios sobre la misma, y hasta optarían por pagar por la App.</p>
Contexto de uso	<p>1. La persona está en su trabajo y se le olvidó traer algún documento importante, así que solicita mediante la App a alguien lo ayude trayendo.</p>

	2. La persona puede estar en una reunión importante de trabajo y ya es hora de retirar a los niños de la escuela, entonces solicita mediante la App a alguien con autobús escolar que lo ayude.
Análisis de las tareas realizadas por el usuario.	<p>Las tareas que necesita realizar el usuario con la App son:</p> <ul style="list-style-type: none"> ● Crear un usuario y registrar la información solicitada. ● Iniciar sesión. ● Escribir y detallar el favor que necesita. ● Elegir si el pago será con un valor monetario, obsequio u otro favor. ● Solicitar el favor descrito. ● Aceptar trato de la persona que lo ayudaría con el favor. ● Calificar a la persona que lo ayudó.
Listado de características o funcionalidades de la App.	<ul style="list-style-type: none"> ● Mostrar información personal y segura de las personas que acepten realizar el favor. ● Permitir calificar y medir el nivel de satisfacción de las personas que lo ayudaron con el favor solicitado. ● Permitir especificar la forma en que pagará por el favor (valor monetario, un obsequio u otro favor). ● Poder aceptar o rechazar ayuda de las personas. ● Registrarse e iniciar sesión en la App.

Perfil 2	Usuario que realiza favores
Características del perfil	<p>Este perfil está compuesto por personas que utilizarían la aplicación principalmente para ayudar a las demás personas y realizar los favores solicitados.</p> <p>Este tipo de usuario por lo general son personas que han terminado sus estudios de nivel secundario o primario, solo unos pocos con un título profesional. La mayoría son estudiantes, personas que se dedican a quehaceres domésticos y personas sin un trabajo fijo o que tienen ingresos económicos medio-bajos. Estas personas no tienen una carga de trabajo tan grande o tienen pocas horas de estudio al día, por lo que el resto del tiempo lo aprovechan para realizar actividades recreativas o de ocio.</p> <p>Este perfil lo comprenden personas de 16 años en adelante que poseen un celular y hacen uso de aplicaciones relacionadas más al entretenimiento como YouTube, Facebook y WhatsApp para conversar con amigos y familia.</p> <p>Uno de los objetivos de los usuarios es buscar alternativas para conseguir ingresos económicos, buscando ofertas de empleo y ofreciendo productos o servicios pequeños en sus redes sociales para así obtener más ingresos. Además, descargan apps sin tomar muy en cuenta la valoración y comentarios; pues comúnmente optan por no pagar por las Apps.</p>

Contexto de uso	1. Un estudiante está muy aburrido esperando que pase la hora libre para comenzar otra clase, y de repente ve en la app que solicitan ayuda para dejar unos documentos en un lugar cerca de la universidad. Él aprovecha el tiempo y se ofrece a ayudar con el favor solicitado y así obtiene dinero extra.
Análisis de las tareas realizadas por el usuario.	Las tareas que necesita realizar el usuario con la App son: <ul style="list-style-type: none"> ● Crear un usuario y registrar la información solicitada. ● Iniciar sesión. ● Ver en la App que una persona que solicitó un favor. ● Revisar el favor o favores que requieren las personas. ● Aceptar o ignorar el favor solicitado. ● Si aceptó ayudar, deberá calificar el trato y recompensa de la persona a la que ayudó.
Listado de características o funcionalidades de la App.	<ul style="list-style-type: none"> ● Permitir ver información personal y segura de las personas que solicitan favores. ● Permitir calificar y medir el nivel de satisfacción del trato y/o recompensa recibida por la ayuda brindada a la persona que pidió el favor. ● Poder aceptar o rechazar solicitudes según su interés. ● Mostrar su calificación personal a las personas que soliciten favores.

2.1.2 Diseño conceptual

En esta fase se elaboran los escenarios de uso a partir de la información recopilada en la fase anterior.

Los escenarios describirán desde la visión del usuario como utilizará la aplicación móvil para pedir o realizar favores en un contexto concreto. Esto servirá para determinar necesidades de los usuarios y de diseño de la App. Una vez definidos estos escenarios, estos servirán más adelante para conceptualizar la estructura de la aplicación y los flujos de interacción.

2.1.2.1 Personajes

Este método o técnica utiliza usuarios ficticios que de manera particular representarían a los usuarios reales de la aplicación y que a su vez tendrán uno de los 2 perfiles definidos en la fase anterior.

En este caso se representará a 4 personajes ficticios, 2 con el perfil de usuario que solicita favores(Jorge, María) y 2 con el perfil de usuario que realiza los

favores(Juan, Jessica), con los cuales se va hacer un detalle de sus características, objetivos, necesidades y de la manera que podrían utilizar la App.

Los personajes se muestran a continuación:

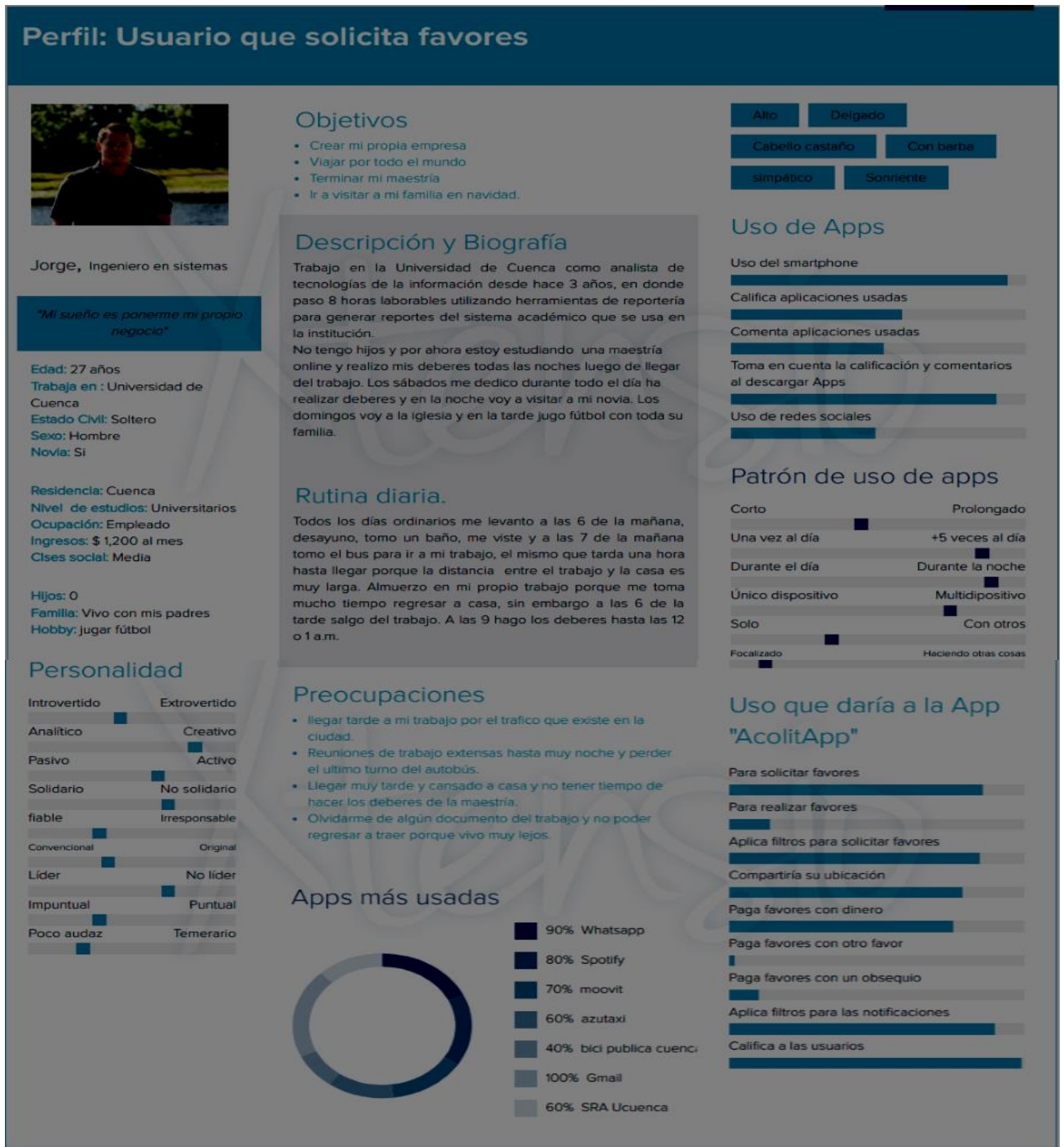


Figura 2.6: Actor 1 (Jorge), con el perfil de usuario que solicita favores.

Perfil: Usuario que ayuda con favores solicitados



JESSICA, estudiante de la carrera de ingeniería comercial.

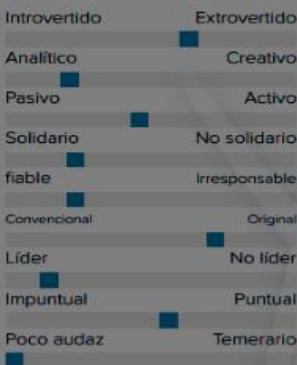
"Leer, estudiar y hacer deporte es lo que más me encanta en la vida"

Edad: 23 años
Estudia en: Universidad Estatal de Cuenca
Estado Civil: Soltera
Sexo: Mujer
Novio: No

Residencia: Cuenca, Ecuador
Nivel de estudios: Universitarios por concluir.
Ocupación: Estudiante
Ingresos: \$ 0, dependo de mi familia
Clase social: Media-baja

Hijos: 0
Familia: Vivo con mis padres y hermanos.
Hobby: Hacer ejercicio y leer

Personalidad



Objetivos

- Terminar el trabajo final y graduarme.
- Conseguir un empleo pronto y a ayudar económicamente a mi familia.
- Mantenerme en forma, cuidando mi salud.
- Salir del País y realizar un postgrado.

Descripción y Biografía

Yo estudio en la Universidad Estatal de Cuenca y hace un mes culminé el último ciclo de mi carrera y ahora empecé a realizar el trabajo final (tesis) para poder graduarse de Ingeniera Comercial. Con respecto a mi vida personal, soy soltera y vivo con mis padres y mis 3 hermanos.

Vivo en la zona rural de la ciudad en una casa que mi padre logró construir con mucho esfuerzo. Hago deporte y cuido mi salud, pues todas las mañanas salgo a correr y a hacer ejercicio, además soy muy buena estudiante y domino muy bien las matemáticas y el inglés.

La universidad en la que estudié todo estos años es pública y por lo tanto fue gratuita durante toda mi carrera; sin embargo, desde hace un mes ha estado buscando trabajo para ya no depender tanto económicamente de mis padres y así pagar mis propios gastos. Sin embargo, no he logrado encontrar trabajo debido a que aún no tengo el título profesional y ni la experiencia en mi área profesional.

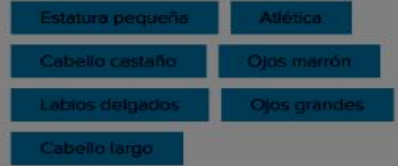
Rutina diaria.

Me he dedicado unas horas al día en realizar la tesis y unas par de horas haciendo cualquier actividad, como ayudando en los quehaceres de la casa y reunirme de vez en cuando con su amiga para escuchar un poco de música, ver películas o hacer ejercicio juntas. También estoy colocado publicidad en Facebook, en la cual ofrezco clases de nivelación en inglés y matemáticas para estudiantes de nivel básico e intermedio. Además busco ofertas de trabajo en la misma red social para ayudar económicamente en mi casa pero no he obtenido buenos resultados.

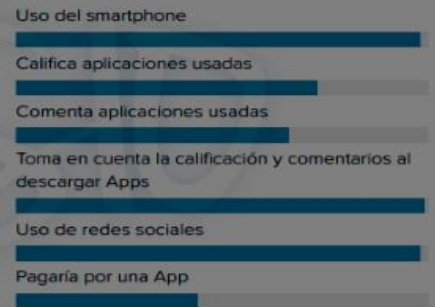
Preocupaciones

- No poder ayudar económicamente a mi familia.
- Depender de mis padres mientras tarde en realizar la tesis o encontrar trabajo.
- No conseguir trabajo por la falta del título y la experiencia laboral.
- No conseguir personas interesadas en las clases que ofrezco brindar.

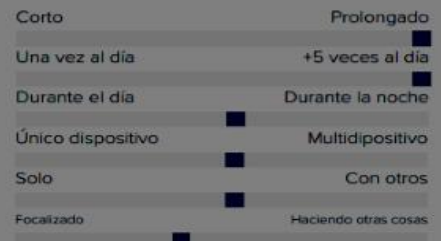
Apps más usadas



Uso de Apps



Patrón de uso de apps



Uso que daría a la App "AcolitApp"



Figura 2.9: Actor 2 (Jessica), con el perfil de usuario que ayuda con favores solicitados.

2.1.2.2 Escenarios de uso

En esta sección se establecerán diferentes escenarios de uso, en los cuales se harán descripciones de los personajes y perfiles de usuario en una situación de uso de la aplicación con ciertos objetivos concretos. A partir de esos escenarios de uso se descubrirán las funcionalidades y flujo en el uso de la aplicación para solicitar y realizar favores.

A continuación, se detallan los diferentes escenarios de uso de la aplicación:

Escenario 1	Registrar usuario
Perfil de usuario	Usuario que realiza favores
Personaje	Jessica
Contexto	Jessica está buscando en su tablet aplicaciones para encontrar empleo y de pronto encuentra la app "AcolitApp", dándose cuenta que muchos usuarios la han usado para solicitar o realizar favores. Entonces se descarga y se registra para poder utilizarla.
Objetivo	Registrar un usuario en la aplicación.
Tareas	<ul style="list-style-type: none"> ● Instalar la aplicación. ● Abrir la aplicación ● Registrar los datos en el formulario respectivo.
Necesidades de información	<ul style="list-style-type: none"> ● Conocer el proceso y los datos necesarios para registrar un nuevo usuario.
Funcionalidades	<ul style="list-style-type: none"> ● Registrar un nuevo usuario. ● Insertar la información básica que se visualizará a los usuarios.
Realización de las tareas	<ul style="list-style-type: none"> ● Abrir la aplicación. ● Seleccionar la opción "Registrar". ● Colocar el correo y contraseña. ● Presionar el botón "siguiente". ● Llenar el formulario con información importante para la seguridad y confianza de las demás personas. ● Presionar en "finalizar".

Escenario 2	Iniciar sesión
Perfil de usuario	Usuario que solicita favores
Personaje	Jorge
Contexto	Jorge decide abrir la app e iniciar sesión para revisar cómo funciona la misma.
Objetivo	Iniciar sesión en la aplicación.

Tareas	<ul style="list-style-type: none"> ● Iniciar sesión en la aplicación
Necesidades de información	<ul style="list-style-type: none"> ● Iniciar sesión para revisar las novedades y funcionalidades que posee la aplicación.
Funcionalidades	<ul style="list-style-type: none"> ● Iniciar sesión.
Realización de las tareas	<ul style="list-style-type: none"> ● Abrir la aplicación. ● Colocar el nombre de usuario y contraseña. ● Presionar en el botón “Iniciar sesión”.

Escenario 3	Solicitar un favor a cambio de un valor monetario
Perfil de usuario	Usuario que solicita favores
Personaje	María
Contexto	María está en su trabajo y recibe una llamada de su hijo contándole que necesita comprar algunos materiales para la clase de pintura de mañana. María no puede dejar su trabajo, así que abre la app y ofrece \$10 a la persona que le ayude comprando los materiales y los entregue en el trabajo.
Objetivo	Solicitar ayuda en la compra y entrega de materiales a domicilio a cambio de un valor monetario.
Tareas	<ul style="list-style-type: none"> ● Iniciar sesión ● Escribir el favor ● Establecer el valor a pagar a cambio del favor ● Solicitar el favor.
Necesidades de información	<ul style="list-style-type: none"> ● Se necesita conocer el favor y el valor a pagar para solicitar el favor
Funcionalidades	<ul style="list-style-type: none"> ● Iniciar sesión. ● Permite redactar el favor a solicitar. ● Permite seleccionar las diferentes maneras de pago o recompensa.
Realización de las tareas	<ul style="list-style-type: none"> ● Iniciar sesión. ● Dar clic en el signo “+”. ● En la pantalla debe escribir el favor a solicitar. ● En la lista desplegable debe seleccionar el tipo de pago (valor monetario, obsequio, otro favor). En ese caso selecciona “valor monetario”. ● Dar clic en el botón “solicitar”.

Escenario 4	Aceptar ayudar con un favor
Perfil de usuario	Usuario que realiza favores.
Personaje	Juan
Contexto	Juan revisa en la app los favores y observa que una persona necesita reparar un mueble de madera por 30 dólares; entonces le parece bien la oferta y decide aceptar y ayudar con el favor solicitado.
Objetivo	Aceptar un favor solicitado mediante la App móvil.
Tareas	<ul style="list-style-type: none"> ● Iniciar sesión. ● Abrir la lista de favores. ● Aceptar el favor solicitado.
Necesidades de información	<ul style="list-style-type: none"> ● Conocer la lista de favores y aceptar un favor realizado.
Funcionalidades	<ul style="list-style-type: none"> ● Iniciar sesión. ● Permitir visualizar favores solicitados por otras personas. ● Ver los detalles del favor solicitado. ● Aceptar el favor solicitado.
Realización de las tareas	<ul style="list-style-type: none"> ● Iniciar sesión ● Seleccionar “favores solicitados”. En la lista, seleccionar un favor. ● Dar clic en el botón “Aceptar”.

Escenario 5	Calificar el nivel de satisfacción del favor realizado
Perfil de usuario	Usuario que solicita favores
Personaje	Jorge
Contexto	Jorge solicitó un favor mediante la aplicación y una persona ya lo ayudó. Pues ha quedado satisfecho con la ayuda recibida, así que en este momento decide calificar a la persona otorgándole 5 estrellas.
Objetivo	Calificar el nivel de satisfacción con respecto a la persona que lo ayudó con el favor.
Tareas	<ul style="list-style-type: none"> ● Iniciar sesión ● Elegir el favor a calificar. ● Calificar el nivel de satisfacción.
Necesidades de información	Se necesita conocer el favor, la persona para colocar la calificación al favor.
Funcionalidades	<ul style="list-style-type: none"> ● Iniciar sesión. ● Seleccionar un favor de la lista de favores y ver su detalle. ● Calificar un favor en particular.

Realización de las tareas	<ul style="list-style-type: none"> ● Iniciar sesión. ● En la lista de favores solicitados, seleccionar el favor a calificar. ● Presionar el botón calificar. ● Seleccionar las estrellas a otorgar (entre 1 y 5 estrellas). ●
---------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Escenario 6	Calificar el trato y recompensa de la persona a la que ayudó
Perfil de usuario	Usuario que realiza favores
Personaje	Jessica
Contexto	Jessica dio clases de nivelación en matemáticas a una persona que le pidió mediante la app. Ella está contenta porque la persona a la que ayudó pagó lo acordado; así que lo califica, otorgándole 5 estrellas.
Objetivo	Calificar el nivel de satisfacción con respecto a la persona a la que ayudó con el favor.
Tareas	<ul style="list-style-type: none"> ● Iniciar sesión ● Elegir el favor a calificar. ● Calificar el nivel de satisfacción.
Necesidades de información	<ul style="list-style-type: none"> ● Se necesita conocer el favor, la persona a la que ayudó para calificar a la persona que lo ayudó
Funcionalidades	<ul style="list-style-type: none"> ● Iniciar sesión. ● Seleccionar un favor de la lista de favores realizados y ver su detalle. ● Calificar un favor realizado.
Realización de las tareas	<ul style="list-style-type: none"> ● Iniciar sesión. ● De las opciones presentadas (favores solicitados, favores realizados), seleccionar “favores realizados”. ● En la lista de favores realizados, seleccionar el favor a calificar. ● Seleccionar las estrellas a otorgar (entre 1 y 5 estrellas).

2.1.2.3 Flujo de interacción

El flujo de interacción muestra las diferentes rutas o decisiones que toma el usuario para cumplir con su objetivo, pues se obtiene una visión global de la interacción, mostrando gráficamente estas rutas y posibles decisiones. Además permite identificar las funcionalidades o información requerida por los usuarios en cada parte del flujo.

A continuación se muestra el flujo de interacción general de la aplicación “AcolitApp”.

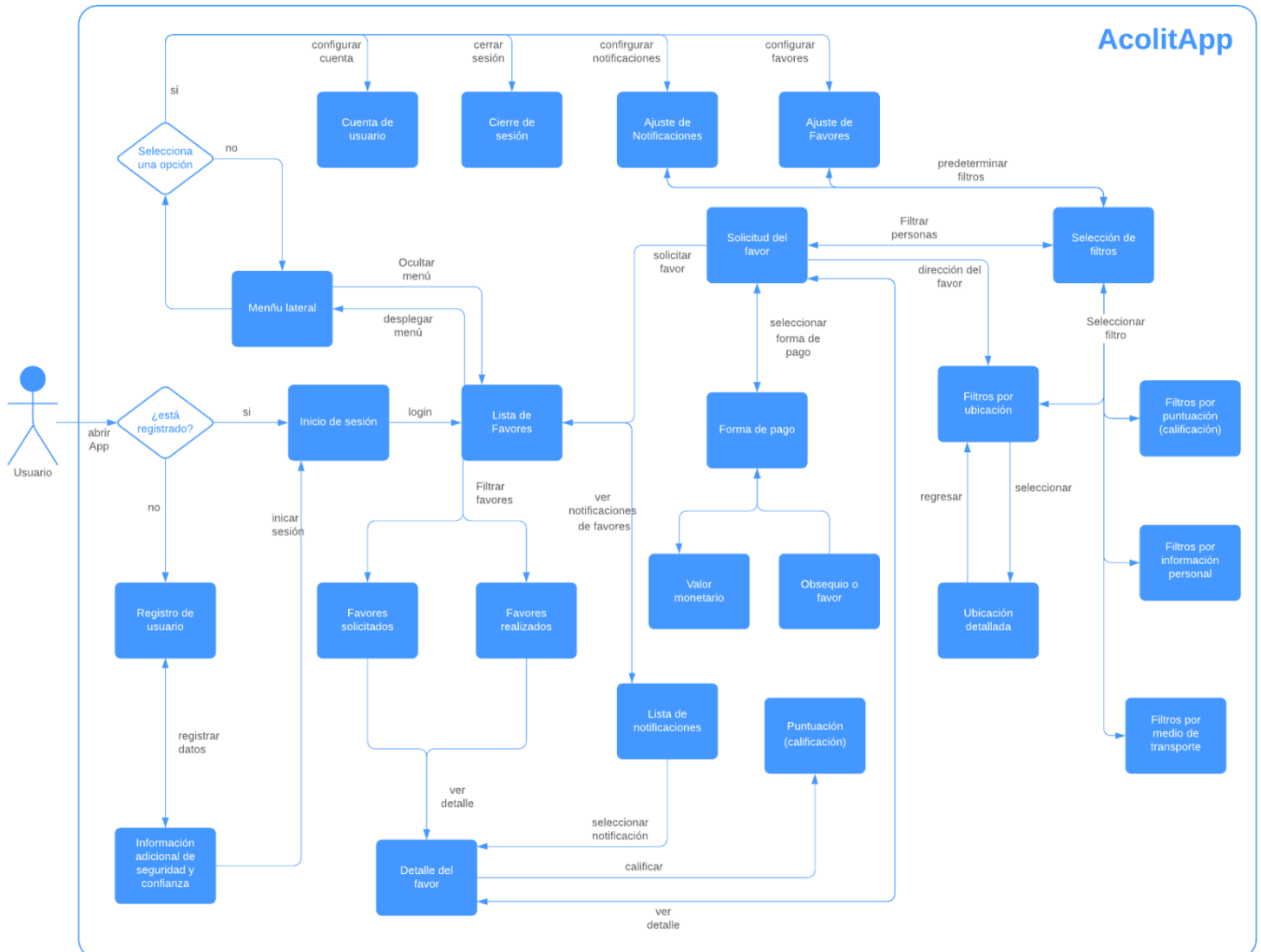


Figura 2.10: Flujo de interacción de la aplicación AcolitApp

2.1.3 Prototipado

En esta fase del Diseño Centrado en el Usuario se realiza el prototipado o diseño de pantallas de la aplicación.

Estos prototipos permiten evaluar y tomar decisiones sobre el diseño final de las pantallas antes de desarrollar la aplicación. Además no se necesitan recursos económicos o herramientas potentes para poder realizar estos prototipos, ya que se comienza realizando bocetos con papel y lápiz, pudiendo borrarlos o modificarlos las veces que sean necesarias según las evaluaciones que se realicen posteriormente sobre las mismas.

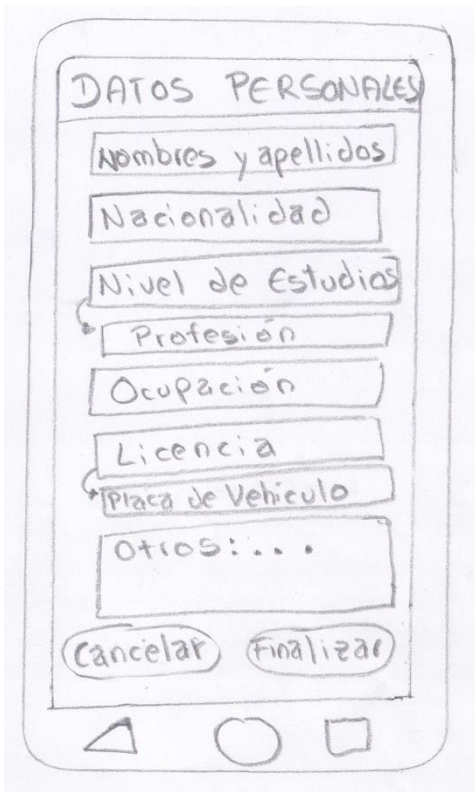
2.1.3.1 Sketches

Los sketches o bocetos iniciales son prototipos de baja fidelidad, los cuales se lo realiza a mano alzada, tratando de esquematizar el diseño de cada una de las interfaces de la aplicación. El objetivo es tener una idea de cómo será la interfaz junto con los elementos y su posición dentro de la misma.

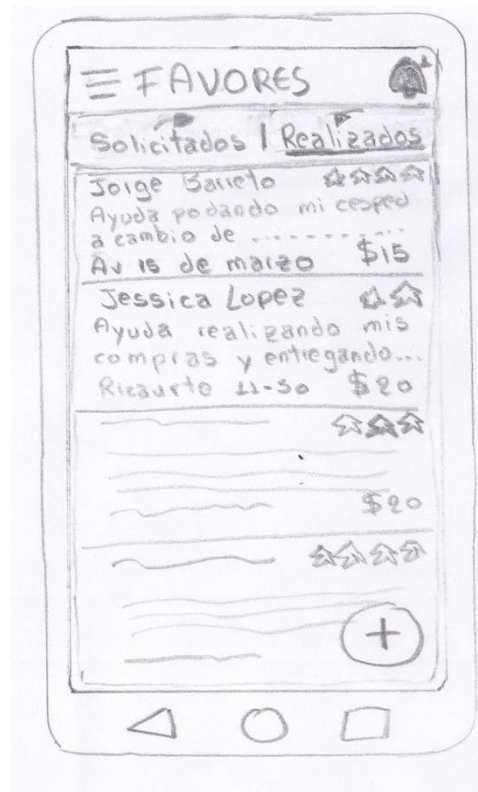
A continuación se presentan los bocetos iniciales de la aplicación AcolitApp:



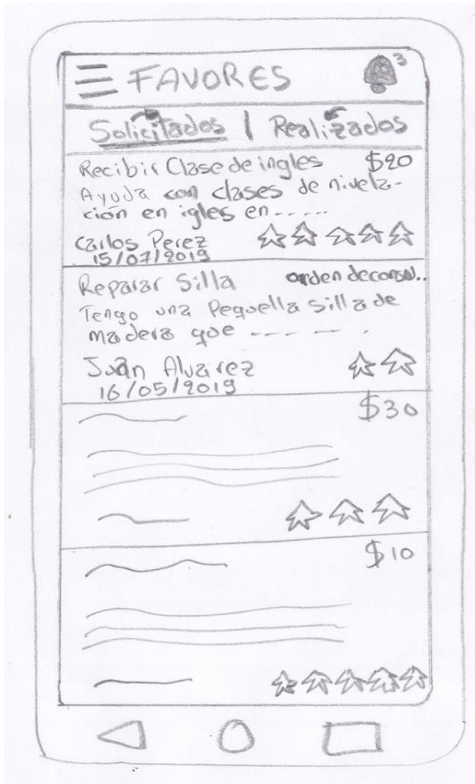
3. Datos personales



4. Favores realizados



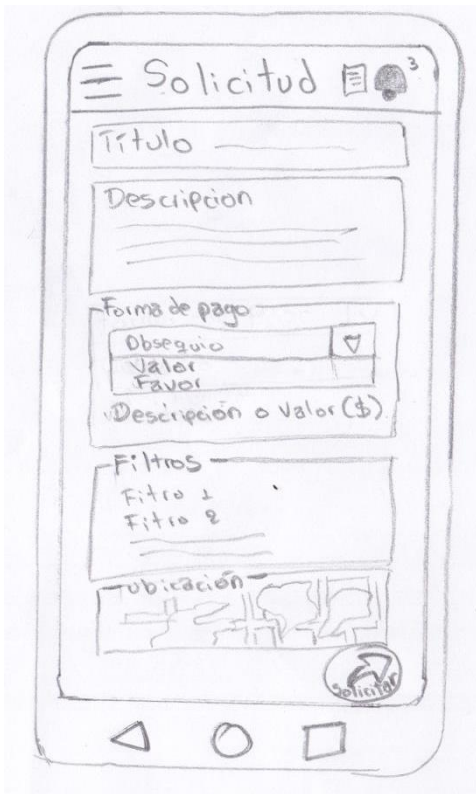
5. Favores solicitados



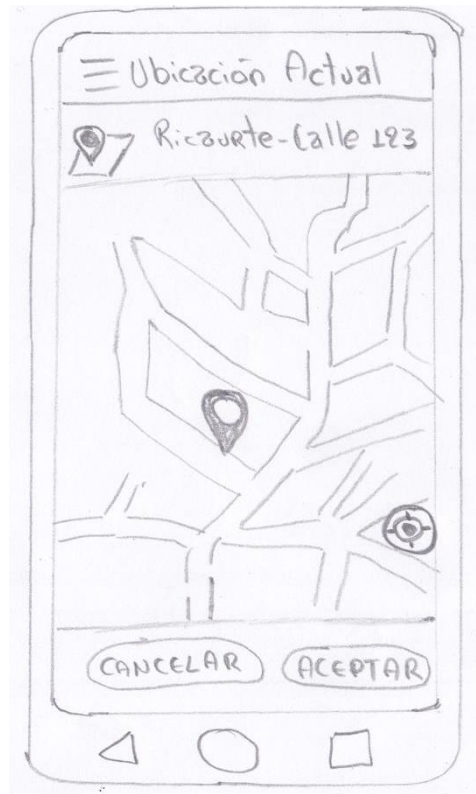
6. Menú lateral



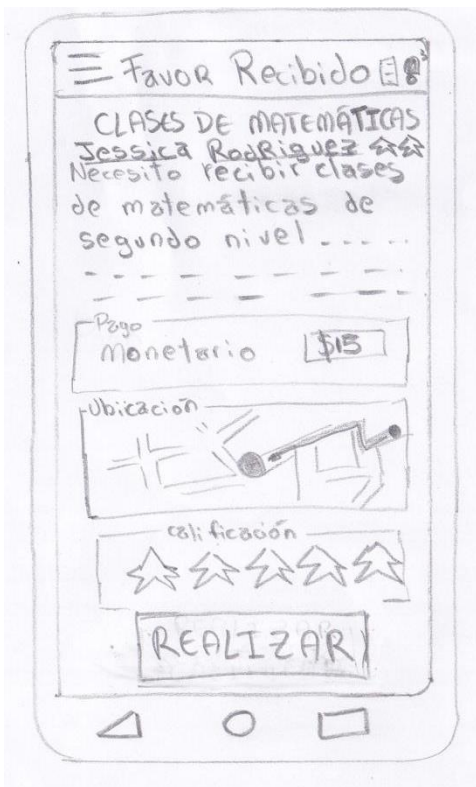
7. Solicitud del favor



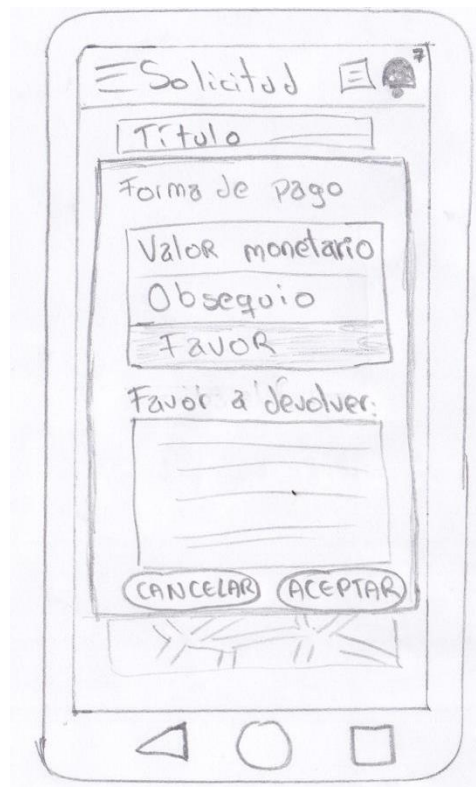
8. Seleccionar Ubicacion



9. Detalle del Favor Recibido



10. Forma de pago



11. Información de la persona



12. Puntuación / Calificación



2.1.3.2 Prototipos horizontales de alta fidelidad

En el prototipado de alta fidelidad se diseñan y construyen las interfaces utilizando herramientas digitales. Dichos diseños son lo más próximo posible a las pantallas que se van a implementar.

Estos prototipos horizontales contienen ya casi todas las funcionalidades y elementos del producto sin que haya interacción todavía entre las diferentes pantallas, lo cual se hará en la etapa de implementación.

A continuación se puede apreciar el prototipo de alta fidelidad de la aplicación, la misma que se realizó con la herramienta online "Marvel App dashboard" disponible en <https://marvelapp.com/>.

1. Inicio de sesión

AcolitApp 13:00

INICIO DE SESIÓN



Usuario _____

Contraseña _____

INICIAR SESIÓN

o

REGISTRAR

2. Registrar

AcolitApp 13:00

REGISTRO

Identificación _____

Email _____

Usuario _____

Contraseña _____

Confirmar contraseña _____

CREAR CUENTA

CANCELAR

3. Datos personales

AcolitApp 13:00

DATOS PERSONALES

Nombres y apellidos _____

Nacionalidad _____

Nivel de estudios _____

Profesión _____

Ocupación _____

Lincencia

Placa de vehículo _____

Otros ... _____

0/120

CANCELAR **FINALIZAR**

4. Favores realizados

AcolitApp 13:00

FAVORES

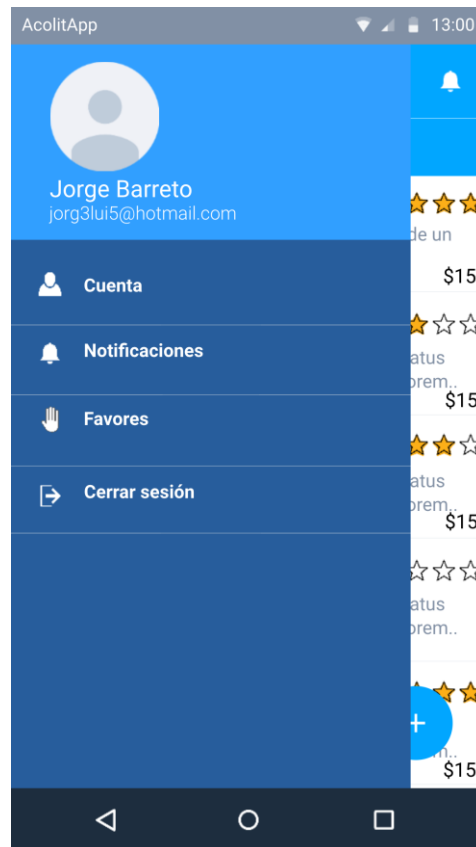
Solicitados | Realizados

- Jorge Barreto** ★★★★★
Ayuda podando mi cesped a cambio de un juego de ...
Av 15 de marzo \$15
- Lourdes viñanzaca** ★★★★☆
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium dolorem..
Av 15 de marzo \$15
- Carlos López** ★★★★★☆
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium dolorem..
Av 15 de marzo \$15
- Carmen Pérez** ★★★☆☆
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium dolorem..
- Luis Bravo** ★★★★★+
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium dolorem..
Av 15 de marzo \$15

5. Favores solicitados



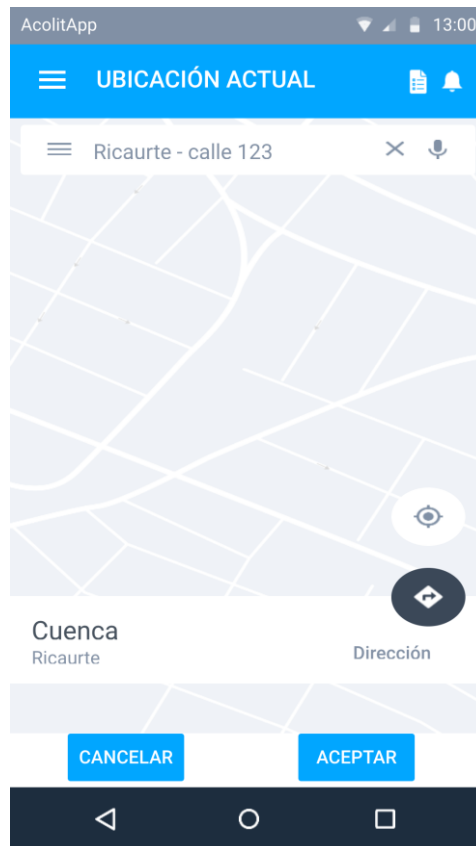
6. Menú lateral



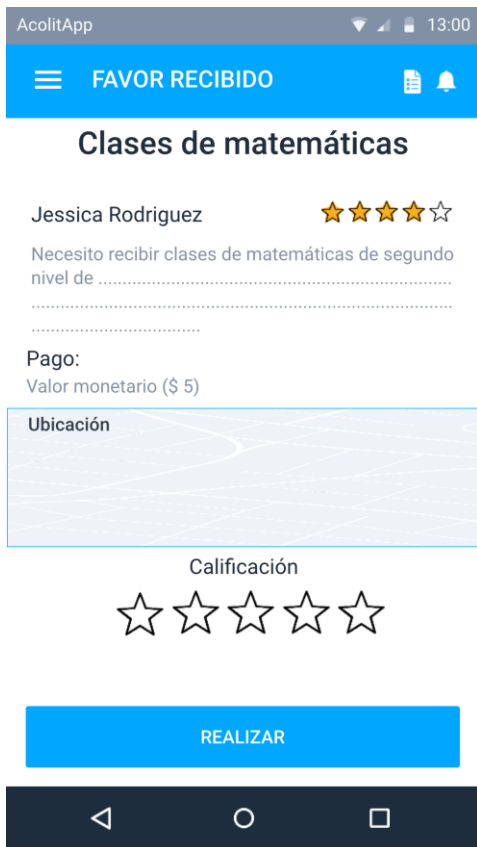
7. Solicitud del favor



8. Seleccionar Ubicación



9. Detalle del favor Recibido



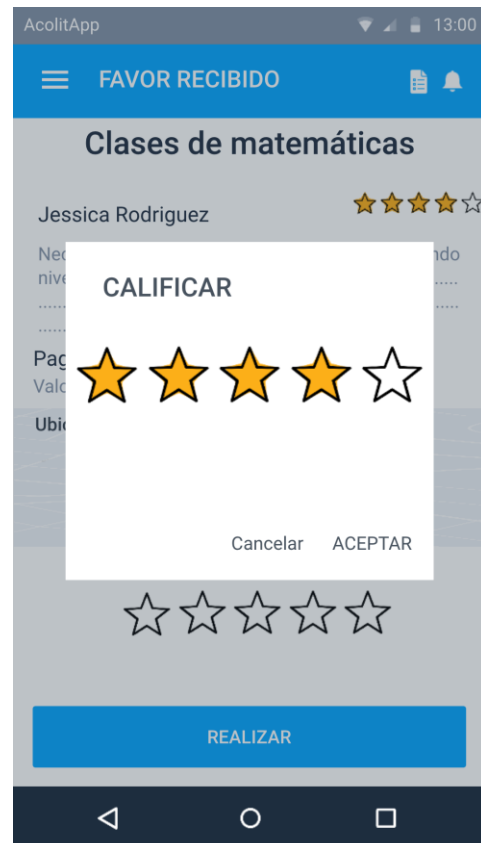
10. Forma de pago



11. Información de la persona



12. Puntuación / Calificación



A continuación se describe cada una de las pantallas diseñadas:

1. **Inicio de sesión:** esta es la pantalla inicial de la aplicación y sirve para iniciar sesión cuando el usuario ya está registrado en la App; por otra parte, también enlaza a la pantalla de registro si el usuario aún no está registrado.
2. **Registrar:** pantalla que permite al usuario registrarse en la aplicación para poder utilizarla. Se pide información necesaria para crear la cuenta (identificación, correo, usuario y contraseña).
3. **Datos personales:** esta pantalla permite registrar datos personales e importantes para brindar seguridad y confianza entre las personas que usan la aplicación. Además esta información también será útil cuando se apliquen filtros solicitando personas que cumplan con cierta información.
4. **Favores realizados:** la pantalla muestra todos los favores que el usuario ha realizado o ayudando a los usuarios que lo solicitaron. Entre la información a mostrar está el nombre de la persona que ayudó, la calificación que obtuvo, el favor realizado, la dirección en la que ayudó y la recompensa o valor recibido por la ayuda prestada.
5. **Favores solicitados:** la pantalla muestra todos los favores que el usuario solicitó y lo ayudaron. Entre la información que se muestra está el título y descripción del favor solicitado, el valor o recompensa entregada, así como el nombre de la persona que lo ayudó, la fecha y la calificación otorgada a dicha persona. Desde esta pantalla se puede acceder a las notificaciones o a la pantalla donde se puede solicitar un favor.
6. **Menú lateral:** esta interfaz contiene las opciones de configuración del menú lateral. Desde esta pantalla se puede acceder a diferentes configuraciones como: configuración de la cuenta, ajuste de notificaciones y favores. También contiene la opción para cerrar sesión.
7. **Solicitud del favor:** esta es la pantalla que permite crear una nueva solicitud de favor y enviarlo a las demás personas. Aquí se describe el favor, el valor monetario o recompensa a entregar y la ubicación de donde se realizará el favor. Además enlaza a la pantalla donde se establecen los diferentes filtros y así enviar la solicitud solo a personas que cumplan con esos criterios.
8. **Seleccionar Ubicación:** en esta interfaz, el usuario podrá elegir la ubicación del favor a realizar o establecer un filtro de ubicación. Aquí se podrá elegir la ubicación actual o seleccionar otro punto en el mapa.
9. **Detalle del favor recibido:** esta pantalla visualiza los detalles del favor recibido a través de una notificación. Se podrá ver la descripción del favor, la persona que lo solicita, el valor monetario o recompensa que brinda, el lugar donde debe realizarlo, entre otros. Además aquí puede aceptar y otorgar ayuda a la persona que lo solicitó, o simplemente ignorar la solicitud.

10. **Forma de pago:** es una pantalla embebida dentro de un cuadro de diálogo que se despliega al momento de establecer la forma de pago del favor que se va a solicitar. La interfaz muestra las distintas formas de pago (valor monetario, obsequio y favor) y dependiendo de cual se seleccione se deberá ingresar información específica.
11. **Información de la persona:** la interfaz visualiza información detallada de una persona en particular, la misma que se ofreció ayudar con el favor solicitado. En esta pantalla se puede aceptar la ayuda de esa persona o simplemente ignorar y seleccionar otra persona.
12. **Puntuación / Calificación:** es una pantalla embebida en un cuadro de diálogo y permite calificar (dar una puntuación de 1 a 5 estrellas) el nivel de satisfacción con respecto a la persona que lo ayudó con el favor o calificar el trato o recompensa recibida por haber realizado dicho favor.

Estos diseños se han ido elaborando de manera iterativa al igual que los bocetos debido a que han ido evolucionando y mejorando gracias a un test sencillo que se explica en la fase de evaluación.

2.1.4 Evaluación

En la fase de evaluación se realizan pruebas sobre los prototipos de las pantallas con el objetivo de mejorarlos, ya que el proceso de DCU es un proceso iterativo, y por lo tanto, hay que ir evaluando los diseños y corrigiendo los errores.

En este caso, se hizo un test inicial y sencillo previo a la implementación de la App con el objetivo de comprobar si las pantallas consideradas, la información presentada y los elementos que se pretende utilizar son los adecuados. También se realizó la planificación para una evaluación más robusta del aplicativo que se realizará cuando las pantallas estén implementadas.

2.1.4.1 Test inicial

Para la evaluación inicial de los prototipos se tuvo contacto con algunas de las personas encuestadas y que fueron parte de la fase 1 del DCU. Las personas involucradas pertenecen a 3 grupos distintos (compañeros de trabajo, familia y amigos del barrio), con los cuales pude reunirme para realizar preguntas puntuales y generar una lluvia de ideas con respecto a los prototipos diseñados.

Los pasos que se siguieron para realizar este test inicial son los siguientes:

1. Se imprimieron los bocetos y los prototipos diseñados en la herramienta digital.
2. Se les explicó el objetivo de la aplicación y las diferentes funcionalidades que se implementarán de acuerdo al análisis realizado sobre encuestas que ellos

respondieron. También se explicó cómo se llevará a cabo el test de usabilidad.

3. Se entregaron a cada persona y de manera aleatoria todos los bocetos o todos los prototipos de alta fidelidad iniciales, para que los revisen y analicen durante 15 minutos.
4. Después de que revisaron las pantallas se lanzó algunas preguntas puntuales a cada persona para saber si nuestro diseño es entendible y de fácil uso.

Las preguntas lanzadas fueron las siguientes:

- Seleccione la pantalla donde usted crea que se solicita un favor.
 - ¿Qué pantalla cree que se mostraría primero al abrir la aplicación?
 - ¿Cuándo cree que se mostrará la pantalla con título “Datos personales”?
 - ¿Qué información cree que falta incluir en la pantalla con título “Datos personales”?
 - Seleccione el elemento (botón, imagen, texto, etc.) que debería presionarse para que aparezca la pantalla que contiene el “menú lateral”.
 - ¿Qué cree que sucedería si se presiona el botón que tiene el signo “+”
 - ¿Qué cree que significan las estrellas que se muestran a lado de cada nombre de la persona?
5. Se les propuso que trabajen en parejas o grupos de 3 personas para que establezcan el orden en el que deben presentar o navegar entre las pantallas para los siguientes casos:
 - **Caso 1:** El usuario quiere solicitar un favor mediante la Aplicación. Necesita que le ayuden comprando algunos materiales a cambio una orden de consumo para 2 personas en su restaurante.
 - **Caso 2:** El usuario quiere calificar el nivel de satisfacción por el favor recibido.
 - **Caso 3:** Al usuario revisa los favores solicitado por otras personas, selecciona y acepta el favor.
 6. Se organizó una lluvia de ideas para que den recomendaciones sobre las pantallas que hacen falta incluir o que deberían modificarse, así como la información que se presenta en cada pantalla y sobre todo si consideran que los elementos utilizados (íconos, títulos, mapas, etc.) son fáciles de entender.

Conclusiones

En conclusión, los pasos realizados para evaluar el prototipo inicial de cada pantalla fueron indispensables para mejorar los diseños propuestos y así tener un modelo que se adapte a la necesidad de los usuarios y que sobre todo sea fácil de usar.

En cuanto a las preguntas realizadas se pudo notar que faltaba una pantalla para establecer los filtros de una determinada categoría, por ejemplo para establecer los filtros según la información personal de la persona, o según la ubicación, calificación, o medio de transporte.

Este proceso también permitió identificar en la pantalla de inicio de sesión la mezcla de dos lenguajes (inglés y español); pues, existía un botón con el texto “Registrar”; y otros con el texto “Login”. De igual manera sucedió con los inputText que tenían los textos “username” y “password”. Por otra parte, se decidió cambiar los títulos de algunas pantallas, así como también algunos elementos o íconos debido a que no daban a entender la funcionalidad que realizarían.

En fin, estos detalles y sugerencias realizadas por las personas fueron consideradas para mejorar el diseño de la aplicación, hacerla más comprensiva e intuitiva y sobretodo facilitar su uso.

2.1.4.2 Planificación de la evaluación

El objetivo de este apartado es la planificación de la evaluación del prototipo en las siguientes iteraciones y sobre todo cuando se implemente la misma, dado que es una fase iterativa que evalúa los diseños y corrige errores a medida que la aplicación avanza. Este proceso se realizará a través de test de usuarios.

Test de usuarios

Con el test de usuarios se logrará conocer con detalle la facilidad de uso, eficiencia y la satisfacción del usuario con respecto a la App. Para esto, se realizará un pre-test con preguntas que se realizó en las encuestas analizadas en la primera fase, y un post-test con preguntas relacionadas a las tareas que forman parte del test.

Cuestionario Pre-test

Antes de que los usuarios realicen las tareas de testing, deberán responder un cuestionario para obtener datos socio-demográficos que serán de gran ayuda para analizar los resultados obtenidos. Si algunos de los usuarios ya contestaron el cuestionario de la primera fase no deberán contestarlo, debido a que contemplará las mismas preguntas de la primera sección y algunas de la segunda sección de este cuestionario.

Las preguntas del cuestionario Pre-test son las siguientes:

Cuestionario Pre-test
● Edad
● Estado civil
● Sexo
● Nivel de estudios

● Ocupación/dedicación
● Número de hijos
● ¿Dispone de algún Smartphone o tableta?
● ¿Cuál es el nivel de uso de aplicaciones en su Smartphone?
● ¿Ha comprado o compraría alguna Aplicación móvil en las tiendas de aplicaciones?
● ¿Cuánto pagaría por una Aplicación móvil?
● ¿Acostumbra puntuar y/o escribir comentarios de aplicaciones o sitios que has visitado?
● ¿Toma en cuenta las valoraciones y/o comentarios de otros usuarios antes de descargar aplicaciones en su teléfono?

Tareas a realizar

Las tareas que realizarán los usuarios son parecidos a los casos del paso 5 del test inicial. Pues, a las personas se les dará una serie de tareas o casos específicos para que la realicen utilizando la aplicación desarrollada.

Entre las tareas o casos a realizar se encuentran los siguientes:

Casos o tareas a realizar
● Tarea 1: registrarse en la aplicación.
● Tarea 2: revisar los favores solicitados por las personas y aceptar ayudar con un favor solicitado.
● Tarea 3: calificar a la persona que ayudó con un favor.
● Caso 1: solicitar un favor para que le ayuden comprando comida a cambio una orden de consumo para 2 personas en su restaurante.
● Caso 2: solicitar un favor para que una persona con motocicleta le lleve las llaves de su oficina ubicada en la Universidad de Cuenca.

Post-test

Para medir la usabilidad y funcionalidad de la aplicación, así como la satisfacción de los usuarios que realizaron el test de la App se realizará el cuestionario Post-test.

Las preguntas del cuestionario Post-test son las siguientes:

Cuestionario Post-test
● De una escala del 1 al 10, siendo 1 “muy fácil” y 10 “muy difícil” ¿Que tan fácil fue utilizar la aplicación?
● ¿Qué tan atractiva es visualmente la aplicación? (nada atractiva, poco atractiva, regular, normal, atractiva, muy atractiva)
● ¿La aplicación le ha permitido cumplir con sus tareas? (sí, no, a veces)
● ¿Exigieron ocasiones en las que no sabía en qué lugar de la aplicación se encontraba? (siempre, casi siempre, a veces, rara vez, nunca).

<ul style="list-style-type: none"> • Los íconos que aparecen en los menús, botones, etc. determinan claramente la función que realizan o información que mostrarán (siempre, casi siempre, algunas veces, rara vez, nunca).
<ul style="list-style-type: none"> • ¿Qué tan fácil le ha resultado realizar las tareas? (muy fácil, fácil, normal, difícil, muy difícil)
<ul style="list-style-type: none"> • ¿Crees que el número de pantallas presentados para realizar las tareas es el correcto? (siempre, a casi siempre, a veces, rara vez, nunca)
<ul style="list-style-type: none"> • ¿La información visualizada en cada pantalla es la correcta? (siempre, a casi siempre, a veces, rara vez, nunca)
<ul style="list-style-type: none"> • ¿Crees que el tamaño de los elementos es el correcto? (siempre, casi siempre, a veces, rara vez, nunca)
<ul style="list-style-type: none"> • ¿Se ha sentido cómodo realizando las tareas? (si, no)
<ul style="list-style-type: none"> • ¿Qué es lo que más y menos te ha gustado de la aplicación?
<ul style="list-style-type: none"> • ¿Añadiría alguna funcionalidad a la aplicación? ¿Cuál?
<ul style="list-style-type: none"> • ¿Eliminaría alguna funcionalidad a la aplicación? ¿Cuál?
<ul style="list-style-type: none"> • ¿Cambiarías algo de la aplicación?, ¿Qué cosa?
<ul style="list-style-type: none"> • ¿Utilizarías la aplicación para solicitar o realizar favores?

2.2 Diseño técnico

En esta sección se usa toda la información obtenida durante todas las fases del diseño centrado en el usuario para definir y detallar los casos de uso del sistema, así como las entidades que componen la aplicación y la arquitectura del mismo.

A continuación se definen los casos de uso y el diseño de la arquitectura:

2.2.1 Definición de los casos de uso

En esta sección se definirán los casos de uso del aplicativo, los mismos que servirán para establecer las funcionalidades de la App.

2.2.1.1 Actores del sistema

Usuario

En este caso solo existe un actor, que es el usuario como tal y representa a todos los usuarios que utilicen la aplicación sin importar el perfil que tenga, debido a que si es un actor que usualmente usa la aplicación para solicitar favores, en ciertas ocasiones también podría ayudar a realizarlos, y viceversa. Además todo usuario podrá utilizar todas las características y funcionalidades de la aplicación.

2.2.1.2 Diagrama UML de los casos de uso

A continuación se muestra el diagrama UML de los casos de uso con todas las interacciones existentes entre el usuario y la aplicación móvil.

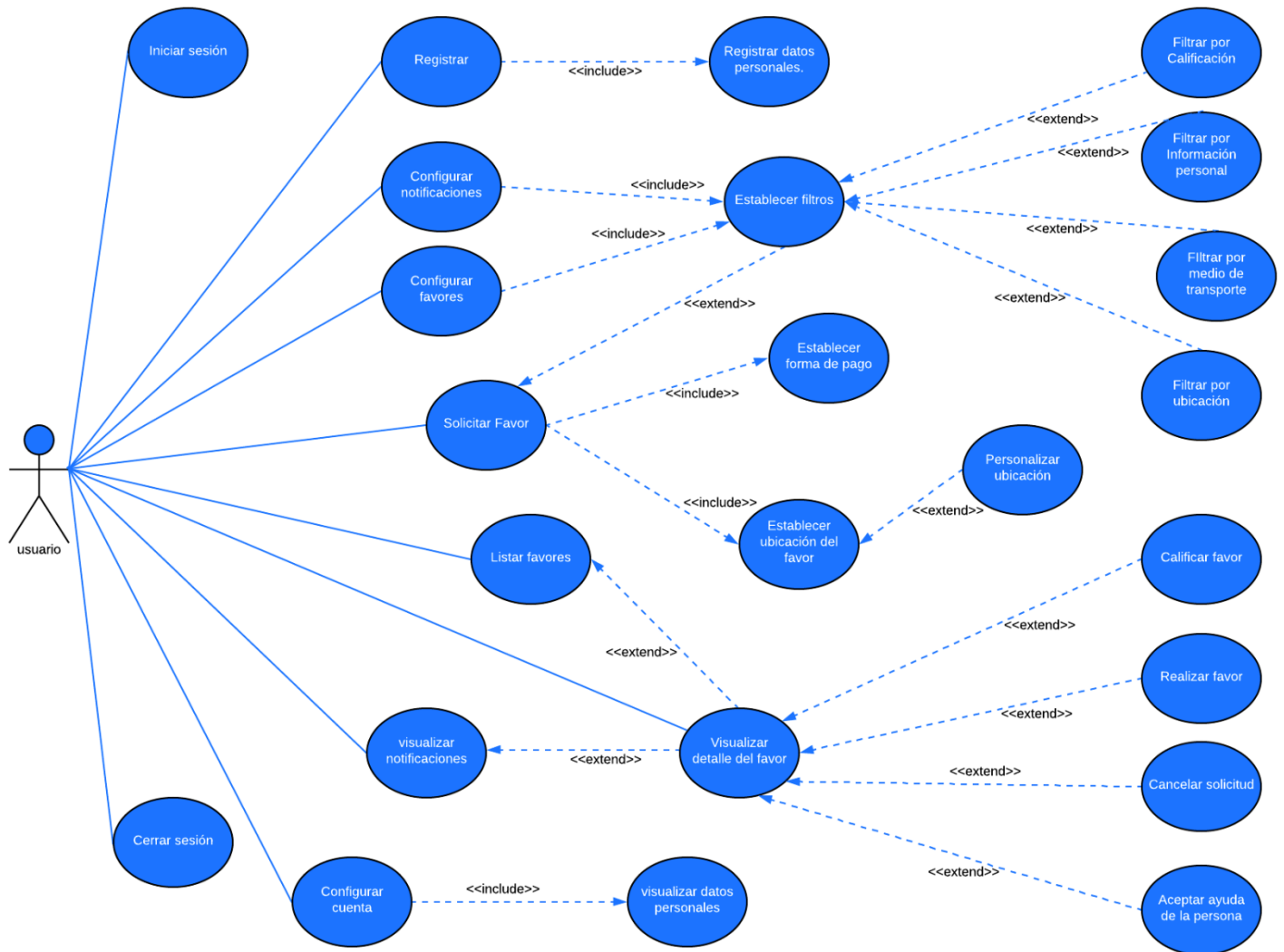


Figura 2.11: Diagrama UML de los casos de uso de la aplicación.

2.2.1.3 Detalle de los casos de uso

A continuación se detalla cada uno de los casos de usos plasmados en el diagrama:

CU-001 Registrar	
Actor	Usuario
Prioridad	Alta
Descripción	Permite registrar un nuevo usuario en el sistema, creando una cuenta de usuario para que pueda iniciar sesión y hacer uso del aplicativo.

Precondiciones	<ul style="list-style-type: none"> No estar registrado previamente con ese usuario, correo e identificación.
Flujo normal	<ol style="list-style-type: none"> Abrir la aplicación. Presionar en el botón “Registrar”. Ingresar los datos solicitados en el formulario (correo, usuario y contraseña). Presionar en el botón “Registrar”. Ingresar al menos los datos obligatorios que aparecen en el formulario de datos personales. Presionar el botón “Finalizar”,
Flujo alternativo	<ol style="list-style-type: none"> 1.1 Si el usuario ya existe, aparece un mensaje indicando que el usuario ya existe. 1.2 Si la contraseña y la confirmación de la misma no existe, aparece un mensaje indicando que las contraseñas son distintas. 1.3 Si no se han llenado los campos obligatorios, muestra un error de los campos obligatorios.
Postcondiciones	<ul style="list-style-type: none"> El usuario debe estar registrado Debe poder iniciar sesión con ese usuario

CU-002 Iniciar Sesión	
Actor	Usuario
Prioridad	Alta
Descripción	Permitir al usuario iniciar sesión para poder acceder y hacer uso de las funcionalidades de la aplicación.
Precondiciones	<ul style="list-style-type: none"> Tener un usuario registrado en la aplicación. Recordar su usuario y contraseña
Flujo normal	<ol style="list-style-type: none"> Ingresar el nombre de usuario y la contraseña. Presionar el botón “Iniciar sesión”. El sistema valida los datos ingresados. Si la validación es correcta, se concede el acceso.
Flujo alternativo	<ol style="list-style-type: none"> 4.1 Si el usuario no existe, aparece un mensaje informando que no existe el usuario. 4.2 Si la contraseña es incorrecta, aparece un mensaje informando que la contraseña no concuerda con el usuario. 4.3 Si los campos están vacíos, muestra un error con los campos obligatorios.
Postcondiciones	<ul style="list-style-type: none"> Acceder al sistema.

CU-003 Cerrar sesión	
Actor	Usuario
Prioridad	Alta
Descripción	Permite al usuario cerrar la sesión activa en la aplicación para poder iniciar con la misma u otra cuenta.

Precondiciones	<ul style="list-style-type: none"> • EL usuario debe tener una sesión activa en la App.
Flujo normal	<ol style="list-style-type: none"> 1. Presionar en el ícono de menú. 2. En el menú lateral, presionar la opción “cerrar sesión”. 3. El sistema visualiza la pantalla de inicio de sesión
Flujo alternativo	1.1 El usuario desliza su dedo hacia la parte lateral izquierda de la pantalla y aparece el menú lateral.
Postcondiciones	<ul style="list-style-type: none"> • Se cierra la sesión del usuario y se muestra la pantalla de inicio de sesión.

CU-004 Ver Información Personal	
Actor	Usuario
Prioridad	Normal
Descripción	Permite al usuario acceder a la información de su cuenta y visualizar sus datos personales.
Precondiciones	<ul style="list-style-type: none"> • La sesión del usuario debe estar activa.
Flujo normal	<ol style="list-style-type: none"> 1. Presionar en el ícono de menú. 2. En el menú lateral, presionar la opción “cuenta”. 3. Se visualizará la información de la cuenta e información personal.
Flujo alternativo	Ninguno
Postcondiciones	<ul style="list-style-type: none"> • La información del usuario es visualizada correctamente.

CU-005 Listar favores.	
Actor	Usuario
Prioridad	Alta
Descripción	Permite listar los favores que el usuario ha realizado así como también los favores que ha solicitado.
Precondiciones	<ul style="list-style-type: none"> • Tener un usuario registrado.
Flujo normal	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Automáticamente muestra esa pantalla porque es la pantalla inicial. 3. Seleccionar la pestaña “Solicitados” para ver los favores solicitados o la pestaña “realizados” para ver los favores realizados. 4. EL sistema filtrará y mostrará los favores según la pestaña seleccionada.
Flujo alternativo	<ol style="list-style-type: none"> 1.1 Tiene la sesión iniciada, y está en la pantalla de favores, se listará por defecto los favores solicitados. 1.2 Si se encuentra en cualquier otra pantalla, presionar en el icono de lista para ver la lista de favores.
Postcondiciones	<ul style="list-style-type: none"> • Los favores deberán estar listados según la pestaña seleccionada.

CU-006 Solicitar favor	
Actor	Usuario
Prioridad	Alta
Descripción	Permite al usuario escribir y solicitar un nuevo favor para que se notifique a los usuarios y puedan ayudarlo.
Precondiciones	<ul style="list-style-type: none"> ● Sesión activa. ● Otorgar permisos de ubicación.
Flujo normal	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. El sistema muestra la lista de favores. 3. Presionar el botón flotante que tiene el signo “+”. 4. Se visualizará la pantalla de solicitud de favores. 5. Llenar el título y descripción del favor. 6. Presionar la opción forma de pago y seleccionar (valor monetario, obsequio, favor). 7. Introducir los datos del pago. 8. Clic en aceptar. 9. Seleccionar la ubicación en el mapa. 10. Presionar el botón “Solicitar”. 11. La solicitud se realizará y los demás usuarios podrán verla.
Flujo alternativo	<p>6.1 El usuario selecciona la forma de pago “valor monetario”, entonces debe introducir el valor.</p> <p>6.2 El usuario selecciona la forma de pago “obsequio” o “favor”, entonces debe ingresar la descripción del obsequio o valor.</p> <p>7.1 El valor o descripción de la forma de pago ingresada no es válida. Se muestra un mensaje con el aviso.</p> <p>7.2 No se seleccionó una forma de pago, el sistema avisa con un mensaje.</p> <p>10.1 El usuario no ha ingresado información en los campos obligatorios, muestra un mensaje informando los campos que son obligatorios.</p>
Postcondiciones	<ul style="list-style-type: none"> ● El favor solicitado debe estar enviado. ● En la lista de favores solicitado debe aparecer el favor enviado.

CU-007 Visualizar detalle del favor.	
Actor	Usuario
Prioridad	Alta
Descripción	Permite revisar el detalle de un favor solicitado o realizado.
Precondiciones	<ul style="list-style-type: none"> ● Mantener la sesión activa del usuario
Flujo normal	<ol style="list-style-type: none"> 1. Abrir la lista de favores. 2. En la lista, presionar el favor que quiere visualizar su detalle. 3. La app visualizará la pantalla con el detalle del favor.
Flujo alternativo	2.1 Selecciona el favor deslizando el dedo por la pantalla.
Postcondiciones	<ul style="list-style-type: none"> ● Debe estar abierta la pantalla con los detalles del favor seleccionado.

CU-008 Ayudar con el favor	
Actor	Usuario
Prioridad	Alta
Descripción	Permite al usuario aceptar el favor solicitado, con el objetivo de hacer el trato y ayudar a la persona que solicitó.
Precondiciones	<ul style="list-style-type: none"> ● Loguearse en el sistema. ● Que el favor no haya sido ya realizado por alguien con anterioridad. Y tenga el estado SOLICITADO.
Flujo normal	<ol style="list-style-type: none"> 1. Visualizar el detalle del favor siguiendo el flujo alternativo 1.1 o 1.2 del caso de uso CU-008. 2. Presionar en el botón “Realizar”. 3. Una vez aceptado por el solicitante, se establece el trato.
Flujo alternativo	<ol style="list-style-type: none"> 1.1 EL botón “Realizar” está deshabilitado debido a que otra persona ya hizo trato con el solicitante. 1.2 El solicitante, ya se hizo trato con otra persona justo cuando se presionó el botón “Realizar”. Se muestra un mensaje informando que no se puede hacer el trato. 3.1 El solicitante no acepta la ayuda ofrecida.
Postcondiciones	<ul style="list-style-type: none"> ● El solicitante verá que el fue alguien se ofreció a ayudar ● Se mostrará en el detalle, un mensaje que confirme que el trato está realizado.

CU-009 Calificar favor	
Actor	Usuario
Prioridad	Normal
Descripción	Permite colocar una calificación o puntaje que representa el nivel de satisfacción con respecto a la persona que lo ayudó o que representa el trato y recompensa recibida por la ayuda otorgada.
Precondiciones	<ul style="list-style-type: none"> ● Haber solicitado o realizado el favor. ● Si solicitó el favor, la otra persona debe haber aceptado la ayuda y establecer el trato.
Flujo normal	<ol style="list-style-type: none"> 1. Visualizar el detalle del favor siguiendo el flujo normal del caso de uso CU-008. 2. Presionar sobre las estrellas que aparecen en el detalle. 3. Seleccionar una de las cinco estrellas visualizadas. 4. La App pintará las estrellas seleccionadas. 5. Presionar el botón “aceptar”.
Flujo alternativo	6.1 El usuario presiona el botón “cancelar”, la calificación no se realizó.
Postcondiciones	<ul style="list-style-type: none"> ● En la pantalla que visualiza el detalle del favor, debe mostrar la calificación asignada.

2.2.2 Diseño de la arquitectura

En este apartado se definirá la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para manejar la información y los diferentes procesos que se realizan con la aplicación. Además se representará la estructura y comunicación existente entre el servidor y la App.

2.2.2.1 Diagrama de la base de datos

Para el diseño de la base de datos se definirán objetos de datos que se han ido identificando en las fases y apartados anteriores. Los datos se almacenarán en una base de datos PostgreSQL y estará situada en el lado del servidor.

El diagrama de esta BD es la siguiente:

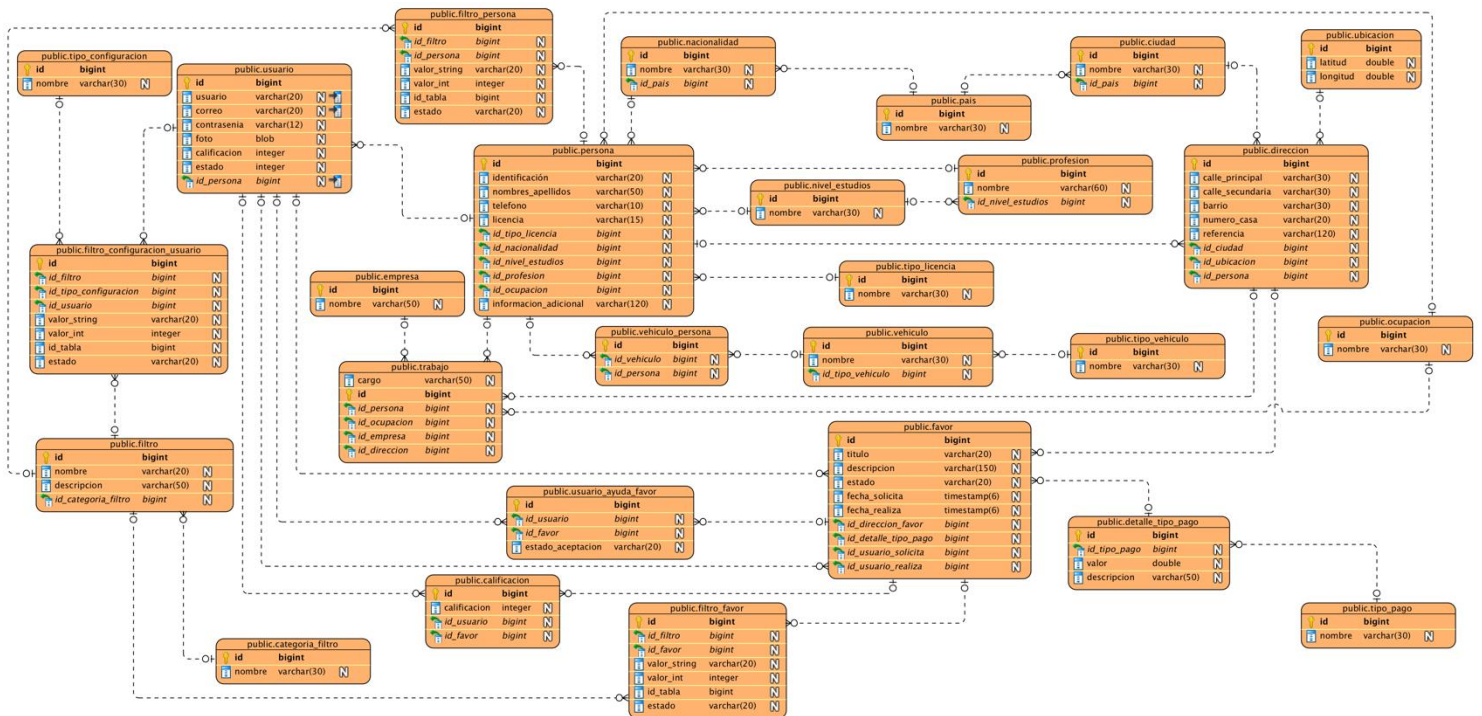


Figura 2.12: Diagrama de la base de datos de la aplicación

2.2.2.2 Diagrama de clases

A continuación se muestra el diagrama de clases de la aplicación:

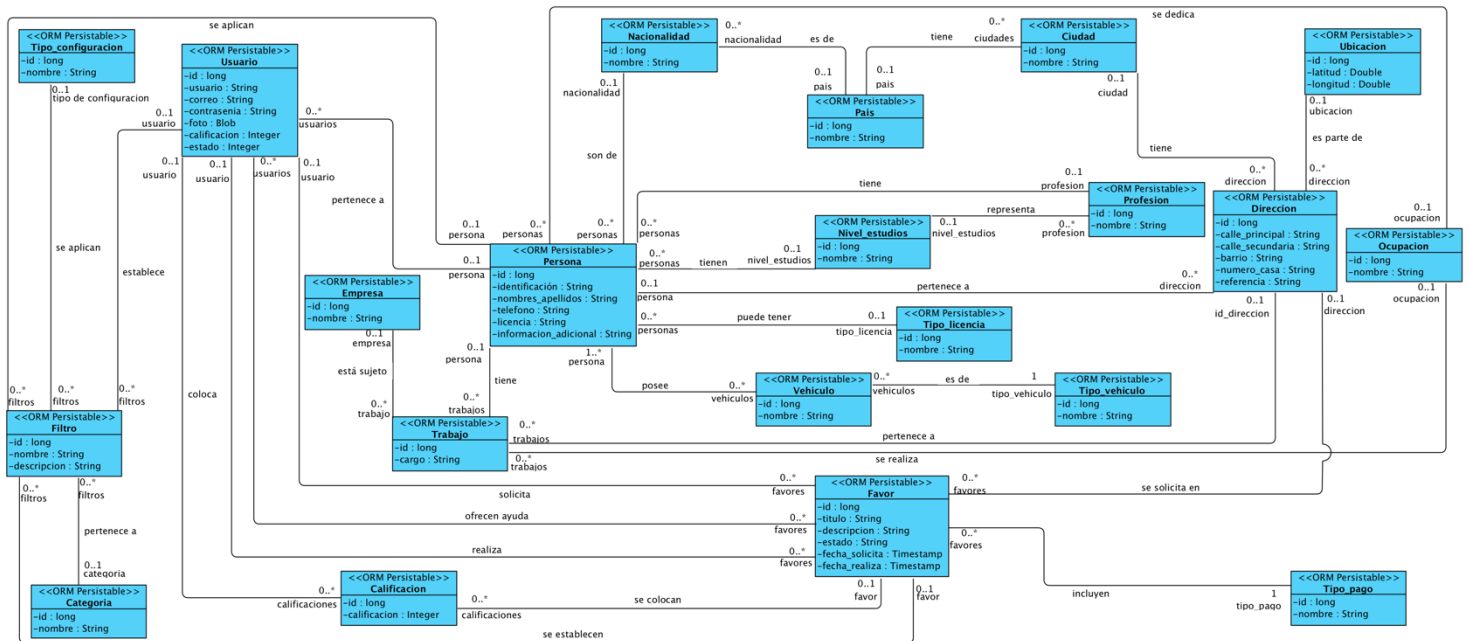


Figura 2.13: Diagrama de clases de la aplicación

Las clases que contienen información de los objetos de datos, estarán dentro de un paquete específico al igual que las clases que contienen la lógica de negocio, objetos de transferencia de datos, servicios, etc.

A continuación se visualiza el diagrama de paquetes:

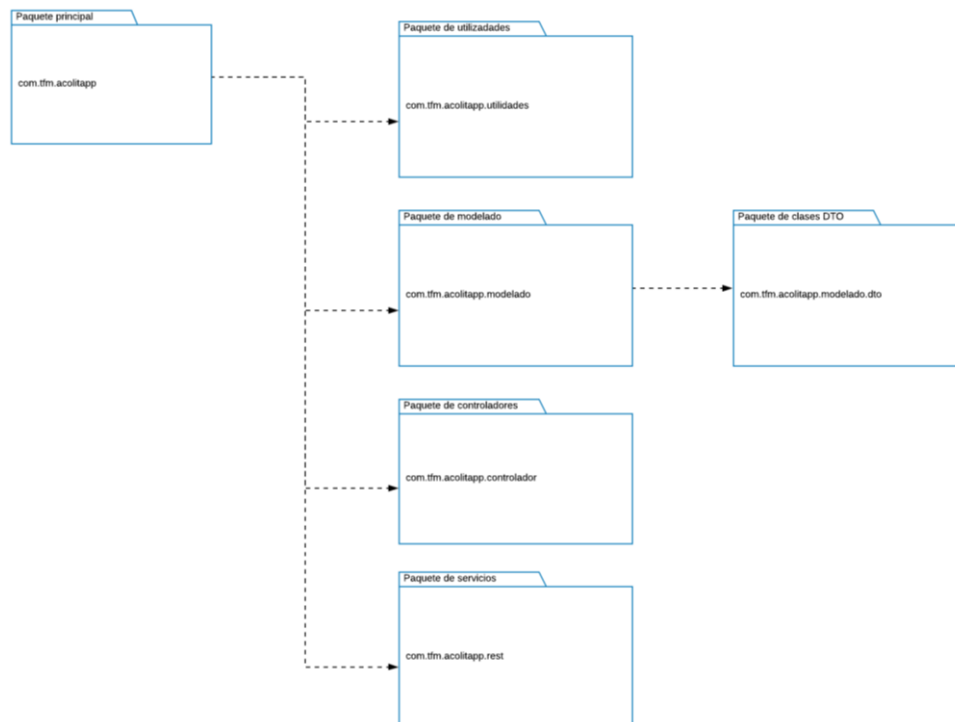


Figura 2.14: Diagrama de paquetes de la aplicación

2.2.2.3 Arquitectura general del sistema

En esta sección se establece la arquitectura general o de alto nivel del sistema. En este caso se decidió utilizar el modelo cliente-servidor, en donde los clientes realizarán peticiones al servidor mediante protocolos de comunicación para que este devuelva una respuesta según lo solicitado por el cliente.

Los 3 componentes principales de la arquitectura cliente-servidor son:

Cliente: el cliente en este caso es la aplicación móvil instalada en los diversos dispositivos móviles con sistemas operativos Android y iOS.

Servidor: es el servidor de aplicaciones y de base de datos, el cual se encuentra separado de la aplicación móvil y es el que tendrá toda la lógica de negocio y sobre todo la base de datos del sistema.

Comunicación: La comunicación se realizará mediante llamadas, utilizando el protocolo http. Se llamarán a servicios REST que estarán implementados en el servidor, mientras que los datos se transferirán usando el formato JSON.

Se ha elegido la arquitectura cliente-servidor porque soporta a muchos clientes conectándose simultáneamente debido a que se pueden implementar métodos síncronos y asíncronos. Además se puede balancear la carga del servidor con los demás servidores cuando la demanda sea muy alta.

Por otra parte, la aplicación usa los servicios de google, por ejemplo para determinar una ubicación, se lo hace utilizando el servicio de google maps.

A continuación se puede visualizar la arquitectura general del sistema:

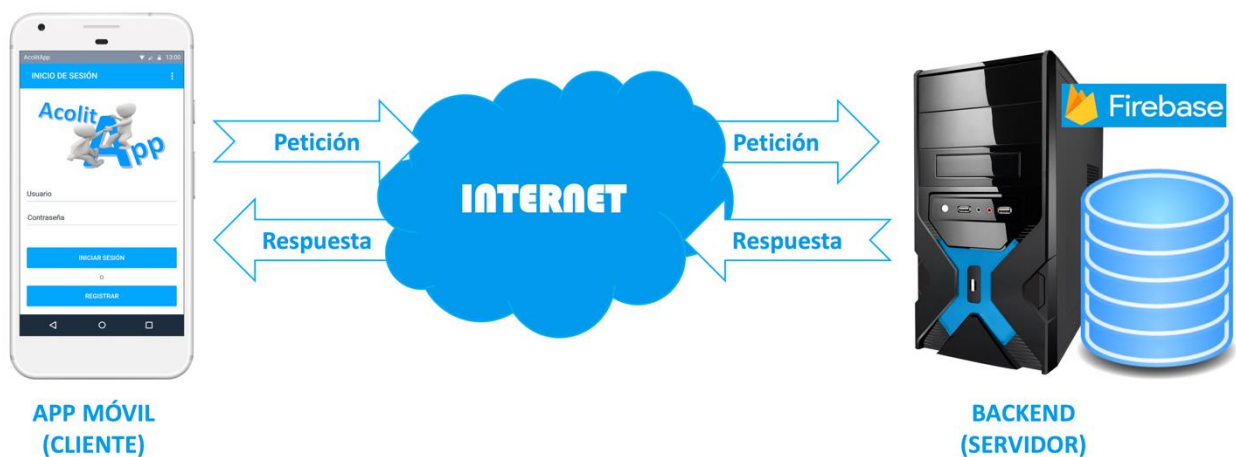


Figura 2.15: Arquitectura cliente-servidor que usará el sistema

2.2.2.4 Arquitectura de la aplicación

En cuanto a la arquitectura de la App, se establece una estructura a bajo nivel, la misma que va a usar la arquitectura MVC (Modelo-Vista-Controlador), con el objetivo de separar los datos y la lógica de negocio y la vista en una aplicación.

A continuación se explica el contenido de la arquitectura MVC:

Modelo: se encargará de los datos y de realizar las consultas en la base de datos. Además realizará las respectivas actualizaciones, búsqueda, consultas, etc.

Vista: se encarga de la presentación visual de los datos, pues son los diseños y los datos mostrados en las diferentes pantallas que el usuario puede visualizar al utilizar la aplicación.

Controlador: se encarga de realizar toda la lógica de negocio, recibiendo las peticiones del usuario, y tratando de establecer la comunicación entre el modelo y las vistas.

A continuación se muestra una un gráfico de la arquitectura MVC a implementar en la Aplicación:

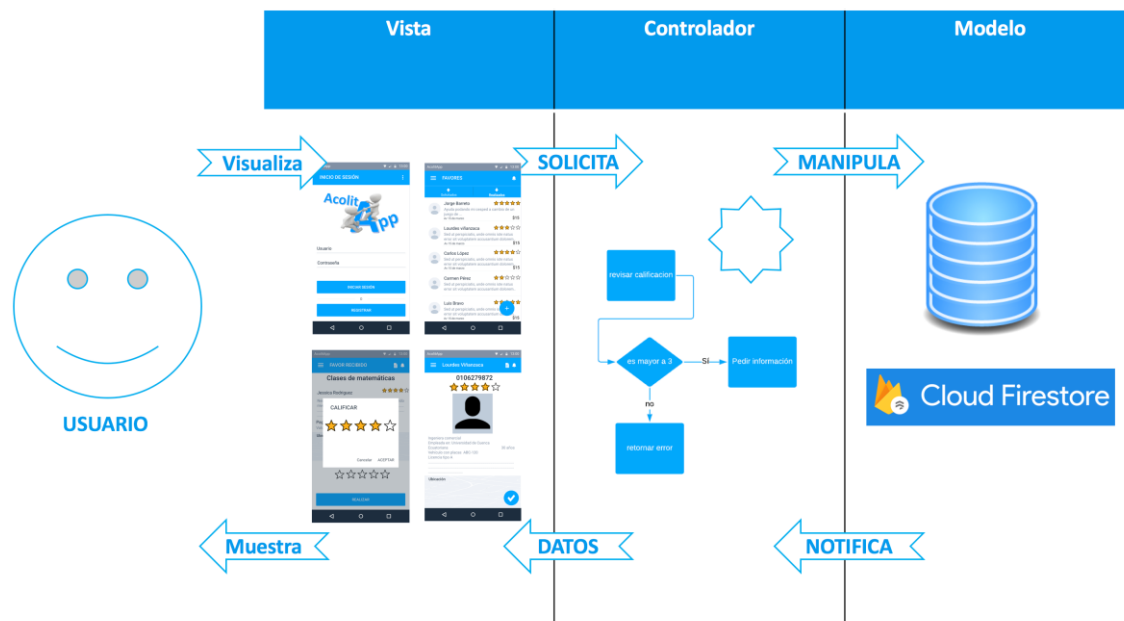


Figura 2.16: Arquitectura MVC (Modelo-Vista-Controlador) de la aplicación.

3. Implementación de la Aplicación

En este apartado se explica la estructura y el proceso que se llevó a cabo en la implementación de la app “acolitapp”. Este proceso se divide en dos etapas (Desarrollo y pruebas).

3.1 Desarrollo

En esta etapa se codificará la aplicación móvil tomando en cuenta el diseño propuesto y los prototipos de alta fidelidad. Además, se especificarán los frameworks, librerías y APIs utilizadas para desarrollar el producto.

3.1.1 Frameworks, librerías y plugins

Para el desarrollo de la aplicación móvil se utilizó varias herramientas tecnológicas fáciles de utilizar, pero muy potentes, las cuales fueron de gran ayuda para resolver diferentes tipos de problemas (autenticación, almacenamiento de datos y archivos, etc). A continuación se especifican las herramientas utilizadas:

HTML 5, CSS 3 y JAVASCRIPT	
¿Qué son?	Son tecnologías básicas pero muy potentes con las cuales se pueden definir la estructura (HTML), el estilo y apariencia (CSS) y la funcionalidad (JS) de una página o aplicación web [12].
¿Por qué se utilizó?	Se utilizaron estas herramientas por el fácil uso de las mismas y porque he ganado un poco de experiencia al utilizarlas en mi trabajo como desarrollador web.
¿En donde se utilizó?	Estas tecnologías se utilizaron en cada pantalla de la aplicación móvil. Pues cada una está estructurada con HTML, CSS y JS.

GIT y GITHUB	
¿Qué son?	Git es un sistema de control de versiones [13], mientras que GitHub es un portal para alojar el código de las aplicaciones y herramientas. [14].
¿Por qué se utilizó?	Se utilizó porque necesitaba respaldar y tener un control de versiones de cada cambio realizado o de las funcionalidades implementadas.
¿En donde se utilizó?	Lo utilicé con la carpeta que tiene el código fuente de la aplicación.

NODE.JS y NPM	
¿Qué son?	Node.js es un entorno en tiempo de ejecución multiplataforma, basado en JavaScript [16], mientras que npm es el sistema de gestión de paquetes por defecto para Node.js [17].

¿Por qué se utilizó?	Se utilizó porque es requerido para realizar aplicaciones con Ionic.
¿En donde se utilizó?	Se utilizó en el proyecto de ionic para compilar y ejecutar la app, y sobretodo para instalar las librerías y plugins de capacitor.

VISUAL STUDIO CODE	
¿Qué es?	Visual Studio Code es un editor de código fuente ligero y potente, con soporte integrado para JavaScript, TypeScript y Node.js [18].
¿Por qué se utilizó?	Se utilizó debido a su interfaz amigable y fácil de utilizar. Posee muchas extensiones que me ayudaron a generar el código de una manera más rápida. Además los colores del texto durante la codificación me ayudaron mucho para encontrar errores.
¿En donde se utilizó?	Toda la app se codificó utilizando este editor de código fuente. Además se utilizó su terminal de comandos para subir cambios al repositorio GIT.

ANGULAR CLI – ANGULAR (version 11)	
¿Qué son?	Angular es un framework para aplicaciones web desarrollado en TypeScript, mientras que Angular CLI es una herramienta que se utiliza para inicializar y desarrollar aplicaciones Angular [20].
¿Por qué se utilizó?	Una aplicación Ionic puede ser implementada con Angular, React o Vue. Se eligió Angular porque he desarrollado con ese framework en mi trabajo y por lo tanto tengo un poco más de experiencia.
¿En donde se utilizó?	Se utilizó angular con la versión 11 en el proyecto de ionic. Además, con Angular se manejó el sistema de rutas, la carga perezosa de los módulos y componentes del aplicativo, etc.

ANDROID STUDIO	
¿Qué es?	Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. [21].
¿Por qué se utilizó?	Se utilizó Android Studio y su SDK con el fin de compilar la aplicación móvil generada en Ionic y ejecutarla en un emulador o dispositivo físico.
¿En donde se utilizó?	Se utilizó en las pruebas de funcionalidad realizadas a la aplicación Acolitapp, pues en Android Studio se ejecutaba la App y se probaba en un emulador o dispositivo físico.

IONIC (versión 5)	
¿Qué es?	Ionic es un framework permite crear y ejecutar apps en diferentes plataformas con la misma base de código [22].
¿Por qué se utilizó?	La propuesta de este trabajo final es implementar una aplicación híbrida y que con el mismo código pueda generarse la App tanto para Android como para iOS. También se tomó esta decisión ya que para desarrollar apps en Ionic se necesita conocimientos en HTML, CSS y Typescript, los cuales poseo. Finalmente, la razón principal es aprender y mejorar mis habilidades en el desarrollo de aplicaciones híbridas.
¿En donde se utilizó?	Prácticamente, ionic fue el framework principal y con el cual se desarrolló toda las Aplicación.

FIREBASE	
¿Qué es?	Firestore facilita la creación de apps de una forma rápida. Brinda servicios de autenticación, almacenamiento, bases de datos, etc. [23].
¿Por qué se utilizó?	Se utilizó para ahorrar tiempo en la parte del servidor backend, pues el tiempo planificado se iba a salir de las manos si implementaba toda la base de datos, el aplicativo backend y los servicios que almacenan y recuperan la información. Con la ayuda de firebase tuve los servicios implementados y solo los llamé desde la aplicación. Por otra parte, firebase resultó relativamente fácil de entender y usar.
¿En donde se utilizó?	En el aplicativo se utilizó tres servicios de firebase: Firestore authentication: se usó para la pantalla de logueo y registro de usuarios. Los usuarios se registran y loguean mediante un correo y una contraseña. Firestore database: Se utilizó para almacenar y recuperar la información de los usuarios y favores. Firestore Storage: Fue de gran ayuda para almacenar la foto de perfil de los usuarios registrados.

3.1.2 Estructura de la Aplicación.

Para el desarrollo de la aplicación se ha creado un proyecto en blanco, utilizando el framework Ionic. Al crear un aplicativo con este framework, nos crea una estructura de archivos básica, desde la cual se pueden generar módulos, componentes, páginas, servicios, pipes, interfaces, clases, enumeradores, guards, etc.

La carpeta “src”, es la que contiene el módulo principal del proyecto y todos los archivos generados durante el desarrollo de la aplicación. A continuación se explica el contenido de cada una de las carpetas que se crearon para esta aplicación:

3.1.2.1 Compartido

Esta carpeta contiene un archivo llamado “**constantes.ts**”, que a su vez tiene palabras y mensajes que se muestran en las diferentes pantallas de la aplicación.

Se creó con la finalidad de usarlas en lugar de escribir quemar texto en las interfaces. De esta manera, si reemplazo el nombre de una acción (por ejemplo: la palabra “modificar” por “actualizar”), entonces no debería cambiar manualmente en cada lugar donde aparece dicha palabra, sino que solo cambiaría esa constante y automáticamente se modificarán en todos los lugares donde se use la misma.

3.1.2.2 Guards

Contiene vigilantes que determinan si se puede cargar o no dicha ruta.

En este caso, se implementaron dos guards:

- **Auth.guard.ts**: permite cargar ciertas rutas o pantallas cuando el usuario está logueado.
- **No-login.guard.ts**: permite cargar ciertas rutas o pantallas cuando el usuario no está logueado.

3.1.2.3 Pipes

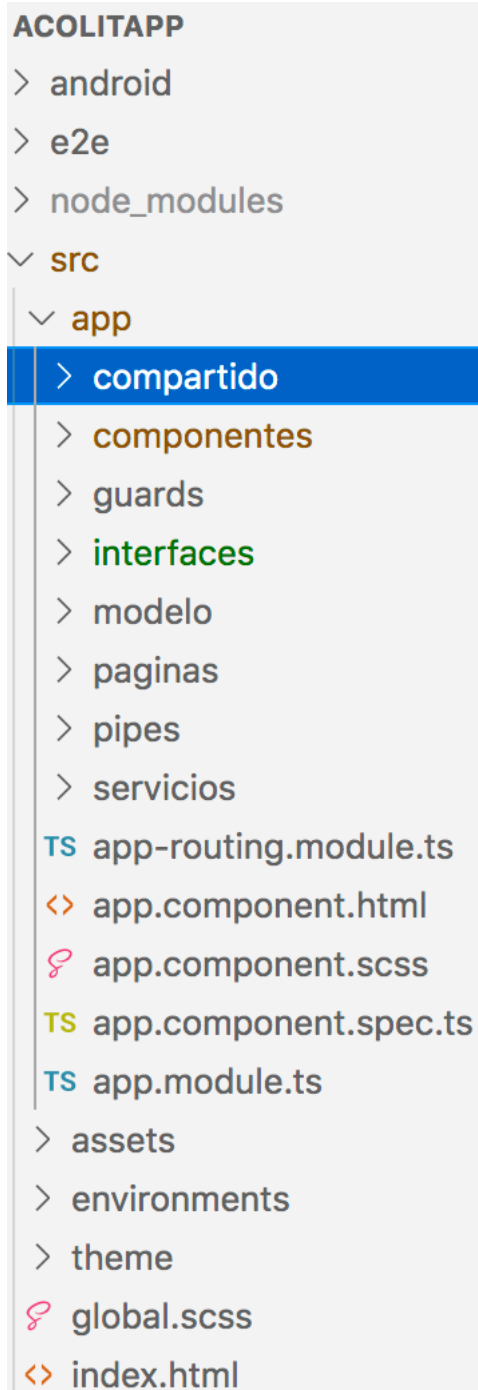
Contiene pipes que retornan la nacionalidad o las ciudades según el país o un color según el estado del favor.

Los pipes implementados son los siguientes:

- **Ciudades-por-pais.pipe.ts**: retorna una lista de ciudades según el país. Se implementó de esta manera porque se tiene 7 países de ejemplo, por lo que no se ve conveniente guardarlos en una base de datos.
- **Color-estado.pipe.ts**: retorna un color según el estado del favor (ejemplo: rojo= “PENDIENTE”). Es necesario que la interfaz muestre un color en el favor para que se pueda saber en qué estado se encuentra el mismo.
- **Nacionalidad-por-pais.ts**: retorna la nacionalidad según el país.

3.1.2.4 Modelado

Tal y como se estableció en la fase de diseño, la aplicación sigue la arquitectura MVC (Modelo Vista Controlador), por esa razón se ha creado la carpeta



“Modelado” con todas las clases que conforman el modelo de datos del aplicativo.

Además existen clases DTO y enumeradores:

- **DTO:** contiene clases que se utilizan para transferir datos en las solicitudes realizadas al backend.
- **Enum:** contienen una lista de valores de algún campo, como son: las formas de pago (Valor monetario, obsequio, favor), estados del favor (solicitado, pendiente, realizando, finalizado), etc.

```

  ✓ modelo
    > dto
    > enum
  TS calificacion.ts
  TS ciudad.ts
  TS favor.ts

```

3.1.2.5 Servicios

También son parte de la arquitectura MVC, pues en estas clases se implementan todos los requests o solicitudes que se realizan al backend para obtener, almacenar, actualizar o eliminar datos.

También existen servicios específicos como:

- **Auth.service.ts:** se encarga de interactuar con los servicios de autenticación de firebase (sign in, sign out, sign up).
- **Storage.service.ts:** se encarga de almacenar o recuperar datos del Local Storage, con el objetivo de tener datos locales accesibles rápidamente sin la necesidad de realizar una solicitud externa.
- **Opciones-menu.service.ts:** recupera los menús que se visualizarán en el menú de la aplicación. Esto se implementó con la finalidad de no tener quemado en el HTML o en el componente los menús. De esta manera, a futuro se podrá recuperar los menús de una base de datos u otro servicio.

```

servicios
  ✓ firebase
  TS auth.service.ts
  ✓ librerias
  TS storage.service.ts
  ✓ menu
  TS opciones-menu.service.ts
  $ detalle-tipo-pago.service.ts
  $ favor.service.ts
  $ nacionalidad.service.ts
  $ nivel-estudios.service.ts

```

3.1.2.6 Componentes

Contiene un conjunto de funcionalidades e interfaces que pueden ser reutilizadas dentro de otras interfaces. Se crearon con la finalidad de no repetir el mismo código en diferentes pantallas, ya que se puede reutilizar.

Entre estos componentes implementados están los siguientes:

- **Cabecera:** visualiza el toolbar en cada una de las pantallas de la aplicación. El toolbar muestra el título de la pantalla actual y el botón que regresa a la pantalla anterior o el botón que despliega el menú lateral.
- **Calificación:** contiene la ventana que se despliega al momento de calificar el favor. Permite calificar al usuario con un valor de 1 a 5 estrellas.

```

  ✓ componentes
    > cabecera
    > calificacion
    > favores-lista
    > item-favor

```

- **Favores-lista:** muestra la lista de favores según el filtro aplicado (todos los favores, favores solicitados o favores realizados)
- **Item-favor:** visualiza los datos de un favor específico. Posee funcionalidades para visualizar el favor, los datos de la persona solicitante o ayudante y la forma de pago por el favor realizado.

3.1.2.7 Páginas

Nuevamente, siguiendo la arquitectura MVC, esta carpeta contiene las interfaces que interactúan directamente con el usuario.

Cada interfaz de usuario está compuesto de 3 archivos, por ejemplo:

- **Favores.page.html:** contiene toda la estructura de la interfaz de usuario.
- **Favores.page.scss:** es el archivo donde se colocan los estilos de la interfaz.
- **Favores.page.ts:** en este archivo se codifica con TypeScript la funcionalidad de la interfaz.

A continuación se detalla las páginas implementadas:

Login: es la pantalla principal de la aplicación, y permite iniciar sesión al usuario o redireccionar a la pantalla de registro cuando aún no está registrado.

Registro: permite al usuario registrarse en la aplicación para poder utilizarla. Además permite subir una fotografía desde la galería o mediante la cámara del dispositivo.

Datos-personales: permite registrar la información personal del usuario para brindar seguridad y confianza entre las personas que usan la App.

Favores: contiene tabs que se pueden seleccionar para ver todos los favores o filtrar los favores solicitados o realizados. Además, desde esta interfaz se puede solicitar un nuevo favor.

Favores-realizados: esta pantalla lista y visualiza todos los favores realizados por el usuario actual (usuario logueado).

Favores-solicitados: esta pantalla lista y visualiza todos los favores solicitados por el usuario actual (usuario logueado).

Favores-todos: esta pantalla lista y visualiza todos los favores que han solicitado los usuarios.

Solicitud: es la interfaz que permite al usuario solicitar un nuevo favor. Pues, deberá llenar el título, descripción y forma de pago por el favor realizado.

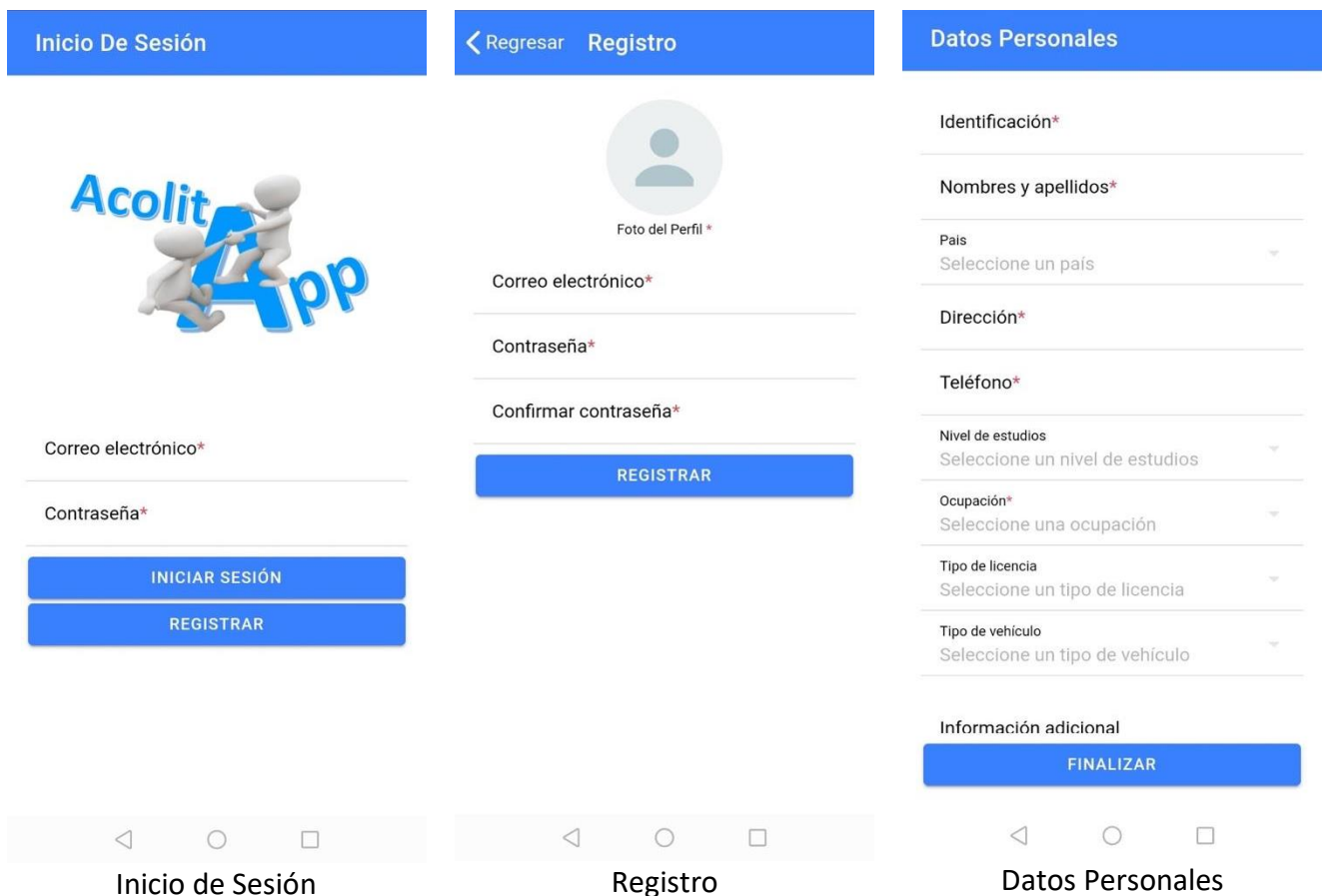
- paginas
- > datos-personales
- > favor-recibido
- > favores
- > favores-realizados
- > favores-solicitados
- > favores-todos
- > informacion-personal
- > login
- > registro
- > solicitud

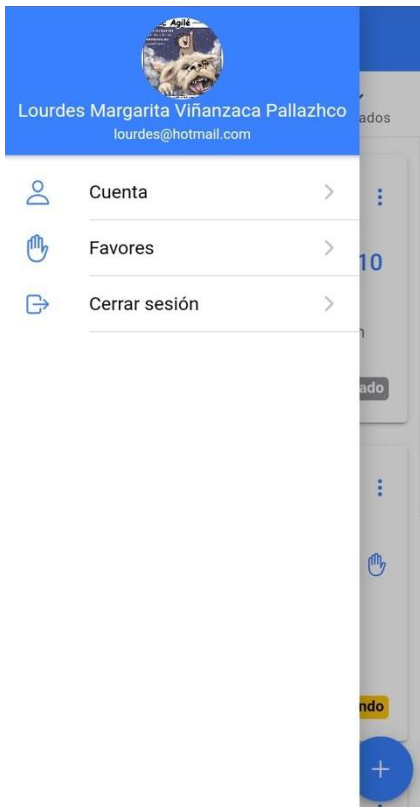
Favor-recibido: interfaz que contiene todos los detalles de un favor específico. En esta pantalla, dependiendo del usuario y del estado del favor, se podrá cancelar, ayudar, rechazar, realizar, finalizar y calificar dicho favor. Además aquí se podrá ver los datos del usuario que solicitó y ayudó con el favor, la calificación brindada por cada usuario y la forma de pago por el favor realizado.

Información-personal: la interfaz visualiza toda la información personal de un usuario en particular. Además podrá ver la foto de dicho usuario y la calificación promedio que tiene por todos los favores realizados.

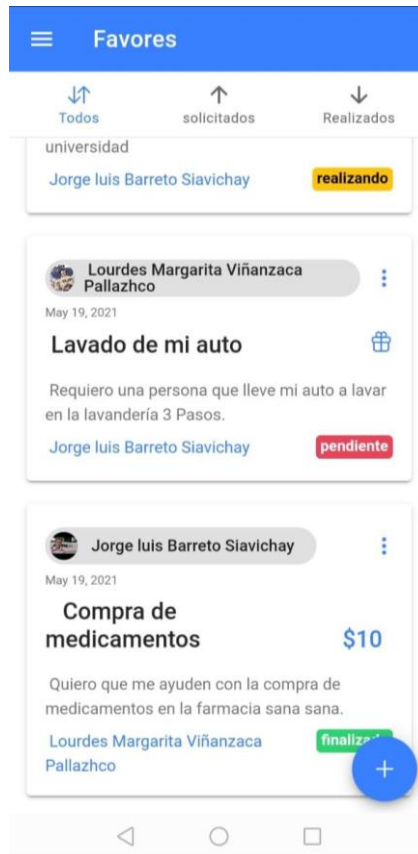
3.1.3 Pantallas Implementadas.

A continuación se presentan las pantallas y funcionalidades más importantes que fueron implementadas:

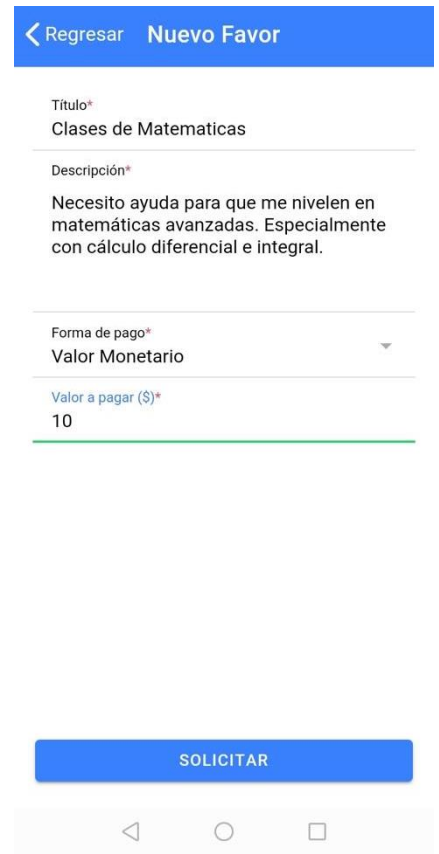




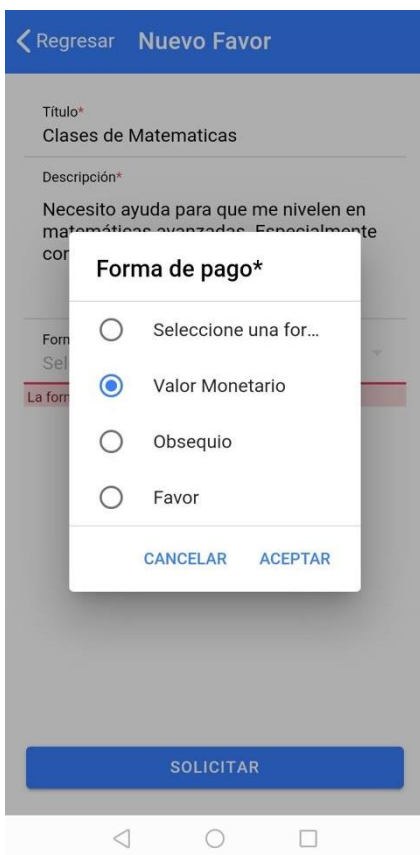
Menú Lateral



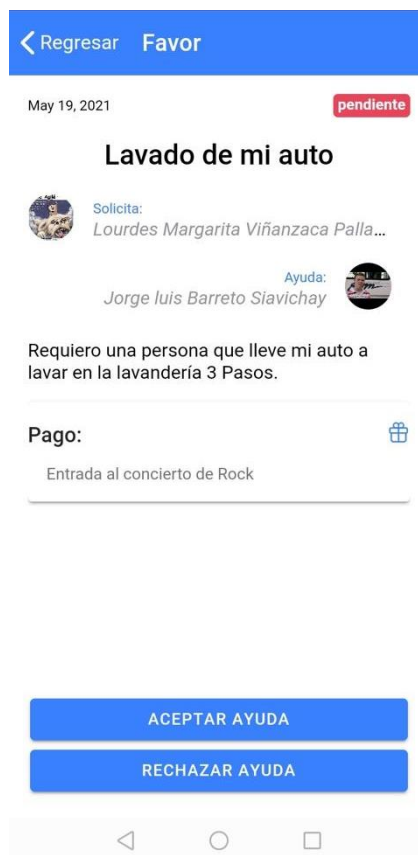
Favores



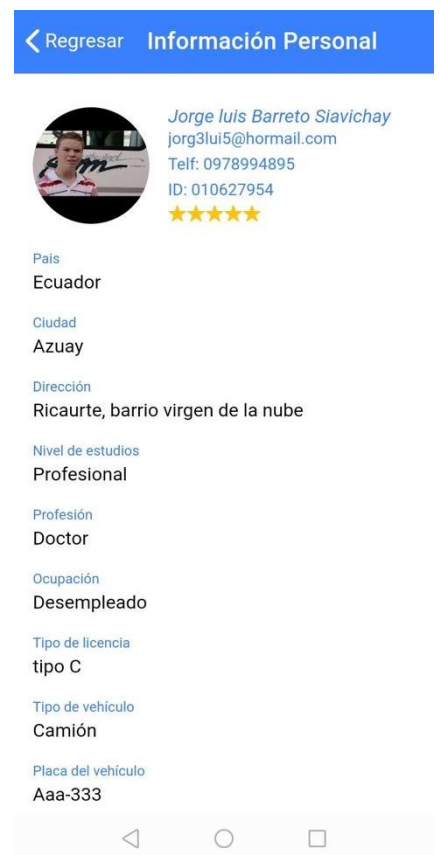
Nuevo favor



Forma de Pago



Favor Recibido



Información Personal



Calificar Favor

3.1.4 Estados y actores del Favor.

En la implementación de la app, se definieron 4 posibles estados que puede tener un favor y 3 actores (tipos de usuarios) que pueden interactuar con este favor. Además el favor siempre va a tener un estado y puede cambiar según la interacción de los usuarios.

La finalidad de estos estados y actores es saber quién es el usuario que solicita y/o ayuda con el favor, además de conocer si el favor ya fue realizado o no.

3.1.4.1 Actores

Los actores que intervienen en un favor son los siguientes:

- **Solicitante:** es la persona que solicita ayuda con un favor registrado.
- **Ayudante:** es la persona que pretende ayudar o ayuda con un favor solicitado por otra persona.
- **Otro:** es la persona que no es solicitante ni ayudante de un favor en particular.

3.1.4.2 Estados

Los estados que puede tener un favor en un determinado tiempo son los siguientes:

- **Solicitado:** Un favor tiene el estado SOLICITADO cuando un usuario solicitó un favor y aún nadie acepta ayudarlo.
- **Pendiente:** Un favor tiene el estado PENDIENTE cuando un usuario aceptó ayudar con el favor, pero el solicitante aún no acepta la ayuda de esa persona.
- **Realizando:** Un favor tiene el estado REALIZANDO cuando un usuario aceptó ayudar y el solicitante aceptó su ayuda.
- **Finalizado:** Un favor tiene el estado FINALIZADO cuando el ayudante o solicitante finalizó el favor (terminó la ayuda).

3.1.5 Aspectos Relevantes del Desarrollo.

Existen aspectos relevantes que deben conocerse sobre la implementación de la aplicación móvil, entre ellos están la autenticación de usuarios, el almacenamiento de datos, el almacenamiento las fotos de perfil, la fotografía subida desde la galería o capturada con la cámara del dispositivo, la forma en que se manejaban los estados del favor, etc. Todos estos detalles se explicarán a continuación:

AUTENTICACIÓN

Herramienta o plugin usado:	Firestore Authentication: https://firebase.google.com/docs
¿Por qué se utilizó?	Se utilizó porque yo solo había implementado la autenticación mediante tokens a lado del servidor, sin embargo el tiempo no me alcanzó para implementarlo y por eso decidí usar algo ya implementado. Pues vi que firebase authentication es un sistema de autenticación, sencillo y fácil de usar.
¿En donde se utilizó?	Se utilizó en el logueo y registro de usuarios. También se utilizó en el menú lateral para que el usuario pueda cerrar la sesión. El registro se hizo mediante un correo y contraseña.
Problemas que se tuvieron:	Problema: El registro del correo era Case Sensitive, por lo que a veces se registraba un correo con mayúsculas y al intentar loguear con minúsculas no encontraba al usuario. Solución: La solución fue transformar en minúsculas el correo ingresado y después registrar al usuario en firebase Authentication.

ALMACENAMIENTO DE LOS DATOS EXTERNOS

Herramienta o plugin usado:	Firestore: https://firebase.google.com/docs/firestore?authuser=1
¿Por qué se utilizó?	Se planteó implementar un servidor con una base de datos relacional, sin embargo, se optó por utilizar una base de datos NoSQL y que esté en la nube, para no tener complicación y desperdiciar tanto tiempo modelando la base de datos, creando el servidor y los servicios que se requieren. Con Firestore ya se dispone de los servicios y se escucha constantemente los cambios en los datos, actualizándolos en la app sin necesidad de actualizar manualmente.
¿En donde se utilizó?	Se utilizó en cada una de las pantallas, para mostrar la información que contiene la misma.

Problemas que se tuvieron:	<p>Problema: Esta parte fue la más complicada del desarrollo, puesto que los cambios realizados en los datos se escuchaban constantemente y no pude controlar eficazmente los métodos asíncronos. Esto provocó inconsistencia en la información visualizada en la pantalla.</p> <p>Solución: La solución fue aumentar la curva de aprendizaje para manejar métodos asíncronos. En ciertos casos se colocó tantos try() y catch(), y condicionales “IF” para controlar la ejecución asíncrona de los métodos.</p>
-----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ABRIR CÁMARA – SELECCIONAR FOTO DE LA GALERÍA

Herramienta o plugin usado	Camera: https://ionicframework.com/docs/native/camera
¿Por qué se utilizó?	Al momento de realizar pruebas con los usuarios vimos muy importante subir la fotografía del usuario con el objetivo de generar un poco más de seguridad y confianza al momento de aceptar la ayuda o ayudar a una persona con un favor. Se realizó con este plugin porque es de Ionic y es sumamente fácil de utilizar.
¿En donde se utilizó?	Se utilizó al momento de registrar al usuario. La foto de perfil se estableció como obligatorio para el registro.
Problemas que se tuvieron:	<p>Problema: Al capturar la foto o subir desde la galería, no se visualizaba en el elemento <ion-avatar>.</p> <p>Solución: Se procedió a transformar y especificar en formato base64.</p>

ALMACENAMIENTO DE ARCHIVOS (FOTO DE PERFIL)

Herramienta o plugin usado:	Cloud Storage: https://firebase.google.com/docs/storage?authuser=1
¿Por qué se utilizó?	Inicialmente, se planteó guardar la foto como un array de bytes en el almacenamiento local, ya que solo hay una foto del usuario logueado; sin embargo, en las pruebas de funcionalidad, esto hizo muy lenta la App, y por lo tanto se investigó otras alternativas como Alfresco y Cloud Storage. Al final se optó nuevamente por la firebase, ya que su implementación es aún más fácil a comparación de la autenticación y Firestore.
¿En donde se utilizó?	Se utilizó al momento que el usuario se registra en la aplicación. Pues, además del correo y contraseña, se requiere la foto de perfil.
Problemas que se tuvieron:	<p>Problema: El único problema fue al subir una fotografía tomada con la cámara del dispositivo, pues no reconocía el formato.</p> <p>Solución: La solución fue convertir la imagen a base64 para que se suba normalmente.</p>

ALMACENAMIENTO DE DATOS LOCALES

Herramienta o plugin usado	Ionic-storage: https://github.com/ionic-team/ionic-storage
¿Por qué se utilizó?	Se tomó la decisión de almacenar como dato local el usuario logueado al sistema, con el objetivo de que en cualquier pantalla se pueda obtener el usuario fácilmente y así obtener los demás datos asociados a ese usuario.
¿En donde se utilizó?	Se utilizó en las pantallas que muestran datos relacionados al usuario logueado. Por ejemplo, para listar los favores que ha realizado o solicitado dicho usuario.
Problemas que se tuvieron:	<p>Problema: La librería no funcionaba en el navegador web.</p> <p>Solución: se usó la librería <i>ionic-storage</i> de angular y no el ionic-storage nativo.</p>

3.2 Pruebas

En esta fase, tomando como base el análisis de los casos de uso establecidos en el diseño, se detallarán las pruebas realizadas en la aplicación móvil con el fin de determinar su funcionamiento. El objetivo de estas pruebas es encontrar errores y solucionarlos antes de desplegar la App en producción.

Existen varios tipos de pruebas que se realizan sobre una aplicación (unitarias, integración, funcionalidad, etc.), sin embargo, en este caso se optó por realizar únicamente las pruebas de funcionalidad, debido a que no se tiene el conocimiento suficiente para realizar pruebas unitarias, especialmente con Angular/Ionic. Por otra parte, por el poco tiempo que se dispone no se vio conveniente realizar una capacitación en este tema.

3.2.1 Entorno de pruebas

Para la preparación y ejecución de las pruebas se ha utilizado un navegador web, un emulador y un dispositivo físico real. A continuación se detalla los mismos:

NAVEGADOR WEB	
Características	Navegador: Google Chrome, Versión: 90, SO: Windows
¿Por qué se utilizó?	Al ser una aplicación móvil híbrida desarrollada con Ionic, tiene la ventaja de ejecutarse en un navegador web. Se ha seleccionado Chrome, porque es el navegador más usado hoy en día.
Configuraciones realizadas	<ol style="list-style-type: none"> 1. Ejecutar la aplicación ionic con el comando: <i>ionic serve</i> 2. Abrir el navegador <input type="checkbox"/> clic derecho->inspeccionar <input type="checkbox"/>seleccionar el ícono "google device toolbar". 3. Seleccionar un dispositivo o el modo responsive.
EMULADOR	
Características	Tipo: SmartPhone Proporcionado por: Android Studio Dispositivo: Nexus 6, Nivel API: 25 (Nougat), SO: Android 7.1.1
¿Por qué se utilizó?	Se utilizó un emulador de Android Studio porque ya se tenía instalado esta app en la computadora. Además la computadora utilizada lo soporta sin problemas a pesar de que el emulador consume muchos recursos; por esta razón no se analizó la posibilidad de usar GenyMotion.
Configuraciones realizadas	<ol style="list-style-type: none"> 1. Abrir Android Studio 2. Seleccionar Tools <input type="checkbox"/>AVD Manager 3. Crear un dispositivo virtual o seleccionar uno creado previamente. 4. Ejecutar la aplicación en el dispositivo virtual
DISPOSITIVO FÍSICO	
Características	Tipo: SmarthPhone Dispositivo: Huawei Y7 2019 SO: Android, Versión: 8.1.0
¿Por qué se utilizó?	Se utilizó porque se debe probar la aplicación en dispositivos físicos reales para verificar su correcto funcionamiento. Este es el único dispositivo al que se tuvo acceso para realizar las pruebas.

1. Abrir Android Studio
2. Generar el archivo APK
3. Instalar el archivo .apk en el dispositivo.

3.2.2 Pruebas Funcionales

Las pruebas de funcionalidad han sido iterativas e incrementales, pues cada vez que se agregaba una nueva funcionalidad se procedía a probar nuevamente la aplicación, y según los resultados, se implementaba una nueva funcionalidad o se corregía la existente para repetir el proceso de pruebas.

Cada vez que se realizaba una nueva iteración probaba tanto las nuevas funcionalidades como las anteriores, para estar seguro de que continúan funcionando correctamente.

3.2.2.1 Definición de casos de pruebas y resultados

CASOS DE PRUEBAS Y RESULTADOS

ID	Caso de prueba	Pasos a seguir	Pre-requisitos	Resultado esperado	Estado	Observaciones
CP-1.1	Registrar un nuevo usuario en la aplicación	1. En la pantalla de Inicio de sesión, pulsar el botón "Registrar". 2. Seleccionar una foto de perfil y seleccionar la opción Galería o Cámara. 3. Tomar la foto o seleccionarla de la galería. 4. Llenar todos los campos obligatorios (foto, correo electrónico, contraseña, y confirmación) 5. Pulsar el botón "Registrar"	-No estar registrado previamente con ese correo.	Muestra un mensaje satisfactorio y navega a la página DATOS PERSONALES.	ADV	El registro de usuario funciona satisfactoriamente; sin embargo, cuando se ejecuta en el navegador web no es posible subir la imagen, debido a que utiliza plugins nativos. A futuro se podrá mejorar y permitir la subida de archivos cuando la app se ejecute desde un navegador.
		FLUJO ALTERNATIVO: -Introducir un email con formato incorrecto -No introducir alguno de los campos obligatorios -Ingresar una contraseña con menos de 6 caracteres. -La contraseña y su campo de confirmación no coinciden. -Ya existe un usuario con el correo ingresado. -Al seleccionar la foto pide permisos al dispositivo		- Muestra un mensaje que informa al usuario sobre el error identificado.	OK	
DP-1.2	Registrar un nuevo usuario en la aplicación	1. Registrar el usuario (CP-1.1) 2. Llenar todos los campos obligatorios (Identificación, nombres y apellidos, dirección, teléfono, ocupación) 3. Pulsar el botón "Finalizar"	-Haber pasado por la pantalla de registro.	Muestra un mensaje satisfactorio y navega a la página FAVORES, listando los favores existentes hasta el momento.	OK	Existían campos que estaban como obligatorios pero que no se visualizaban en pantalla, lo cual no se podía saber con exactitud el campo faltante. Se solucionó en la segunda iteración.
		FLUJO ALTERNATIVO -No introducir alguno de los campos obligatorios - No ingresar mínimo 5 caracteres en el campo "nombres y apellidos"		- Muestra un mensaje que informa al usuario sobre el error identificado.	OK	
CP-2.1	Iniciar sesión con un usuario registrado en el aplicativo	1. Abrir la aplicación 2. Ingresar el correo y la contraseña. 3. Presionar el botón "Iniciar sesión". El sistema valida los datos ingresados. Si la validación es correcta, se concede el acceso.	-Tener un usuario registrado en la aplicación. -Recordar su usuario y contraseña	Accede a la página FAVORES, listando los favores existentes hasta el momento.	OK	Hubo inconvenientes en 2 ocasiones. En la primera, existían usuarios que a pesar de pasar la pantalla de registro, estos datos nunca se almacenaron. Se solucionó en la iteración 1. En ciertas ocasiones, se mostraba un mensaje para anunciar que el usuario no estaba registrado, sin embargo fue debido a que el campo correo era CASE SENSITIVE. Se modificó para que siempre se guarde con minúsculas(Solucionado en la tercera iteración).

		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - El usuario no existe. - La contraseña es incorrecta. -No se ingresa un correo válido. -No se llenan los campos requeridos. 		- Muestra un mensaje que informa al usuario sobre el error identificado.	OK	
CP-3.1	Ver la información personal del usuario logueado.	<p>1. Iniciar sesión con un usuario registrado (CP-2.1)</p> <p>2. Presionar en el ícono de menú.</p> <p>3. En el menú lateral, presionar la opción "cuenta".</p> <p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - Ninguno 	- Tener un usuario registrado e iniciar sesión.	La información del usuario debe estar reflejada en la pantalla INFORMACIÓN PERSONA.	ADV	<p>Hubo información personal que no se mostraba a pesar de haberla ingresado. Pues había faltado incluirlos a nivel del HTML (Solucionado en la iteración 4).</p> <p>Para visualizar la imagen del usuario en toda la pantalla del dispositivo se utilizó un plugin del dispositivo móvil, el mismo que no funciona cuando se ejecuta la app en un navegador web. Se tomará en cuenta para realizar estas mejoras a futuro.</p>
CP-4.1	Ver la lista de todo los favores (solicitados, realizados o todos)	<p>1. Iniciar sesión con un usuario registrado (CP-2.1).</p>	<ul style="list-style-type: none"> - Tener un usuario registrado. - Debe existir al menos un favor realizado o solicitado por cualquier usuario. 	Debe listar todos los favores solicitados hasta el momento por cualquier persona.	OK	
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - No existen favores solicitados o realizados. - Seleccionar la opción (Tab) "Realizados". - Seleccionar la opción (Tab) "Solicitados". 		<ul style="list-style-type: none"> -Si no existen favores solicitado por cualquier persona, entonces muestra la lista vacía. -Si seleccionó un Tab, entonces debe filtrar y mostrar los favores según el tab seleccionado. 	ADV	<p>En este apartado hubo muchos problemas, pues al seleccionar un Tab, no se filtraban los favores según el tab seleccionado, y a veces los filtraba mal. Se trató de solucionar y controlar los métodos asíncronos que escuchaban los cambios en la base de datos. Al final se solucionó en la iteración 8; sin embargo se estará pendiente por si vuelve a suceder algún otro tipo de problema.</p>
CP-5.1	Cerrar la sesión del usuario logueado	<p>1. Iniciar sesión con un usuario registrado (CP-2.1)</p> <p>2. Presionar en el ícono de menú.</p> <p>3. En el menú lateral, presionar la opción "Cerrar sesión".</p>	EL usuario debe tener una sesión activa en la App.	Se cerrará la sesión activa del usuario y mostrará la pantalla de inicio de sesión.	OK	<p>Al inicio se pasó por alto eliminar el usuario del localStorage, lo cual permitía acceder a ciertas páginas a pesar de estar deslogueado. Se solucionó en la iteración 4.</p>
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - Ninguno 			OK	
CP-6.1	Solicitar un nuevo favor	<p>1. Iniciar sesión con un usuario registrado (CP-2.1)</p>	Sesión activa.	- En la lista de favores solicitado debe aparecer el favor solicitado recientemente.	OK	<p>El problema que se presentó fue nuevamente con los métodos asíncronos, pues al cambiar un dato del favor en tiempo real, el servidor</p>

		<p>2. Pulsar el botón que tiene el signo más (+).</p> <p>3. Llenar los campos requeridos (Título, descripción, forma de pago, descripción o valor del pago).</p> <p>4. Pulsar el botón "Solicitar"</p>		<p>-El favor recién solicitado debe tener sus datos correctos. El estado del favor debe ser SOLICITADO.</p>		<p>ya devolvía nuevamente el registro con el cambio actualizado. Esto provocó que a veces no se muestren los datos que se modificaron recientemente. Se solucionó en la iteración 6.</p>
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - Selecciona la forma de pago "Valor monetario". - Selecciona la forma de pago "Obsequio". - Selecciona la forma de pago "Favor". - No llena la información requerida. - Ingresa un valor de pago menor a cero. 		<p>-Si selecciona la forma de pago "Favor u obsequio", debe ingresar su detalle.</p> <p>-Si selecciona la forma de pago "Valor monetario", debe aparecer el campo para ingresar un valor (número).</p> <p>-Si no ingresa un campo requerido, mostrará un mensaje de error.</p>	OK	
CP-7.1	Ver detalle de un favor solicitado o realizado.	<p>1. Listar Favores (CP-4.1)</p> <p>2. Pulsar sobre el título del favor.</p>	<ul style="list-style-type: none"> - Existir al menos un favor solicitado por cualquier persona. - Mantener la sesión activa del usuario 	<p>Debe mostrar la pantalla FAVOR, la cual contiene todos los datos del favor.</p>	OK	<p>No hubo mayores inconvenientes, pues los datos que se visualizaban eran los correctos.</p>
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> -Visualiza el favor dando clic en el ícono que tiene los 3 puntos. -Visualiza el favor deslizando el dedo hacia la izquierda sobre el favor. 		<p>-Si selecciona en los 3 puntos, deberá seleccionar la opción: Visualizar.</p> <p>-Si se desliza el favor, entonces debe pulsar sobre el ícono del ojo.</p>	OK	
CP-8.1	Un usuario Ayudar con un favor solicitado	<p>1. Listar favores (CP-4.1)</p> <p>2. Ver favor (CP-71).</p> <p>3. Pulsar en el botón "AYUDAR".</p> <p>4. Pulsar la opción "SI, AYUDAR"</p>	<p>Existir al menos un favor solicitado por otra persona con el estado SOLICITADO. El favor no debe haber sido ya realizado por alguien con anterioridad.</p>	<p>-Deberá cambiar el estado del favor a PENDIENTE.</p> <p>-Debe visualizarse el nombre del usuario que va ayudar el favor.</p> <p>-El botón inferior debe tener el texto "Cancelar ayuda"</p>	OK	<p>Se presentaron pequeños problemas al cambiar el estado del favor, debido a que en ocasiones se no se redirecciona a la lista de favor. Esto se logró solucionar de inmediato en la iteración 2.</p>

		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - Pulsa la opción "No" - El favor tiene un estado pendiente. - El favor tiene un estado Realizando. 		<p>Si el usuario presiona "No", entonces no se hará ningún cambio al favor.</p> <p>Si el favor tiene el estado PENDIENTE, entonces el usuario podrá Aceptar la ayuda en caso de haber solicitado o cancelar la ayuda si el usuario intentaba ayudar.</p> <p>Si el favor tiene el estado REALIZANDO, entonces podrá FINALIZAR dicho favor.</p>	OK	
CP-9.1	Visualizar la información de la persona que realiza y/o solicita el favor	<p>1. Listar favores (CP-4.1)</p> <p>2. Dar clic en el nombre de la persona que realiza o solicita un favor específico.</p>	Existir un favor solicitado. Para ver el perfil de la persona que va a ayudar o ayudó con el favor, debe haber aceptado dicho favor.	Se mostrará la pantalla INFORMACIÓN PERSONAL con la información de ese usuario.	OK	No hubo ningún inconveniente al mostrar la información de un usuario.
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - Ver Favor (CP-71) 		Si se abrió el favor, entonces podrá presionar en el nombre del usuario para ver dicha información.	OK	
CP-10.1	El usuario que solicitó el favor edita los datos del favor solicitado.	<p>1. Listar favores (CP-4.1)</p> <p>2. Ver favor (CP-71). Debe seleccionar un favor que el mismo usuario solicitó.</p> <p>3. Llenar o modificar los campos requeridos</p> <p>4. Pulsar el botón "Guardar".</p>	EL usuario debe haber solicitado el favor que va a editar.	Aparece un mensaje que confirma los cambios realizados.	OK	El único problema que se presentó, fue que al editar, no reconocía los campos requeridos. Sin embargo, en la iteración 2 se solucionó.
		<p>FLUJO ALTERNATIVO</p> <ul style="list-style-type: none"> - No llena la información requerida. - Ingresa un valor de pago menor a cero. 		Muestra al usuario un mensaje con el error identificado.	OK	

3.2.2.2 Pruebas del flujo y los estados del favor.

Es muy importante realizar pruebas entre varios usuarios al mismo tiempo, pues se necesitan al menos 3 personas para que puedan interactuar entre sí y ver cómo los favores solicitados y realizados cambian de estado según la acción que realice otro usuario.

Primero, debe estar claro que todos los favores tienen un estado en cierto momento, y un flujo que deben seguir según el estado del mismo. Para empezar, un favor puede pasar por uno y hasta por cuatro diferentes estados (Solicitado, Pendiente, Realizando, Finalizado) desde el momento que se solicita, hasta el momento en que se finaliza.

Para explicar de mejor manera estos estados, se definirán 3 actores:

Solicitante: es la persona que solicita ayuda con un favor registrado.

Ayudante: es la persona que pretende ayudar o ayuda con un favor solicitado por otra persona.

Otro: es la persona que no es solicitante ni ayudante de un favor en particular.

Ahora, con los cuatro estados y los tres actores definidos, se mostrará una tabla de cómo pueden cambiar dichos estados y seguir un flujo:

PRUEBAS DEL FLUJO Y LOS ESTADOS DEL FAVOR				
ESTADO	SIGNIFICADO	SOLICITANTE	AYUDANTE	OTRO
SOLICITADO	Un favor tiene el estado SOLICITADO cuando un usuario solicitó un favor y aún nadie acepta ayudarlo.	CANCELAR: El solicitante puede cancelar el favor solicitado. Al presionar esta opción, el favor se eliminará y por ende, nadie podrá verlo.	Todavía no existe ayudante	AYUDAR: cualquier persona puede ayudar con el favor solicitado. Al presionar esta opción, el estado del favor cambia a pendiente y el usuario se convierte en ayudante .
PENDIENTE	Un favor tiene el estado PENDIENTE cuando un usuario aceptó ayudar con el favor, pero el solicitante aún no acepta la ayuda de esa persona.	RECHAZAR: El solicitante puede rechazar la ayuda ofrecida por el ayudante. Al presionar esta opción, el estado del favor cambia a solicitado para que otra persona pueda ayudarlo. ACEPTAR: El solicitante puede aceptar la ayuda del usuario. Al presionar esta opción, el estado del favor cambia a realizando .	CANCELAR: El ayudante puede cancelar la ayuda ofrecida a un favor. Al presionar esta opción, el usuario ya no es ayudante y el estado del favor cambia a solicitado . De esta manera, ahora otro usuario puede convertirse en ayudante	A otro usuario NO le aparece ninguna opción, debido a que el favor está pendiente de aceptación o rechazo. Le volverá a aparecer si el Solicitante rechaza la ayuda, caso contrario ninguna otra persona podrá ofrecer ayuda con este favor.
REALIZADO	Un favor tiene el estado REALIZANDO	FINALIZAR: El solicitante puede finalizar para confirmar	FINALIZAR: El ayudante puede finalizar para confirmar	A otro usuario NO le aparece ninguna opción, puesto que el

	cuando un usuario aceptó ayudar y el solicitante aceptó su ayuda.	que el favor fue culminado. Al presionar esta opción, el estado del favor cambia a finalizado y el solicitante podrá calificar al ayudante .	que el favor fue culminado. Al presionar esta opción, el estado del favor cambia a finalizado y el ayudante podrá calificar al solicitante .	favor ya está siendo realizado.
FINALIZADO	Un favor tiene el estado FINALIZADO cuando el ayudante o solicitante finalizó el favor (terminó la ayuda).	CALIFICAR: EL solicitante puede calificar al ayudante siempre y cuando el ayudante haya finalizado el favor y por ende ya le haya calificado. Al presionar esta opción, tanto el solicitante como el ayudante se habrán calificado mutuamente.	CALIFICAR: EL ayudante puede calificar al solicitante siempre y cuando el solicitante haya finalizado el favor y por ende ya le ha calificado. Al presionar esta opción, tanto el ayudante como el solicitante se habrán calificado mutuamente.	A otro usuario NO le aparece ninguna opción, puesto que el favor ya fue realizado por otro usuario.

Las pruebas de flujo fueron exitosas, pues esto se realizó después de que se corrigieron la mayoría de errores encontrados en las iteraciones anteriores.

3.2.3 Evaluación de la Aplicación

Finalmente, se realizó 9 iteraciones con el objetivo de tener una aplicación sin errores, eficiente y que cumpla con el objetivo principal; sin embargo, puede existir pequeños problemas que no se detectaron en ese momento y que algún usuario nuevo lo detecte.

Por otra parte, se envió la aplicación a 5 personas para que la revisaran y usarán, con la finalidad de encontrar nuevos errores, inconsistencias o mejorar la usabilidad de la misma.

Los usuarios que evaluaron la app, nos dieron las siguientes recomendaciones o sugerencias:

- “Sería muy bueno implementar un chat en la aplicación para coordinar el proceso del favor entre el solicitante y el ayudante”.
- “Quisiera que me llegue una notificación cada vez que el favor que estoy solicitando o ayudando cambia de estado”.
- “Sería interesante que los favores puedan ver sólo personas que viven cerca de mi lugar de residencia o el lugar donde marque al solicitar el favor”.
- “Sería bueno poner la hora y día en que requiere que se haga el favor”.
- “No estaría mal poder llamar a la persona desde la app cuando yo acepte su ayuda, con el objetivo de coordinar el favor.”
- “Creo que es primordial que se apliquen más filtros a los favores, a parte de solo filtrar los favores recibidos y solicitados”.
- “Se debería modificar la posición de las estrellas, ya que la mayoría entiende que las estrellas que están a lado de la foto representan las estrellas que le ha dado el otro usuario y no las estrellas que ha otorgado el mismo”.

Algunos de los cambios sugeridos por los usuarios se pudieron realizar, sin embargo hay sugerencias que requieren más tiempo para su implementación, por lo que se tomará en cuenta para futuros cambios.

Por otra parte, la aplicación fue evaluada y se verificó que cumple con su objetivo principal y sobretodo con los requisitos funcionales, aunque algunas cosas no se hayan implementado de la forma que se definió en el diseño.

A continuación se muestra una tabla con los requisitos funcionales establecidos en la planificación y el cumplimiento de los mismos.

REQUERIMIENTOS FUNCIONALES		
Requisito	Estado	Porcentaje
La app permitirá registrar nuevos usuarios o iniciar sesión a los usuarios previamente registrados.	Desarrollado	100%
El usuario podrá listar los favores que ha solicitado.	Desarrollado	100%
El usuario podrá listar los favores en los que ha brindado ayuda (favores realizados).	Desarrollado	100%
La app permitirá al usuario solicitar nuevos favores	Desarrollado	100%
El usuario podrá elegir la forma en la que pagará por un favor realizado (un valor monetario, una recompensa o devolver con otro favor).	Desarrollado	100%
El usuario podrá ver los favores que han solicitado los demás usuarios.	Desarrollado	100%
Un usuario podrá aceptar o no las diferentes propuestas de favores y así llegar a un acuerdo con la persona que lo solicitó.	Desarrollado	100%
El usuario que solicita un favor, podrá ver la información de las personas que pretenden ayudarlo, y así seleccionar a la persona que considere más adecuada.	Desarrollado	100%
Tanto el solicitante como la persona que aceptó realizar el favor podrán calificarse mutuamente según el nivel de satisfacción del favor realizado.	Desarrollado	100%
La app permitirá colocar filtros y listar los favores que el usuario desea visualizar.	Desarrollado	90%

4. Conclusiones

4.1 Conclusiones del trabajo

Tras realizar este trabajo final de Máster, adquirí experiencia en afrontar todos los retos que presentan el desarrollar un proyecto completo y aprendí ciertas lecciones que a continuación me dispongo a valorar.

Personalmente, la realización de este proyecto me permitió afianzar y poner en práctica la mayor parte de los conocimientos adquiridos durante el Máster y alcanzar nuevas habilidades y capacidades con las herramientas, métodos y metodologías utilizadas para llevar a cabo el trabajo final. En concreto, pude profundizar en la implementación de aplicaciones móviles híbridas y la utilización de plataformas, frameworks, servicios y plugins que facilitan su desarrollo. Además, la documentación disponible en la web fue de gran ayuda a la hora de resolver dudas que iban surgiendo a lo largo del proyecto.

Uno de los motivos para usar Ionic fue aprovechar la oportunidad para aprender una nueva tecnología a parte de las aprendidas a lo largo del Máster; sin embargo, aprenderla, tuvo una curva de aprendizaje muy grande. Afrontar dicha curva ha sido muy duro porque actividades y pantallas sencillas de diseñar e implementar tomaron un tiempo adicional al planificado, no obstante, me dejó una satisfacción al ver que la app iba tomando forma y funcionando como se esperaba.

En mi opinión, conseguí que la app sea funcional, agradable y que cumpla con los requisitos funcionales establecidos, por lo que me siento muy orgulloso. Por otra parte, estoy consciente de que pudo haberse hecho cosas aún mejores y es lo que tanto anhelaba porque cada vez me motivaba más y me surgían ideas para que la aplicación se vea más atractiva y de alta calidad. Sin embargo, por el poco tiempo disponible traté de centrarme en el objetivo principal de la app para a futuro incluir todas las nuevas funcionalidades que deseo implementar.

La planificación del trabajo fue lo que me dejó una gran lección, pues inicialmente al definir el alcance del proyecto, personalmente me sentí muy motivado y con ganas de desarrollar una app profesional, de calidad, amigable y lista para ser usada por el mundo entero, sin pensar en todo el tiempo, recursos, conocimientos y esfuerzo que se requiere para conseguir una aplicación de esta dimensión. A pesar de todo, puse mi mayor esfuerzo para cumplir con los objetivos y entregar los hitos en las fechas planificadas.

Como conclusión final, quiero enfatizar la satisfacción que siento al haber implementado una versión beta de la app "Acolitapp", la cual funciona y cumple con el objetivo propuesto (una persona puede solicitar un favor y otra podrá ayudarla). Sin duda, acerté al elegir el área "Desarrollo de aplicaciones móviles multiplataforma" para realizar el TFM, puesto que es un ámbito que me apasiona y donde pude consolidar gran cantidad de conocimientos adquiridos a lo largo del Máster.

4.2 Logro de los objetivos planteados

Al momento de revisar los objetivos, requisitos funcionales y no funcionales definidos en el plan de trabajo, pude verificar que estos se cumplieron de manera satisfactoria, aunque un número reducido de objetivos se hayan cumplido de manera sin dedicarle mucho tiempo y mayor esfuerzo. Por ejemplo, para cumplir con el requisito no funcional *“Construir pantallas interactivas y fáciles de usar (usabilidad)”* se hizo algunos cambios a las pantallas previamente diseñadas, debido a las sugerencias de los usuarios que probaron la app, sin embargo, otros cambios no se aplicaron por la falta de tiempo y la complejidad que conlleva realizarlos.

Un requisito funcional que también se cumplió pero que se pudo haber hecho de mejor manera es la pantalla que permite iniciar sesión, puesto que el usuario registrado no puede iniciar sesión si es que no se acuerda la contraseña, debido a que no se implementó una forma de recuperar la contraseña con la que se registró. Esta funcionalidad no se implementó debido a que existían objetivos y requisitos más importantes, y por ende se dio una mayor prioridad.

Por otra parte, la aplicación cumple con todos los requisitos funcionales. Esto fue posible gracias a cambios que se hicieron en los diseños de algunas pantallas, la utilización de herramientas y servicios que facilitaban y agilizaban el desarrollo, y sobretodo descartar la implementación de un servidor backend y base de datos relacional, lo cual hubiese tomado mucho tiempo.

4.3 Seguimiento de la planificación y metodología propuesta

Con respecto a la planificación del proyecto, se pudo ejecutar todas las fases del trabajo y entregar los hitos correspondientes de cada fase en el tiempo previsto inicialmente, cumpliendo con los plazos establecidos en dicha planificación. El único inconveniente que se tuvo fue al entregar la PEC3, puesto que se entregó una semana después debido a los errores o bugs que se encontraron y a los cambios que pedían los usuarios para mejorar el tema de la usabilidad. A pesar de este retraso, la entrega final se hizo en la fecha prevista, pues la memoria del proyecto estaba casi lista y con visto bueno, por lo que no se hizo un mayor cambio en ella.

En cuanto a la metodología seguida, puedo decir que ha sido la más adecuada, pues se siguió el ciclo de vida de desarrollo de software de una manera incremental e iterativa, lo que permitió hacer pruebas y corregir errores por cada funcionalidad que se iba implementando. Además, se eligió bien usar Ionic porque ya tenía conocimientos previos de desarrollo web y sobretodo porque las pruebas de la app pude realizarlas directamente en el navegador web, para posteriormente realizarlas en un emulador y dispositivo Android real.

Por otra parte, fue una buena decisión aplicar el método Kanban Personal, porque fue de gran ayuda para mantener un control y organización de las tareas

que se iban a realizar, las que se estaban realizando y las que se culminaron. Al establecer prioridades y tiempos en las tareas, se pudo aplicar mayor esfuerzo en ciertas ocasiones para cumplir con lo planificado.

Por último, decidir entregar el archivo APK en lugar de subir a Play Store fue lo mejor, porque nos quitamos una preocupación de encima y aprovechamos ese tiempo en implementar las funcionalidades de la aplicación.

En relación a los cambios introducidos en la app desarrollada para garantizar el éxito del trabajo quiero destacar los siguientes:

- No se creó una base de datos relacional ni se implementó un servidor backend desde cero, puesto que toma demasiado tiempo llevarlo a cabo y realmente enfocarnos en el servidor no es el objetivo del Máster. Así que aprovechamos las ventajas que ofrece firebase para implementar la autenticación y el almacenamiento de datos y archivos.
- Se evitó la implementación de notificaciones push y se decidió mostrar una lista con todos los favores solicitados por los usuarios, debido a que me resultó muy difícil implementar notificaciones push de un usuario a otro, pues si destinaba más tiempo en ello y seguía intentando implementarlas, entonces se me iba a salir de las manos la planificación del proyecto y no lo iba a poder entregar a tiempo.
- Se cambió el diseño de algunas pantallas, haciéndolas más simples. Por ejemplo, se puso un campo “dirección” en lugar de obtener la ubicación mediante la API de Google Maps. Este y otros cambios en las pantallas se hicieron con el objetivo de dar prioridad a los requisitos funcionales definidos en la planificación.

En fin, considero que debemos ser más realistas al momento de planificar un proyecto, pues debe ser la fase que más tiempo debemos dedicar para que el trabajo no se desfase y pueda terminarse con éxito en las fechas establecidas.

4.4 Líneas de trabajo futuro

Por el momento la aplicación móvil “Acolitapp” está en su versión beta, la cual muestra el objetivo y funcionalidad básica de la misma (Solicitar favores y ayudar en los favores solicitados); sin embargo, por falta de tiempo no se pudo implementar las funcionalidades tal y como se tenía pensado. Por ende, una de las primeras acciones será implementar de mejor manera esas funcionalidades.

También existen aspectos y funcionalidades que podemos añadir para complementar y enriquecer a la aplicación. Entre ellas están las siguientes:

- Implementar y detectar la ubicación del usuario para que los favores solicitados sean vistos solo por usuarios que pertenecen a cierta zona o sector.

- Diseñar interfaces responsivas, para que se adapten a cualquier tamaño de pantalla.
- Implementar notificaciones push para los favores solicitados o recibidos.
- Implementar un chat entre la persona que solicita y la persona que ayuda con el favor.
- Permitir seleccionar la fecha y hora del favor a realizar.
- Permitir recuperar el usuario o contraseña de un usuario.
- Establecer diferentes tipos de filtros para permitir buscar favores que una persona desea realizar.
- Poder enviar notificaciones de un favor solicitado según los filtros aplicados por el usuario (ubicación, personas con vehículo, personas con cierto nivel de estudio, usuarios con cierta calificación, etc.).
- Mejorar el diseño de las pantallas para que sean más atractivas y fáciles de usar.

Por otra parte, se pretende generar la aplicación para las plataformas iOS y Android.

Al final se desea desplegar y publicar la aplicación en las tiendas Appstore y Google Play, con el objetivo de que las personas puedan descargarla y utilizarla para facilitar su vida o ganar dinero extra ayudando con los favores solicitados.

5. Glosario

TFM: Trabajo Fin de Máster.

Acolitar: jerga dicha en Ecuador que hace referencia a ayudar, o apoyar a alguien. Utilizado generalmente cuando se pide ayuda o desea que lo acompañen a cualquier lugar.

Android: Sistema operativo basado en Linux. Fue diseñado para dispositivos móviles. Desarrollado por Android Inc., y en 2005 adquirido por Google.

iOS: Sistema operativo móvil de la multinacional Apple Inc. No permite la instalación de iOS en hardware de terceros.

Google Play: Tienda de aplicaciones oficial de la plataforma Android.

Diagrama de Gantt: Herramienta gráfica que muestra el tiempo de dedicación previsto para cada tarea a lo largo del proyecto.

App (Application): Término utilizado para referirse a las aplicaciones de dispositivos móviles.

Plugin: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

JSON (JavaScript Object Notation): Es un formato de texto sencillo para el intercambio de datos.

Diseño Centrado en el Usuario (DCU): Filosofía de diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso.

Android Studio: Entorno de desarrollo integrado para la plataforma Android.

Backend: Parte de un sistema que se encarga de procesar la entrada desde la parte que interactúa con los usuarios y devolverle la información que se requiere.

Firebase: Es una plataforma y servicio de backend para el desarrollo de aplicaciones web y móviles, lanzada por Google.

Modelo Vista Controlador (MVC): Patrón arquitectónico de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

Ionic framework: Kit de herramientas de interfaz de usuario para el desarrollo de aplicaciones móviles híbridas.

App híbrida: una aplicación construida para ser utilizada o implementada en distintos sistemas operativos móviles, tales como, Android, iOS, etc.

Responsive: Se refiere a un diseño web adaptativo que busca la correcta visualización de una misma página o aplicación en dispositivos de distinto tamaño.

Caso de uso: es la descripción de una acción, actividad o funcionalidad.

6. Referencias

- [1] El País. ¿Cómo construir cadenas de favores con el móvil? - [01 mar, 2019]
https://elpais.com/tecnologia/2017/01/30/actualidad/1485777001_190227.html
- [2] Le Cool Barcelona. Helpin'; la app para intercambiar favores - [03 mar, 2019]
<https://barcelona.lecool.com/interviews/entrevista-carlos-mauri-helpin/>
- [3] Cultura Geek. FavForMe: la primera app para pedir favores - [05 may, 2020]
<http://culturageek.com.ar/favforme-aplicacion-ofrecer-pedir-favores/>
- [4] Google Play. Aplicación móvil FavForMe - [05 may, 2020]
<https://play.google.com/store/apps/details?id=com.appstract.favforme>
- [5] El Confidencial. Estrategias para pedir lo que quieras y obtener siempre un "sí" - [06 may, 2020]
https://www.elconfidencial.com/alma-corazon-vida/2013-01-02/estrategias-para-pedir-lo-que-quieras-y-obtener-siempre-un-si_208870/
- [6] Applesfera. Focus, app que te ayuda a centrarte en el trabajo - [02 may, 2020]
<https://www.applesfera.com/aplicaciones-ios-1/focus-app-que-te-ayuda-a-centrarte-trabajo-app-semana>
- [7] Tu Blog Tecnológico. Apps que ayudan a mejorar la productividad en el trabajo - [01 mar, 2021]
<http://tublogtecnologico.com/apps-para-mejorar-la-productividad-en-el-trabajo/>
- [8] Google Play. Uber - Pide tu viaje - [04 mar, 2020]
https://play.google.com/store/apps/details?id=com.ubercab&hl=es_419
- [9] Google Play. Aplicación móvil Cabify - [05 mar, 2020]
https://play.google.com/store/apps/details?id=com.cabify.rider&hl=es_EC
- [10] Kanban Tool. Tablero kanban Personal - [05 abr, 2020]
<https://kanbantool.com/es/tablero-kanban-personal>
- [11] Facile Things. ¿Cómo gestionar tus proyectos con kanban personal? - [03, julio, 2020]
<https://facilethings.com/blog/es/personal-kanban>
- [12] Geo Innova. HTML, CSS y JAVASCRIPT. Lenguajes para el desarrollo de páginas web - [03 may, 2021]
<https://geoinnova.org/blog-territorio/html-css-y-javascript-lenguajes-para-el-desarrollo-de-paginas-web/>
- [13] GIT. Software de control de versiones - [03 may, 2021]

<https://git-scm.com/>

- [14] GitHub. Contenedor de proyectos utilizando Git – [05 may, 2021]
<https://github.com/>
- [15] Postman API Client. Documentación - [06 may, 2021]
<https://www.postman.com/product/api-client/>
- [16] Node.js. Entorno en tiempo de ejecución multiplataforma - [12 may, 2021]
<https://nodejs.org/es/about/>
- [17] Npm. Sistema de gestión de paquetes para Node.js - [12 may, 2021]
<https://docs.npmjs.com/about-npm>
- [18] Visual Studio Code. Editor de código fuente - [14 may, 2021]
<https://code.visualstudio.com/docs>
- [19] Angular. Framework de aplicaciones web en TypeScript - [15 may, 2021]
<https://angular.io/docs>
- [20] Angular CLI. interfaz de líneas de comandos para Angular - [15 may, 2021]
<https://angular.io/cli>
- [21] Android Studio. Entorno de desarrollo para Android - [16 may, 2021]
<https://developer.android.com/docs>
- [22] Ionic Framework. Framework de desarrollo (apps híbridas) - [18 may, 2021]
<https://ionicframework.com/docs/>
- [23] Firebase. Plataforma de desarrollo de apps web y móviles - [20 may, 2021]
<https://firebase.google.com/docs>
- [24] Material UOC: Recursos otorgados en el Máster de Desarrollo de aplicaciones para dispositivos móviles, a lo largo de todo el proyecto.
- Diseño centrado en el usuario.
 - Tecnología y desarrollo en dispositivos móviles.
 - Exposición de contenidos en video.
 - Presentación de documentos y elaboración de presentaciones.
 - Redacción de textos científico-técnicos.
 - Trabajo final de máster.
 - Módulo 1. Introducción al trabajo final.
 - Módulo 2. El trabajo final como proyecto.
 - Módulo 3. La gestión del proyecto a lo largo del trabajo final.

7. Anexos

En este capítulo se indican los apartados que son demasiado extensos para incluir dentro de la memoria, que tienen un carácter autocontenido y que complementan el presente documento.

Los anexos que conforman este Trabajo Final de Máster son los siguientes:

Anexo I – Encuesta realizada (01Encuesta.pdf): En este documento se muestra la lista de preguntas realizadas a las personas durante la fase de análisis.

Anexo II – Resultados de la encuesta (02ResultadosEncuesta.pdf): Este documento muestra los resultados obtenidos después de realizar la encuesta a las personas.

Anexo III – Análisis completo de los resultados de la encuesta (03AnalisisResultadosEncuesta.pdf): El documento presenta todo el análisis que se hizo sobre los resultados de las encuestas realizadas para definir a los usuarios de la aplicación.

Anexo IV – Diagrama de clases y entidad relación (04DiagramasClasesER.vpp): Este archivo contiene el diagrama de clases y entidad-relación realizado en la fase de diseño, pero que se no se logró implementar.

Anexo V – Recursos incluidos en la memoria (05Recursos.zip): Este archivo contiene todas las imágenes (diseño de pantallas, arquitecturas, diagramas, etc.) que se incluyeron en la memoria del trabajo final.

Anexo VI – Instrucciones de configuración y ejecución (06InstruccionesCompilacionEjecucion.pdf): Este documento contiene las instrucciones sobre cómo compilar el proyecto con Ionic y generar el archivo .APK para dispositivos Android. Además incluye indicaciones para ejecutar la app en un emulador o dispositivo físico.

Anexo VII – Versión empaquetada de la app (07acolitapp.apk): Este archivo es la aplicación generada en su versión beta para dispositivos móviles Android. La aplicación móvil también se puede descargar desde la siguiente URL: https://drive.google.com/file/d/1H3EiZ52utC7S1Ou9sunC_m3MoA6Zfm5T/view?usp=sharing

Anexo VIII – Manual de Usuario (08ManualUsuario.pdf): Este anexo contiene el manual de usuario, que explica cómo utilizar la aplicación móvil Acolitapp.

Anexo IX – Código fuente de la app (09CodigoFuente.zip): Este archivo contiene todo el código fuente y recursos utilizados en la aplicación híbrida realizada con Ionic. El código fuente también puede descargarse del repositorio GitHub (<https://github.com/jorg3lui5/acolitapp.git>).

Anexo X – Video de presentación del trabajo (10Presentacion.avi): Es el video que contiene la presentación final de trabajo.