



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN
TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE
GUATEMALA “YodonoGt”**

Oscar Francisco Oliva Monterroso

Asesorado por el Ing. Everest Darwin Medinilla Rodríguez

Guatemala, febrero de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN
TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE
GUATEMALA "YodonoGt"**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

OSCAR FRANCISCO OLIVA MONTERROSO

ASESORADO POR EL ING. EVEREST DARWIN MEDINILLA RODRÍGUEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. Luis Fernando Espino Barrios
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE GUATEMALA “YodonoGt”

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 20 de febrero de 2019.

Oscar Francisco Oliva Monterroso



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Guatemala, 23 de julio de 2019.

Ing. Carlos Alfredo Azurdia Morales
Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Respetable Ing. Azurdia:

Por medio de la presente hago de su conocimiento que el estudiante Oscar Francisco Oliva Monterroso que se identifica con CUI No. 2494076980101 y código estudiantil No. 201403651 de la Facultad de Ingeniería, de la Universidad de San Carlos de Guatemala, de la Carrera de Ingeniería en Ciencias y Sistemas, ha finalizado en su totalidad su trabajo de graduación que recibe el nombre de APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE GUATEMALA. "YodonoGt".

Sin otro particular, me es grato suscribirme.

Atentamente,

F:

Medinilla Rodríguez, Everest Darwin

emedin@gmail.com

5966 3689

Everest Darwin Medinilla Rodríguez
Ingeniero en Ciencias y Sistemas
Colegiado 4,332



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 31 de julio de 2019

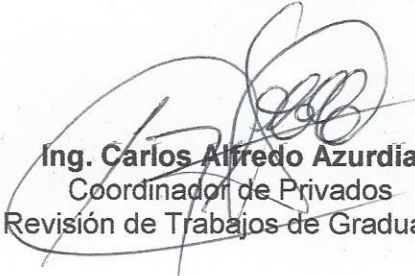
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **OSCAR FRANCISCO OLIVA MONTERROSO** con carné **201403651** y CUI **2494 07698 0101** titulado **“APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN TRES CATEGORIAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE GUATEMALA –YodonoGt-”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdía
Coordinador de Privados
y Revisión de Trabajos de Graduación



ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE GUATEMALA "YodonoGt"**, realizado por el estudiante, OSCAR FRANCISCO OLIVA MONTERROSO aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

Carlos Gustavo Alonzo
Msc. Ing. *Carlos Gustavo Alonzo*
Director
Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 07 de febrero de 2020

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **APLICACIÓN PARA LA PUBLICACIÓN, CONSULTA Y MONITOREO DE DONATIVOS EN TRES CATEGORÍAS: MEDICINA, ROPA Y MATERIAL DIDÁCTICO, EN LA CIUDAD DE GUATEMALA "YodonoGT"**, presentado por el estudiante universitario: **Oscar Francisco Oliva Monterroso**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.



Inga. Aurelia Anabela Cordova Estrada

Decana



Guatemala, febrero de 2020.

AACE/asga
cc

ACTO QUE DEDICO A:

- Dios** Porque todo logro alcanzado ha sido por Él y para Él.
- Mis padres** Jeannette Monterroso Morales y Oscar Francisco Oliva Ramos, porque nunca me hizo falta nada, y siempre he tenido su cariño y apoyo incondicional.
- Mi hermana** María José Oliva Monterroso, por ser un ejemplo por seguir en el trascurso de mi vida.
- Mis tíos** Claudia Lorena Bracamonte Gil y Edgar Esduardo Oliva Ramos, por ser una importante influencia en mi carrera.
- Mi abuelo** Oscar Fernando Oliva Fernández, por sus consejos y cariño incondicional.
- María Marcelina Saso** Por ser una parte fundamental en mi vida.
- Yoc**

AGRADECIMIENTOS A:

Dios	Por ser mi guía, y acompañarme en todo momento.
Familia	Por su paciencia y apoyo durante el transcurso de la carrera.
Mis amigos	Por apoyarme y brindarme su amistad, porque pude aprender de cada uno de ellos.
Mi asesor de tesis	Everest Darwin Medinilla Rodríguez, por su tiempo y guía para realizar mi trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	VII
RESUMEN.....	IX
OBJETIVOS.....	XI
INTRODUCCIÓN	XIII
JUSTIFICACIÓN.....	XV
1. MARCO TEÓRICO.....	1
1.1. Modelo de intermediación y su impacto en Guatemala	1
1.1.1. Análisis de modelos de intermediación existentes....	1
1.1.2. Empresas u organizaciones que implementan este modelo de negocio.....	1
1.1.3. Relación del modelo de intermediación con la tecnología escogida.....	2
1.2. Aplicaciones móviles	2
1.2.1. Tipos de aplicaciones móviles	2
1.2.1.1. Aplicaciones móviles nativas	3
1.2.1.2. Aplicaciones móviles web.....	4
1.2.1.3. Aplicaciones móviles híbridas.....	6
1.3. Herramientas de desarrollo móvil	7
1.3.1. Plataformas de desarrollo móvil.....	7
1.3.1.1. Ionic	7
1.3.1.2. Xamarin	8
1.3.1.3. Flutter	9
1.3.1.4. Matriz de selección	10

1.4.	Firebase	11
1.4.1.	Cloud Storage	12
1.4.2.	Principales servicios ofrecidos	12
1.5.	<i>Cloud Computing</i>	13
1.6.	Microinteracciones	14
1.6.1.	Estados del arte	14
1.6.2.	Relevancia y estructura de las microinteracciones	15
1.7.	Geolocalización	17
1.7.1.	Tipos de geolocalización	17
1.7.2.	Google Maps	18
2.	IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN ..	19
2.1.	Antecedentes	19
2.2.	Identificación del problema.....	20
2.3.	Propuesta de solución.....	20
2.4.	Usuarios de la aplicación	22
3.	DISEÑO DE PLATAFORMA PARA CONSULTA Y PUBLICACIÓN DE DONATIVOS.....	23
3.1.	Diseño de arquitectura por utilizar.....	23
3.2.	Prototipo.....	24
3.2.1.	Pantalla inicial	24
3.2.2.	Pantalla selección de tipo de acceso	25
3.2.3.	Pantalla de bienvenida	26
3.2.4.	Pantalla menú principal	27
3.2.5.	Pantalla de perfil.....	28
3.2.6.	Pantalla selección tipo de donativo	29
3.2.7.	Pantalla registro de donativo	30

3.2.8.	Pantalla de lista de donativos	33
3.2.9.	Pantalla de lista de traslados	34
3.2.10.	Pantalla de ubicación.....	35
3.2.11.	Servicios de ruta y lugar	36
3.2.12.	Pantalla de visualización de perfiles	37
3.2.13.	Pantalla registro de recolección.....	38
3.2.14.	Pantalla confirmación de entrega	39
3.2.15.	Pantalla registro de entrega.....	40
3.2.16.	Pantalla calificación del voluntario	41
3.3.	Inversión	41
4.	DOCUMENTACIÓN	43
4.1.	Requisitos.....	43
4.1.1.	Implementación de la aplicación desarrollada	43
4.2.	Herramientas para el desarrollo de la aplicación	43
4.2.1.	Visual Studio Code	43
4.2.2.	Flutter SDK.....	43
4.2.3.	Plugin Flutter y Dart	44
4.2.4.	Google USB Driver	44
4.2.5.	Cloud Firestore	44
4.2.6.	Android SDK.....	44
4.3.	Hardware	45
4.4.	Software	46
4.5.	Tutorial de desarrollo y referencias	46
4.5.1.	Instalación en Windows	46
4.5.1.1.	Obtener SDK de Flutter	46
4.5.1.2.	Actualizar path.....	47
4.5.1.3.	Correr flutter doctor.....	47
4.5.1.4.	Descargar e instalar Android Studio	48

- 4.5.1.5. Configurar dispositivo Android..... 49
- 4.5.1.6. Descargar e instalar VS Code50
- 4.5.1.7. Instalar los complementos Flutter y
Dart50

CONCLUSIONES.....51

RECOMENDACIONES53

BIBLIOGRAFÍA.....55

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Tipos de aplicaciones móviles.....	3
2.	Servicios Firebase.....	12
3.	Representación de la relación del diseño de interacción con otras disciplinas.....	15
4.	Elementos que conforman la estructura de una microinteracción	16
5.	Diseño de arquitectura	23
6.	Mockup pantalla inicial	24
7.	Mockup pantalla selección de tipo de acceso	25
8.	Mockup pantalla de bienvenida	26
9.	Mockup menú principal	27
10.	Mockup pantalla de perfil	28
11.	Mockup selección de tipo de donativo.....	29
12.	Mockup pantalla registro de donativo.....	30
13.	Mockup características de donativos	31
14.	Mockup características de donativos II	32
15.	Mockup pantalla de lista de donativos.....	33
16.	Mockup pantalla de lista de traslados	34
17.	Mockup ubicación y recolección.....	35
18.	Mockup servicios de ruta y lugar	36
19.	Mockup pantalla de visualización de perfiles	37
20.	Mockup pantalla registro de recolección	38
21.	Mockup confinación de entrega	39
22.	Mockup de registro de entrega.....	40

23.	Mockup de calificación del voluntario.....	41
24.	Flutter doctor.....	48
25.	Instalación Android Studio	49

TABLAS

I.	Ventajas y desventajas aplicaciones móviles nativas.....	4
II.	Ventajas y desventajas de aplicaciones móviles web.....	5
III.	Ventajas y desventajas aplicaciones móviles híbridas.....	7
IV.	Ventajas y desventajas de Ionic.....	8
V.	Ventajas y desventajas de Xamarin.....	9
VI.	Ventajas y desventajas de Flutter.....	10
VII.	Matriz de selección de plataformas.....	10

GLOSARIO

Android	Sistema operativo para dispositivos móviles, basado en GNU Linux desarrollado por Google, gratuito y multiplataforma.
API	<i>Application Programming Interface</i> . Especifica como deben de interactuar los componentes de software con los demás.
IOS	Sistema operativo de celulares y tabletas de la empresa Apple Inc. basado en OS X.
Microinteracción	Son los detalles funcionales e interactivos de un producto.
Tracking	Acciones con las que se siguen y se miden índices utilizados para cuantificar el comportamiento de determinadas variables en distintos momentos del tiempo.
Widget	Un <i>widget</i> es una descripción inmutable de parte de una interfaz de usuario.
YodonoGt	Nombre de la aplicación.

RESUMEN

YodonoGT es un sistema descentralizado para la publicación, consulta y monitoreo de donativos en tres categorías: medicina, ropa y material didáctico, para instituciones sin fines de lucro.

Para el desarrollo de la aplicación móvil se detectó la necesidad de insumos en lugares de la ciudad de Guatemala, como son: escuelas, albergues y hospitales; los cuales no cuentan con recursos suficientes para atender a todas las personas.

El sistema está diseñado para conectar personas y promover una cultura de solidaridad e incrementar el voluntariado dentro de la ciudad de Guatemala, unificando personas que estén en la posibilidad de hacer donaciones de ropa, medicina o material didáctico y personas con tiempo y transporte para trasladar los donativos a lugares de almacenamiento y distribución.

El sistema provee un módulo *tracking* que será el encargado de mostrar en tiempo real el proceso en el cual se encuentran los donativos, el nombre de los voluntarios y donantes que estarán suscritos, todo dentro de una interfaz agradable para sus usuarios.

OBJETIVOS

General

Proveer un sistema descentralizado para la publicación, consulta y monitoreo de donativos en tres categorías: medicina, ropa y material didáctico en la ciudad de Guatemala, para instituciones sin fines de lucro.

Específicos

1. Proveer un medio en el cual las personas tengan la disposición de publicar sus donativos, con características básicas e información de contacto.
2. Desarrollar un medio en el cual las personas tengan la disposición de trasladar los donativos a los lugares destinados.
3. Implementar un sistema de búsqueda por preferencia para determinar la ubicación o el tipo de donativo.
4. Documentar un módulo que permita monitorear el proceso en el cual se encuentra el donativo.
5. Documentar y publicar los lugares a los cuales se entregarán los donativos.

INTRODUCCIÓN

Un donativo es algo que se da de forma voluntaria y sin esperar alguna remuneración económica. Tanto en la ciudad de Guatemala como en el resto del mundo existen personas dispuestas a colaborar con otras más necesitadas, simplemente por el servicio altruista intrínseco al ser humano.

Lamentablemente, según La Comisión Económica para América Latina y el Caribe (CEPAL) para 2017, el número de personas que se encuentra viviendo en la pobreza llegó a 184 millones (30,2 % de la población), por lo que se les dificulta el acceso a medicamento, vestimenta o alimento. Debido a esta gran necesidad, se crea un sistema que ayuda a estas personas a través de donativos entregados a terceros.

En la ciudad de Guatemala hay personas que tienen medicina sobrante de tratamientos médicos, ropa sin usar o libros que ya no utilizan y que les gustaría donar, pero no encuentran los medios adecuados para hacerlo, ya sea por falta de tiempo, transporte o información. Así, se ve la necesidad de implementar un sistema que pueda conectar personas que donan y recolectan donativos, con lugares que almacenan estos donativos para su posterior distribución.

JUSTIFICACIÓN

En la ciudad de Guatemala se puede apreciar en televisión o medios escritos a personas realizando donaciones, ya sea, a gente necesitada de recursos, como ropa y medicina, u organizaciones que se encargan de distribuir las. Sin embargo, también existe una gran cantidad de personas con el deseo de colaborar con donaciones, pero deciden no involucrarse cuando se les presenta algún impedimento, como la falta de tiempo, transporte o información.

Otro factor por el cual algunas personas se rehúsan a realizar donaciones es por la trazabilidad, debido a que parte o la totalidad de los donativos no llega a los lugares que se acordaron desde un principio.

En la ciudad de Guatemala no existe conocimiento generalizado sobre la publicación, consulta y monitoreo de donativos, por lo que se ve necesario implementar una plataforma descentralizada, que atraiga personas que quieran realizar donaciones, que tengan disponibilidad de tiempo y transporte, e instituciones que reciban, almacenen y distribuyan los donativos a las personas con necesidad de estos recursos.

Además, para solventar problemas expuestos anteriormente, se ve necesario implementar un medio óptimo de transparencia, donde se conozcan en todo momento los datos personales de las personas involucradas en los traslados con el objetivo de fomentar una cultura de solidarismo.

1. MARCO TEÓRICO

1.1. Modelo de intermediación y su impacto en Guatemala

Un modelo basado en la intermediación permite a los intermediarios ser los creadores del mercado: juntando a los interesados en ofrecer algún producto o servicio con los interesados en comprar o adquirir el servicio, facilitando los intercambios

1.1.1. Análisis de modelos de intermediación existentes

Se trata de facilitadores de transacciones, dedicados a poner en contacto a vendedor con comprador. En este modelo, la información de los consumidores y sus hábitos de compra son muy valiosas, en especial cuando la información se analiza minuciosamente y se utiliza para dirigir campañas de mercadeo.

1.1.2. Empresas u organizaciones que implementan este modelo de negocio

Entre las empresas más reconocidas que están implementando este modelo de negocio, se puede mencionar a Uber, que es la empresa de taxis más potente del mundo y no tiene ni un solo taxi en propiedad. Alibaba es el comercio *online* más potente y no tiene productos en su inventario. Airbnb ofrece millones de habitaciones en todo el mundo y no posee ninguna de ellas. Facebook es el medio de comunicación más consultado diariamente y no produce ningún contenido.

1.1.3. Relación del modelo de intermediación con la tecnología escogida

En este trabajo de graduación se utiliza como base un modelo de intermediación, debido a que conecta personas interesadas en ofrecer un servicio (voluntarios) con personas interesadas en hacer entrega de donativos (donantes). Sin embargo, cabe mencionar que el objetivo de estos modelos de negocio disruptivos busca obtener ingresos, mientras que la aplicación desarrollada es para personas e instituciones sin fines de lucro, debido a que en ningún momento cobrará por los servicios, por lo que sí es totalmente gratuita.

La meta de la aplicación es generar el deseo de colaborar con personas que no tienen recursos como ropa, material didáctico o medicinas, de una forma transparente y accesible.

1.2. Aplicaciones móviles

Una aplicación móvil, en esencia no deja de ser un software; es una aplicación informática creada con el objetivo de ser ejecutada en teléfonos o dispositivos inteligentes. Las aplicaciones móviles facilitan la vida cotidiana, ya que, permiten a los usuarios realizar un conjunto de tareas o instrucciones de cualquier tipo.

1.2.1. Tipos de aplicaciones móviles

Cuando se necesita entregar una aplicación móvil, existen tres soluciones principales disponibles, cada una con sus propias fortalezas y debilidades las cuales se pueden agrupar en aplicaciones nativas, web móviles e híbridas.

Figura 1. Tipos de aplicaciones móviles



Fuente: Y2K Webs. https://y2kwebs.com/wp-content/uploads/2016/06/apps_hibridwebnative1.jpg. Consulta: abril de 2019.

1.2.1.1. Aplicaciones móviles nativas

Normalmente, el código nativo es la solución que la mayoría de los desarrolladores piensa cuando se necesita crear una aplicación móvil. Para crear una aplicación nativa, los desarrolladores deben escribir en el idioma predeterminado para cada plataforma móvil específica, que es Objective-C o Swift para dispositivos iOS, Java para Android y C # o XAML para Windows Universal.¹

Este tipo de desarrollo viene con varias ventajas fuertes sobre las otras opciones. Primero, las herramientas de desarrollo están estrechamente integradas en la plataforma del dispositivo. Los desarrolladores pueden trabajar en IDE configurados para crear aplicaciones móviles para esa plataforma: Xcode para iOS y Android Studio para Android. Segundo, dado que el desarrollo se realiza en el marco nativo, todas las API y características nativas están

¹ GRIFFITH, Chris. *Mobile App Development with Ionic, Revise Edition*. <https://learning.oreilly.com/library/view/mobile-appdevelopment/9781491998113/ch01.html>. Consulta: abril de 2019.

disponibles para el desarrollador sin la necesidad de soluciones de puente adicionales. La tercera ventaja es que el rendimiento de la aplicación será lo mejor posible. Dado que la aplicación se ejecuta de forma nativa, no hay capas intermedias de código que puedan afectar el rendimiento.²

Tabla I. **Ventajas y desventajas aplicaciones móviles nativas**

Ventajas	Desventajas
<ul style="list-style-type: none"> • Acceso completo al hardware del dispositivo. • Mejor experiencia del usuario gracias a la explotación de las características del móvil. • Permite el envío de notificaciones a los usuarios. • Actualización de la aplicación constante. 	<ul style="list-style-type: none"> • Diferentes configuraciones para cada plataforma de destino. • Suele ser más caro desarrollar este tipo de aplicación. • El código utilizado para una aplicación no es reutilizable entre las diferentes plataformas. • Dependencia del modelo del dispositivo.

Fuente: elaboración propia.

1.2.1.2. **Aplicaciones móviles web**

Si se regresa a 2007, cuando fue el anuncio por primera vez del iPhone, no existían las aplicaciones de terceros, ni siquiera una tienda virtual, sino que, lo único que se podría adquirir eran aplicaciones de terceros disponibles como aplicaciones web móviles. Si bien es cierto que en la actualidad este no es el caso, aún existe la posibilidad de crear aplicaciones web móviles. Estas aplicaciones se cargan a través del navegador web móvil del dispositivo. El desarrollo de una aplicación web y una aplicación móvil llega a ser realmente

² GRIFFITH, Chris. *Mobile App Development with Ionic, Revise Edition*. <https://learning.oreilly.com/library/view/mobile-appdevelopment/9781491998113/ch01.html>. Consulta: abril de 2019.

parecida, ya que realmente lo que se busca es crear aplicaciones utilizando tecnologías web que serán entregadas a través del navegador del dispositivo.

Una de las ventajas de implementar esta solución, es que se puede tener un mayor alcance con la aplicación. Más allá de IOS y Android, plataformas móviles adicionales están disponibles, además se tiene un acceso directo al servidor web, el proceso de aprobación de la aplicación en ocasiones poder ser lento o presentar complicaciones, para aplicaciones nativas no es un problema.

El hecho de que estas aplicaciones se ejecuten dentro del navegador nativo trae consigo una serie de limitaciones. Primero, el navegador no tiene acceso a todas las capacidades del dispositivo. Por ejemplo, no hay capacidad para que el navegador acceda a la lista de contactos en el dispositivo. En segundo lugar, los usuarios están acostumbrados a ir a una tienda virtual para descargar la aplicación, mientras que con una aplicación móvil web se debe ingresar a un URL lo que sería algo poco común para los usuarios.

Tabla II. **Ventajas y desventajas de aplicaciones móviles web**

Ventajas	Desventajas
<ul style="list-style-type: none"> • Posibilidad de utilizar el mismo código de base, en múltiples plataformas. • Proceso de desarrollo más sencillo y económico. • No se necesita ninguna aprobación de terceros para poder ser publicada. • Siempre se puede contar con la última versión. • Se puede encontrar muchas estructuras de código ya preconstruídas. 	<ul style="list-style-type: none"> • Requiere de conexión a internet para su funcionamiento. • Pobre acceso y utilización de las características del hardware del dispositivo. • Dificultad para volverse viral debido a la falta de costumbre de utilizar una URL para acceder a la aplicación. • La experiencia del usuario y el tiempo de respuesta es menor que en una app nativa.

Fuente: elaboración propia.

1.2.1.3. Aplicaciones móviles híbridas

Una aplicación híbrida es una aplicación móvil nativa que utiliza un navegador web sin cromos (a menudo llamado *WebView*) para ejecutar la aplicación web. Esta solución utiliza una envoltura de aplicación nativa que interactúa entre el dispositivo nativo y la vista web. Las aplicaciones híbridas tienen una serie de ventajas. Al igual que las aplicaciones web móviles, la mayoría del código se puede implementar en múltiples plataformas. Al desarrollar en un lenguaje común, mantener el código base es más fácil. A diferencia de las aplicaciones web móviles, hay acceso completo a las funciones del dispositivo, generalmente a través de algún tipo de sistema de *plug-in*.

Sin embargo, esta solución tiene desventajas reales. Dado que la aplicación es solo una aplicación web, está limitada por el rendimiento y las capacidades del navegador en el dispositivo. El rendimiento puede variar ampliamente. Los dispositivos más antiguos a menudo tenían navegadores móviles con un rendimiento muy bajo, lo que significa que el rendimiento de la aplicación no era el ideal.

Aunque esta solución es una aplicación nativa, la comunicación entre *WebView* y las funciones nativas del dispositivo se realiza a través de complementos. Esto introduce otra dependencia en su proyecto y no garantiza que la API esté disponible a través de este método. Finalmente, los otros componentes de la interfaz de usuario nativo no están disponibles dentro de *WebView*.

Tabla III. **Ventajas y desventajas aplicaciones móviles híbridas**

Ventajas	Desventajas
<ul style="list-style-type: none"> • Posibilidad de distribución en las tiendas de iOS y Android. • Posibilidad de reutilización de código en múltiples plataformas. • Permite una instalación nativa pero construida con JavaScript, HTML y CSS. • Acceso a parte del hardware del dispositivo. 	<ul style="list-style-type: none"> • Falta de relación entre sistema operativo y diseño de la aplicación. • Experiencia del usuario más propia de la aplicación web que de la app nativa.

Fuente: elaboración propia.

1.3. Herramientas de desarrollo móvil

El ecosistema de aplicaciones móviles posee una gran variedad de herramientas de software que facilitan el flujo de trabajo y aumentan la eficiencia y productividad.

1.3.1. Plataformas de desarrollo móvil

Se presenta un breve análisis de tres plataformas de desarrollo móvil, exponiendo individualmente las ventajas y desventajas que cada una posee, luego se muestra una matriz de selección para evaluar la mejor solución basada en criterios compartidos.

1.3.1.1. Ionic

El Framework Ionic fue lanzado por primera vez en 2003, y en la actualidad es una de las plataformas más populares para el desarrollo de aplicaciones

móviles. Ionic se proporciona bajo la licencia MIT y está disponible en el sitio web oficial de Ionic Framework.

La característica principal de Ionic Framework es proporcionar los componentes de la interfaz de usuario que no están disponibles para el desarrollo de aplicaciones basadas en web. Por ejemplo, una barra de pestañas es un componente de interfaz de usuario común que se encuentra en muchas aplicaciones móviles. Pero este componente no existe como un elemento HTML nativo. Estos componentes se construyen con una combinación de HTML, CSS y JavaScript, y cada uno se comporta y se parece a los controles nativos que está recreando.³

Tabla IV. **Ventajas y desventajas de Ionic**

Ventajas	Desventajas
<ul style="list-style-type: none">• Una de sus principales ventajas es que es multiplataforma, se puede codificar una vez y generar apps para Android, IOS y Windows.• Framework libre y de código abierto.	<ul style="list-style-type: none">• No utiliza las características nativas de los dispositivos, en su lugar utiliza <i>webview</i>, lo que hace disminuir su rendimiento y fluidez.

Fuente: elaboración propia.

1.3.1.2. Xamarin

Xamarin.Forms es un kit de herramientas de interfaz de usuario multiplataforma que permite a los desarrolladores crear eficazmente diseños de interfaz de usuario nativa que se pueden compartir entre aplicaciones de iOS,

³ MINNICK, Brandon. *Introducción a Xamarin*. <https://docs.microsoft.com/es-es/xamarin/get-started/>. Consulta: abril de 2019.

Android y Plataforma universal de Windows. Para obtener acceso a la documentación completa de Xamarin.Forms.⁴

“Xamarin es una plataforma de desarrollo que permite codificar aplicaciones nativas, multiplataforma iOS, Android y Winsows Phone en C#”⁵

Tabla V. **Ventajas y desventajas de Xamarin**

Ventajas	Desventajas
<ul style="list-style-type: none"> • Codificación reducida y facilidad de comprensión. • Se escribe el código solo una vez. • Mantiene la experiencia de usuario propia de cada plataforma. • Posibilidad de crear código XAML y aprovechar potencia que provee MVVM. 	<ul style="list-style-type: none"> • Creación de UX que demande demasiado esfuerzo gráfico.

Fuente: elaboración propia.

1.3.1.3. Flutter

Flutter fue anunciado como parte del Mobile World Congress 2018, como un nuevo Framework para crear aplicaciones nativas tanto para Android como para iOS. Flutter es utilizado por desarrolladores y organizaciones de todo el mundo, es gratuito y de código abierto. Todo indica que será una plataforma prometedora, ya que la facilidad de codificación es uno de sus pilares.

⁴ MINNICK, Brandon. *Introducción a Xamarin*. <https://docs.microsoft.com/es/xamarin/get-started/>. Consulta: abril de 2019.

⁵ HERMES, Dan. *Xamarin Mobile Application Development: Cross-Platform C# and Xamarin Forms Fundamentals*. https://www.theseus.fi/bitstream/handle/10024/111535/Jaaskelainen_Timo.pdf?sequence=1&isAllowed=y. Consulta: abril de 2019.

Tabla VI. **Ventajas y desventajas de Flutter**

Ventajas	Desventajas
<ul style="list-style-type: none"> • Un solo proyecto que permitirá crear aplicaciones para Android y iOS. • Rapidez de desarrollo. • Calidad nativa. • Tecnología fácilmente personalizable. 	<ul style="list-style-type: none"> • Debido a su novedad aún no se encuentra con mucha documentación.

Fuente: elaboración propia.

1.3.1.4. **Matriz de selección**

Se muestra una matriz de selección para evaluar la mejor solución basada en criterios compartidos que se adaptan de mejor manera a los objetivos de la aplicación.

Tabla VII. **Matriz de selección de plataformas**

Solución	Variable	Lenguaje	Desempeño	GUI	Mercado y soporte	Reusabilidad	Popularidad	Precio	Total
	Peso %	8 %	19 %	15 %	20 %	13 %	5 %	20 %	100 %
IONIC	nota	9	5	8	8	7	7	6	
	ponderada	0,72	0,95	1,2	1,6	0,91	0,35	1,2	6,93
XAMARIN	nota	9	7	7	8	7	7	6	
	ponderada	0,72	1,33	1,05	1,6	0,91	0,35	1,2	7,16
FLUTTER	nota	8	10	9	6	6	6	8	
	ponderada	0,64	1,9	1,35	1,2	0,78	0,3	1,6	7,77

Fuente: elaboración propia.

Criterios de evaluación:

- Lenguaje: el criterio se basa a la curva de aprendizaje del lenguaje, su extensibilidad y soporte.
- Desempeño: se base en cómo funcionan las aplicaciones desde una perspectiva de rendimiento en tiempo de ejecución.
- GUI: cómo se juzgan a las aplicaciones dentro de los primeros segundos de su uso por lo tanto la GUI de una aplicación debe ser fácil de usar y atractiva.
- Mercado y soporte: se basa en tener un ecosistema dinámico con cantidad considerable de usuario o desarrolladores utilizando la herramienta.
- Reusabilidad: cantidad de código escrito que se puede reutilizar entre plataformas.
- Popularidad: cantidad y calidad de aplicaciones creadas a nivel mundial por la herramienta.
- Precio: cantidad de dinero que debe ser invertido para usar parte o totalidad de los componentes de cada herramienta.

Las tres plataformas presentadas tienen punteos similares, sin embargo, para este trabajo de graduación y el desarrollo de la aplicación móvil, se consideraron ciertos criterios más relevantes como lo son el precio, el desempeño y la *GUI* debido a que se consideran puntos clave para una mejor experiencia de usuario y de menor impacto en la inversión económica, por lo que se escogió Flutter como la mejor solución.

1.4. Firebase

Firebase se compone de un conjunto de herramientas dedicadas a la creación y mantenimiento de aplicaciones de alta calidad.

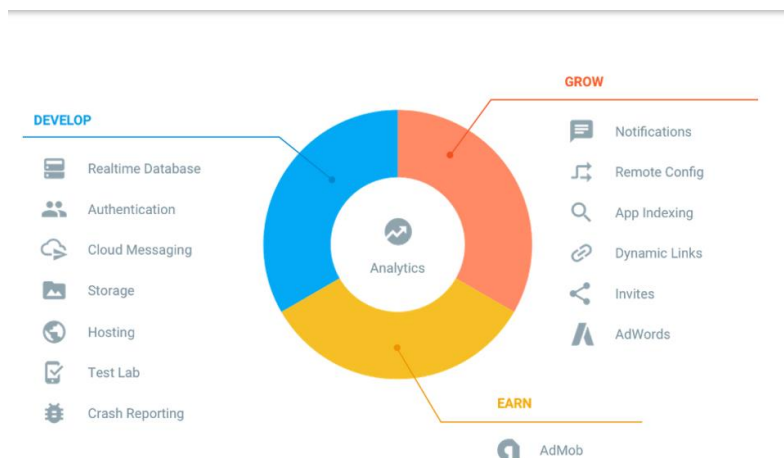
1.4.1. Cloud Storage

Cloud Storage para Firebase es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para tus apps de Firebase, sin importar la calidad de la red. Se puede utilizar usar los SDK para almacenar imágenes, audio, video y otros tipos de contenido generado por el usuario.⁶

1.4.2. Principales servicios ofrecidos

Se presenta una lista de los servicios más relevantes que ofrece Firebase para trabajar de una manera más eficiente y sencilla en nuestra aplicación.

Figura 2. Servicios Firebase



Fuente: RAMIREZ, Vicente. <http://s3.amazonaws.com/cdn.wp.m4ecnet/wp-content/uploads/2017/02/04190313/Firebase.png>. Consulta: abril de 2019.

⁶ Firebase. *Cloud Storage*. <https://firebase.google.com/docs/storage/?hl=es-419>. Consulta: mayo de 2019.

- Base de datos Realtime: sincroniza los datos de una base de datos local con la propia de Firebase en la nube, los cuales se almacenan en un formato JSON, con la posibilidad de aplicación de reglas para permitir request con token, se puede modificar en tiempo real, y los cambios se verán reflejados en la aplicación en tan solo segundos.
- Autenticación: provee la facilidad de inicio de sesión y gestión de cuentas dentro de cuentas desde la plataforma de Firebase. Firebase guarda datos de la conexión en la nube de una forma segura.
- Almacenamiento: utiliza una base de datos llamada Cloud Firestore que cuenta con flexibilidad y escalabilidad para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform.
- *Hosting*: se puede subir una aplicación y esta automáticamente contará con SSL y HTTP2, sin necesidad de configuraciones, para que el contenido esté disponible de forma segura.

1.5. **Cloud Computing**

El *cloud computing* es un término general que se refiere a la prestación de servicios alojados a través de Internet. Se trata de una tecnología avanzada que hace que todos los archivos, programas e información estén almacenados en Internet, como en una “nube”, siendo completamente irrelevante las capacidades de almacenaje de los ordenadores instalados por el cliente y prescindiendo así de los discos duros.⁷

⁷ Infotelecom. *¿Qué es el cloud computing?* <<https://infotelecom.es/blog/que-es-el-cloud-computing/>>. Consulta: abril de 2019.

1.6. Microinteracciones

Las microinteracciones son los detalles funcionales e interactivos de un producto, y los detalles, como dijo Charles Eames⁸, no son solo los detalles; Ellos son el diseño. Los detalles pueden hacer que interactuar con el producto sea más fácil y placentero, incluso si no los recordamos conscientemente. Algunas microinteracciones son práctica o literalmente invisibles. Algunas acciones microinteractivas pueden ser frustrantes, algunas aburridas y olvidadas, mientras que las mejores son atractivas e inteligentes.

1.6.1. Estados del arte

Con los primeros estudios sobre usabilidad y psicología cognitiva realizados por Donald A. Norman, junto a la investigación de Ben Shneiderman sobre Interacción Hombre-Computadora (IHC) y los aportes del pionero Bill Moggridge dentro de la disciplina del Diseño de Interacción, han llevado a aceptar que la tecnología digital no sólo se trata de ofrecer alta tecnología, llena de sofisticadas funciones de gran desempeño, sino que además deben conjugar una disposición de interfaces y usos, en las formas más convenientes para el ser humano.

Derivados de tales estudios e investigaciones surgieron otros fundamentos y disciplinas que resultaron convenientes en el entendimiento de la interacción. Estas interrelaciones están presentadas a través de la figura 3, la cual muestra con claridad la estrecha relación entre estas múltiples disciplinas, y señala la inclusión de varios fundamentos identificados como significativos en un proceso de diseño con enfoque incluyente.

⁸ EAMES, Charles. *See 100 Quotes by Charles Eames, (Eames Office, 2007).*
<https://www.homeanddecor.com.sg/uncategorized/10-quotes-by-charles-and-rayeames/>
Consulta: marzo de 2019.

Es necesario señalar que la misma figura aborda con delimitación lo relacionado con el desarrollo de software, por lo que fue adaptada de su versión original para buscar una mejor explicación, y no confundir con la inclusión de otras disciplinas más relacionadas con el diseño de otro tipo de productos.⁹

Figura 3. **Representación de la relación del diseño de interacción con otras disciplinas**



Fuente: SANTANA, Pedro César. http://incytde.org/web/wp-content/uploads/2015/03/RCS_VOL_76-1.pdf#page=45. Consulta: abril 2019.

1.6.2. Relevancia y estructura de las microinteracciones

Con la conceptualización anterior es posible conducir la atención en la microinteracciones. A pesar de que pueden pasar desapercibidas por los usuarios por su mínimo tamaño o casi invisibilidad, las microinteracciones son muy importantes, ya que pueden ser útiles para:

⁹ SANTANA, Pedro César. *Research on Computing Science 76 (2014)* http://incytde.org/web/wp-content/uploads/2015/03/RCS_VOL_76-1.pdf#page=43. Consulta: marzo de 2019.

- Ejecutar una tarea simple.
- Interconectar dispositivos.
- Interactuar con datos simples o básicos.
- Controlar algún proceso en ejecución.
- Ajustar un parámetro de configuración.
- Visualizar o crear una pequeña pieza de contenido como un mensaje de estado.

En una aplicación las microinteracciones son frecuentemente las responsables de convertirla en un producto predilecto o, en otros casos, simplemente en algo tolerado.

Su trascendencia puede ir aún más allá, hacer la vida de las personas más sencilla, divertida e, inclusive, más interesante, si son bien construidas. Las microinteracciones se conforman según la siguiente estructura de elementos¹⁰

Figura 4. **Elementos que conforman la estructura de una microinteracción**



Fuente: SANTANA, Pedro César. http://incytde.org/web/wp-content/uploads/2015/03/RCS_VOL_76-1.pdf#page=47. Consulta: abril 2019.

¹⁰ SANTANA, Pedro César. *Research on Computing Science 76 (2014)* http://incytde.org/web/wp-content/uploads/2015/03/RCS_VOL_76-1.pdf#page=46. Consulta: marzo de 2019.

1.7. Geolocalización

Se puede definir como geolocalización a la capacidad de determinar la posición geográfica o ubicación de algún dispositivo inteligente a través de coordenadas. La geolocalización es un término que ha ganado más popularidad en la actualidad ya que varias herramientas usadas en todo el mundo prestan servicios que requieren de estas capacidades.

1.7.1. Tipos de geolocalización

- GPS (sistema de posicionamiento global): es una red que se encuentra compuesta aproximadamente por 30 satélites, de los cuales 4 de estos se muestran para el dispositivo y cada uno de ellos envía señales sobre su ubicación cada cierto tiempo.
- GMS (sistema global para comunicaciones móviles): en todo el planeta se encuentran torres que brindan servicio telefónico, teniendo en cuenta cada una de las torres, la aproximación dentro cada una, el tiempo que tarda la señal al trasladarse de torre en torre y la fuerza de transmisión, se logra calcular la ubicación de nuestro dispositivo.
- Wifi: cada red wifi encendida emite una señal identificativa llamada MAC, sabiendo en todo momento a qué conexión está asociado alguien se puede determinar la localización en su teléfono u ordenador.

1.7.2. Google Maps

Google Maps se define como un servidor aplicaciones de mapas en la web que pertenece a Alphabet Inc. se encarga de ofrecer a los usuarios toda la información que necesiten sobre su ubicación actual. Además provee imágenes de mapas desplazables, fotografías por satélite del mundo e incluso el trazado de recorridos para llegar al lugar que se desee desde un punto inicial que puede ser la ubicación actual de los usuarios.

2. IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.1. Antecedentes

En la actualidad es una realidad que los mapas tradicionales están muriendo, ya no se necesita mirar un mapa con cientos de datos y recursos, ahora es el mapa el que se adapta a los gustos y a las relaciones de las personas.

En el pasar de los años, se ve cómo la tecnología ha revolucionado la forma en que las personas se localizan, además se le puede dar muchos usos de los que se puede mencionar dos:

- Geolocalización *online* como negocio
- Herramienta de comunicación

Según Donald W. Pfaff en su libro *El cerebro altruista: por qué somos naturalmente buenos*,¹¹ el cerebro altruista demuestra, desde el enfoque de la neurociencia, que las personas son naturalmente buenas porque están biológicamente predispuestas a ello.

Además, se menciona que el cerebro altruista constituye el estudio más importante sobre cómo y por qué medios, por mecanismos puramente físicos los seres humanos empatizan entre sí y responden de manera altruista. Esa

¹¹ PFAFF, D. W. *El cerebro altruista: por qué somos naturalmente buenos*. <https://dialnet.unirioja.es/download/articulo/6549740.pdf>
Consulta: abril de 2019.

amabilidad espontánea es el comportamiento natural, independientemente de los condicionamientos religiosos o culturales.

El autor menciona que el comportamiento está orientado a ayudar a las demás personas, sentir empatía es una de las reacciones naturales del cerebro.

2.2. Identificación del problema

A continuación, se listan las causas por las cuales se determinó desarrollar una aplicación, con el fin de conectar personas, en favor de realizar donativos y trasladarlos a instituciones sin fines de lucro:

- Falta de transporte, tiempo o información para donar; muchas personas tienen la disposición, pero por alguno de los factores mencionados, no lo hacen.
- Otro factor por el cual las personas se rehúsan a realizar donaciones es por la inseguridad, debido a que parte o la totalidad de los donativos, nunca llega a los lugares acordados en un principio.
- En la ciudad de Guatemala no existe conocimiento generalizado sobre la publicación, consulta y monitoreo de donativos.

2.3. Propuesta de solución

Implementar una aplicación multiplataforma descentralizada, que atraiga personas que quieran realizar donaciones, que tengan la disponibilidad de tiempo y transporte, e instituciones que reciban, almacenen y distribuyan los donativos a personas con necesidad de medicina, ropa o material didáctico.

Para el desarrollo de la aplicación se listaron las siguientes historias de usuario:

Voluntario

- Descripción: como voluntario, quiero tener una interfaz sencilla y entendible para poder listar los donativos que están pendientes de traslado.
- Descripción: como voluntario, quiero ver mi historial de donativos entregados para tener un control de cuánto he contribuido.
- Descripción: como voluntario, quiero poder visualizar mi reputación y comentarios de mis entregas.
- Descripción: como voluntario, quiero reservar el donativo y comprometerme a entregarlo en el plazo indicado.
- Descripción: como voluntario, quiero visualizar la ubicación de los donativos para facilitar la recolección.
- Descripción: como voluntario, quiero tener control sobre quién me entregará el donativo, conocer el lugar de destino, y contacto para poder comunicarme con los donantes.

Donante

- Descripción: como donante, quiero tener una interfaz amigable, para poder publicar los donativos y también la ubicación donde me encuentro.

- Descripción: como donante, quiero ver mi historial de donativos hechos para tener un control de cuánto he contribuido.
- Descripción: como donante, quiero visualizar mi reputación y comentarios de mi servicio.
- Descripción: como donante, quiero tener control sobre quién recogerá el donativo.
- Descripción: como donante, quiero llevar un control del proceso en el que se encuentra el donativo.
- Descripción: como donante, quiero tener un registro para validar que los donativos han sido entregados al lugar acordado.

2.4. Usuarios de la aplicación

La aplicación está diseñada para personas que viven en la ciudad de Guatemala y cuentan con teléfonos de sistema Android.

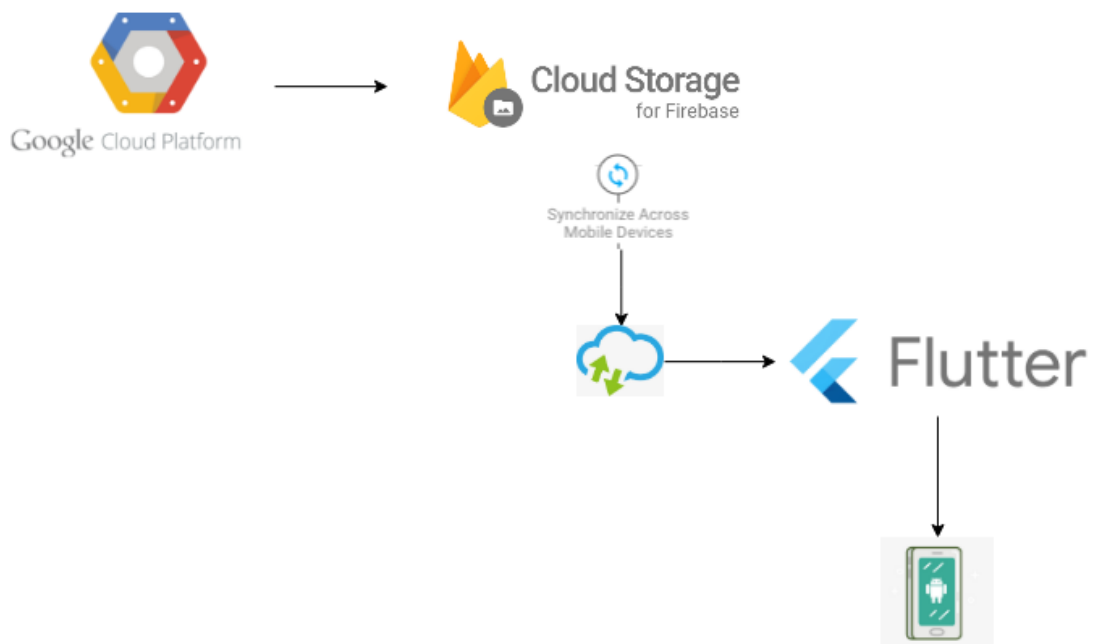
La aplicación no generará ingresos para los intermediarios o los suscritos en ella. Está diseñada para toda aquella persona que tenga a su disposición objetos para donar, en las categorías: medicina, ropa y material didáctico. Además para quienes tengan tiempo y transporte para trasladar esos donativos a instituciones sin fines de lucro.

3. DISEÑO DE PLATAFORMA PARA CONSULTA Y PUBLICACIÓN DE DONATIVOS

3.1. Diseño de arquitectura por utilizar

El diseño de la arquitectura muestra a una aplicación móvil Android, diseñada con el Framework Flutter, estos a su vez se encuentran conectados por medio de internet a Firebase y su base de datos, los proyectos añadidos a Firebase están asociados con proyectos de la nube de *Google Platform*, por lo cual se tiene acceso a todas las herramientas que este provee.

Figura 5. Diseño de arquitectura



Fuente: elaboración propia.

3.2. Prototipo

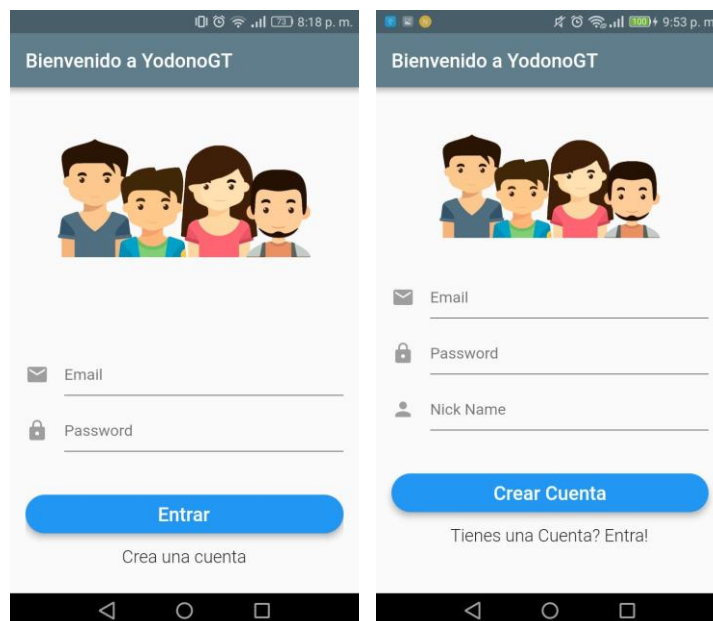
Se busca crear una aplicación que tenga una interfaz amigable y de fácil navegación, aprovechando al máximo las características que ofrecen las herramientas de desarrollo.

3.2.1. Pantalla inicial

La pantalla inicial ofrece dos opciones:

- Crear cuenta: si el usuario aún no tiene una cuenta, la aplicación da la posibilidad de crear una nueva con un correo y una contraseña.
- Entrar: si el usuario ya posee una cuenta, procederá a escribir su correo y contraseña para entrar al sistema.

Figura 6. Mockup pantalla inicial



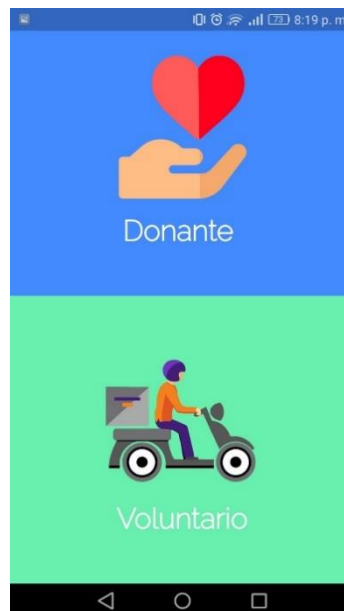
Fuente: elaboración propia.

3.2.2. Pantalla selección de tipo de acceso

Tipo de acceso, se brinda la opción de ingresar ya sea como voluntario o como donante, la aplicación permite que con la misma cuenta se pueda acceder a cualquier tipo de categoría con el fin de motivar a los usuarios, a optar por ambos roles cuando se tenga la disponibilidad.

- Donante: permitirá al usuario crear nuevas donaciones, con el objetivo de que un voluntario vea la publicación y se comprometa a recoger el donativo y llevarlo la institución establecida.
- Voluntario: mostrará una lista de los donativos disponibles, en el cual los usuarios pueden seleccionar según sus preferencias para su posterior traslado.

Figura 7. **Mockup pantalla selección de tipo de acceso**



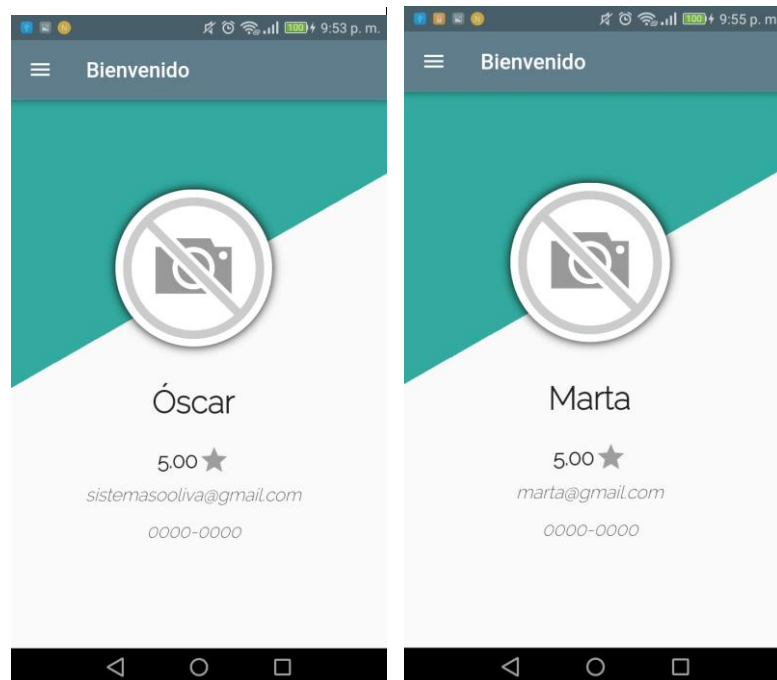
Fuente: elaboración propia.

3.2.3. Pantalla de bienvenida

En esta sección se muestran los perfiles de los usuarios recientemente creados, uno como donante y otro como voluntario, el perfil cuenta con lo siguiente:

- Foto de perfil: una foto que identificará al usuario de la aplicación.
- Punto de valoración: según la calificación que reciban por otros usuarios, se verá reflejado en su valoración.
- Correo electrónico.
- Número de teléfono.

Figura 8. Mockup pantalla de bienvenida



Fuente: elaboración propia.

3.2.4. Pantalla menú principal

El menú para donantes y para voluntario es similar, cuenta con cinco secciones, en las que comparten la sección de: bienvenidos, perfil y salir.

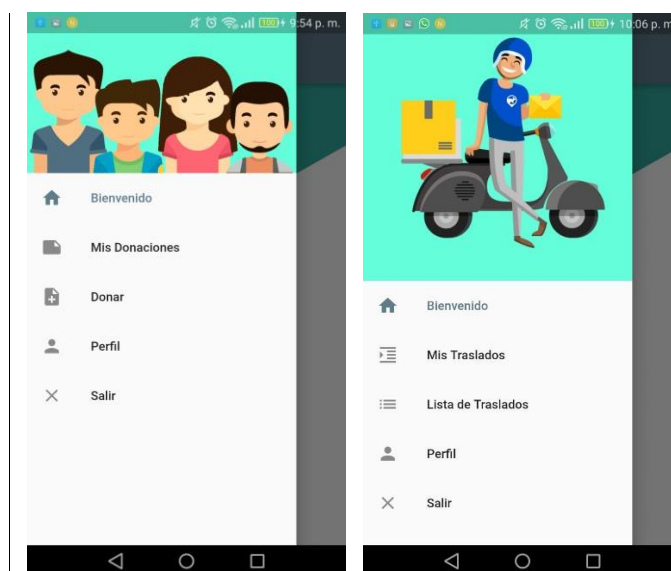
Para el menú de donante se cuenta con dos secciones específicas:

- Donar: permite crear los donativos
- Mis donaciones: muestra una lista de todos los donativos realizados

Para el menú del voluntario se cuenta con dos secciones específicas:

- Lista de traslados: muestra una lista general de todos los donativos.
- Mis traslados: Lista de donativos que el voluntario se ha comprometido a trasladar.

Figura 9. **Mockup menú principal**



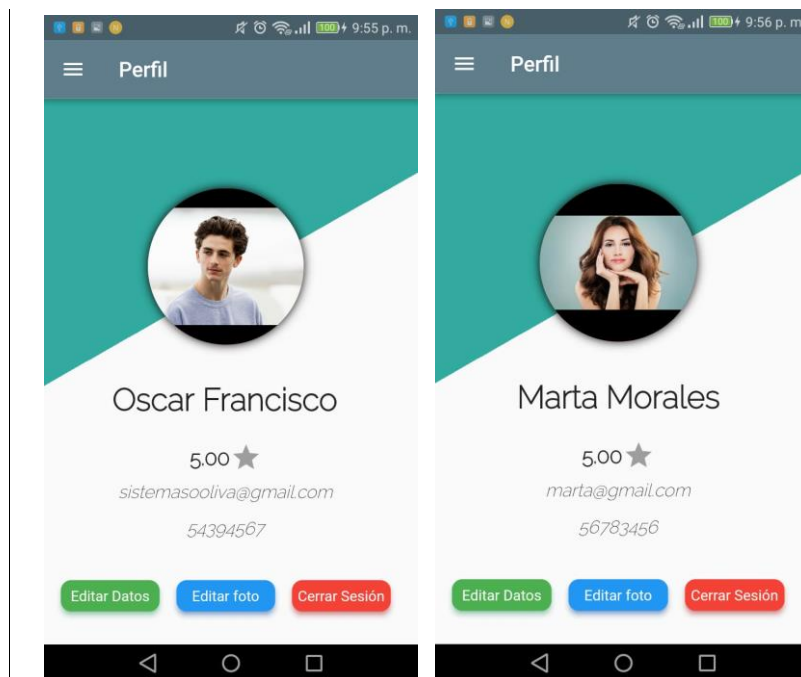
Fuente: elaboración propia.

3.2.5. Pantalla de perfil

En la pantalla de perfil, se podrá editar los datos personales de los usuarios; cuenta con tres opciones:

- Editar datos: se podrán editar los datos como el nombre y el número de teléfono del usuario.
- Editar foto: permitirá subir o actualizar la foto de perfil.
- Cerrar sesión: permitirá salir del sistema.

Figura 10. Mockup pantalla de perfil

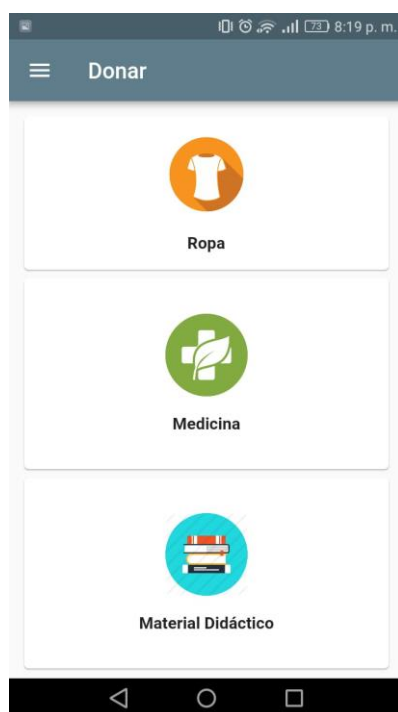


Fuente: elaboración propia.

3.2.6. Pantalla selección tipo de donativo

Esta sección aplica para los donantes, ya que la aplicación permitirá escoger entre tres categorías: ropa, medicina y material didáctico.

Figura 11. **Mockup selección de tipo de donativo**

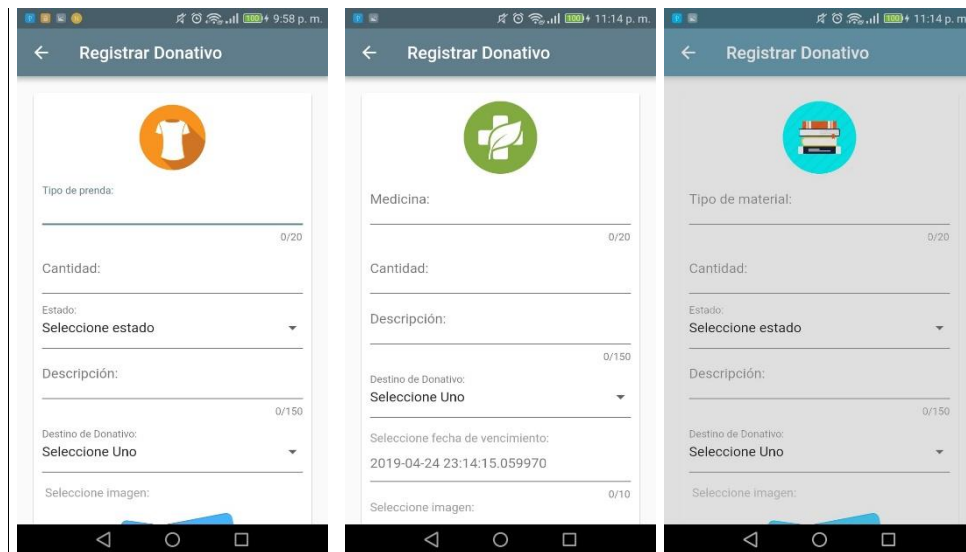


Fuente: elaboración propia.

3.2.7. Pantalla registro de donativo

En la pantalla registro de donativo, se desplegará un formulario el cuál se debe de llenar con las características del objeto que se quiere donar, según el donativo escogido se le brindarán características específicas como fecha de vencimiento en el caso de medicina.

Figura 12. Mockup pantalla registro de donativo



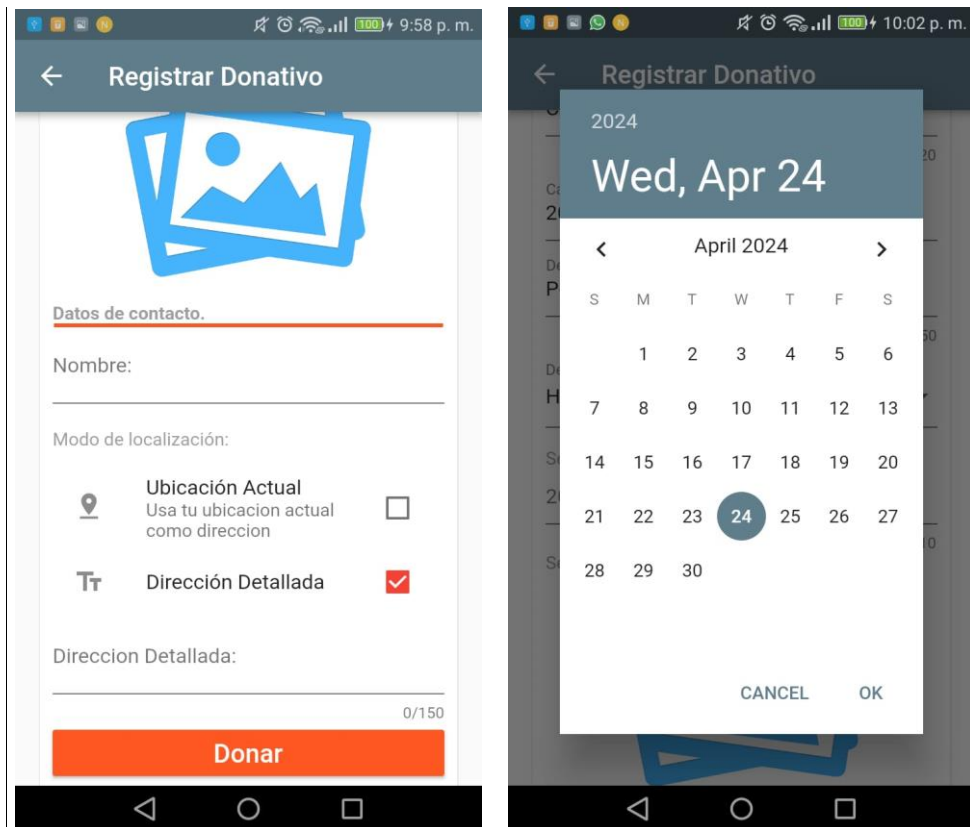
Fuente: elaboración propia.

Al momento de crear un donativo en la parte inferior de la pantalla habrá una sección llamada “Modo de localización” la cual brinda dos opciones:

- Ubicación actual: la cual se base en geolocalización y mapas para ubicar el donativo.
- Dirección detallada: la cual desplegará un cuadro de texto para ingresar manualmente la dirección.

Cuando se escoge la opción de registrar un donativo de medicina, la aplicación desplegará un calendario en el cual se podrá escoger la fecha de vencimiento dinámicamente.

Figura 13. **Mockup características de donativos**

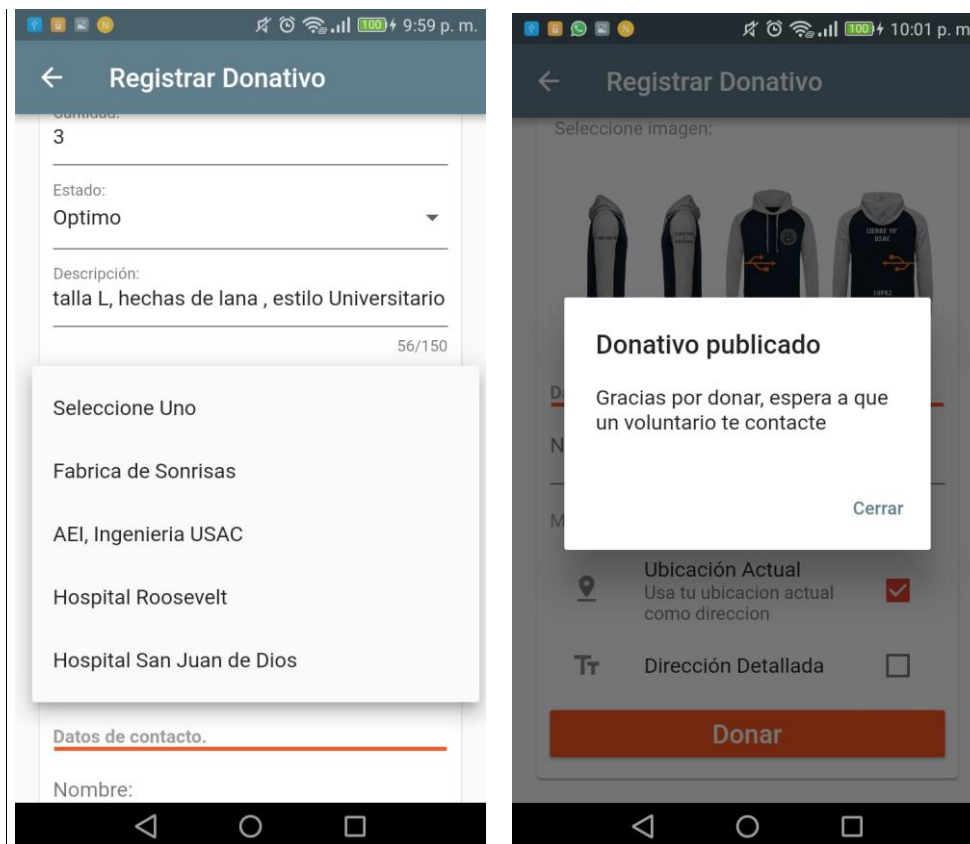


Fuente: elaboración propia.

El donante es el encargado de especificar hacia donde quiere que se traslade el donativo, de un menú desplegable el cual mostrará las opciones disponibles.

Una vez finalizado el llenado del formulario se procederá a presionar el botón de “Donar” el cual registrará el donativo en la base de datos.

Figura 14. **Mockup características de donativos II**



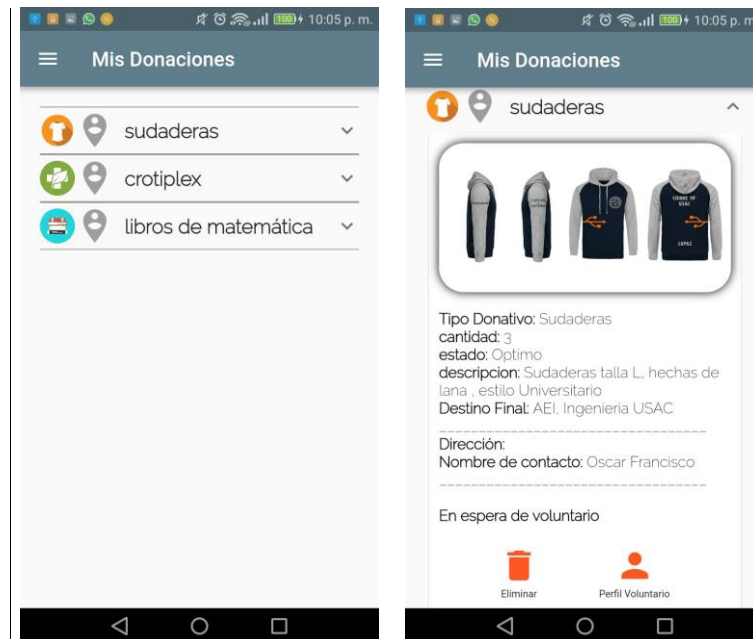
Fuente: elaboración propia.

3.2.8. Pantalla de lista de donativos

Tanto para el donante como para el voluntario aparecerá una lista de los donativos disponibles. Al comenzar se podrá apreciar dos iconos:

- Icono de donativo: mostrará a la categoría que pertenece.
- Icono de *tracking* de proceso: mostrará según el color en el que se encuentre el proceso en el que se encuentra el donativo.
 - Gris: donativo en espera de voluntario para su traslado
 - Amarillo: donativo confirmado por voluntario
 - Azul: donativo en posesión del voluntario
 - Verde: donativo entregado a la institución correspondiente

Figura 15. Mockup pantalla de lista de donativos



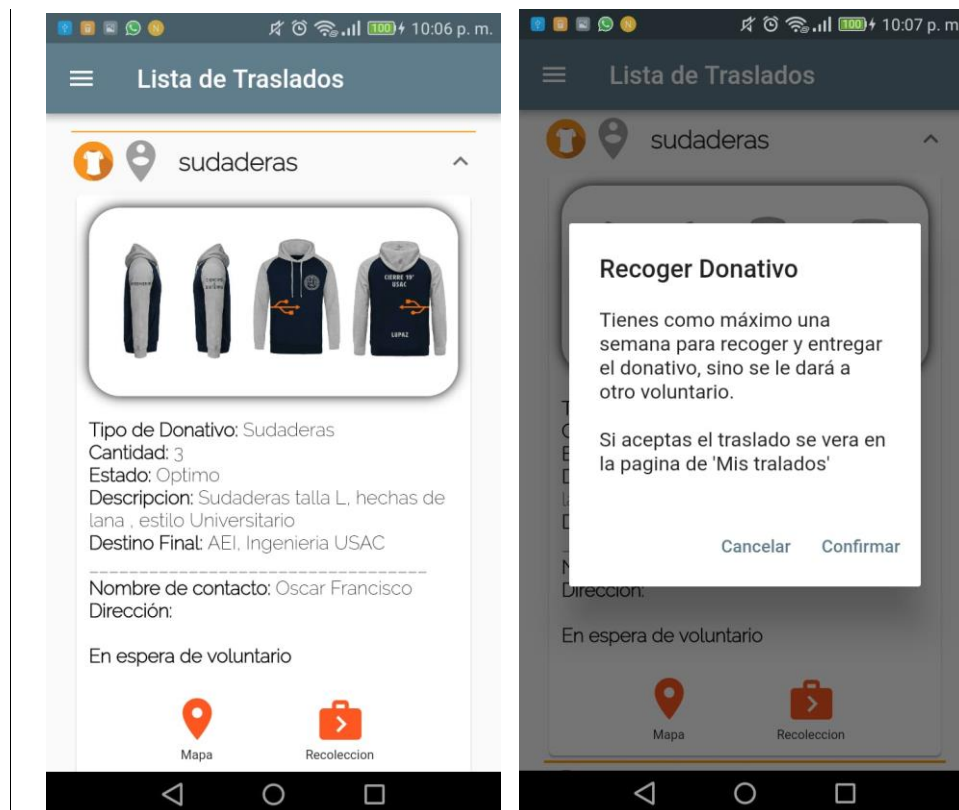
Fuente: elaboración propia.

3.2.9. Pantalla de lista de traslados

El voluntario será el que tome acción debido a que una vez pueda visualizar la lista de los donativos publicados, seleccionará el que le parezca más conveniente según sus criterios, posee dos opciones:

- Mapa: Nos mostrará a través de un mapa de Google, la ubicación exacta del donativo, si el donante habilitó la opción.
- Recolección: Iniciará el proceso de recolección del donativo, y mostrará una advertencia al voluntario en la cual se le indica el proceso que debe seguir.

Figura 16. Mockup pantalla de lista de traslados

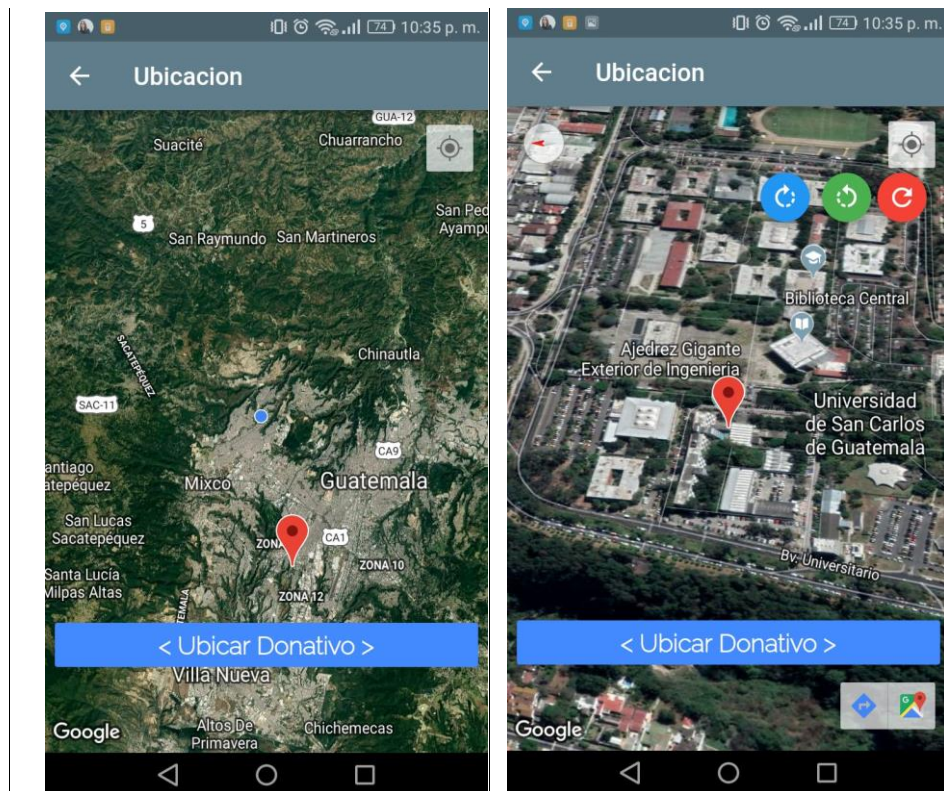


Fuente: elaboración propia.

3.2.10. Pantalla de ubicación

El voluntario podrá visualizar con suma precisión la ubicación del donativo, primero se le dará una vista general de Guatemala y una marca donde se encuentra el donativo, al presionar el botón “ubicar donativo” la aplicación hará un zoom hasta la posición del donativo, donde se podrá visualizar detalladamente el lugar de recolección.

Figura 17. Mockup ubicación y recolección



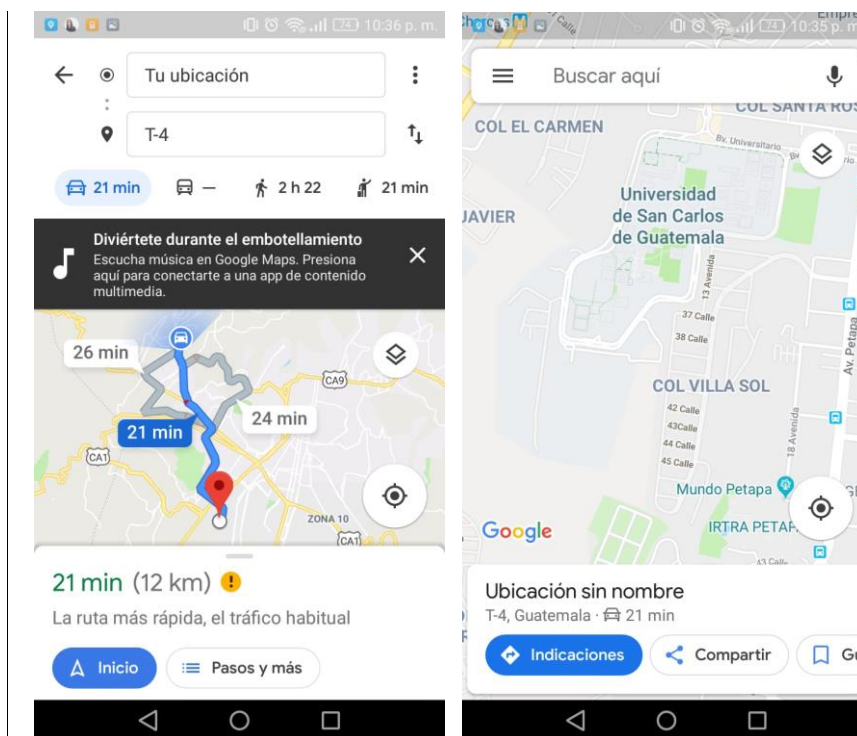
Fuente: elaboración propia.

3.2.11. Servicios de ruta y lugar

Una vez localizado el donativo en el mapa, la aplicación brinda dos opciones:

- Rutas: la cual brindará la opción de ubicar el donativo desde nuestra ubicación actual vía automóvil, además, indica el tiempo para llegar y la ruta que hay que seguir.
- Lugares: brinda información detallada sobre los lugares cercanos que se encuentran cerca del donativo, con el objetivo de conocer destinos atractivos en algún caso se desee visitar la zona en otra ocasión.

Figura 18. Mockup servicios de ruta y lugar



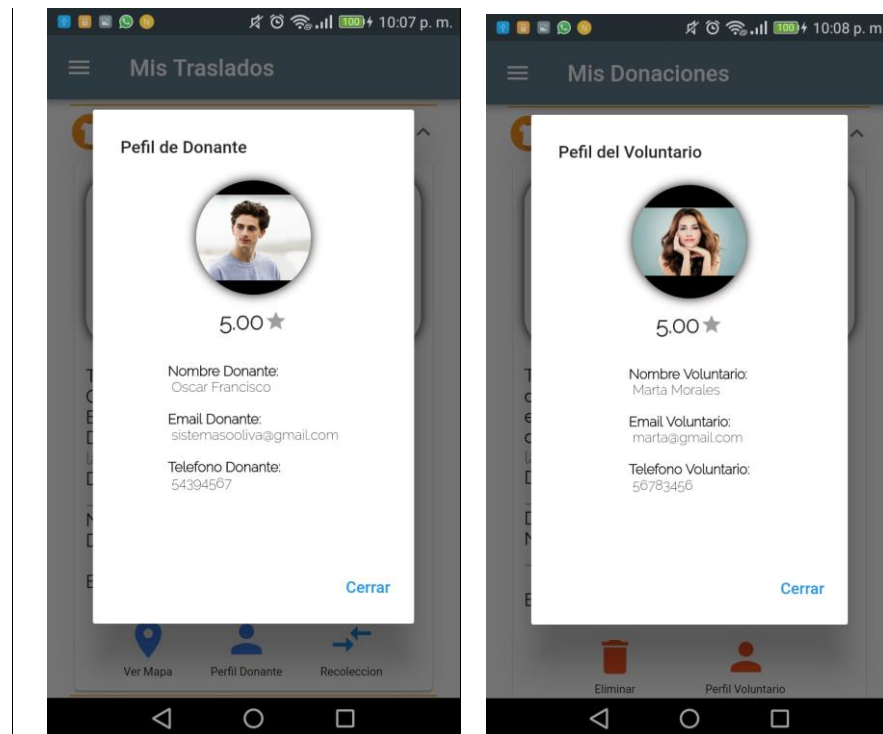
Fuente: elaboración propia.

3.2.12. Pantalla de visualización de perfiles

Una vez confirmado el inicio de proceso de recolección por parte del voluntario, tanto donante como voluntario podrán apreciar los datos personales del usuario con el que se han emparejado. Estos datos se mencionan a continuación:

- Foto de perfil
- Valoración
- Nombre
- Correo electrónico
- Teléfono

Figura 19. Mockup pantalla de visualización de perfiles



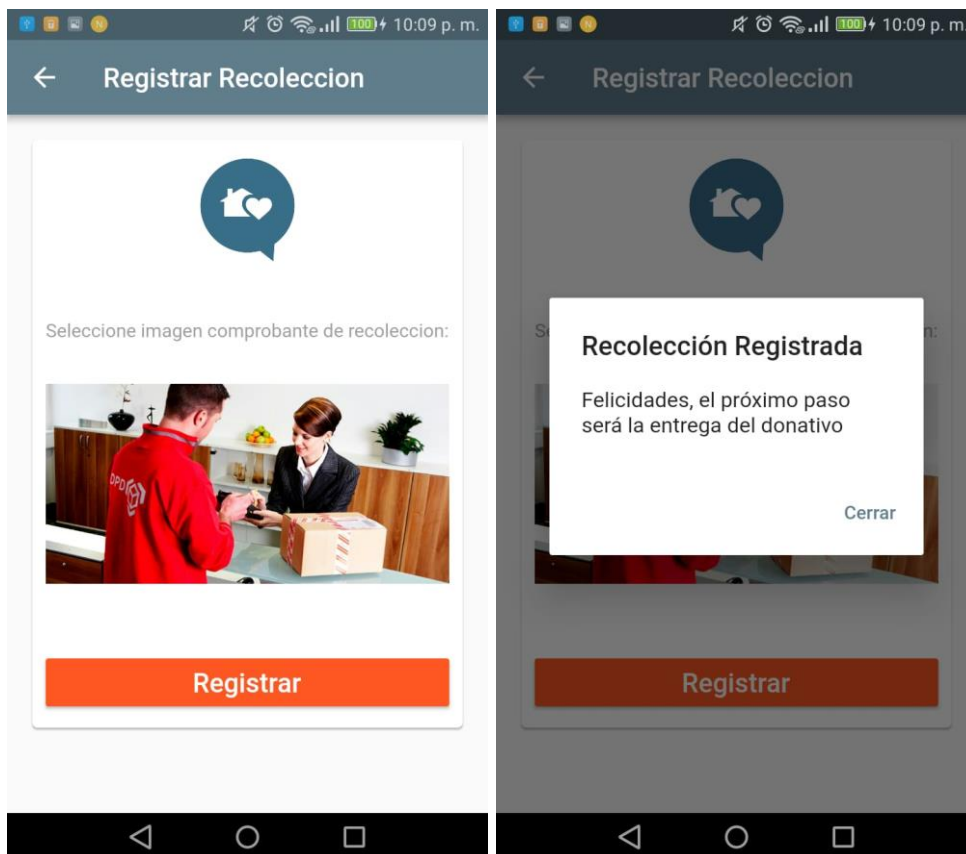
Fuente: elaboración propia.

3.2.13. Pantalla registro de recolección

En esta pantalla, se mostrará un formulario el cuál contendrá un registro de que el voluntario llegó a recoger el donativo hasta su ubicación. El formulario cuenta con los siguientes campos:

- Foto de comprobación: debe ser tomada en el momento de la recolección mostrando al donante y el donativo.

Figura 20. Mockup pantalla registro de recolección



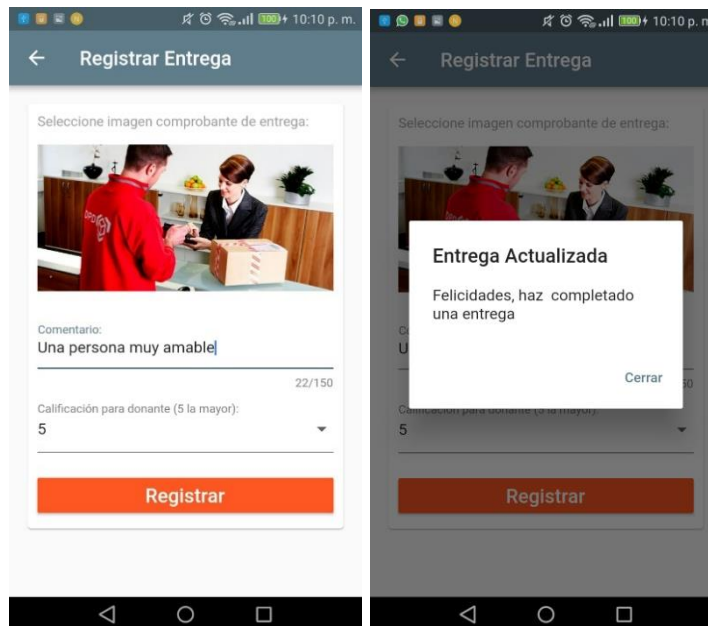
Fuente: elaboración propia.

3.2.14. Pantalla confirmación de entrega

En esta pantalla, se desplegará un formulario en cual servirá como constancia de que el voluntario entregó el donativo al lugar especificado.

- Foto de comprobación: debe ser tomada en el momento de la entrega, mostrando al voluntario y a una persona encargada del recibir el donativo.
- Comentario: comentario dedicado al donante, explicado si hubo alguna complicación o felicitación.
- Calificación: el voluntario podrá calificar de uno a cinco; es cinco la mejor puntuación y uno la peor.

Figura 21. Mockup confinación de entrega



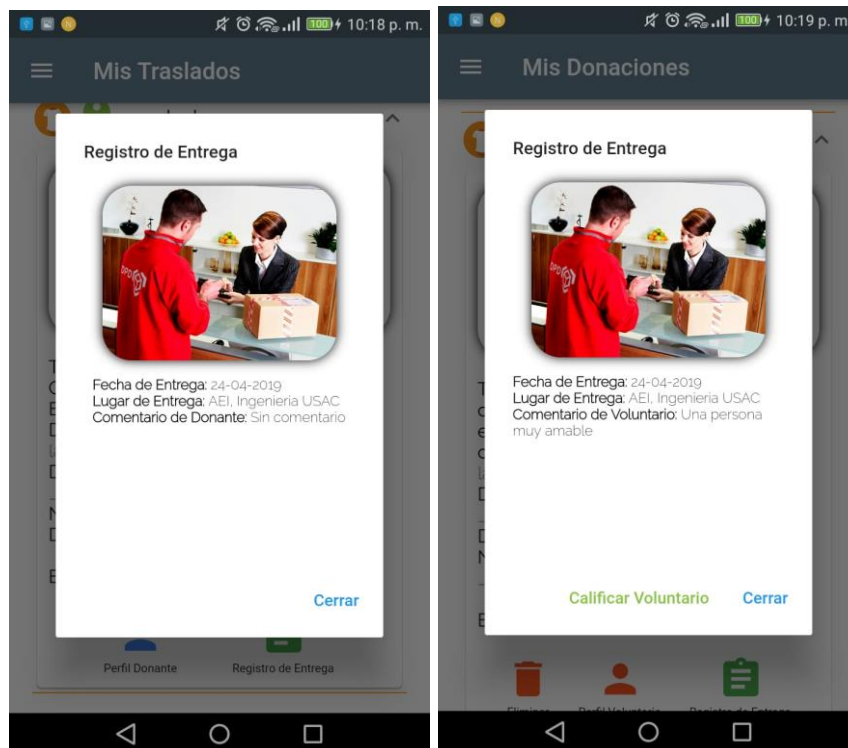
Fuente: elaboración propia.

3.2.15. Pantalla registro de entrega

Esta es la pantalla del último proceso, debido a que se muestra el registro de entrega del donativo, el que es el mismo tanto para donantes como para voluntarios. En él se muestran datos generales que se mencionan a continuación:

- Foto de comprobación: debe ser tomada en el momento de la entrega, mostrando al voluntario y a una persona encargada del recibir el donativo.
- Fecha de entrega.
- Lugar de destino.
- Comentario de donante/voluntario.

Figura 22. Mockup de registro de entrega



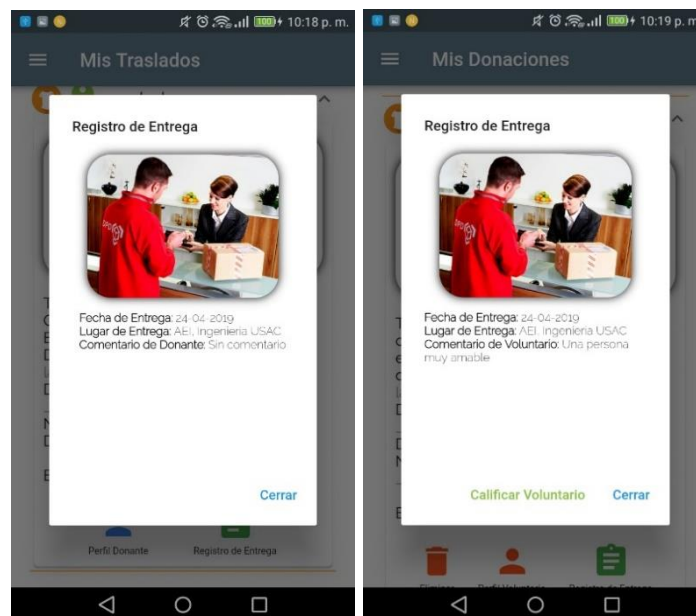
Fuente: elaboración propia.

3.2.16. Pantalla calificación del voluntario

Al igual que el voluntario, el donante puede calificar al voluntario al finalizar el proceso, en el registro de entrega se le desplegará la opción de calificar al voluntario y, posteriormente, un formulario en el que se le solicitan los siguientes campos al donante:

- Comentario: comentario sobre la interacción con el voluntario.
- Calificación: el donante podrá calificar según su conformidad en la interacción con el voluntario.

Figura 23. Mockup de calificación del voluntario



Fuente: elaboración propia.

3.3. Inversión

- Precio por subir aplicación a tienda virtual (Play Store): Q. 200,00

4. DOCUMENTACIÓN

4.1. Requisitos

Requisitos necesarios para el funcionamiento óptimo de la aplicación, en caso no se cumplan, no se garantiza el correcto funcionamiento de la aplicación.

4.1.1. Implementación de la aplicación desarrollada

- Android con Android 4.1 (nivel de API 16)
- Opciones de geolocalización habilitadas
- Opciones de acceso a multimedia habilitadas

4.2. Herramientas para el desarrollo de la aplicación

Herramientas utilizadas para diseñar, testear y desarrollar la aplicación móvil.

4.2.1. Visual Studio Code

Visual Studio Code es un editor de código gratuito, multiplataforma, súper rápido y liviano desarrollado por Microsoft para Windows, Linux y OS X.

4.2.2. Flutter SDK

Flutter SDK es un nuevo Framework para crear aplicaciones nativas tanto para Android como para iOS. Flutter utiliza código DART, es utilizado por

desarrolladores y organizaciones de todo el mundo, es gratuito y de código abierto. Todo indica que será una plataforma prometedora, ya que la facilidad de codificación es uno de sus pilares.

4.2.3. Plugin Flutter y Dart

Son *plugins* que deben ser agregados al editor de texto Visual Studio Code, Estos complementos le brindan finalización de código, resaltado de sintaxis, asistencia de edición de widgets, soporte de ejecución y depuración, y más.

4.2.4. Google USB Driver

Es un controlador que se utiliza para depurar dispositivos Google como el Nexus. Estos controladores son necesarios debido a que si no se instala y se quiere entrar en modo desarrollador a la aplicación lanzará una excepción.

4.2.5. Cloud Firestore

Cloud Firestore se compone de bases de datos de documentos NO SQL que permite sincronizar, almacenar, y consultar datos rápidamente para apps móviles y web a escala global, es donde se lleva todo el control y almacenamiento de los datos de YodonoGT. Además, permite sincronización en tiempo real.

4.2.6. Android SDK

El SDK (*software development kit*) de Android provee herramientas necesarias para el desarrollo de una aplicación basada en lenguaje Android, la cual será utilizada para un teléfono inteligente, por ende, es indispensable para el desarrollo de la aplicación.

4.3. Hardware

Flutter depende de que estas herramientas de línea de comandos estén disponibles en su entorno.

- Windows
 - Microsoft® Windows® 7/8/10 (32 o 64 bits)
 - 2 GB de RAM (8 GB de RAM recomendado)
 - 2 GB de espacio libre mínimo (4 GB recomendado)
 - Resolución mínima de 1,280 x 800
 - Java 8

- Mac
 - Mac® OS X® 10.8.5 o superior
 - 2 GB de RAM (8 GB de RAM recomendado)
 - 2 GB de espacio libre mínimo (4 GB recomendado)
 - Resolución mínima de 1,280 x 800
 - Java 6

- Linux
 - GNOME o KDE Desktop
 - Ubuntu® 16.04
 - 64 bits / 32 bits
 - GNU C (glibc) 2.1 o superior
 - 2 GB de RAM (8 GB de RAM recomendado)
 - 2 GB de espacio libre mínimo (4 GB recomendado)
 - Resolución mínima de 1,280 x 800
 - Java 8

Requisitos obtenidos de la página oficial de Developer Android y página oficial de documentación de Flutter.

4.4. Software

- Visual Studio Code
- Flutter SDK
- Android SDK
- Windows / Linux

4.5. Tutorial de desarrollo y referencias

Se presenta una lista de instrucciones para poder instalar y configurar las herramientas de desarrollo, es importante tener en cuenta que, debido a la constante actualización de componentes y librerías, se debe instalar la versión más reciente.

4.5.1. Instalación en Windows

Se presenta una lista de instrucciones para poder instalar y configurar las herramientas de desarrollo en un sistema operativo Windows.

4.5.1.1. Obtener SDK de Flutter

- Descargar el paquete de instalación para obtener la última versión estable del SDK de Flutter:

https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_v1.2.1-stable.zip.

- Extraer el archivo zip y colocar el contenido flutter en la ubicación de instalación deseada para el SDK de Flutter (por ejemplo C:\src\flutter, no instalar Flutter en un directorio C:\Program Files\ que requiera privilegios elevados).
- Localizar el archivo flutter_console.bat dentro del flutter directorio. Comienzar haciendo doble clic.

4.5.1.2. Actualizar path

- Desde la barra de búsqueda de Inicio, escribir 'env' y seleccionar Editar variables de entorno para su cuenta.
- En variables de usuario, comprobar si hay una entrada llamada "Ruta".
 - Si la entrada existe, agregar la ruta completa para flutter\bin usar ";" como separador de los valores existentes.
 - Si la entrada no existe, crear una nueva variable de usuario nombrada Path con la ruta completa flutter\bin como su valor.

4.5.1.3. Correr flutter doctor

Desde una ventana de consola que tiene el directorio Flutter en la ruta ejecutar el siguiente comando para ver si hay alguna dependencia de plataforma que necesite para completar la configuración:

Figura 24. **Flutter doctor**

```
C:\Users\oscar>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.2.1, on Microsoft Windows [Version 10.0.17763.437], locale es-GT)
[!] Android toolchain - develop for Android devices (Android SDK version 28.0.3)
    ! Some Android licenses not accepted.  To resolve this, run: flutter doctor --android-licenses
[✓] Android Studio (version 3.3)
[✓] VS Code (version 1.33.1)
[!] Connected device
    ! No devices available

! Doctor found issues in 2 categories.
```

Fuente: elaboración propia.

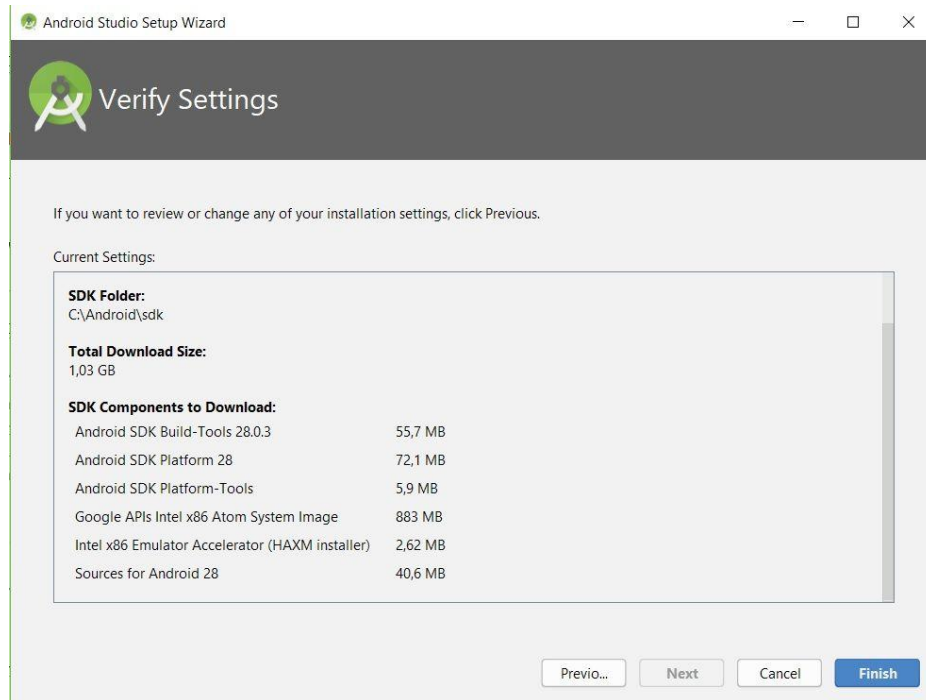
Este comando verifica el entorno y muestra un informe del estado de la instalación de Flutter.

4.5.1.4. Descargar e instalar Android Studio

Flutter necesita una instalación completa de Android Studio para resolver algunas de sus dependencias de la plataforma móvil Android.

- Descargar e instalar Android Studio:
<https://developer.android.com/studio/#downloads>.
- Iniciar Android Studio y seguir las instrucciones del asistente de configuración.

Figura 25. **Instalación Android Studio**



Fuente: elaboración propia.

4.5.1.5. **Configurar dispositivo Android**

Para ejecutar y probar aplicaciones Flutter en un dispositivo móvil Android, se necesitará un dispositivo Android 4.1 (nivel 16 de la API) o superior.

- En el dispositivo móvil, habilitar las opciones de desarrollo y la depuración USB. detalles de cómo configurar las opciones para programadores en el dispositivo: <https://developer.android.com/studio/debug/dev-options>.
- Para Windows: instalación del controlador USB de Google. <https://developer.android.com/studio/run/win-usb> en Android Studio desde el SDK Manager como se explica a continuación.

- Hacer clic en Tools > Android > SDK Manager.
 - Hacer clic en la pestaña SDK Tools.
 - Seleccionar Google USB Driver package y hacer clic en Install.
 - Una vez que finalice el proceso, los archivos del controlador se descargarán en el directorio: <sdk>\extras\google\usb_driver\.
-
- Utilizando un cable USB, conectar el teléfono al ordenador. Si lo solicita, autorizar al equipo a tener acceso al dispositivo.

 - En el terminal, ejecutar el comando Flutter devices para verificar que Flutter reconoce el dispositivo Android conectado.

4.5.1.6. Descargar e instalar VS Code

VS Code es un editor ligero con ejecución de aplicaciones Flutter y soporte de depuración: <https://code.visualstudio.com/>.

4.5.1.7. Instalar los complementos Flutter y Dart

- Inicio del código VS.
- Invocar Ver> Paleta de comandos.
- Escribir "instalar" y seleccione Extensiones: Instalar extensiones.
- Escribir "flutter" en el campo de búsqueda de extensiones, seleccionar Flutter en la lista y haga clic en Instalar. Esto también instala el complemento Dart requerido.

CONCLUSIONES

1. YodonoGt es un sistema descentralizado para la publicación, consulta y monitoreo de donativos en tres categorías: medicina, ropa y material didáctico en la ciudad de Guatemala, para instituciones sin fines de lucro.
2. Provee un medio en el cual las personas tienen la disposición de publicar sus donativos, con características básicas e información de contacto.
3. Desarrolla un medio en el cual las personas tienen la disposición de trasladar los donativos a los lugares destinados.
4. Implementa un sistema de búsqueda por preferencia para determinar la ubicación del donativo.
5. Implementa un módulo que permite monitorear el proceso en el cual se encuentra el donativo.
6. Publica los lugares a los cuales se entregarán los donativos.

RECOMENDACIONES

1. Divulgar YodonoGt en la comunidad universitaria para proliferar su uso.
2. Crear una opción que permita agregar instituciones sin fines de lucro para participar dentro de la aplicación.
3. Separar acceso de voluntarios y donantes, y crear dos aplicaciones distintas según su uso.

BIBLIOGRAFÍA

1. CASCÓN BAÑOS, Juan. *Flutter un nuevo framework para crear aplicaciones nativas*. [en línea]. <<https://www.frikipandi.com/tecnologia/20180228/flutter-nuevo-framework-crear-aplicaciones-nativas-tanto-android-ios-disponible/>>. [Consulta: abril de 2019].
2. CONTRERAS, Jorge. *Comisión Económica para América Latina y el Caribe*. [en línea]. <<https://www.cepal.org/es/comunicados/la-pobreza-america-latina-se-mantuvo-estable-2017-pero-aumento-la-pobreza-extrema>>. [Consulta: abril de 2019].
3. EAMES, Charles. *See 100 Quotes by Charles Eames, (Eames Office, 2007)*. [en línea] <<https://www.homeanddecor.com.sg/uncategorized/10-quotes-by-charles-and-rayeames/>>. [Consulta: marzo de 2019].
4. GRIFFITH, Chris. *Mobile App Development with Ionic, Revise Edition*. [en línea]. <<https://learning.oreilly.com/library/view/mobile-app-development/9781491998113/ch01.html>>. [Consulta: abril de 2019].
5. HERMES, Dan. *Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals*. [en línea] <<https://docs.microsoft.com/es/xamarin/get-started/>>. [Consulta: abril de 2019].

6. Infotelecom. *¿Qué es el cloud computing?* [en línea]. <<https://infotelecom.es/blog/que-es-el-cloud-computing/>>. [Consulta: abril de 2019].
7. LÓPEZ BELTRÁN, Gersón. *La importancia del dónde.* [en línea] <https://www.academia.edu/21820901/Geolocalizaci%C3%B3n_online_la_importancia_del_d%C3%B3nde>. [Consulta: abril de 2019].
8. MINNICK, Brandon. *Introducción a Xamarin.* [en línea]. <<https://docs.microsoft.com/es-es/xamarin/get-started/>>. [Consulta: abril de 2019].
9. PEÑA, Jorge. *¿QUÉ ES FIREBASE Y QUÉ NOS APORTA?* [en línea]. <<https://arpenstechnologies.com/es/blog/aplicaciones-movil/que-es-firebase-y-que-nos-aporta/>>. [Consulta: abril de 2019].
10. SANTANA, Pedro César. *Research on Computing Science 76 (2014)* [en línea]. <http://incytde.org/web/wpcontent/uploads/2015/03/RCS_VOL_76-1.pdf#page=46>. [Consulta: marzo de 2019].
11. SANTIAGO, Raúl. *Mobile learning: nuevas realidades en el aula.* [en línea] <https://www.researchgate.net/publication/299584978_Mobile_Learning_Nuevas_realidades_en_el_aula>. [Consulta: abril de 2019].