



Koffer

31/08/2019

```
if ( Nicky Kennedy Rufino Correa && Javier Ruiz Romera && Daniel Pando Moreno ) {  
    Boolean approved = true;  
    int note = 10;  
    approveWithHonors(approved, note) ;  
} return suspense;
```



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Índice

Resumen	4
Introducción	5
Usuario	5
Transportista	5
Contexto y justificación del Trabajo	5
Objetivos del Trabajo	5
Enfocamiento y método a seguir	6
Estrategia más apropiada	6
Planificación del proyecto	6
Breve resumen de productos obtenidos	6
Breve descripción de los capítulos de la memoria	7
Investigación	7
Público objetivo	7
Scrum	7
Trello	7
Java	7
Android Studio	7
Sprints	7
Conclusión	7
Anexo	7
Glosario	7
Bibliografía	7
Investigación	8
Público objetivo	9
SCRUM	10
Product Owner	10
ScrumMaster	10
Development team	10
Stakeholders	10
Trello	11
Java	12
Planificación	13
Sprints	14

Sprint 1	14
Objetivos	14
Reunión	14
Retrospectiva	14
Presentación	14
Sprint 2	15
Objetivos	15
Reunión	15
Retrospectiva	15
Presentación	15
Sprint 3	16
Objetivos	16
Reunión	16
Retrospectiva	16
Presentación	16
Sprint 4	17
Objetivos	17
Reunión	17
Retrospectiva	17
Presentación	17
Conclusión	18
Anexo	19
Funcionamiento	19
Servicios	19
Geolocalización de maletas	20
Diagramas	21
Diagrama de clases (funcionamiento general)	21
Diagrama de estados (funcionamiento maleta)	21
Diagrama de casos de uso (cliente y transportista)	22
Diseño	23
Portada	23
Home Usuario y Transportista	24
Servicio	25
Mapa	26
Más opciones Usuario y Transportista	27
Glosario	28
Bibliografía	30

Firestore	30
Android Studio/Java	30
SCRUM/Trello	30
Paypal	30
Links del proyecto GitHub/Trello	30

Resumen

Esta es una aplicación hecha en colaboración con CFGS Administració i Finances para la empresa ficticia creada, **KOFFER** y consiste en una app desarrollada en Android que permite al usuario solicitar un servicio de transporte de maletas en la cual la empresa recoge la maleta designada y la traslada a donde indique el usuario.

Donde el usuario puede ver donde están sus maletas mediante la geolocalización de la misma cuando está entrega a el destino designado. A su vez la app también sirve para que los transportista puedan hacer sus funciones designadas como aceptar encargos de maletas y entregarlas.



Introducción

El proyecto **Koffer** es una aplicación que se divide en 2 partes; una para el usuario y otra para el trabajo de la empresa el transportista.

Usuario

El **usuario** puede solicitar un servicio el cual pide que le recojan la/s maleta/s en un punto específico.

A la hora de solicitar el servicio, el usuario tendrá que rellenar un formulario con sus datos personales, para así cuando el transportista acepte este servicio pueda ver la información del cliente el cual ha solicitado dicho servicio. Una vez rellenada toda la información, se le permite al usuario hacer el pago con la plataforma **PayPal**

Transportista

El **transportista** puede aceptar las solicitudes de servicio de los *usuarios*, para así completar el encargo de recogida y/o entrega de maleta/s, mientras se registra las ubicaciones de las diferentes maletas, para mostrarle al usuario la ubicación en tiempo real de su maleta, y así saber en todo momento dónde se encuentra ubicada.

1. Contexto y justificación del Trabajo

Como ya hemos comentado antes, el proyecto nació fruto de un grupo de estudiantes de CFGS que necesitaban una aplicación para su empresa ficticia llamada Koffer donde necesitaban una serie de funcionalidades concretas acorde al enfoque de su empresa que se dedicaba al transporte de maletas.

2. Objetivos del Trabajo

Los objetivos de este proyecto consiste en saber utilizar la herramientas a nuestra disposición para la elaboración de una aplicación en Android donde podemos administrar las tareas y el tiempo de forma adecuada para la fecha límite impuesta por el cliente.

En este proyecto hemos aprendido a utilizar la métrica SCRUM y la herramienta Trello para poder llevarla a cabo. En la cual hemos aprendido a gestionar las tareas de un proyecto en un tiempo pequeño haciendo así que se trabaje de forma eficiente cada semana.

Además hemos aprendido como manejar con base de datos remota con Firebase, haciendo así que se pueda mostrar, modificar, crear y borrar datos o base de datos en línea y en tiempo real.

También hemos aprendido cómo manejarnos con un hipotético cliente futuro ya que hemos trabajado con otro ciclo obteniendo así experiencia para futuros proyectos en empresas o por nuestra cuenta.

3. Enfocamiento y método a seguir

Una de las posibles estrategias es reutilizar código de prácticas ya hechas anteriormente durante el curso, para así ahorrar tiempo a la hora de investigar sobre como usar o implementar algunas funcionalidades a nuestro proyecto.

Otra posible estrategia es dedicar mayor parte del tiempo obteniendo documentación en foros y viendo tutoriales en *Internet*, para que a la hora de trabajar tengamos una base de conocimientos bastante amplia sobre el tema del cual vamos a hacer el proyecto.

4. Estrategia más apropiada

La estrategia de dedicar mayor parte del tiempo a documentarse y ver tutoriales en *Internet* puede ser la más adecuada, ya que invertimos tiempo para después tener eficacia a la hora de trabajar e ir directos hacia el objetivo.

5. Planificación del proyecto

No hace falta explicar nada aquí, ya que hay un apartado más adelante dedicado a este tema en concreto.

6. Breve resumen de productos obtenidos

No es necesario entrar en detalle, la descripción detallada se hará en el resto de capítulos.

7. Breve descripción de los capítulos de la memoria

7.1. Investigación

Se menciona las tecnologías y métrica usadas en el proyecto.

7.2. Público objetivo

Se explica a quién va dirigida la app.

7.3. Scrum

Se explica porque hemos utilizado Scrum.

7.4. Trello

Se explica porque hemos utilizado Trello

7.5. Java

Se explica porque hemos utilizado Java para desarrollar la app.

7.6. Android Studio

Se explica porque hemos utilizado Android studio y no otro IDE.

7.7. Sprints

Se explica que hemos hecho en los sprints.

7.8. Conclusión

Se exponen las ideas y opiniones del proyecto.

7.9. Anexo

Se desarrolla en más profundidad el funcionamiento y diagramas.

7.10. Glosario

Palabras utilizadas en este documento.

7.11. Bibliografía

Toda la información extraída para el proyecto.

Investigación

A la hora de empezar el proyecto nos hemos tomado el tiempo necesario para poder investigar sobre las tecnologías y las métricas que utilizaremos durante el transcurso del mismo.

Al analizar la situación nos hemos decantamos por utilizar una serie de lenguajes y tecnologías en la cual estábamos familiarizados por la comodidad y su utilización anteriormente.

Decidido usar el lenguaje de programación **Java** para desarrollar en aplicación en **Android**, a su vez decidimos que para desarrollar en android utilizaríamos **Android Studio**.

Para preservar los datos de los usuarios dentro de la app y el funcionamiento de la misma decidimos utilizar **Firebase**, una herramienta que nos proporciona **Google** y que nos permite utilizar una serie de herramientas de forma gratuita y bastante completas como puede ser en nuestro caso; una **Base de datos** y asistencia de **autenticación de usuarios**.

A su vez decidimos que utilizaremos la métrica **SCRUM** para monitorizar y organizar nuestro proyecto pudiendo así comunicarnos con nuestros clientes para poder enseñar lo hecho. La herramienta seleccionada ha sido **Trello** ya que es una específica para trabajar con **SCRUM**.



Público objetivo

El principal público objetivo puede ir dirigido tanto para gente joven y mayor entre 17-90 años ya que es un aplicación de asistencia de maletas, es decir, ayuda a las personas a transportar sus maletas de forma rápida y segura.

Así que esta aplicación tiene un mercado muy amplio donde la aplicación podría ser utilizada por usuario que desean viajar y no preocuparse por su maleta. Ya que una vez contratan el servicio no se tienen que preocupar más de sus maletas hasta que el cliente llegue a su destino, pues la empresa se encargaría de toda la logística de transportar las maletas donde deseen.



SCRUM

I. Product Owner

El product Owner es el rol que se encarga de descubrir las necesidades del cliente y buscar soluciones a dichas necesidades. Para ello se utiliza el product backlog donde se almacenan toda la lista de tareas para cubrir las necesidades del cliente.

II. ScrumMaster

El rol de Scrum Master es el que se encarga de llevar a cabo toda la planificación del product backlog, sprint backlog, y de facilitar la planificación en cada inicio del sprint, donde hará que el Product Owner y el equipo de desarrollo puedan definir el objetivo del sprint a través de la selección de los elementos del product backlog.

III. Development team

El equipo de desarrollo es el que está compuesto por personal técnico que se encargan de llevar a cabo las tareas de cada sprint y que de manera conjunta desarrollan el producto del proyecto.

IV. Stakeholders

Los Stakeholders son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del "sprint".

Trello

Es un gestor de proyectos online que permite organizarse mejor las rutinas de trabajo, priorizar, generar avisos de citas y muchas otras opciones.

Cuando estábamos haciendo la investigación sobre cuál herramienta utilizar para gestionar SCRUM nos hemos topado con dos grandes herramientas que nos permiten hacer lo mismo pero con grandes diferencias en su funcionamiento, **Trello** y **Jira**.

Hemos elegido la herramienta **Trello** frente a Jira por dos fundamentos claros, uno de ellos es que ya habíamos trabajado con la misma un tiempo atrás y sabíamos sus funciones.

¿Porque hemos elegido **Trello** antes que **Jira**?

Aunque Jira es un herramienta muy potente frente a Trello en algunas situaciones Trello nos ofrece unas herramientas sencilla de usar y de aprender además de la sencillez de su interfaz y funcionamiento. Mientras que Jira puede llegar a tener más funciones su interfaz no es tan amigable con el usuario y tienes que tomarte un tiempo para acostumbrarte a ella.

En resumen, trello es un herramienta que conocemos y sencilla que nos permite administrar nuestro proyecto de forma sencilla y rápida para que nos podamos centrar en el proyecto.



Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

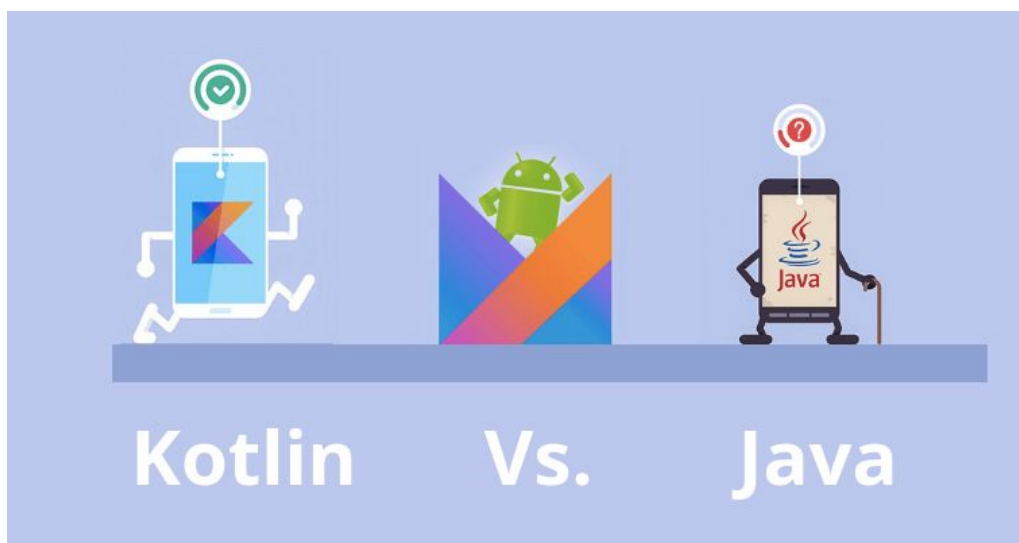
A la hora de elegir un lenguaje de programación para Android nos hemos topado con dos lenguajes principales, la vieja conocida **Java** y el nuevo y flamante **Kotlin**.

Hemos elegido el lenguaje de programación **Java** frente a **Kotlin** por dos fundamentos claros, uno de ellos es que ya habíamos trabajado con la misma un tiempo atrás y la otra es que no poseíamos el tiempo necesario para aprender un nuevo lenguaje de programación en un periodo tan corto de tiempo.

¿Porque hemos elegido **Java** antes que **Kotlin**?

Hemos decido **Java** por las razones mencionadas anteriormente haciendo así que la elección sea bastante obvia, haciendo que la decisión final se inclinara por hacer el proyecto en menos tiempo. También ya que es el lenguaje que principalmente hemos tocado durante todo el curso.

En resumen, hemos elegido **Java** aunque hoy en día no es el preferido para desarrollar en Android. Pero de todos modos sigue siendo el más usado hoy en día para desarrollar en Android.



Planificación

Antes de empezar a programar, nos hemos puesto dos semanas de plazo para hacer investigación sobre las herramientas que vamos a utilizar y en qué lenguaje y framework programar (edit: estas dos semanas fueron pocas horas, ya que hacíamos 1 o 2 horas semanales de proyecto). Una vez finalizada esta investigación, empezamos a documentarnos sobre lo que implementar en la aplicación, para así no tener dudas y/o problemas a la hora de programar y/o diseñar.

Mientras vamos avanzando con el proyecto, hacemos reuniones semanales con el grupo de administración, para así tener *feedback* con ellos, y cambiar cosas de la aplicación que no les parece bien o no les gusta y/o implementar cosas nuevas que quieren que tenga la aplicación.

En el proceso de desarrollo de la aplicación, seguimos haciendo investigación sobre métodos, librerías o diseño que queremos implementar. Las dos primeras semanas no dieron suficiente, como era de esperar.

Hemos optado por implementar la metodología *SCRUM* en nuestro proyecto, pero no al 100%, ya que no se adapta al completo a nuestras necesidades y/o incumbencias, por lo que está implementado de una manera no real, ya que utilizamos partes de la metodología que se podrían adaptar a nuestro proyecto, y las que no, las dejamos a un lado.



Sprints

Sprint 1

I. Objetivos

Los objetivos que nos pusimos para la primera semana eran principalmente tener cuentas creadas de correo, GitHub, Firebase, etc, y el proyecto creado y registrado en Firebase para trabajar con él. Además de esto, nos dio tiempo a crear un diseño minimalista de lo que iba a ser la aplicación.

II. Reunión

Al poner las cosas en común que habíamos hecho en este sprint, nos pusimos de acuerdo en un problema que teníamos, que era la seguridad de la base de datos, ya que estábamos almacenando la contraseña del usuario sin encriptar. Al final de la reunión quedamos en la solución de que *Firebase* podía hacer esto por nosotros, así que optamos por la opción de no almacenar la contraseña dentro de la base de datos y dejar que *Firebase* se encargue.

III. Retrospectiva

A la hora de crear el login y registro, nos dimos cuenta que no era buena idea almacenar datos que pudiesen comprometer la seguridad del usuario, como puede ser la contraseña. Por esta razón alargamos estas dos tareas al siguiente sprint, para poder solucionar este problema de seguridad.

IV. Presentación

En la primera semana nos reunimos con el grupo de administración para saber que querían exactamente en la aplicación. Nos explicaron cómo iba a ser su proyecto para así nosotros tener una idea de cómo empezar y hacia dónde llevar la aplicación.

Sprint 2

I. Objetivos

Los objetivos de este sprint han sido, crear las *activities* y/o *fragments* que necesita nuestra aplicación, además de diseñar la base de datos en tiempo real de *Firebase*, para más tarde no tener problemas de organización y/o redundancia de datos. Una vez hecha esta tarea, testeamos la base de datos, para ver si lo que habíamos diseñado estaba correcto.

II. Reunión

En la reunión de este sprint, fue fácil sacar los errores y/o problemas que habíamos tenido. Uno de ellos era el diseño de la aplicación, ya que los colores que nos habían pasado el grupo de administración no ayudaban a distinguir entre botones y campos de textos. Así que en la presentación hablamos de este problema y llegamos a un acuerdo de cambiar los colores.

III. Retrospectiva

El grupo de administración nos pasaron los colores que querían que tuviese la aplicación. Una vez puestos, enseguida nos dimos cuenta que no se veían bien, ya que eran unos colores apagados, oscuros, además de tener problemas para hacer que se viesen bien los botones, campos de texto, etc, así que dejamos este asunto para la reunión del siguiente sprint.

IV. Presentación

En la segunda semana tuvimos una pequeña reunión con el grupo de administración para enseñarles el diseño genérico que iba a tener la aplicación, para así, tener un poco de feedback con ellos. Pareció que les gustó.

Sprint 3

I. Objetivos

Una vez hecho el diseño de la aplicación y de la base de datos, debíamos ponernos con lo fuerte de la aplicación, que era almacenar los servicios creados por el usuario para el transporte de maletas en la base de datos, para que así el transportista pueda ver los servicios que están activos y poder aceptarlos. Además de esto, implementamos la librería de *PayPal*, para que el usuario pudiese pagar de alguna forma el servicio de maletas, y así nosotros simular tener beneficios por nuestro trabajo.

II. Reunión

Al tener la reunión y poner todo en común, teníamos un problema, y era que no éramos capaces de mostrar los pedidos activos del usuario en el mapa, para que más tarde pueda ver si su maleta ha sido entregada o está aún en camino.

III. Retrospectiva

En algunas *activities* y/o *fragments* a la hora de introducir datos, el usuario no era capaz de ver dichos, así que optamos por implementar un *ScrollView* en aquellas *activities* y/o *fragments* que tenían este pequeño problema.

IV. Presentación

En la tercera semana, al enseñar la aplicación al grupo de administración, le comentamos que el logo de la aplicación no tenía buena resolución, ya que sea veía un poco pixelado, pero no hubo suerte de que nos pasaran dicho logo con más resolución, así que lamentablemente se quedó ese logo. Por lo demás pareció que seguían contentos por el camino que estaba tomando la aplicación.

Sprint 4

I. Objetivos

En este último sprint, debíamos de terminar la aplicación para entregarla a tiempo al grupo de administración, además de hacer la memoria del proyecto. Los objetivos enfocados a la aplicación eran simples; mostrar en los marcadores del mapa si está el servicio entregado o no, modificar pequeños detalles del diseño de la aplicación y grabar un pequeño video tipo '*demo*' que el cliente nos comentó si podíamos hacerlo.

II. Reunión

En esta última reunión, al haber finalizado la aplicación, nos pusimos de acuerdo para comentar que habíamos hecho mal y el que bien. De esta manera nos sirve para nuestro próximo proyecto, no cometer los mismos errores.

III. Retrospectiva

Al haber grabado el video de la aplicación, el diseño final de la aplicación no estaba terminado, pues nos arrepentimos de haber entregado dicho video sin el diseño final, ya que por falta de tiempo, por parte del grupo de administración, no pudimos hacerlo.

IV. Presentación

Al hacer la presentación final de la aplicación, pareció que quedaron bastante satisfechos con el producto final. Pues la aplicación tenía las funciones que debía tener, y el diseño que al final optamos por entregar, gustó.

Conclusión

Como teníamos claro al principio de empezar, hemos aprendido y mejorado mucho más haciendo nuestro propio proyecto que durante todo el curso. El por qué es fácil de explicar.

Al ser un proyecto que nace la idea de nosotros mismos, cambia la visión del hecho de trabajar, le dedicas más tiempo, se hace con más mimo, si se encuentra un problema o sale un error, no importa buscarle solución, ya que uno mismo quiere que el producto final esté bien hecho.

Aunque al principio pueda dar miedo hacer un proyecto tan grande en *'colaboración'* con el grupo de CFGS de administración sin tener casi experiencia en el desarrollo de aplicación para android, acaba siendo una experiencia bastante buena.

El hecho de haber tenido la oportunidad de hacer una *'colaboración'* con el grupo de administración, nos ha servido para saber cómo sería trabajar con un cliente, el cual te dice como quiere el producto y para cuando lo quiere. Si es cierto que constantemente hemos estado preguntando pequeñas cosas al cliente; cosas que no han quedado claras, o que simplemente no se han hablado porque el cliente las daba por hechas, etc.

Aunque el desarrollo de la aplicación hemos tenido problemas técnicos y/o de conocimientos, lo hemos sabido superar buscando información y probando por nuestra cuenta tratando de ver una solución no solo fácil, sino adecuada.

Al mismo tiempo hemos obtenido experiencia a la hora de desarrollar una app en la que, simulamos estar en un empresa, ya que hemos tenido muchos altibajos en el desarrollo, por el hecho de que hemos sido nosotros los que hemos hecho todo desde cero, tanto el diseño, la lógica como el funcionamiento de la misma. Sin duda hemos aprendido tanto de forma profesional como individual.

Por últimas palabras, nos ha gustado hacer esta *'colaboración'* en nuestro proyecto, porque hemos sentido que nos hemos acercado a la realidad de hacer un trabajo con un cliente real, ya sea en una empresa o nuestra propia.

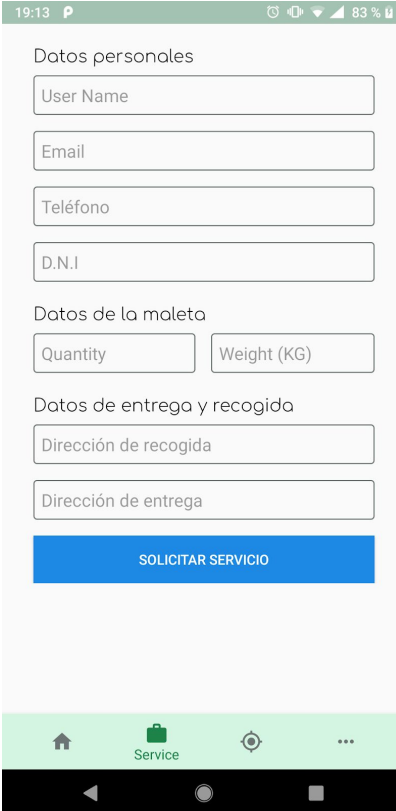
Anexo

Funcionamiento

1. Servicios

La funcionalidad principal de la aplicación es la de contratar el servicio, es el apartado donde el cliente puede realizar una solicitud para que la empresa realice un transporte de maletas desde un punto indicado a otro.

Además se ha introducido el método de pago vía PayPal y tarjeta de crédito para dar diferentes posibilidades de pago.



The screenshot shows a mobile application interface for requesting a service. The form is titled "SOLICITAR SERVICIO" and is divided into several sections:

- Datos personales:** Includes input fields for "User Name", "Email", "Teléfono", and "D.N.I".
- Datos de la maleta:** Includes input fields for "Quantity" and "Weight (KG)".
- Datos de entrega y recogida:** Includes input fields for "Dirección de recogida" and "Dirección de entrega".

A blue button labeled "SOLICITAR SERVICIO" is positioned below the form fields. The bottom navigation bar features a home icon, a "Service" icon with a suitcase, a location pin icon, and a menu icon. The status bar at the top shows the time as 19:13 and a battery level of 83%.

2. Geolocalización de maletas

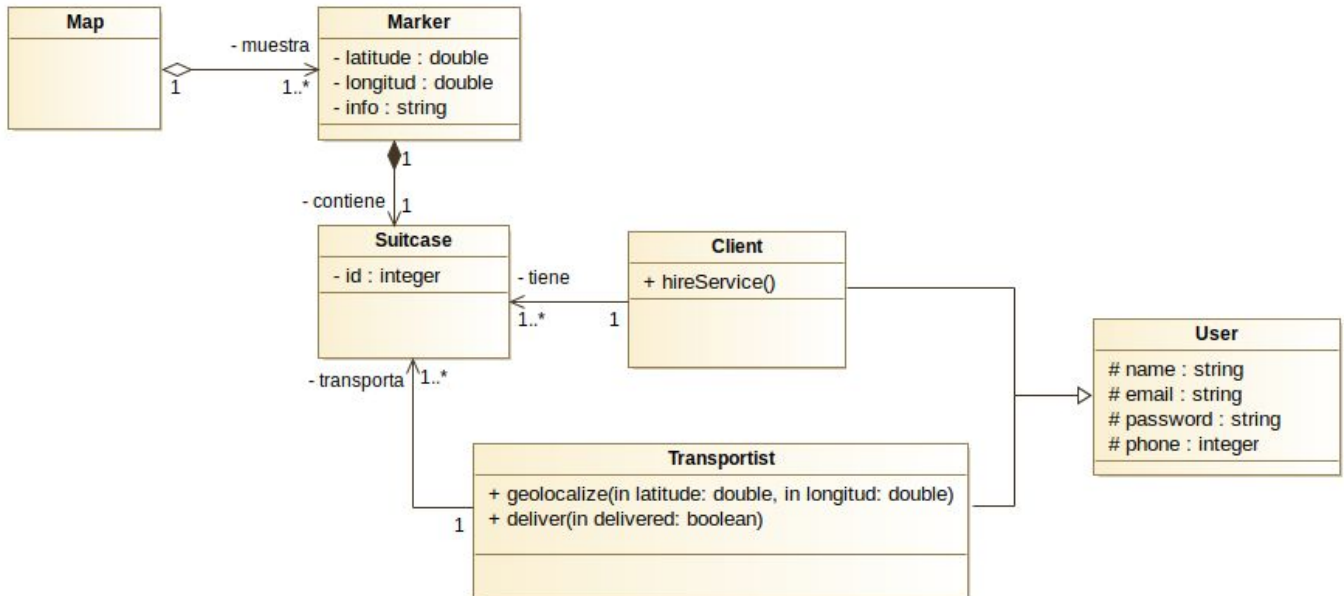
Una de las principales funcionalidades en la aplicación es la posibilidad de que el cliente pueda realizar un seguimiento de sus pedidos y poder ver su ubicación actual, de esta forma el cliente en todo momento estar mas tranquilo sabiendo que sus maletas están siendo repartidas y no están desaparecidas.

El funcionamiento es simple, cuando un transportista selecciona uno de los paquetes que tiene asignados y selecciona entregar, entonces la aplicación geolocaliza al transportista y sube su ubicación a Firebase y cambia el parámetro **delivered** a verdadero. De este modo cuando el paquete está entregado sale en el mapa del cliente.

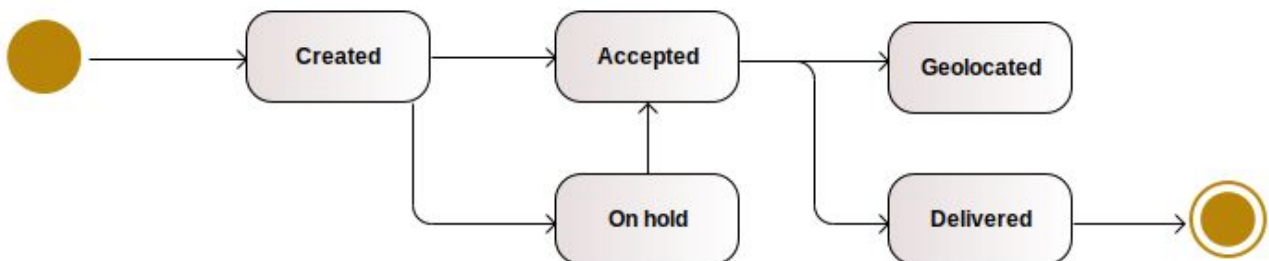


3. Diagramas

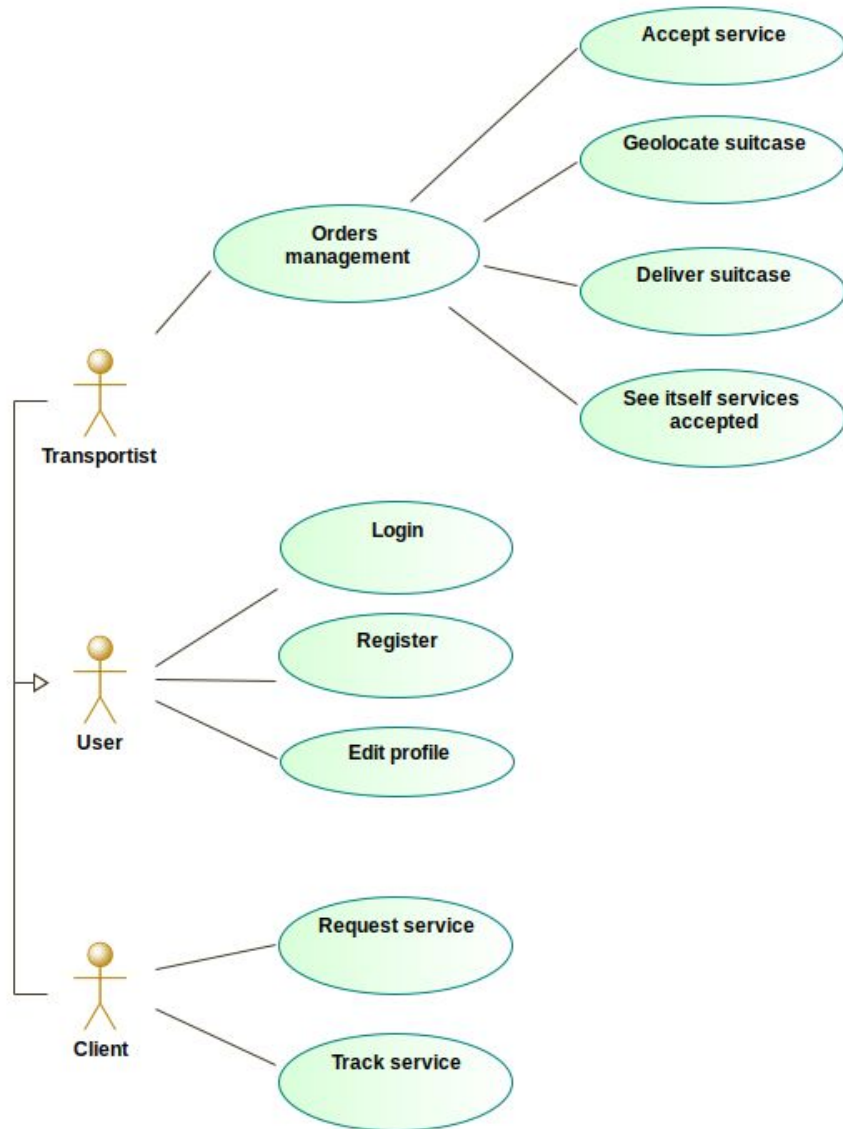
3.1. Diagrama de clases (funcionamiento general)



3.2. Diagrama de estados (funcionamiento maleta)



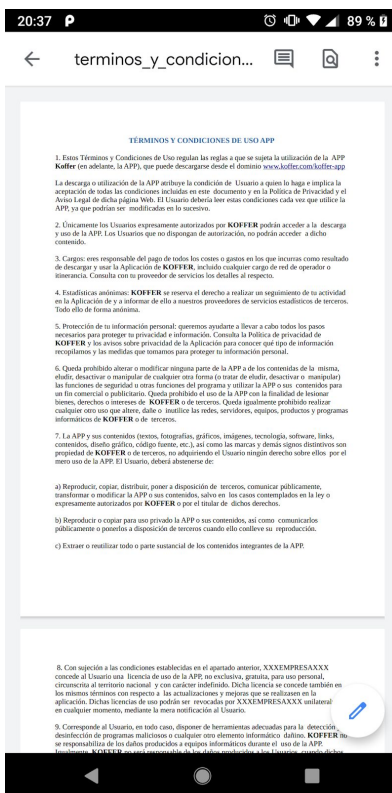
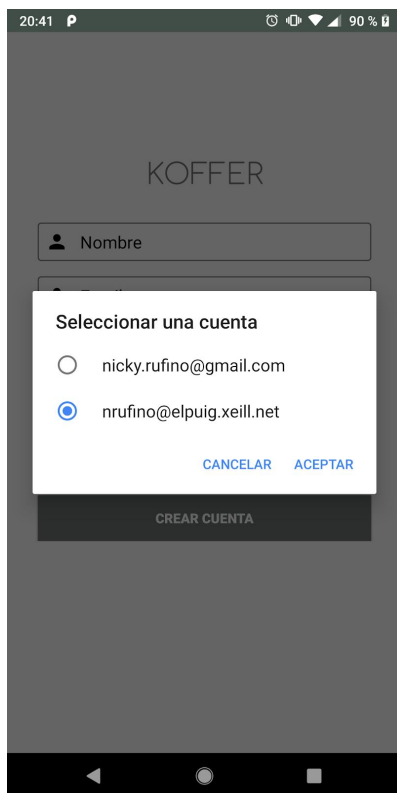
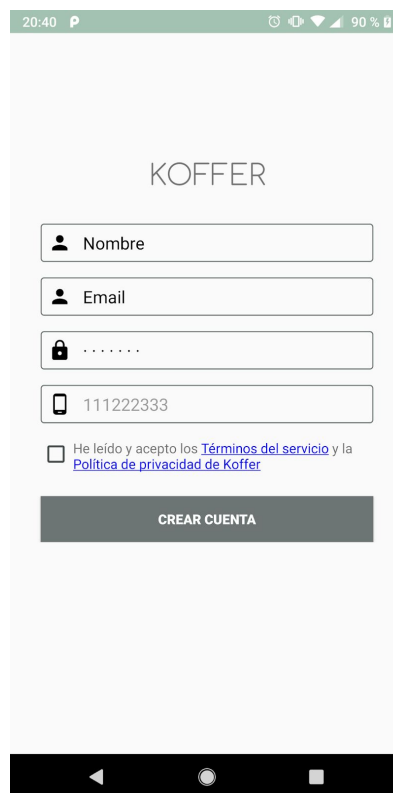
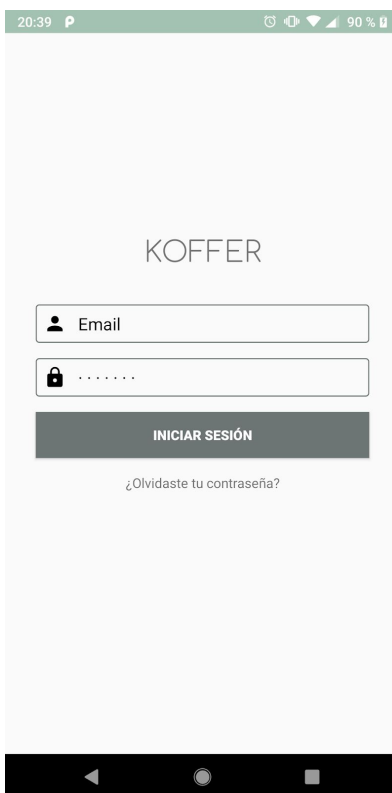
3.3. Diagrama de casos de uso (cliente y transportista)



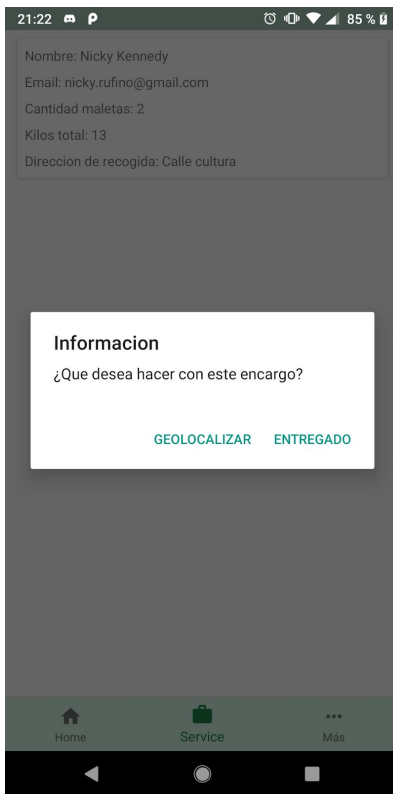
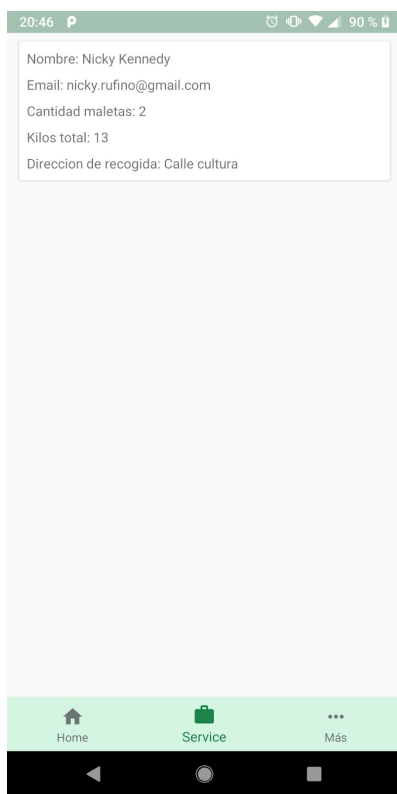
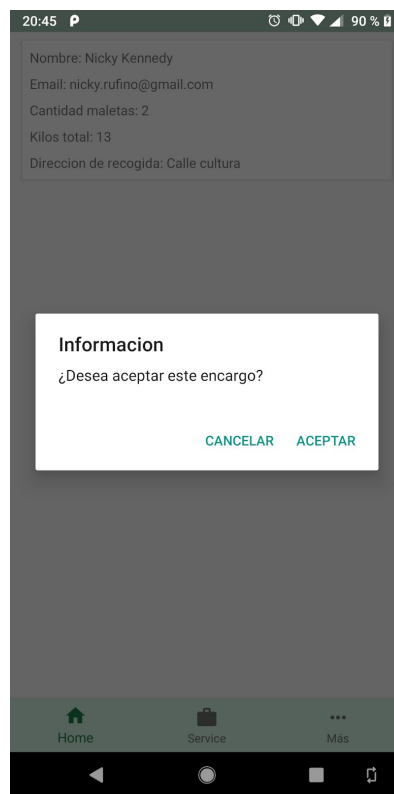
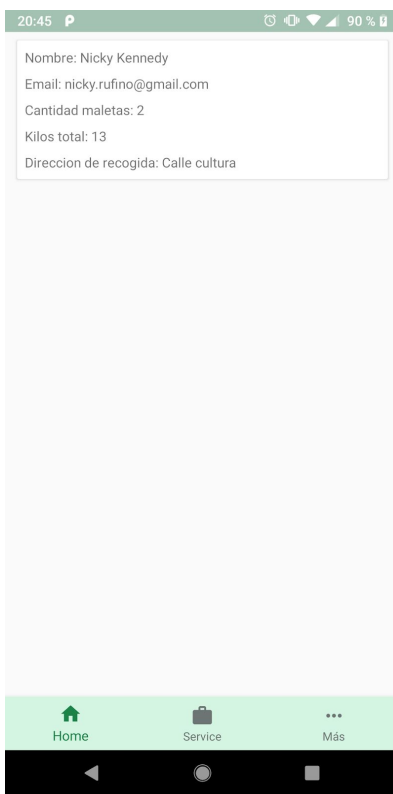
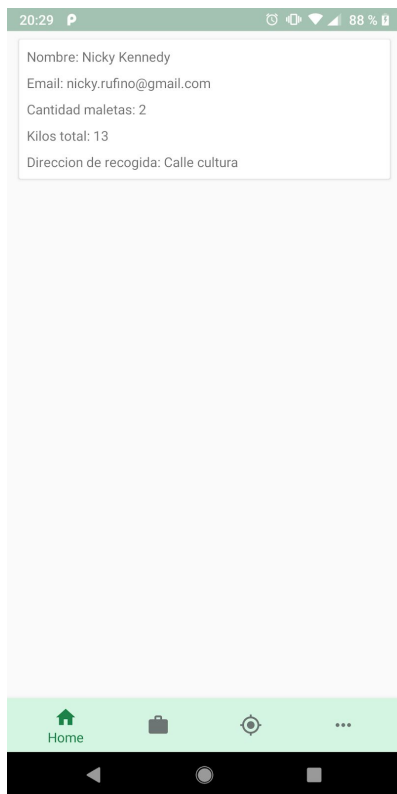
4.

5. Diseño

5.1. Portada



5.2. Home Usuario y Transportista



5.3. Servicio

20:20 P 📶 86%

Datos personales

Nicky Kennedy

nicky.rufino@gmail.com

123456789

X9853286V

Datos de la maleta

2 13

Datos de entrega y recogida

Calle cultura

El Prat

SOLICITAR SERVICIO

🏠 Service 📍 ⋮

20:20 P 📶 86%

Tu pedido

Pay to Koffer 10,00 €

Pagar con **PayPal**

Pagar con tarjeta

Entorno de pruebas

21:26 P 📶 86%

Iniciar sesión con PayPal

PayPal

personal@correo.com

.....

Iniciar sesión

¿Has olvidado la contraseña?

¿No tienes una cuenta PayPal? Registrarse

Entorno de pruebas

21:27 P 📶 86%

Confirmar

Pay to Koffer 10,00 €

PayPal

personal@correo.com

Cerrar sesión

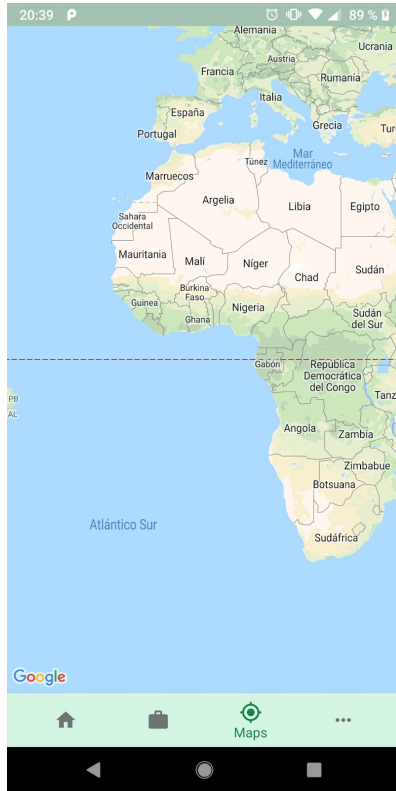
Forma de pago preferida

Saldo de PayPal >

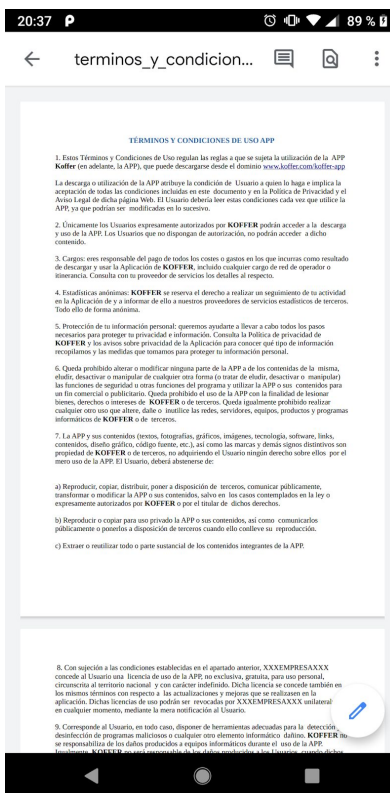
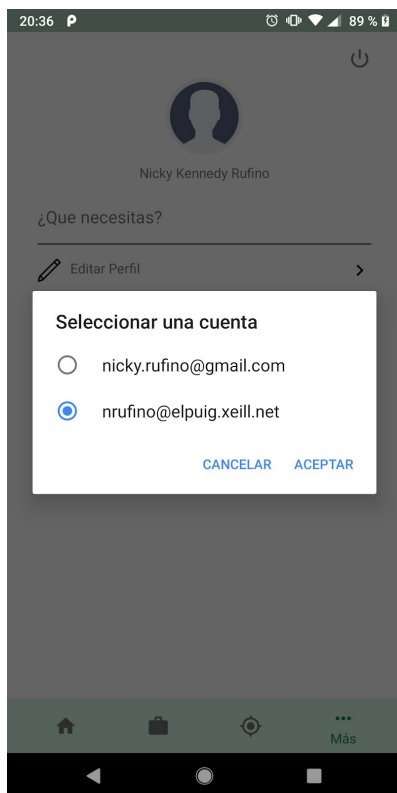
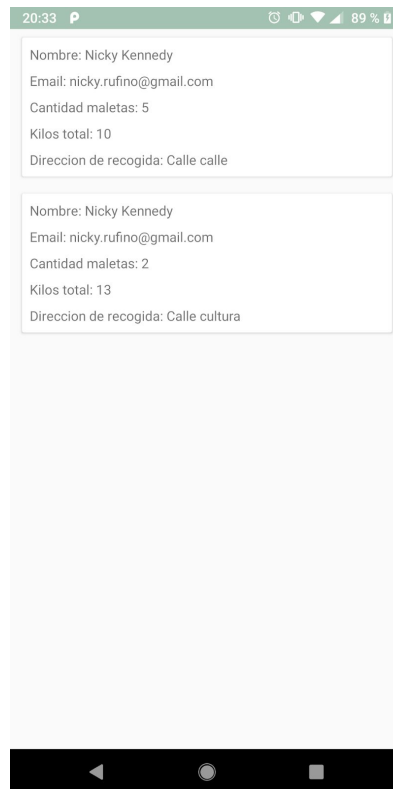
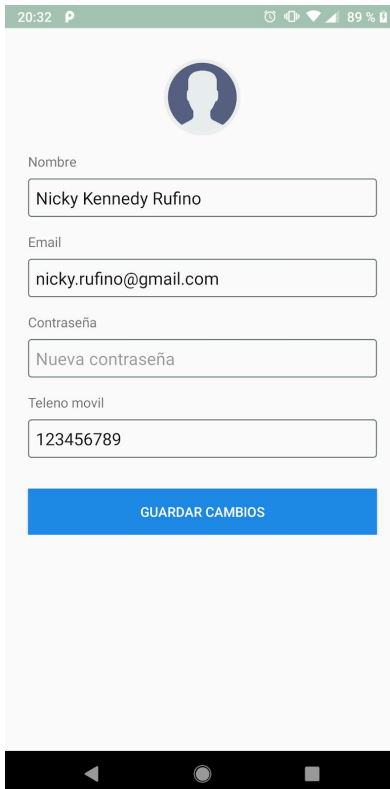
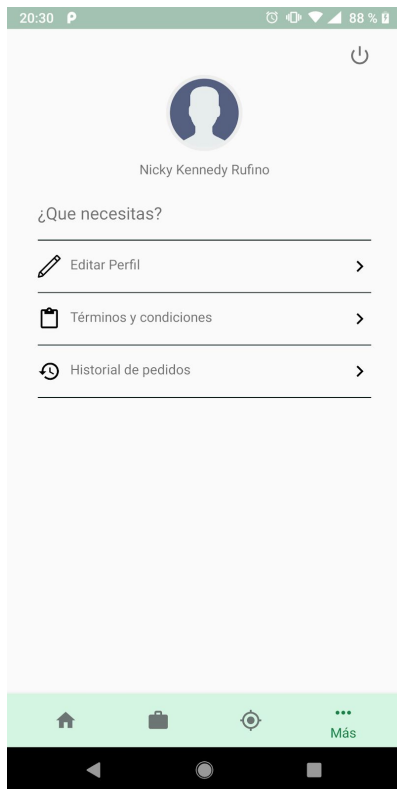
Pagar

Entorno de pruebas

5.4. Mapa



5.5. Más opciones Usuario y Transportista



Glosario

JAVA

Es un lenguaje de programación y una plataforma informática. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable.

Android Studio

Es el entorno de desarrollo integrado oficial para la plataforma *Android*. Fue anunciado en 2013 en la conferencia *Google I/O*, y reemplazó a *Eclipse* como el IDE oficial para el desarrollo de aplicaciones para Android.

JDK

Es un software que provee herramientas de desarrollo para la creación de programas en Java. Trae herramientas útiles como el compilador (javac), desensamblador de binarios (javap), debugger, entre otras.

Firestore

Es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles ubicadas en la nube.

Google Maps

Es un servidor de aplicaciones de mapas en la web que pertenece a Alphabet Inc. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle

Git

Es un software de control de versiones pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

GitHub

Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

Trello

Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y android para organizar proyectos.

SCRUM

Scrum es el nombre (no un acrónimo) con el que se denomina a los marcos de desarrollo ágiles. Permite flexibilizar las cargas de trabajo haciendo que el desarrollo sea incremental.

PayPal

Es una compañía que opera como un procesador de pagos para vendedores en línea, sitios de subastas y otros usuarios comerciales en línea e incluso comercios físicos.

XML

El eXtensible Markup Language es un metalenguaje que permite definir lenguajes de marcas utilizado para almacenar datos en forma legible. Este lenguaje para dar forma y diseño a la aplicación.

Activity

Es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción

Fragment

Representa un comportamiento o una parte de la interfaz de usuario en una Activity. Se puede combinar múltiples fragmentos en una sola actividad para crear una IU multipanel y volver a usar un fragmento en múltiples actividades.

Walkthrough

Es un tutorial que permite enseñar al usuario cuales y/o como utilizar las funcionalidades que tiene la aplicación.

Bibliografía

1. Firebase

<https://firebase.google.com/>

<https://firebase.google.com/docs>

[Youtube \(Tutoriales\)](#)

2. Android Studio/Java

<https://developer.android.com/studio>

<https://developer.android.com/docs>

[Youtube \(Tutoriales\)](#)

3. SCRUM/Trello

[SCRUM](#)

[SCRUM \(Wikipedia\)](#)

[Qué es SCRUM](#)

[Guía fundamental de SCRUM](#)

[Trello](#)

[Trello Help](#)

4. Paypal

[PayPal](#)

[PayPal \(docs\)](#)

5. Links del proyecto GitHub/Trello

[Koffer](#)

[Trello for peasants](#)

[Trello \(link only for V.I.P\)](#)