

Transpdroid l'app per a Transportistes

Carlos Bonache Casado

Grau d'Enginyeria Informàtica

Desenvolupament d'aplicacions per a dispositius (Android)

Joan Vicent Orenge Serisuelo

Carles Garrigues Olivella

10/01/2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Transpdroid l'app per a Transportistes</i>
Nom de l'autor:	<i>Carlos Bonache Casado</i>
Nom del consultor/a:	<i>Joan Vicent Orenga Serisuelo</i>
Nom del PRA:	<i>Carles Garrigues Olivella</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Desenvolupament d'aplicacions dispositius mòbils (Android)</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Aplicació, Android, logística</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>La finalitat del treball és fer tot el procés de disseny i desenvolupament d'una aplicació per al sistema operatiu Android, amb l'objectiu de facilitar el treball als repartidors de paquets com els d'Amazon, Correos, etc.</p> <p>Actualment els repartidors tenen moltes entregues i poc temps per realitzar-les, per això es vol crear una aplicació mòbil per optimitzar les entregues i que faciliti les funcions dels repartidors i que també millori l'experiència de cara a l'usuari final.</p> <p>Per tal d'assolir la totalitat del projecte s'utilitzarà el model de desenvolupament en cascada.</p> <p>El producte que s'obtindrà com a resultat d'aquest treball de fi de grau seran dues aplicacions. La primera aplicació serà la part Servidor que contindrà la informació i els serveis; i l'altra aplicació serà la part Client amb una interfície per al repartidor per poder utilitzar totes les funcionalitats i comunicar-se amb el servidor.</p>	

Abstract (in English, 250 words or less):

The purpose of the work is to make the entire process of designing and developing an application for the Android operating system, with the aim of facilitating the work of distributors of packages such as Amazon, Correos, etc.

Nowadays, the distributors have many deliveries in a short period of time to realize them. It's for that reason that I found necessary create a mobile application to optimize the deliveries and facilitate the functions of the distributors. This also improves the experience towards the end user.

In order to achieve the whole project, the cascade development model will be used.

The product that will be obtained as a result of this end-of-degree project will be two applications. The first application will be the Server part that will contain information and services;The second one will be the Client part with an interface for the distributor to be able to use all the functionalities and communicate with the server.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit	3
1.4 Planificació del Treball.....	4
1.4.1 Anàlisi de requeriments	4
1.4.1.1 Arquitectura <i>Android</i>	5
1.4.1.2 Arquitectura <i>Kotlin</i> . (Architecture in Kotlin)	8
1.4.2 Diagrama de Gantt	9
1.5 Breu sumari de productes obtinguts	9
1.6 Breu descripció dels altres capítols de la memòria.....	9
2. Disseny centrat en l'usuari	11
2.1 Usuaris i context d'ús	11
2.2 Disseny conceptual	12
2.2.1 Escenaris d'ús	12
2.3 Prototipatge	17
2.3.1 Sketches.....	17
2.3.2 Prototipus horitzontal d'alta fidelitat	18
2.3.3 Solucions de disseny proposades	19
2.4 Avaluació.....	20
3. Disseny tècnic de l'aplicació.....	22
3.1 Definició dels casos d'ús	22
3.2 Disseny de l'arquitectura	29
3.2.2 El diagrama UML corresponent al disseny de les entitats i classes.	30
3.2.3 Un diagrama explicatiu de l'arquitectura del sistema.....	31
4. Desenvolupament	32
4.1 Entorn de desenvolupament.....	32
4.2 Estructura aplicació Android ^[20] ^[21]	36
L'estructura es compon principalment per tres grans mòduls:.....	37
4.3 Estat actual del projecte	39
4.4 Proves	41

4.3.1 Proves d'integració.....	41
5. Conclusions.....	44
6. Glossari	46
7. Bibliografia.....	47
8. Annexos	48
8.1 Llistat de preguntes realitzades en les entrevistes	48
8.2 Resum de les conclusions de les entrevistes	48
8.3 Perfils d'usuaris	49
8.4 Escenaris d'us	50
8.5 Manual instal·lació servidor web.	52
8.6 Manual compilar codi Android i crear APK.	53
8.7 Manual instal·lar apk.	54
8.8 Manual Postman.....	55

Llista de figures

Il·lustració 1 - Cicle de vida app Android	7
Il·lustració 2 - Diagrama de Gantt	9
Il·lustració 3 - Flux d'interacció	16
Il·lustració 4 - Sketches	17
Il·lustració 5 - Inicia sessió	18
Il·lustració 6 - Detall d'un enviament	18
Il·lustració 7 - Llistat d'enviaments	18
Il·lustració 8 - Mapa del destí d'un l'enviament	18
Il·lustració 9 - Escanejar amb codi QR	19
Il·lustració 10 - Signatura	19
Il·lustració 11 - Signatura	19
Il·lustració 12 - Diagrama casos d'ús Repartidor	22
Il·lustració 13 - Diagrama casos d'ús Client Final	28
Il·lustració 14 - Diagrama casos d'ús Gestor aplicació web	28
Il·lustració 15 - Diagrama UML del disseny de la base de dades	29
Il·lustració 16 - El diagrama UML de les entitats i classes	30
Il·lustració 17 - Diagrama de l'arquitectura del sistema	31
Il·lustració 18 – Estructura projecte Android	36
Il·lustració 19 - Estructura aplicació Android (Kotlin) ^[21]	37
Il·lustració 20 - Model View Presenter (MVP) ^[21]	37
Il·lustració 21 – Domain Layer estructura ^[21]	38
Il·lustració 22 - Data Layer estructura ^[21]	38
Il·lustració 23 - Inicia sessió	39
Il·lustració 24 - Llistat de paquets	39
Il·lustració 25 - Detall d'un paquet	40
Il·lustració 26 - Signatura	40
Il·lustració 27 – Exemple petició permisos	40
Il·lustració 28 - Mapa del destí d'un	40
Il·lustració 29 - Escanejar amb codi QR	41
Il·lustració 30 - Flux d'interacció Frontent Web	50
Il·lustració 31 - Flux d'interacció Panell Administració	51

Il·lustració 32 – Android Studio	53
Il·lustració 33 – Android Studio carregar APK	54
Il·lustració 34 - Postman plugin de Chrome.	55
Il·lustració 35 - Importar col·lecció Postmant	55

1. Introducció

1.1 Context i justificació del Treball

Actualment, els repartidors de gairebé totes les companyies de transport tenen poc temps per realitzar i formalitzar els enviaments. És una necessitat latent, ja que aquest procés s'ha intentat millorar des de fa temps.

Antigament, a l'hora de realitzar els enviaments es realitzaven fent registres d'entrega en format paper i fins que el repartidor no entregava aquest registre i es registrava a la central no quedava realment registrat. Aquesta manera de treballar feia que l'experiència tant del repartidor com del client final fos bastant dolenta, tot i que ara i hi ha moltes més solucions algunes companyies de transport segueixen treballant així.

Després algunes empreses varen començar a utilitzar PDA per solucionar aquests problemes però les PDA's tenien un preu molt elevat i no totes podien comprar aquestes eines per a tots els treballadors.

Finalment, les mateixes empreses que es podien permetre utilitzar les PDA's varen crear cada una aplicació mòbil molt senzilla per validar els lliuraments i tenir a l'usuari final sempre informat de l'estat del seu enviament.

És per totes aquestes causes que hi ha la necessitat d'incorporar una solució per a totes les companyies de transport, siguin petites o grans amb un producte que estigui a l'abast de tots.

Aquest TFG consisteix a analitzar i desenvolupar una aplicació que pugui millorar tant les necessitats del repartidor com l'experiència del client final. L'aplicació ha de facilitar i cobrir les necessitats del repartidor i ha de millorar l'experiència de l'usuari final a l'hora de rebre l'enviament.

La necessitat que pretén cobrir aquest projecte és reduir costos i riscos per part de les empreses de transport. A més, es planteja una millora del servei als usuaris amb el coneixement de l'estat dels seus enviaments. Això inclou, unes millores de comunicació entre client final i l'empresa de transport, reduint així els problemes que puguin realitzar-se.

Per tant, el resultat que es vol obtenir en aquest projecte és el desenvolupament d'un prototip d'una aplicació mòbil que serveixi per a totes les companyies destinades al sector de logística, millora'n l'experiència de l'usuari final i que a més a més li doni més facilitats al repartidor amb informació dels enviaments i el seu destí.

1.2 Objectius del Treball

Com ja s'ha esmentat, l'objectiu general del treball és l'obtenció d'un prototip d'una aplicació mòbil per la plataforma Android, destinada al sector de logística que permetrà la gestió i seguiment dels enviaments.

A més a més, això ens proporciona l'oportunitat de treballar sobre un projecte des de les fases inicials fins a l'entrega final d'un producte, donant una experiència que no s'obté només amb la teoria.

Com a objectius principals específics:

- Atenen als requisits i a les necessitats dissenyar i desenvolupar una aplicació específica per dispositius mòbils per cobrir les necessitats del repartidor i l'usuari final que sigui escalable en un futur.
- Adquirir coneixements necessaris sobre la plataforma Android i el seu desenvolupament d'aplicacions.
- Adquirir coneixements necessaris sobre l'arquitectura *Kotlin*.
- Desenvolupar de la manera més òptima un prototip del producte.
- Realitzar les proves pel correcte funcionament d'aquest producte.
- Treure les conclusions del cicle del projecte i si cal plantejar el següent.
- Posar en pràctica els coneixements adquirits durant la carrera.

Per complir els objectius serà necessari el següent abast del projecte:

- Una part servidor.
 - Un mòdul *CRUD* d'enviaments.
 - Un mòdul *CRUD* d'estats d'enviament.
 - Un mòdul *CRUD* d'usuaris/repartidors.
 - *API REST OAuth2*.
 - Mètode d'autenticació.
 - Llistat d'enviaments.
 - Informació dels enviaments.
 - Mitjançant ID.
 - Mitjançant el codi QR (Quick Response).
 - Firmar com rebut un enviament.
 - Donar d'ubicació del destí d'un enviament.
 - Una pàgina *Frontend* perquè el client final pugui comprovar l'estat del seu enviament.
 - Enviament de correu per cada canvi d'estat.
- Una part client Android
 - Arquitectura neta Kotlin
 - Connexió via API REST al servidor
 - Pantalles necessàries:
 - Inicia sessió.
 - Llistat d'enviaments.
 - Detall d'un enviament.
 - Mapa de la ubicació del destí d'un enviament.
 - Signatura.
 - Escanejar amb codi QR

1.3 Enfocament i mètode seguit

Aquest TFG té un enfocament principalment acadèmic i es tracta d'una aplicació prototip. Tot i així, aquesta aplicació conté la possibilitat de poder adaptar-se en un escenari real. Les dades que oferirà seran per poder testejar l'aplicació i en cap cas podrien ser reals, així que farem una simulació del que podria ser un escenari real. Per aquesta raó, la següent memòria quedarà merament en l'àmbit acadèmic i en cap cas es portarà a l'àmbit real.

La metodologia de treball a seguir consisteix a analitzar el problema actual i buscar una solució modular i escalable amb la finalitat d'aconseguir un prototip ampliable a futures implementacions.

A causa de la falta d'experiència en el desenvolupament Android en *Kotlin*, s'implementarà de forma progressiva per poder comprovar en cada pantalla que l'aplicació funciona i anar afegint les diverses pantalles amb les seves funcionalitats una a una, per poder aïllar errors i poder provar de manera ordenada i esglaonada.

1.4 Planificació del Treball

1.4.1 Anàlisi de requeriments

Els recursos necessaris per a la realització del projecte són els següents:

1. Servidor Web.

Es requereix un servidor Web per muntar tota la part del servidor. Aquest servidor ha estat cedit per *3&Punt Solucions Informàtiques*^[1] que és l'empresa on actualment treballa.

Aquest servidor està configurat per suportar la versió 5.4 del *framework* Laravel^[2] amb els següents components:

- Sistema Operatiu Ubuntu^[3] 14.04.3 LTS
- Servidor Apache/2.4.7
- Servidor MySQL 5.5.46
- PHP 5.6.31-2
 - Aquest PHP té les extensions requerides per *Laravel*^[2].
 - OpenSSL PHP Extension
 - PDO PHP Extension
 - Mbstring PHP Extension
 - Tokenizer PHP Extension
 - XML PHP Extension

2. Part Client.

Es requereix per aquest desenvolupament una arquitectura neta de *Kotlin*. En aquest cas utilitzarem la estructura facilitada en un GitHub de *HelmMobile*^[4].

3. Un ordinador per desenvolupar les aplicacions. En aquest cas, tindrè disponible el meu ordinador personal i a vegades el del meu lloc de treball que per motius de seguretat, serà necessari per pujar els canvis al servidor.

Els dos ordinadors tenen un '*Dual-Boot*' amb el sistema Operatiu W10 i Centos^[5] 6 amb entorn gràfic Gnome.

4. IDE Netbeans^[6] 8.1 (PHP) per desenvolupar la part Servidor.

5. IDE Android Studio^[7] 2.3.3 per desenvolupar la part Client.

- Fare us dels plugins:
 - com.android.application
 - kotlin-android
 - kotlin-android-extensions

6. Per realitzar les proves del funcionament de l'enviament de correu seran amb *mailtrap.io*

7. Per realitzar les proves del funcionament de la API s'utilitzarà el plugin de Google Chrome Postman.

8. Per realitzar les proves utilitzaré un dispositiu mòbil per testejar l'aplicació, no és 100% necessari per realitzar aquest projecte, ja que hi ha emuladors que ho fan.

Les característiques del mòbil son:

- Marca: Samsung.
- Modelo: Galaxy J5 (2016)
- SO: Android 6.0.1

Mas informació en la pàgina oficial de Samsung:
<http://www.samsung.com/es/smartphones/galaxy-j5-j500fn/SM-J500FZDAPHE/>

1.4.1.1 Arquitectura *Android*.

Els components principals del sistema operatiu Android són els següents:

- **Aplicacions:** En Android totes les aplicacions estan escrites en llenguatge de programació Java.
- **Marc de treball d'aplicacions:** Els desenvolupadors d'Android tenen accés complet a les *APIs* del *Framework* que s'utilitzen per crear les aplicacions. L'arquitectura està dissenyada per simplificar la reutilització de components i així qualsevol aplicació pot publicar les seves funcionalitats i qualsevol altra aplicació pot després fer ús d'aquestes funcionalitats.
- **Biblioteques:** Android inclou un conjunt de biblioteques de C/C++ usades per diversos components del sistema. Aquestes característiques s'exposen als desenvolupadors a través del marc de treball d'aplicacions d'Android, algunes d'aquestes biblioteques són: Biblioteques de gràfics, System C library, SQLite, entre d'altres.
- **Runtime Android:** Android inclou un set de biblioteques base que proporcionen la major part de les funcions disponibles a les biblioteques base del llenguatge Java. Android corre el seu propi procés, amb la seva pròpia instància de la màquina virtual Dalvik. La Màquina Virtual està basada en registres i corre classes compilades pel compilador de Java. Des de la versió 5.0 utilitza el ART, que compila totalment al moment d'instal·lació de l'aplicació.
- **Nucli Linux:** Android depèn de Linux per als serveis base del sistema com seguretat, gestió de memòria, gestió de processos, pila de xarxa i model de controladors. El nucli també actua com una capa d'abstracció entre el maquinari i la resta de la pila de programari.

Anatomia d'una aplicació Android

Dins d'una aplicació Android hi ha quatre components principals: *Activities*, *Listeners*, *Services* i *Content Providers*. Totes les aplicacions en Android estan formades per alguns d'aquests elements o combinacions d'ells.

- **Activity (Activitat)**

Els *Activities* són l'element constituent més comú en Android. Per implementar una Activitat s'utilitza una classe per cada que s'estén de la classe base *Activity*. Cada classe mostrarà una interfície d'usuari, composta per visites. Cada vegada que es canviï de Vista, es canviarà l'Activitat. Per passar de vista en vista, Android utilitza una classe especial anomenada *Intent*.

- **Intent**

Intent és un objecte del tipus missatge que descriu què vol fer una aplicació. Les dues parts més importants d'un *Intent* són l'acció que es vol realitzar i la informació necessària que es proporciona per poder realitzar l'acció, la qual s'expressa en format URI.

- **Listeners**

Els *Listeners* s'utilitzen per reaccionar a esdeveniments externs. Els *listeners* no tenen *UI* i per llançar un avís no cal que l'aplicació s'estigui executant, en cas necessari, Android la iniciarà automàticament si s'activa el *listeners* per algun esdeveniment.

- **Services**

Un *Service*, o Servei, és bàsicament un codi que s'executa durant molt de temps i sense necessitat d'UI, com pot ser un gestor de descàrrega, en el qual s'indiquen els continguts a descarregar i posteriorment l'usuari pot accedir a una nova vista sense que el gestor s'interrompi. En cas que hi hagi múltiples serveis alhora, se'ls pot indicar diferents prioritats segons les necessitats.

- **Content Providers**

En Android, les aplicacions poden guardar la seva informació en fitxers, BBDD SQLite, etc., però en cas que el que es vulgui sigui compartir aquesta informació amb altres aplicacions, el necessari és un *Content Provider*. Un *Content Provider* és una classe que implementa un conjunt estàndard de mètodes que permet a altres aplicacions guardar i obtenir la informació que maneja dit *Content Provider*.

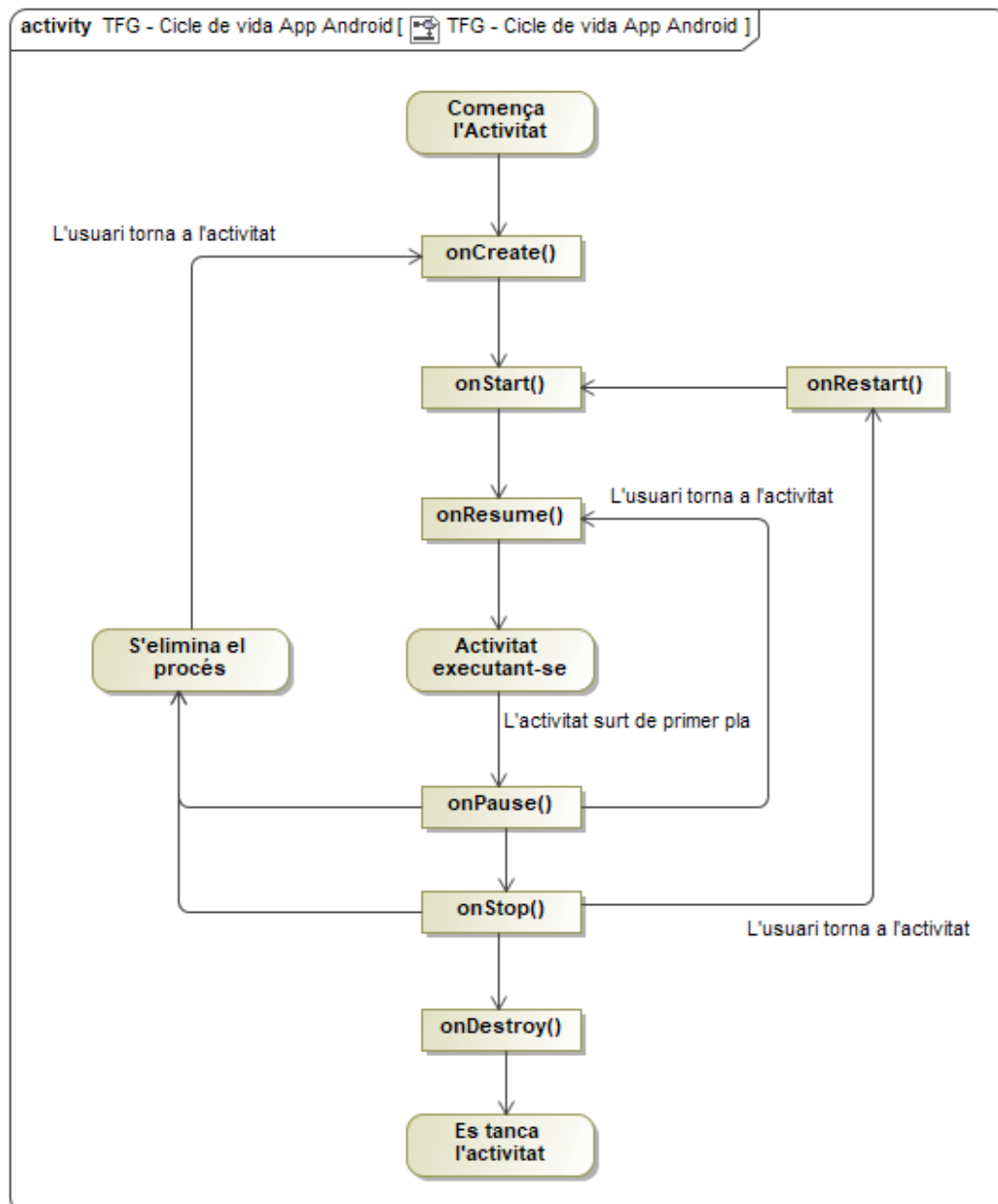
- **Android Manifest**

En Android hi ha un fitxer XML anomenat *AndroidManifest* que, encara que no formi part del codi principal de l'aplicació, és necessari per al seu correcte funcionament. Aquest fitxer és el fitxer de control que li diu al sistema què ha de fer amb tots els components anteriorment esmentats en aquest apartat que pertanyen a una aplicació en concret.

Cicle de vida d'una aplicació Android^[16]

En Android cada aplicació corre en el seu propi procés, el qual és creat per l'aplicació quan s'executa, i roman fins que l'aplicació deixa de treballar o el sistema necessita memòria per a altres aplicacions. Una característica fonamental d'Android és que el cicle de vida d'una aplicació no està controlat per la mateixa aplicació sinó que ho determina el sistema a partir d'una combinació d'estats com poden ser quines aplicacions estan funcionant, quina prioritat tenen per a l'usuari i quanta memòria queda

Per comprendre millor el cicle de vida d'una aplicació Android, en la següent figura es mostra el diagrama de flux del cicle, indicant els mètodes que es duen a terme.



II-lustració 1 - Cicle de vida app Android

Diagrama creat amb MagicDraw Personal Edition 18.5

1.4.1.2 Arquitectura *Kotlin*. (Architecture in Kotlin)

Kotlin és un llenguatge de programació madur i que conté un plugin per l'IDE Android Studio que funciona molt bé.

Les raons per les quals s'ha escollit aquest llenguatge són:

- És un llenguatge que està basat en Java.
- Utilitza el mateix compilador.
- Fa poc que Google l'ha convertit en un llenguatge oficial, per tant es una tecnologia nova que pot aportar nous coneixements.
- Dóna més seguretat en les variable null.
- Dóna facilitats a l'hora de programar.
- Aporta més llibreries per facilitar el seu ús.

Altres raons varen ser donades per companys de feina i els caps de l'empresa que se'ls va realitzar una sèrie de preguntes per poder plantejar el millor possible la part Android del projecte. Una de les solucions va ser aquesta, un llenguatge que amb Android havia agafat prou força anomenat Kotlin.

Vaig investigar^[12] i vaig veure que dóna bastants facilitats a l'hora de programar en Android i que estigues totalment integrat al IDE Android Studio és un punt molt a favor per utilitzar-lo.

Seguidament hem van recomanar utilitzar l'estructura pujada al GitHub per part de l'empresa Helm Mobile Development^[13] col·laborador directe amb la meua empresa 3ipunt. Aquesta arquitectura està molt bé per perquè conté un exemple per poder seguir i una base per poder començar l'aplicació Android.

Hi ha altres alternatives per programar en Android, però aquesta era la que més factible es veia, ja que ha sortit fa poc com a llenguatge oficial i els del més escollit per als desenvolupadors Android i es contempla l'opció de posicionar-se com una de les millors maneres de programar en Android.

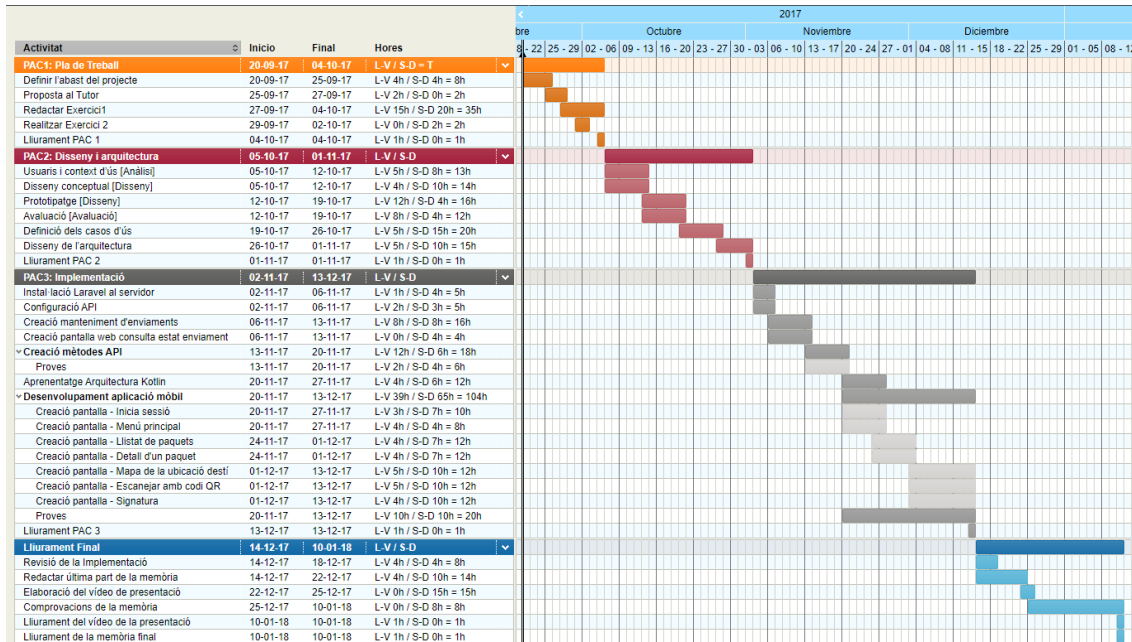
1.4.1.3 Framework Laravel.

Laravel és un Framework PHP amb una base (core) Symfony, això vol dir que conté totes les funcionalitats de Symfony però la capa de programació ha sigut canviada per la de Laravel. Les raons per les quals he escollit Laravel per fer aquest projecte són:

- És programari lliure i conté molta documentació.
- Programar directament en Symfony és una mica més costos, ja que gairebé ho has de fer tu tot i en canvi Laravel et dóna una facilitat a l'hora de programar que s'agraeix molt, per exemple a l'hora de crear i mapejar la base de dades amb els models.
- És un framework MVC.
- Té molts *bundles* que pots instal·lar. Un *bundle* és un mòdul que et dóna un servei. Per exemple en aquest projecte s'utilitzarà el bundle passport^[14] per poder realitzar la part de la API.
- L'experiència que puc aportar amb aquest framework és bastant alta i això facilitarà que pugui dedicar-me molt més a desenvolupar l'aplicació mòbil.

1.4.2 Diagrama de Gantt

El següent diagrama de Gantt mostra les tasques i subtasques necessàries per completar el TFG. Les tasques han sigut programades partint de la base de les entregues programades definides al pla docent.



II-lustració 2 - Diagrama de Gantt

El diagrama de Gantt^[7] l'he creat *online* mitjançant *tomsplanner.es*.

1.5 Breu sumari de productes obtinguts

Els productes obtinguts al llarg del cicle de vida del projecte seran els següents:

- Part Servidor, una aplicació web amb una *API REST Oaut2* en *PHP* amb el *Framework Laravel*^[2], que contindrà la informació i els serveis necessaris, per obtenir les dades de l'aplicació mòbil. També s'obindrà un panell administratiu amb mòduls CRUD per poder generar el registres a la base de dades i una pantalla perquè l'usuari final pugui obtenir informació de l'enviament.
- Part Client, una aplicació mòbil en Android amb l'arquitectura Kotlin que utilitzarà el repartidor per poder realitzar el seu treball de manera més eficaç.
- Memòria del projecte.
- Presentació virtual.

1.6 Breu descripció dels altres capítols de la memòria

En els següents capítols d'aquesta memòria s'aniran descrivint les tasques realitzades per a l'obtenció dels objectius plantejats en l'apartat 1.2, seguint com a base la vida del producte tal com explica l'assignatura "Gestió de projectes". Així que l'estructura dels següents capítols seria:

Capítol 2: Disseny centrat en l'usuari. En aquest capítol es descriuen l'anàlisi del problema per tal de trobar la millor solució. A partir d'això es defineixen els objectius del treball.

Capítol 3: Disseny tècnic de l'aplicació. Aquí agafant com a base l'anàlisi es comencen a explicar les decisions preses respecte a la mateixa aplicació, funcionalitats, arquitectura, Disseny gràfic i aspecte visual de la interfície.

Capítol 4: Desenvolupament. Capítol que explica com s'ha dut a terme el desenvolupament de l'aplicació i les seves proves.

Capítol 5: Demostració. Es mostren les pantalles reals que componen la interfície de l'aplicació, tant del Servidor com del Client.

Capítol 6: Conclusions. S'exposen les conclusions finals al resultat del producte creat, les possibles ampliacions i millores possibles.

2. Disseny centrat en l'usuari

2.1 Usuaris i context d'ús

Tal com s'ha descrit anteriorment l'aplicació d'aquest projecte està centrat en l'usuari repartidor, per tant s'utilitzarà el mètode d'indagació "Entrevistes en profunditat".

S'utilitzarà aquest mètode en aquest projecte perquè així s'aconsegueix diferents punts de vista amb requeriments variats segons el repartidor entrevistat. Podria haver escollit el mètode d'indagació "Dinàmiques de grup" o "Enquestes", però en la primera opció es podria perdre punts de vista dels participants i l'altra opció donava més informació estadística i es vol obtenir les necessitats del projecte d'una manera més concreta no fent estadístiques.

Per portar a terme les entrevistes es planteja una sèrie de preguntes amb temes com la forma actual de treballar de cada repartidor i els seus problemes o necessitats que possiblement no estan cobertes actualment o que es poden optimitzar. Així que mitjançant un seguit d'entrevistes s'aconseguirà uns resultats per cada repartidor entrevistat que donaran una visió dels requeriments del projecte.

Els resultats obtinguts dels diferents entrevistats no contindran molta diferència entre ells, però si podrien estar enfocats de diverses maneres, ja que poden contenir diferents solucions o formes de veure el mateix problema. És per això que la conclusió que s'obtindrà dels resultats serà un agrupament de les necessitats més importants les quals es transformaran en objectius principals del projecte i aquelles altres es mirarà d'analitzar-los i tractar d'esbrinar com fer perquè sigui útil per al conjunt de repartidors.

El llistat de les preguntes estan en l'annex 8.1.

I els resultats obtinguts estan en l'annex 8.2.

Perfils d'usuari identificats

L'aplicació mòbil que es vol desenvolupar està dirigida a un perfil d'usuari que treballa en el sector logístic com a repartidor. Així que el contingut de la metodologia escollida l'enfocarem molt cap aquest perfil i ens centrarem en el model d'usuari del repartidor i les seves necessitats.

Repartidor

Al ser un perfil d'usuari d'un treballador les seves característiques del perfil poden ser molt diverses:

- **Característiques demogràfiques:** Normalment a prop del lloc de treball.
- **Interessos:** Poder realitzar tots els enviaments en la jornada de treball.
- **Motivacions:** Pot tenir diverses motivacions, ja que depenen de l'usuari
- **Experiència amb l'ús de la tecnologia mòbil:** Pot tenir diverses experiències, ja que depenen del usuari i aquest perfil de treball conté un ampli rang d'edat i estudis.

- **Contexts d'ús:** Faran ús de l'aplicació durant la seva jornada laboral.
- **Anàlisi de tasques:** Les tasques a realitzar per assolir els seus objectius seran:
 - Iniciar sessió
 - Visualitzar els enviaments
 - Veure el detall de l'enviament
 - Veure el destí de l'enviament
 - Fer que l'usuari que recull l'enviament signi.
 - Fer que el repartidor en cas d'una incidència pugui trucar al client final.
 - Fer que el repartidor pugui reportar incidències.
- **Llistat de característiques:**
 - Accés amb usuari i contrasenya a l'aplicació.
 - Un llistat dels enviaments que porta en aquest moment el repartidor. Els enviaments entregats s'identificaran d'alguna manera per poder diferenciar-los amb els que encara no han sigut entregats.
 - Un lector de codi QR per facilitar la identificació de l'enviament.
 - Un mapa per veure el destí de l'enviament.
 - Un lloc on pugui registrar la persona que recull l'enviament.
 - Una breu informació dels detalls de l'enviament.
 - Un lloc on informar les incidències. Els enviaments poden tenir incidències, com que no hi hagi ningú per recollir-lo o que la direcció fos errònia.
 - Un lloc que en algun cas justificat es pugui trucar al receptor de l'enviament.

Per raons de disseny, aquest projecte té més perfils d'usuari però aquests no seran usuaris de l'aplicació mòbil, només el receptor de l'enviament signarà que ha rebut l'enviament però no realitzarà cap acció dintre de l'aplicació.

Per aquesta raó hem creat un annex amb una fitxa per cadascun dels diferents perfils d'usuari identificats que no faran ús de l'aplicació mòbil però si faran ús d'altres serveis del projecte.

2.2 Disseny conceptual

2.2.1 Escenaris d'ús

Com que l'objectiu d'aquest treball és el desenvolupament d'una app que sigui una eina per al repartidor l'únic usuari que hi ha és el repartidor i no fa falta distingir entre diferents tipus de repartidors, ja que tots desenvolupen el mateix tipus de treball. Potser els repartidors han de enviar el paquet en un temps determinat o porten diferents tipus de mercaderies, però tots aquest tenen en comú les funcionalitats del projecte.

Repartidor

L'usuari repartidor farà ús d'aquesta aplicació per desensenyar el seu treball. Amb aquesta eina podrà veure els enviaments que té en el seu vehicle, el detall d'aquest enviament amb dades com el pes total, les mides i quants paquets hi ha. També per si no sap on està el destí de l'enviament, podrà visualitzar

l'adreça en el mapa. A més és responsabilitat seva fer que la persona a qui entrega l'enviament li faciliti el nom i cognoms o el DNI i una firma abans de poder donar-li el paquet o paquets d'aquest enviament i en cas que la direcció fos errònia o que no hi hagués ningú per recollir el paquet ho podrà notificar immediatament.

Per l'altra part, per raons de disseny aquest projecte té dos escenaris d'ús que són complementaris a l'aplicació mòbil, però no realitzarà cap acció dintre de l'aplicació. Per aquesta raó hem creat un annex amb cadascun dels escenaris d'ús de cada perfil d'usuari.

Escenaris d'ús

Perfil	Repartidor.
Context	Comença el seu horari laboral.
Objectius	Entrar dins l'aplicació.
Tasques per assolir l'objectiu	Introduir usuari i contrasenya en l'aplicació.
Necessitats d'informació	Saber l'usuari i la contrasenya.
Funcionalitats necessàries	Un formulari d'inici de sessió.
Desenvolupament de la tasca	Omplir el formulari correctament.

Perfil	Repartidor.
Context	Comprova els paquets que ha de repartir aquesta jornada.
Objectius	Veure el llistat d'enviaments que té pendents i realitzats.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del llistat d'enviaments.
Necessitats d'informació	No requereix informació addicional.
Funcionalitats necessàries	Un llistat d'enviaments d'un usuari repartidor en concret
Desenvolupament de la tasca	Obrir l'aplicació i tenir una sessió oberta.

Perfil	Repartidor.
Context	Arriba al lloc indicat i vol identificar el paquet sense buscar-lo en el llistat
Objectius	Identificar un enviament per codi QR.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del lector QR.
Necessitats d'informació	Requereix un codi QR per llegir.
Funcionalitats necessàries	Un lector de QR.
Desenvolupament	Fer clic a la icona de QR i enfocar el codi correctament.

de la tasca	
Perfil	Repartidor.
Context	No sap on ha d'entregar el següent enviament i o consulta en l'aplicació.
Objectius	Trobar en un mapa el destí d'un enviament.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del detall d'un enviament.
Necessitats d'informació	Requereix trobar l'enviament al llistat o utilitzar el lector de QR.
Funcionalitats necessàries	Un mapa del tipus google maps on és maqui el destí de l'enviament. I un enllaç a l'aplicació pròpia de google (google maps).
Desenvolupament de la tasca	Un cop en el detall de l'enviament fer clic al botó "Veure destí" del detall d'un enviament. Es mostra una pantalla amb el mapa i una ruta, però si es vol més informació, es pot obtenir fent un clic llarg a sobre del mapa i s'obre l'aplicació oficial de google maps amb més informació i amb el destí ja establert.

Perfil	Repartidor.
Context	Arriba a la direcció indicada i fa entrega de l'enviament.
Objectius	Entregar l'enviament.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla de la signatura de l'enviament identificat.
Necessitats d'informació	Requereix el nom i cognom o el DNI de qui recull l'enviament i una signatura digital.
Funcionalitats necessàries	Un "canvas" per firmar i un "input text" per recollir les dades de qui recull l'enviament.
Desenvolupament de la tasca	Fer clic al botó "Signar enviament" del detall d'un enviament. Demandar el nom i cognoms o el DNI de qui recull l'enviament i fer que signi en la pantalla. Fer clic al botó "Firmar"

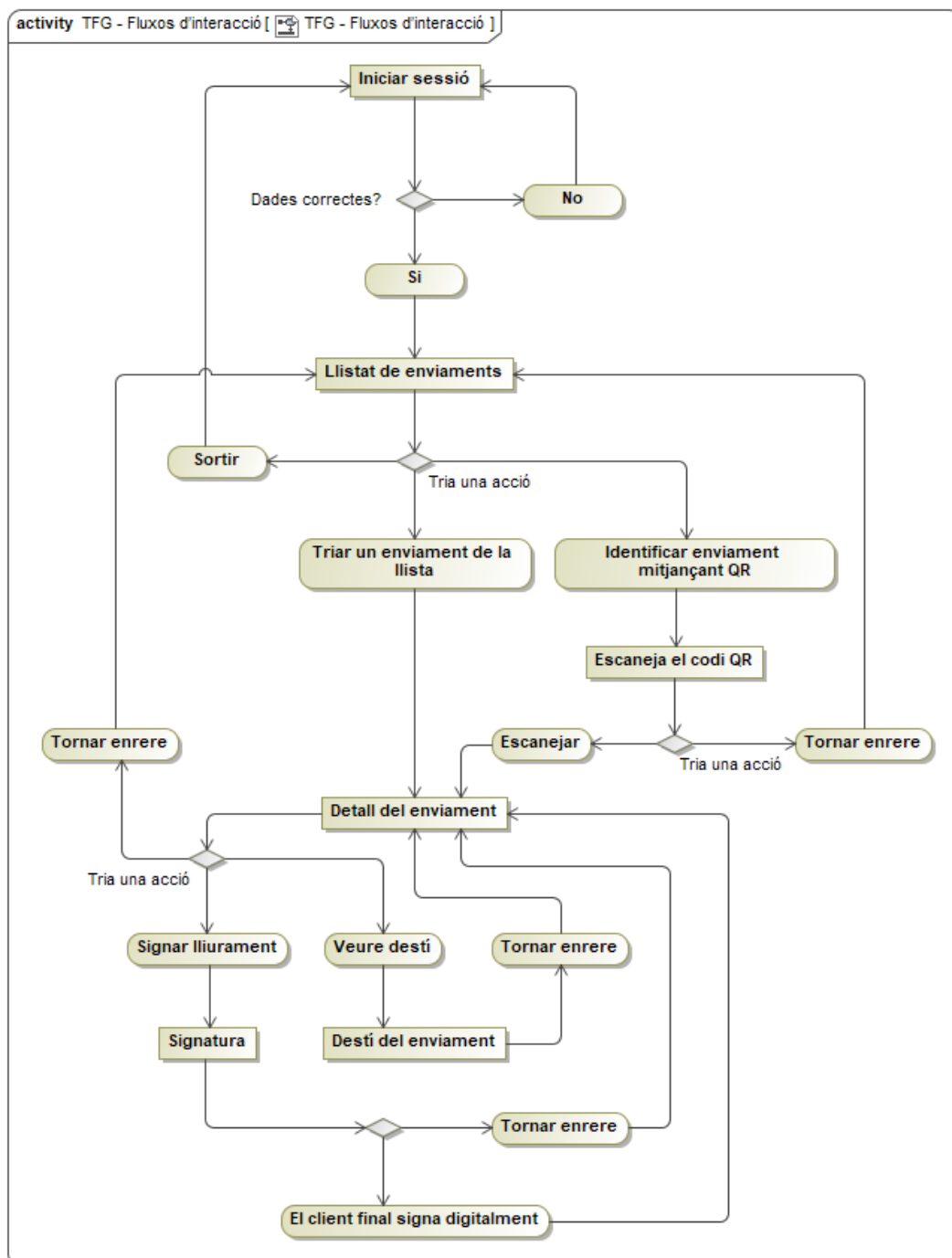
Perfil	Repartidor.
Context	Arriba a la direcció indicada i la direcció és errònia.
Objectius	Informar que la direcció indicada és errònia.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del detall de l'enviament indicat.
Necessitats d'informació	Comprovar que la direcció que s'indica en l'enviament és errònia.
Funcionalitats necessàries	Un botó que canviï l'estat de l'enviament a "Direcció errònia".
Desenvolupament de la tasca	Identificar l'enviament i anar a la pantalla del detall. Fer clic al botó "Direcció errònia"

Perfil	Repartidor.
Context	Arriba a la direcció indicada i no troba a ningú per realitzar l'entrega.
Objectius	Informar que el client no s'ha trobat.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del detall de l'enviament indicat.
Necessitats d'informació	Comprovar que no hi ha ningú a qui fer l'entrega.
Funcionalitats necessàries	Un botó que canviï l'estat de l'enviament a "Client no trobat".
Desenvolupament de la tasca	Identificar l'enviament i anar a la pantalla del detall. Fer clic al botó "Client no trobat"

Perfil	Repartidor.
Context	Per raons justificades s'ha de contactar amb el client que s'ha de fer l'entrega.
Objectius	Trucar al client de l'entrega.
Tasques per assolir l'objectiu	Tenir una sessió iniciada en l'aplicació. Anar a la pantalla del detall de l'enviament indicat.
Necessitats d'informació	Que l'usuari hagi facilitat el seu número de telèfon.
Funcionalitats necessàries	Un botó que directament marqui el número de telèfon del client a fer l'entrega.
Desenvolupament de la tasca	Identificar l'enviament i anar a la pantalla del detall. Fer clic al botó amb el símbol de telèfon.

2.2.2 Fluxos d'interacció

A continuació es mostra un diagrama de flux d'interacció entre l'usuari amb el perfil Repartidor i les diverses pantalles, que representa la navegació per l'aplicació per realitzar les tasques necessàries.



Il·lustració 3 - Flux d'interacció

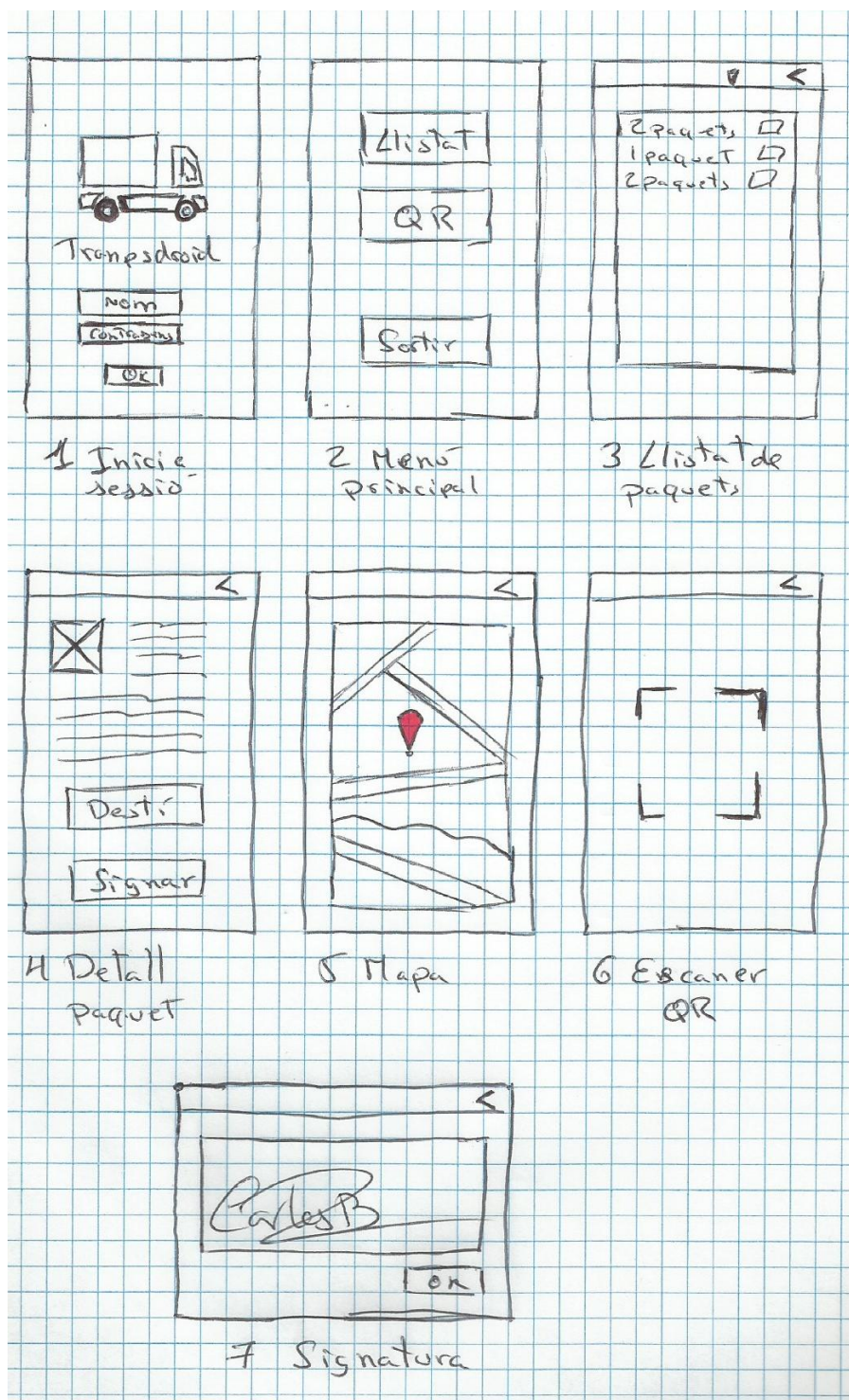
Diagrama creat amb MagicDraw Personal Edition 18.5

Com s'ha comentat anteriorment, per raons de disseny aquest projecte té dos fluxos d'interacció que són complementaris a l'aplicació mòbil, però no realitzarà cap acció directa dintre de l'aplicació.

2.3 Prototipatge

2.3.1 Sketches

De cara a l'elaboració del prototip primer s'ha dissenyat a mà alçada els esbossos de les diferents pantalles o requeriments de l'aplicació.



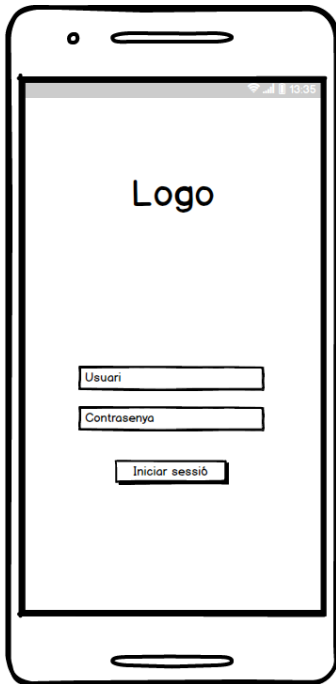
II-lustració 4 - Sketches

2.3.2 Prototipus horitzontal d'alta fidelitat

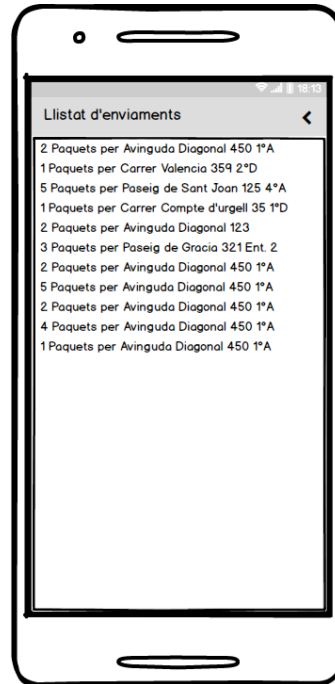
Per aquesta part s'ha creat uns Mockups.

S'ha escollit fer-ho amb mockups perquè representa millor el producte final i que en alguns casos poden contenir funcionalitat representativa.

I per l'aplicació mòbil aquests són els Mockups de les pantalles tretes del flux d'interacció creat anteriorment. S'ha intentat fer una visió completa, però intuïtiva per facilitar el seu ús i aprenentatge.



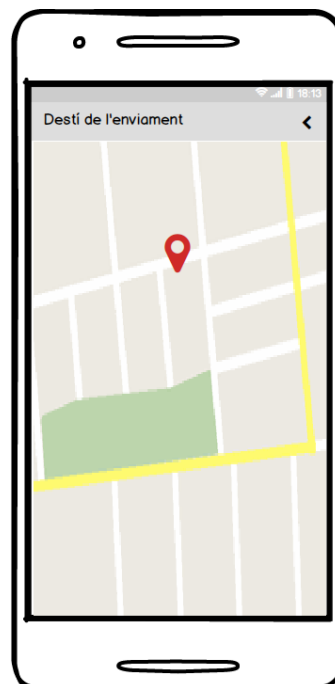
Il·lustració 5 - Inicia sessió



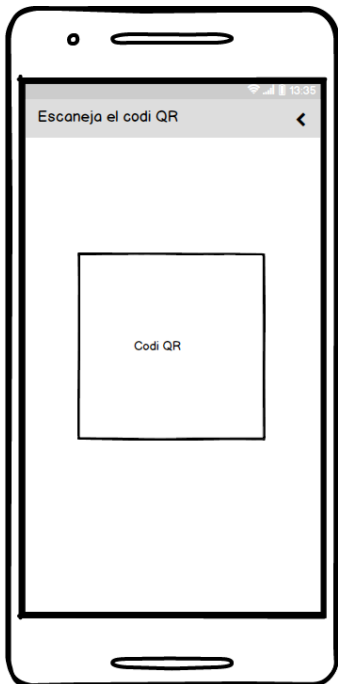
Il·lustració 7 - Llistat d'enviaments



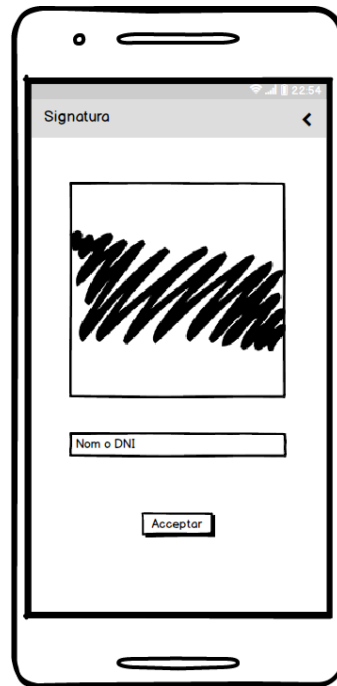
Il·lustració 6 - Detall d'un enviament



Il·lustració 8 - Mapa del destí d'un l'enviament



II-lustració 9 - Escanejar amb codi QR



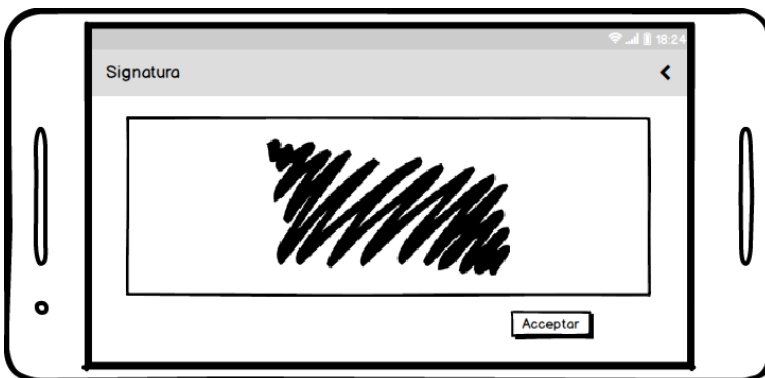
II-lustració 10 - Signatura

2.3.3 Solucions de disseny proposades

A partir dels sketches s'ha intentat respectar la idea principal, però per millorar l'experiència de l'usuari s'ha implementat els següents canvis:

- S'ha canviat tota l'aplicació en vista vertical, ja que només hi havia una pantalla en vista horitzontal. Per tant la pantalla de "Signatura" serà en vertical.
- S'ha afegit un camp més a la pantalla "Signatura", ja que podem oferir més informació de la persona que recull l'enviament, que no sempre serà la persona a qui va adreçada l'enviament.

Per les altres pantalles es proposa una vista simple i fàcil i intuïtiva, això es fa per ajudar al repartidor a ser àgil amb l'aplicació, ja que el seu objectiu és facilitar el treball al repartidor i no donar-li més treball.



II-lustració 11 - Signatura

2.4 Avaluació

Per realitzar l'avaluació de l'app a través del test d'usuaris, s'han de buscar els candidats que compleixin el perfil d'usuari Repartidor.

Per començar s'ha definit una sèrie de preguntes d'informació sobre l'usuari que realitzaria el test.

- 1) Edat?
- 2) Estudis?
- 3) Disposa d'un dispositiu amb plataforma Android?
- 4) Quant temps portes treballant com a repartidor?
- 5) Has utilitzat una aplicació semblant?
- 6) Que esperes d'aquesta aplicació?
- 7) Acostuma a descarregar i provar aplicacions per mòbil?

Per portar a terme l'avaluació, s'escolliran 15 usuaris que responguin a les anteriors preguntes.

Per tal de fer l'avaluació el més real possible es facilitarà a tots el mateix model de dispositiu amb la versió d'Android indicada en el projecte.

Per això s'ha definit les tasques que els usuaris haurien de realitzar.

- 1) L'empresa ha incorporat una nova eina per als repartidors, aquesta eina és una app mòbil i s'ha facilitat a cada repartidor el seu usuari i contrasenya, identificat en la nova aplicació.
- 2) Arribes a l'oficina central i et donen els enviaments que t'han assignat per aquesta jornada, identifica mitjançant el llistat de l'aplicació que tots els enviaments són correctes.
- 3) Identifica mitjançant l'aplicació aquest enviaments per el seu codi QR.



- 4) De l'enviament anterior escriu a continuació les dades corresponents a l'enviament (client, número de paquets, etc.)
- 5) Per una causa justificada et veus amb la necessitat de trucar al client de l'entrega, realitza mitjançant l'aplicació una trucada al client.
- 6) Identifiques el següent enviament a entregar i no coneixes el camí per arribar, consulta la millor ruta amb l'aplicació.
- 7) Ja has arribat al lloc del destí i la direcció és errònia notifica-ho mitjançant l'aplicació.
- 8) Ja has arribat al lloc de destí i no hi ha ningú per recollir l'enviament notifica-ho mitjançant l'aplicació.
- 9) Ja has arribat al lloc de destí, realitza l'entrega de l'enviament.

Per acabar s'ha definit unes preguntes referents a les tasques.

- 1) És una aplicació àgil?
- Si/No
- 2) És una aplicació prou intuïtiva?
- Si/No
- 3) El llistat és útil quan hi ha molts enviaments?
- Si/No
- 4) Quina sembla que és la millor manera d'arribar al detall d'un enviament?
- Llistat/Lector QR
- 5) De l'1-5 quanta informació dona la pantalla del mapa per veure el destí de l'enviament?
- 6) S'ha realitzat les tasques de la forma esperada?
- Si/No
- 7) De l'1-5 quina dificultat ha suposat fer totes les tasques?
- 8) Compleix les necessitats bàsiques per poder treballar amb aquesta aplicació com si fos una eina?
- Si/No
- 9) Què milloraries de l'aplicació? Descriviu breument com a màxim 5 característiques que milloraries de l'aplicació.
- 10) Finalment de l'1-10 quina puntuació li donaries a l'aplicació?

3. Disseny tècnic de l'aplicació

Aquest projecte s'ha dissenyat amb l'objectiu de crear una app mòbil que es connecti mitjanant una API a un servidor que conté totes les dades. La base que s'utilitzarà per a l'aplicació mòbil serà Kotlin i per la connexió entre l'aplicació i la a API REST s'utilitzarà Retrofit.

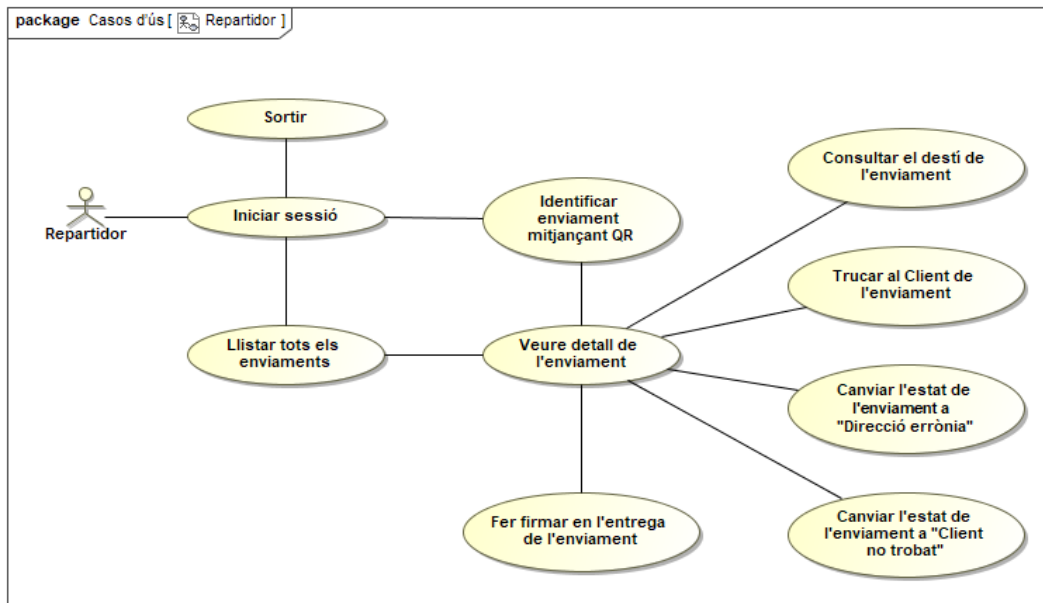
Aquesta aplicació conté els següents productes:

- Aplicació mòbil Android.
 - Autenticació.
 - Pantalles.
 - Lector QR.
 - Mapa Google.
- Una aplicació Web en Laravel.
 - API REST OAuth2.
 - Frontend
 - Backend

En els anteriors casos de disseny centrat en l'usuari s'ha diferenciat entre el que seria l'aplicació mòbil i els serveis creats, per tal de facilitar i ampliar la part servidor web s'ha descrit sempre en l'annex. En aquest apartat com és un disseny més tècnic no se separarà per obtenir una vista més global del projecte.

3.1 Definició dels casos d'ús

- Diagrama casos d'ús Repartidor



Il·lustració 12 - Diagrama casos d'ús Repartidor

Diagrama creat amb MagicDraw Personal Edition 18.5

Identificador	CU-001
Nom	Iniciar Sessió
Prioritat	Alta
Descripció	És el primer cas un cop l'aplicació està iniciada i no es detecta cap usuari guardat en les "Dades de l'aplicació".
Actors	Repartidor
Pre-Condicions	No existeix informació guardada de l'usuari en "Dades de l'aplicació". Requereix connexió a l'API.
Flux	1- L'usuari inicia l'aplicació fent clic sobre la icona i el sistema mostra la primera pantalla, la pantalla d'inici de sessió. 2- L'usuari posa el seu usuari i contrasenya i prem el botó "Iniciar Sessió". 3- El sistema fa una petició al servidor i comprova si és vàlid o no l'usuari i contrasenya. 4a- Si és vàlid, el sistema retorna el "token" i el "refreshToken" en les "Dades de l'aplicació". 4b- Si no és vàlid, el sistema retorna un error. 5- El sistema guarda el "token" i el "refreshToken" en les "Dades de l'aplicació" i es prepara per el CU-002.
Post-Condicions	Es guarda el "token" i el "refreshToken" en les "Dades de l'aplicació".
Notes	M'entres el "token" sigui vàlid, no es torna a demanar la validació. Amb el "refreshToken" es va refrescant i actualitzant el "token" de l'usuari.

Identificador	CU-002
Nom	Llistar tots els enviaments.
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan sobre l'aplicació hi ha un usuari vàlid guardat en les "Dades de l'aplicació".
Actors	Repartidor
Pre-Condicions	Requereix tenir una sessió iniciada i vàlida en l'aplicació. Requereix connexió a l'API.
Flux	1- L'usuari inicia l'aplicació fent clic sobre la icona. 2- El sistema fa una petició al servidor i retorna una resposta en format JSON amb els enviaments d'aquest repartidor. 3- El sistema mou el focus de l'aplicació cap a la pantalla on mostra un llistat d'enviaments amb la informació rebuda.
Post-Condicions	Es guarda en memòria cau la resposta durant un temps predeterminat.
Notes	Es força la petició quan hi ha hagut un canvi d'estat en algun dels enviaments

Identificador	CU-003
Nom	Identificar enviament mitjançant codi QR
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix fer clic sobre la icona del QR en el llistat d'enviaments.
Actors	Repartidor
Pre-Condicion	Requereix tenir una sessió iniciada i vàlida en l'aplicació. L'aplicació necessita permís per utilitzar la càmera del mòbil.
Flux	1- L'usuari fa clic sobre la icona del QR de la pantalla del llistat d'enviaments. 2- El sistema inicia la captura del codi. 3- El sistema llegeix el codi i es prepara per el CU-004.
Post-Condicion	El codi es guarda temporalment per poder fer la petició al servidor de què l'enviament a retornar el seu detall.
Notes	El lector es capaç de llegir QR i codi de barres

Identificador	CU-004
Nom	Veure detall de l'enviament
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar un enviament en el llistat d'enviaments o mitjançant la identificació QR.
Actors	Repartidor
Pre-Condicion	Requereix tenir una sessió iniciada i vàlida en l'aplicació. Requereix que el repartidor tingui assignat algun enviament. Requereix connexió a l'API.
Flux	1a- L'usuari selecciona un enviament del llistat de la pantalla del llistat d'enviaments. 1b- L'usuari prèviament a escanejat un codi QR. 2- El sistema fa una petició al servidor i retorna una resposta en format JSON amb el detall de l'enviament indicat. 3- El sistema mou el focus de l'aplicació cap a la pantalla on mostra el detall de l'enviament amb la informació rebuda.
Post-Condicion	Es guarda en memòria cau la resposta durant un temps predeterminat
Notes	

Identificador	CU-005
Nom	Consultar el destí de l'enviament
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar el botó "Veure destí".
Actors	Repartidor
Pre-Condicion	Requereix tenir una sessió iniciada i vàlida en l'aplicació. L'aplicació necessita permís per utilitzar el GPS. Requereix una API key de Google vàlida. Requereix connexió a l'API.
Flux	1- L'usuari selecciona el botó "Veure destí" de la pantalla del detall d'un enviament. 2- El sistema fa una petició al servidor i retorna una resposta en format JSON amb la direcció correctament formatada de l'enviament. 3- El sistema mou el focus de l'aplicació cap a la pantalla on mostra un mapa amb la localització del lloc amb la informació rebuda. 4- L'usuari si vol més informació, pot fer una pulsació llarga en el mapa i el sistema farà una crida a l'aplicació Google Maps amb un "Ir a" amb la direcció destí ja omplerta.
Post-Condicion	Sense Post-Condicion
Notes	La direcció ha de ser real i correcte si no l'API de Google no retorna bé el lloc.

Identificador	CU-006
Nom	Trucar al client de ha de rebre l'enviament
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar d'un enviament.
Actors	Repartidor
Pre-Condicion	Requereix tenir una sessió iniciada i vàlida en l'aplicació. L'aplicació necessita permís per utilitzar les trucades del mòbil. Requereix connexió a l'API.
Flux	1- L'usuari selecciona la icona de telèfon de la pantalla del detall d'un enviament. 2- El sistema comprovà que hi ha permisos per utilitzar les trucades del mòbil. 3- Es fa una petició de trucada al sistema operatiu.
Post-Condicion	Sense Post-Condicion
Notes	El repartidor farà ús d'aquesta funcionalitat en cas d'una incidència amb l'enviament.

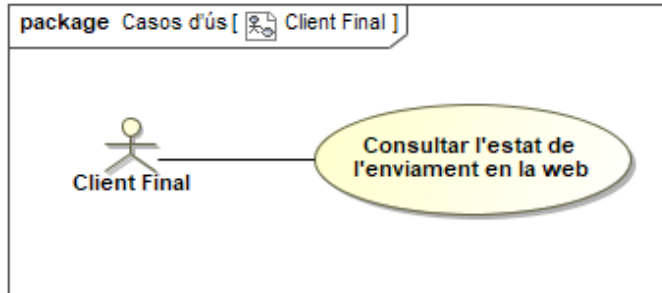
Identificador	CU-007
Nom	Canviar l'estat de l'enviament a "Client no trobat"
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar el botó "Client no trobat".
Actors	Repartidor
Pre-Condicions	Requereix tenir una sessió iniciada i vàlida en l'aplicació. Requereix que el repartidor comprovi que cap persona pot recollir l'enviament. Requereix connexió a l'API.
Flux	1- L'usuari selecciona el botó "Client no trobat" de la pantalla del detall d'un enviament. 2- El sistema fa una petició al servidor via API per demanar que s'actualitzi l'estat de l'enviament a "Client no trobat".
Post-Condicions	El servidor web canvia l'estat de l'enviament a "Client no trobat" i envia un correu.
Notes	Quan el repartidor torni a l'oficina central l'estat de l'enviament canviarà a "En oficina Central" i s'assignarà al repartidor corresponent per fer l'entrega un altre dia.

Identificador	CU-008
Nom	Canviar l'estat de l'enviament a "Direcció errònia"
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar el botó "Direcció errònia".
Actors	Repartidor
Pre-Condicions	Requereix tenir una sessió iniciada i vàlida en l'aplicació. Requereix que el repartidor comprovi si la direcció de l'enviament es vàlida. Requereix connexió a l'API.
Flux	1- L'usuari selecciona el botó "Direcció errònia" de la pantalla del detall d'un enviament. 2- El sistema fa una petició al servidor via API per demanar que s'actualitzi l'estat de l'enviament a "Direcció errònia".
Post-Condicions	El servidor web canvia l'estat de l'enviament a "Direcció errònia" i envia un correu.
Notes	Quan el repartidor torni a l'oficina central l'estat de l'enviament canviarà a "En oficina Central" i s'intentarà solucionar la incidència.

Identificador	CU-009
Nom	Fer firmar en l'entrega de l'enviament
Prioritat	Normal
Descripció	Acció que duu a terme el sistema quan l'usuari decideix seleccionar el botó "Signar enviament".
Actors	Repartidor
Pre-Condicions	Requereix tenir una sessió iniciada i vàlida en l'aplicació. Requereix que el client final pugui facilitar les dades personals i una signatura. Requereix connexió a l'API.
Flux	1- L'usuari selecciona el botó "Signar enviament" de la pantalla del detall d'un enviament. 2- El sistema mou el focus de l'aplicació cap a la pantalla on mostra un "canvas" on l'usuari final ha de signar amb un camp per posar el nom o el DNI. 3- El sistema fa una petició al servidor via API per demanar que s'actualitzi l'estat de l'enviament a "Entregat".
Post-Condicions	El servidor web canvia l'estat de l'enviament a "Entregat" i envia un correu.
Notes	La imatge de la firma no es guardarà.

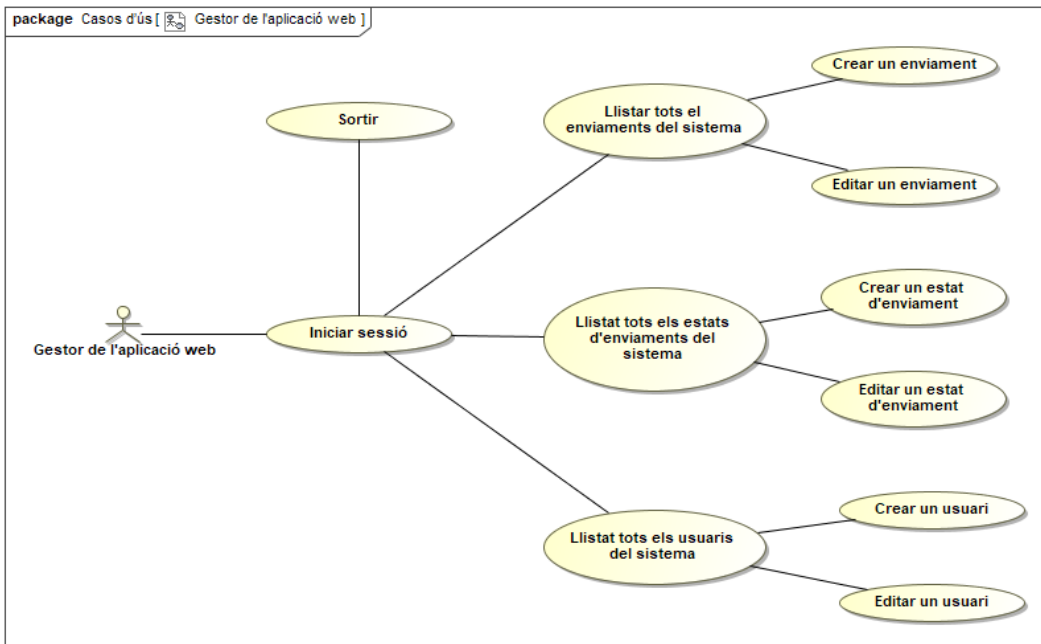
Identificador	CU-010
Nom	Sortir o canviar d'usuari en l'aplicació
Prioritat	Normal
Descripció	És quan el repartidor tanca la seva sessió en el dispositiu. És el primer cas un cop l'aplicació està iniciada i no es detecta cap usuari guardat en les "Dades de l'aplicació".
Actors	Repartidor
Pre-Condicions	Requereix tenir una sessió iniciada i vàlida en l'aplicació.
Flux	1- L'usuari selecciona el botó "Sortir". 2- El sistema elimina les dades de l'usuari guardat en les "Dades de l'aplicació" i es prepara per el CU-001.
Post-Condicions	El servidor web canvia l'estat de l'enviament a "Entregat" i envia un correu.
Notes	La imatge de la firma no es guardarà.

- Diagrama casos d'ús Client Final



II-lustració 13 - Diagrama casos d'ús Client Final
Diagrama creat amb MagicDraw Personal Edition 18.5

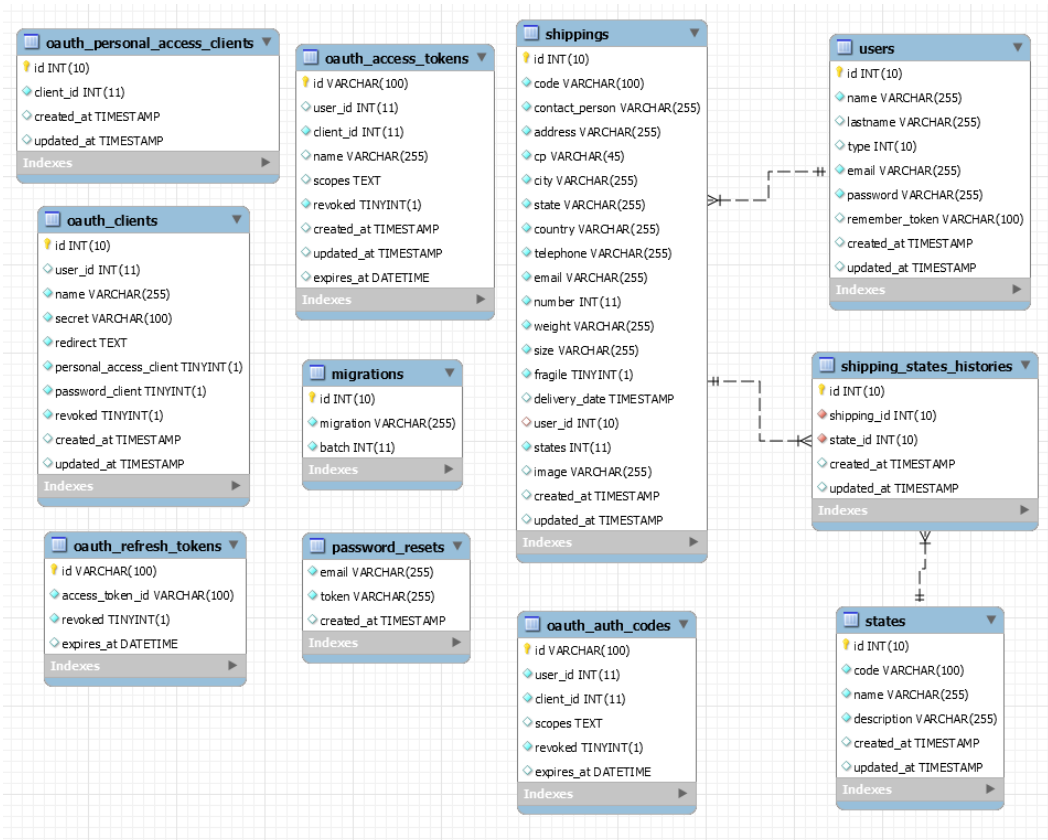
- Diagrama casos d'ús Gestor aplicació web



II-lustració 14 - Diagrama casos d'ús Gestor aplicació web
Diagrama creat amb MagicDraw Personal Edition 18.5

3.2 Disseny de l'arquitectura

3.2.1 Diagrama UML del disseny de la base de dades.



Il·lustració 15 - Diagrama UML del disseny de la base de dades
Diagrama creat amb MySQL Workbench 6.3

Les taules del *framework* Laravel per defecte són:

- users
- migrations
- password_reset

Les taules del *bundle* de Laravel passport són:

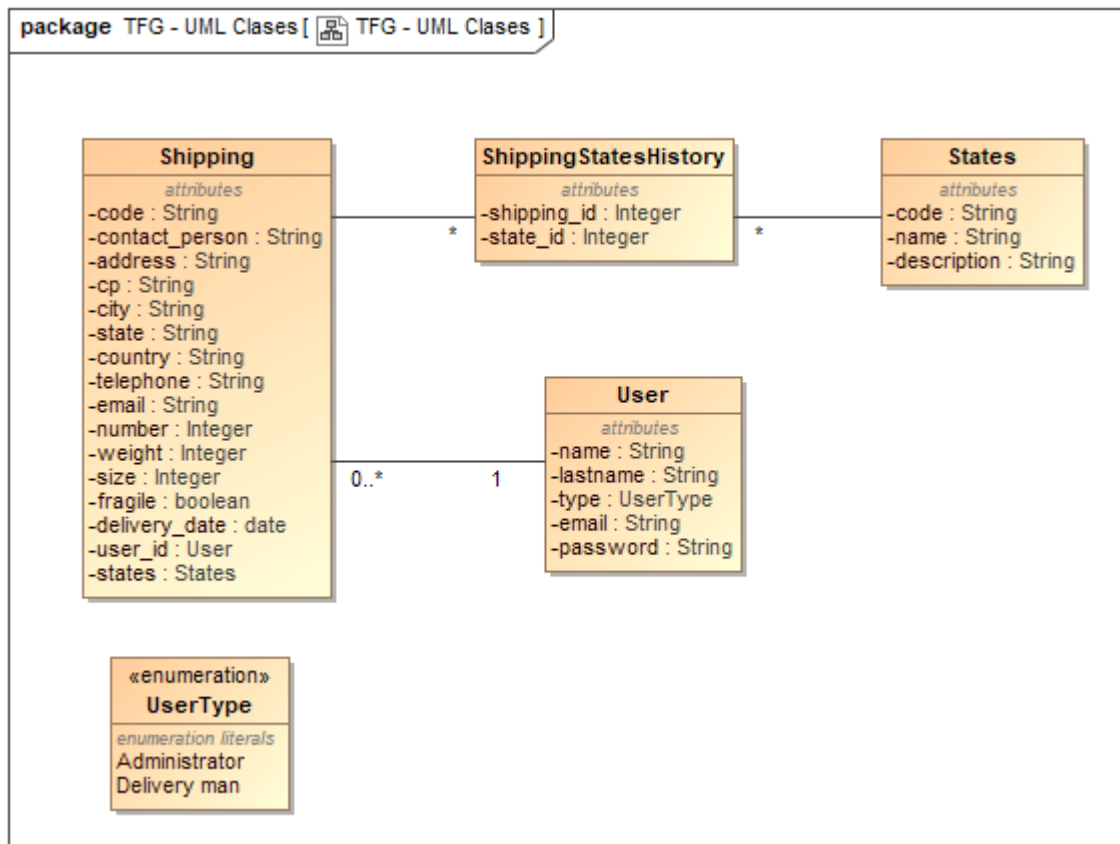
- oauth_access_tokens
- oauth_clients
- oauth_refresh_tokens
- oauth_auth_codes
- oauth_personal_access_clients

Les taules pròpies per aquest projecte són:

- shippings
- shipping_states_histories
- states

3.2.2 El diagrama UML corresponent al disseny de les entitats i classes.

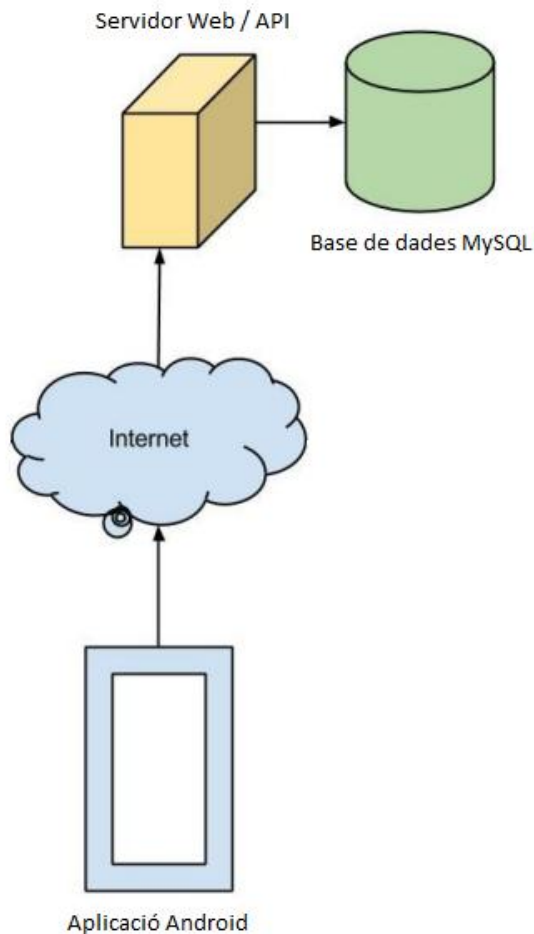
A continuació es mostra les classes relacionades:



II-lustració 16 - El diagrama UML de les entitats i classes
Diagrama creat amb MagicDraw Personal Edition 18.5

3.2.3 Un diagrama explicatiu de l'arquitectura del sistema.

L'app estarà connectada a internet constantment per rebre la informació del servidor, ja que aquest la té emmagatzemada a la base de dades que mitjançant uns serveis API l'aplicació mòbil pot accedir a la informació que necessita per crear les pantalles.



Il·lustració 17 - Diagrama de l'arquitectura del sistema

4. Desenvolupament

4.1 Entorn de desenvolupament.

Per al desenvolupament d'aquest projecte s'ha utilitzat les següents eines i tecnologies. Aquestes eines han facilitat molt el treball a l'hora de realitzar el projecte i verificar el seu correcte funcionament.

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

Android Studio és l'IDE que s'ha utilitzat per crear el codi i compilar l'aplicació Android, la base de l'IDE és IntelliJ IDEA.

Va molt bé per poder detectar errors a l'hora de programar Android. Per exemple té una visualització dels errors tipogràfics o relacionals, també detecta els errors a l'hora de compilar i pot capturar els errors en temps d'execució.

A vegades dona errors/warnings de compilació que són difícils d'esbrinar on s'ha comès l'error. En canvi si és un error en execució, gairebé sempre mostra un error bastant específic.

També té un mòdul per gestionar versions com per exemple amb git, això ens facilita fer un "clone" del repositori realitzar "comit", "pull" i "push". A més conté un Terminal per poder executar comandes.

I per dissenyar les pantalles té un apartat de manera visual amb un "Drag and Drop" o si vols mitjançant codi.

Netbeans: <https://netbeans.org/>

NetBeans és l'IDE que s'ha utilitzat per crear el codi del servidor Web.

Aquest és un entorn de desenvolupament integrat lliure, que aporta a l'usuari un editor molt complet i compacte. Va molt bé per poder detectar errors a l'hora de programar PHP. Una de les coses més útils és la visualització dels errors tipogràfics o relacionals i com navegar entre elles. Per aquest projecte hem utilitzat una extensió de Laravel integrat al propi Netbeans i el gestor de versions que ofereix tot el que necessites per fer ús del Git.

MySQL Workbench: <https://www.mysql.com/products/workbench/>

És l'eina visual de disseny de bases de dades que s'ha utilitzat per realitzar les següents tasques:

- Integrar desenvolupament de programari.
- Administrar la base de dades.
- Disseny la base de dades.
- Creació de Backups i manteniment per al sistema de base de dades MySQL.

Chrome: <https://www.google.com/chrome/browser/desktop/>

Per navegar i verificar el funcionament de la part del projecte "Servidor Web" s'ha utilitzat Google Chrome que és un navegador web desenvolupat per Google, aquest navegador conté aplicació i eines molt útils a l'hora de programar per l'entorn Web.

Postman (Aplicació Chrome): <https://www.getpostman.com/>

Podríem dir que és la millor App de Chrome a l'hora de testejar i validar el funcionament d'una API.

Es poden crear col·leccions de crides a diferents API's utilitzant qualsevol tipus de peticions (GET, POST, PUT, DELETE, etc.).

És molt configurable, ja que pots afegir capçaleres, els paràmetres en el "body" i tot el que necessitis.

Quan es realitza una petició et mostra una sortida on pots validar que el mètode de l'API funciona correctament i el que retorna.

GitHub: <https://github.com/>

Per recopilar el codi de les aplicacions en un únic lloc s'ha utilitzat els repositoris de GitHub. GitHub és una plataforma de desenvolupament col·laboratiu per allotjar projectes utilitzant el sistema de control de versions Git. El codi dels projectes allotjats en GitHub s'han emmagatzemat de forma pública.

WAMP: <http://www.wampserver.com/>

Per poder desenvolupar l'aplicació Web s'ha utilitzat un paquet de software per Windows anomena WAMP.

WAMP conte els següents programes:

- Servidor Apache
- Gestor de bases de datos MySQL
- PHP

Laravel: <https://laravel.com/>

Per la part del Servidor Web s'ha utilitzat Laravel. Laravel és un framework de codi obert per a desenvolupar aplicacions i serveis web amb. Amb laravel s'haconsegueix desenvolupar codi PHP amb MVC. A més té una amplia gama de extensions que es poden instal·lar per facilitar-nos el desenvolupament.

Laravel passport: <https://laravel.com/docs/5.4/passport>

És una extensió oficial de Laravel que ens ofereix tota una base i estructura per començar a crear la nostra API REST. Al principi pot semblar una mica complicar d'instal·lar, però aporta moltes funcionalitats i estabilitat a l'API en desenvolupament.

Simple QrCode: <https://www.simplesoftware.io/docs/simple-qr-code>

És una extensió no oficial de Laravel que ens ofereix la funcionalitat de poder mostrar una imatge en format QR de la cadena de text escollida. Aquesta funcionalitat s'ha utilitzat en el servidor Web per mostrar-la en el correu electrònic i en la pàgina d'editar de l'enviament.

Datatables: <https://datatables.net/>

Datatables és un plugin de JQuery que ofereix la creació dinàmica de taules amb diversos tipus de fonts de dades. Per exemple en aquest projecte s'ha utilitzat en el panell de gestió per poder mostrar les dades de cada mòdul creat. Per agafar les dades de la base de dades s'ha utilitzat AJAX per fer les peticions al controlador.

Mailtrap: <https://mailtrap.io/>

Per la gestió de correu s'ha utilitzat Mailtrap que proporciona un servidor SMTP fals per poder fer proves a l'hora del desenvolupament, comprovant que els correus electrònics s'enviïn correctament. Aquest entorn és tancat i mai surten fora els correus electrònics posis l'adreça que posis arriba sempre al "INBOX" que s'ha configurat de prova.

KotlinCleanArchitecture:

<https://github.com/HelmMobile/KotlinCleanArchitecture>

KotlinCleanArchitecture és una base d'un projecte en llenguatge Kotlin que s'ha utilitzat com la base d'aquest projecte Android. A més d'una base estable aporta funcionalitats pròpies que han sigut útils en desenvolupament de l'aplicació. Aquesta versió no és oficial, ja que ha sigut creada a partir de diferents bases i posteriorment modificada per HelmMobile.

Anko: <https://github.com/Kotlin/anko>

Anko és una llibreria per a Kotlin de Android. Aquesta llibreria fa que el desenvolupament d'aplicacions en Android sigui més ràpida i senzilla. A més Anko fa que el codi sigui net i fàcil de llegir, permetent desenvolupar funcionalitats de manera més fàcil que per part de l'SDK d'Android.

Dexter: <https://github.com/Karumi/Dexter>

Dexter és una llibreria per a Android que simplifica el procés de sol·licitud de permisos en temps d'execució. S'ha utilitzat per administrar els permisos en l'aplicació com per exemple accés al GPS o a les trucades.

Retrofit: <https://github.com/square/retrofit>

Retrofit es un client HTTP de tipus segur per a Android. S'ha utilitzat per fer les crides a la API.

ZXing Android Embedded:

<https://github.com/journeyapps/zxing-android-embedded>

Aquesta és la Llibreria que s'ha utilitzat per realitzar la funcionalitat d'escanejar els codis de barres i QR.

API Google Maps:

<https://developers.google.com/maps/documentation/android-api/start>

API de Google que s'ha fet servir per mostrar el mapa i poder crear mitjançant el "Polyline" una ruta que marqui l'origen (Ubicació GPS) fins al destí de l'enviament.

Enllaços d'ajuda per implementar algunes funcionalitats.

Una recopilació d'enllaços que s'han utilitzat per realitzar diferents funcionalitats del projecte.

- Creació d'un canvas en Android:

<https://examples.javacodegeeks.com/android/core/graphics/canvas-graphics/android-canvas-example/>

- Cridar al Google Maps (Aplicació Android) des de Transpdroid

<https://stackoverflow.com/questions/2662531/launching-google-maps-directions-via-an-intent-on-android>

- Manuals per implementar el lector de QR:

<https://code.tutsplus.com/es/tutorials/android-sdk-create-a-barcode-reader--mobile-17162>

<https://www.numericlabz.com/android-qr-code-scanner-using-zxingscanner-library-tutorial/>

Materials Grafics

Pagines oficials del materials utilitzats en el projecte.

- Icona QR:

<https://materialdesignicons.com/icon/qrcode-scan>

- Icones d'estats:

<https://www.flaticon.com/packs/shipping-delivery-8>

Repositoris Github.

El codi de les aplicacions que implementen les funcionalitats definides per aquest Treball està en un repositori de GitHub

- GitHub AppMovel

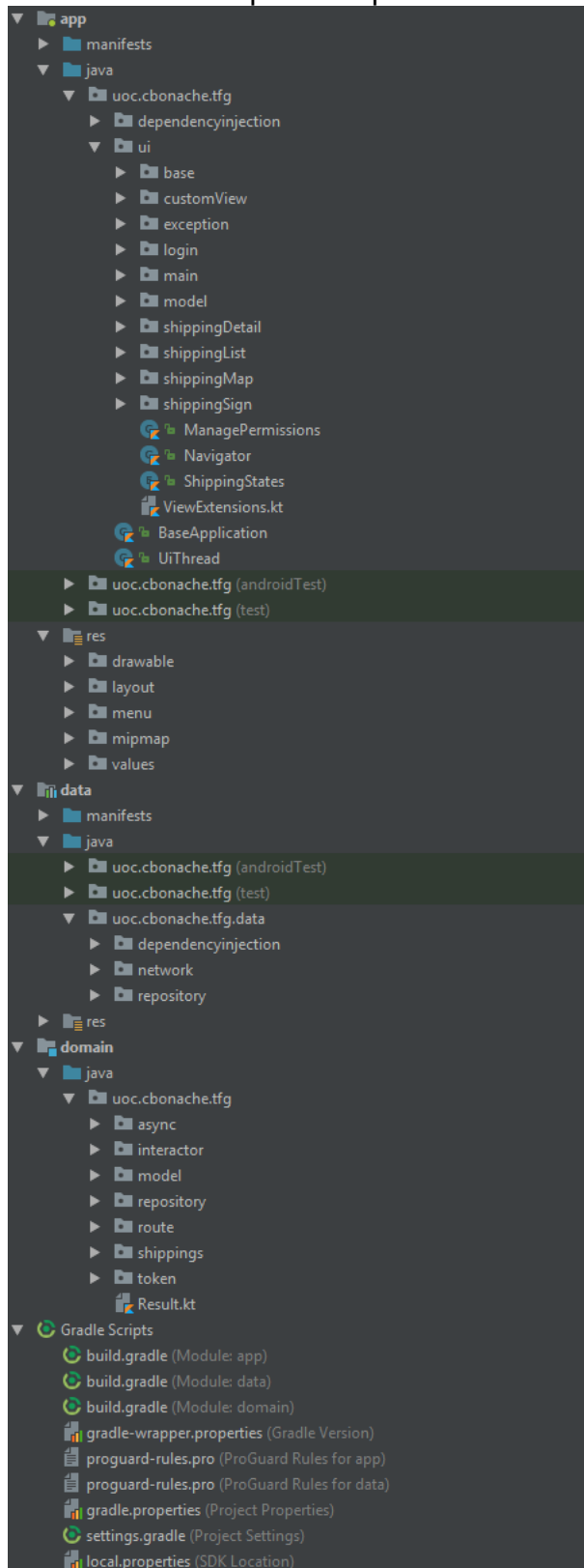
<https://github.com/cbonacheuoc/Transpdroid-Android-App/tree/develop>

- GitHub Servidor Web

<https://github.com/cbonacheuoc/Transpdroid-Server-API>

4.2 Estructura aplicació Android^[20] [21]

L'estructura de l'aplicació que s'ha utilitzat en aquest projecte en la següent.

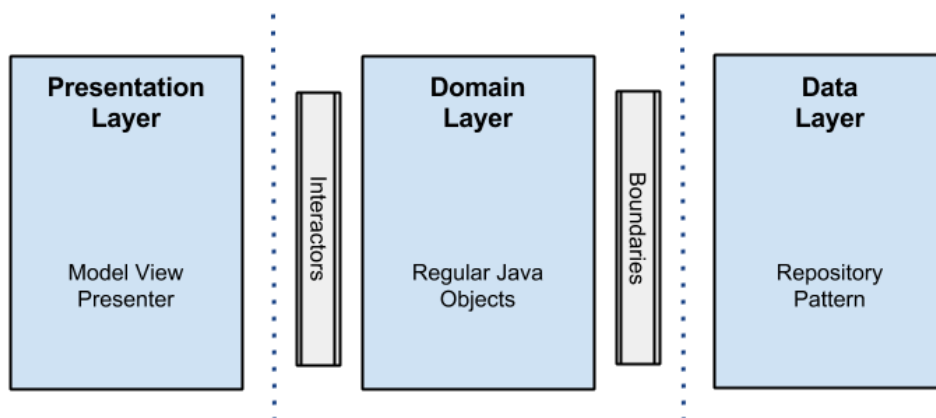


II-lustració 18 – Estructura projecte Android

S'ha de dir que aquesta estructura es donada gracies a l'arquitectura Kotlin utilitzada.

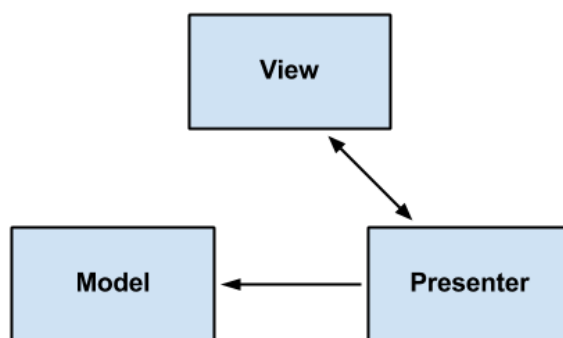
L'estructura es compon principalment per tres grans mòduls:

- App (Presentation Layer)
- Domain (Domain Layer)
- Data (Data Layer)



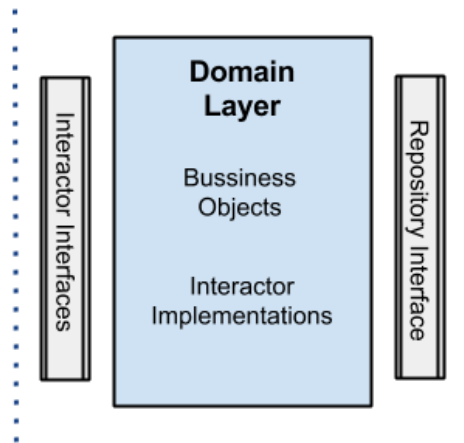
II-lustració 19 - Estructura aplicació Android (Kotlin) ^[21]

El mòdul **App** conté la lògica relacionada amb vistes i animacions. No utilitza més que un *Model View Presenter*. En aquest mòdul els *fragments* i les *activities* són només les vistes, no hi ha cap lògica dins d'ells que no sigui la lògica de la interfície d'usuari, també conte tot el material de gràfic (Icones).



II-lustració 20 - Model View Presenter (MVP) ^[21]

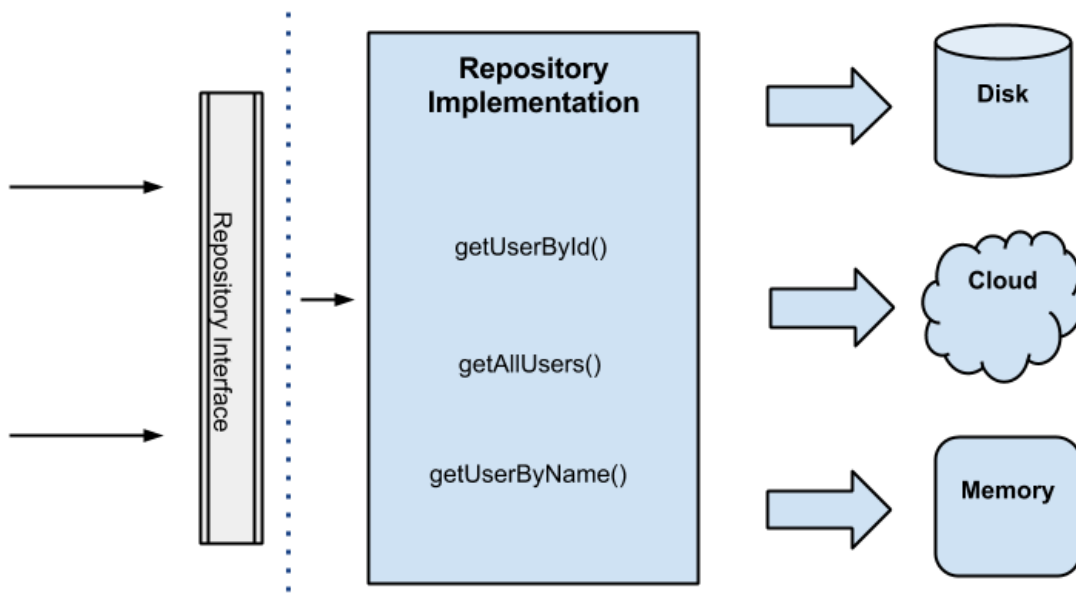
El mòdul de **Domain** és on passa tota la lògica de l'aplicació. També és on s'implementa els *interactors* (Casos d'ús). Aquesta capa és un mòdul de Java pur sense dependències d'Android. Tots els components externs utilitzen interfícies quan es connecten als "*business objects*".



II-lustració 21 – Domain Layer structure^[21]

El mòdul de **Data** és el mòdul que obté totes les dades necessàries per a l'aplicació provenen d'aquesta capa a través d'una implementació d'un Repository, aquesta és la interfície que es troba a la capa de domini i utilitza un *Repository Pattern* amb un strategy que, a través d'una *factory* s'obté les fonts de dades diferents segons les condicions indicades.

El benefici que s'obté utilitzant aquest mètode és que l'origen de les dades és transparent per al client, no hi ha diferència si les dades provenen de la memòria, el disc o del *cloud*, només s'obtiniran les dades sense importar el seu origen.



II-lustració 22 - Data Layer structure^[21]

4.3 Estat actual del projecte

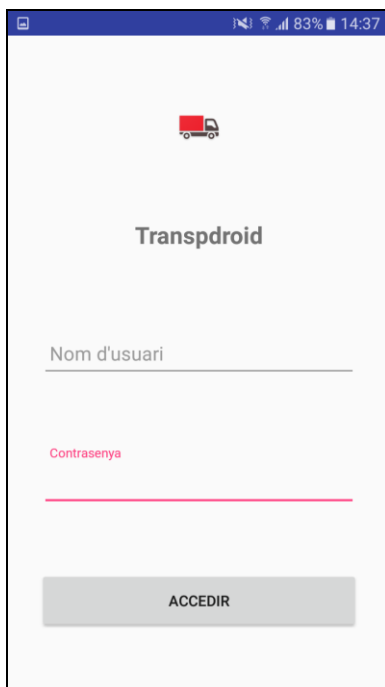
S'ha implementat gairebé completament el Servidor Web. Les funcionalitats que queden pendents son:

- Fer que un usuari que no sigui del tipus repartidor pugui iniciar sessió en la part Backend de la web.
- Ja que es un prototip i no tenim enviaments diaris i fer enviaments de prova suposa fer enviaments per molts dies, no es limita actualment els enviaments pel seu dia de enviament per poder veure els enviaments a l'app de Android.

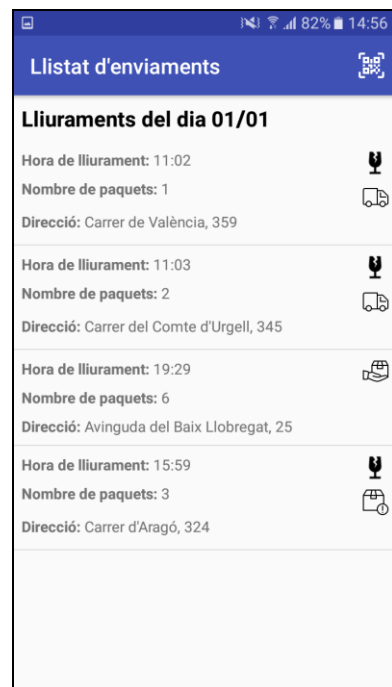
Per la part de l'aplicació mòbil només faltarien finalitzar uns detalls, com per exemple:

- Poder tancar la sessió.

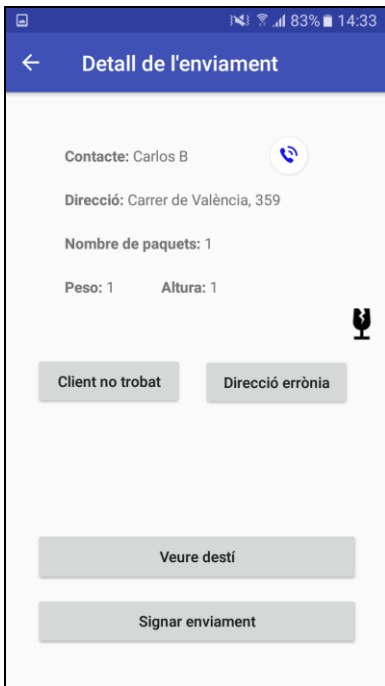
A continuació les captures real de la aplicació final.



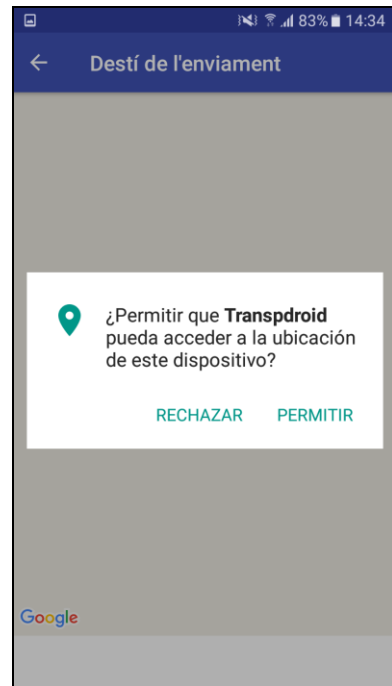
Il·lustració 23 - Inicia sessió



Il·lustració 24 - Llistat de paquets



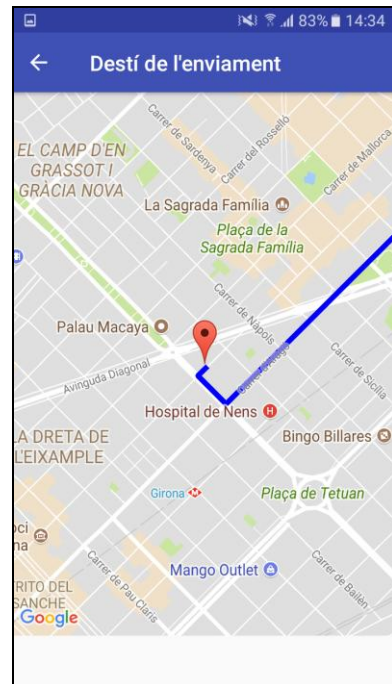
Il·lustració 25 - Detall d'un paquet



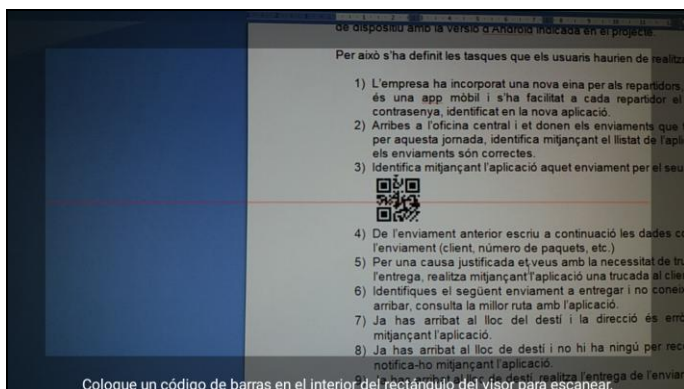
Il·lustració 27 - Exemple petició permisos



Il·lustració 26 - Signatura



Il·lustració 28 - Mapa del destí d'un l'enviament



II-Il·lustració 29 - Escanejar amb codi QR

4.4 Proves

L'aplicació s'ha estat testejat durant tot el desenvolupament tant en dispositiu Samsung Galaxy J5 com a simulador d'Android Studio a l'hora de desenvolupar.

L'aplicació mòbil és bastant senzilla, ja que a part de les funcionalitats de llegir el QR, mostrar mapa i signar només fa crides a l'API.

És per això que les proves del funcionament dels serveis de l'API s'han testejat en l'aplicació Postman.

En Annex 8.8 hi ha un petit tutorial de com fer servir aquesta aplicació i verificar el correcte funcionament dels serveis de l'API

Posteriorment, amb les Proves d'Integració, s'ha comprovat el funcionament del servei en qüestió.

4.3.1 Proves d'integració.

En el següent apartat es detalla les accions que haurà de realitzar l'aplicació, ordenant aquestes, en funció de la pantalla en què ens situem. Es detalla l'acció que farà l'usuari i el resultat que ha d'executar:

Pantalla Iniciar Sessió			
Numero	Acció	Resultat esperat	Verificació
1	Fer clic sobre el botó "Accedir" sense dades	Mostra error, falten dades	OK
2	Fer clic sobre el botó "Accedir" amb dades incorrectes	Mostra error genèric	OK
3	Fer clic sobre el botó "Accedir" amb dades correctes	Carrega el llistat d'enviaments d l'usuari	OK
Nº d'accions Totals	Correctes	Errònies	% Verificació
3	3	0	100%

Pantalla llistat d'enviaments			
Numero	Acció	Resultat esperat	Verificació
1	Fer clic sobre la icona QR	Mostrar un lector de QR per llegir el codi	OK
2	Fer clic sobre un enviament del llistat	Carrega el detall de l'enviament escollit	OK
3	Fer clic sobre la icona Sortir	Tanca la sessió oberta	KO
Nº d'accions Totals	Correctes	Errònies	% Verificació
3	2	1	66.66%

Pantalla lector QR			
Numero	Acció	Resultat esperat	Verificació
1	Col·locar la càmera enfocant a un QR vàlid	Carrega el detall de l'enviament escollit amb el codi QR	OK
2	Col·locar la càmera enfocant a un QR invàlid	Mostra error i torna al llistat d'enviaments	OK
Nº d'accions Totals	Correctes	Errònies	% Verificació
2	2	0	100%

Pantalla detall de l'enviament			
Numero	Acció	Resultat esperat	Verificació
1	Fer clic sobre la icona del telèfon	Fa una trucada al client	OK
2	Fer clic sobre el botó "Client no trobat"	Canvia l'estat de l'enviament i torna al llistat d'enviaments	OK
3	Fer clic sobre el botó "Direcció errònia"	Canvia l'estat de l'enviament i torna al llistat d'enviaments	OK
4	Fer clic sobre el botó "Veure destí"	Carrega el mapa amb una ruta per anar al destí de l'enviament	OK
5	Fer clic sobre el botó "Signar enviament"	Carrega la pantalla de signar enviament	OK
Nº d'accions Totals	Correctes	Errònies	% Verificació
5	5	0	100%

Pantalla mapa enviament			
Numero	Acció	Resultat esperat	Verificació
1	Fer clic llarg sobre el mapa	Obre el Google Maps amb la direcció de l'enviament	OK
Nº d'accions Totals	Correctes	Errònies	% Verificació
1	1	0	100%

Pantalla signar enviament			
Numero	Acció	Resultat esperat	Verificació
1	Fer clic sobre el botó "Firmar"	Canvia l'estat de l'enviament i torna al llistat d'enviaments	OK
Nº d'accions Totals	Correctes	Errònies	% Verificació
1	1	0	100%

5. Conclusions

Les conclusions del projecte realitzat són personalment bones, ja que s'ha aconseguit, partint d'una base lliure crear un prototip de l'aplicació amb les funcionalitats que es desitjaven. Tot això tenint en compte que no es tenia molta experiència s'ha desenvolupat una aplicació mòbil en la seva totalitat, en la qual s'ha pogut posar en pràctica tots els coneixements de les assignatures del grau. Aquest conjunt de coneixement que utilitzats conjuntament amb l'experiència laboral han fet possible poder avançar i arribar a la finalització del projecte.

S'ha hagut d'estudiar el funcionament de l'Arquitectura Kotlin i aprendre a manejar la SDK d'Android, això ha ocasionat més d'un contratemps, ja que a vegades no era fàcil trobar d'on provenia l'error.

Les funcionalitats bàsiques no s'han vist afectades, però si s'han desenvolupat més funcionalitats que les que en un moment es pensaven que eren necessàries per al seu funcionament.

La planificació ha sigut afectada per les valoracions de temps en algunes de les funcionalitats, tant positivament com negativament, per això encara que hi ha hagut modificacions s'ha pogut finalitzar correctament l'aplicació.

La metodologia ha estat adequada, ja que tot i alguns contratemps, s'ha pogut ajustar els esforços per obtenir les funcionalitats de l'aplicació, en aquest sentit les metodologies àgils ens ofereixen una referència immediata del progrés del projecte i ens permeten avaluar la prioritat de les tasques i dedicar-li més recursos.

Línies de treball futur

Les ampliacions i següents actuacions són diverses:

En general és podria treure les dades de test i deixar l'aplicació millorada amb els petits canvis que s'han de fer en l'API. En la app mobil s'hauria de terminar el tancar usuari i guardar temporalment la signatura i el DNI de l'enviament en els servidors com a comprovant quant s'entrega l'enviament.

Una millora possible seria incorporar noves funcionalitats depenent de com s'organitzi cada empresa de transports, però una nova funcionalitat podria ser el control de temps que porta conduint un repartidor sense fer una parada, ja que com a norma general, per evitar la fatiga cal descansar 20 minuts cada 2 hores de conducció o cada 150/200 quilòmetres.

Per una altra, l'aplicació web és pot ampliar, fent la part del client que registra el enviament i dona tota la informació per que un repartidor vagi a recollir al seu domicili el paquet i el repartidor l'envií a la oficina central per ser enviat al seu destí. Aquesta modificació es faria de una manera fàcil afegint un formulari al Frontend amb les dades necessàries per al enviament i una passarel·la de

pagament. En l'aplicació Android es veuria que l'estat es de recollir i no de entregar i per recollir només fa falta el lector QR per identificar el paquet i que el repartidor el porti a la oficina central.

6. Glossari

1. **CRUD**(Create, Read, Update and Delete): Són les quatre funcions bàsiques d'emmagatzemament persistent.
2. **Framework**: és un entorn de treball és una estructura conceptual i tecnològica d'assistència definida.
3. **Backend**: És la part que processa l'entrada des del frontend.
4. **Frontend**: És la part del programari que interactua amb els usuaris.
5. **Mockup**: és un prototip si proporciona almenys una part de la funcionalitat d'un sistema i permet proves del disseny.
6. **Core**: Nucli de l'aplicació.
7. **API**: Representa la capacitat de comunicació entre components de programari.
8. **REST**: Estil d'arquitectura programari per a sistemes en aquest cas basades en HTTP.
9. **OAuth2**: Estàndard obert que permet fluxos simples d'autorització per a llocs web o aplicacions informàtiques.
10. **JSON**: Format de text lleuger per a l'intercanvi de dades
11. **Kotlin**: Llenguatge de programació de tipatge estàtic que corre sobre la Màquina Virtual de Java
12. **Listener**: Funció que escolta si hi ha hagut algun esdeveniment indicat.
13. **Presenter**:
<https://developer.android.com/reference/android/support/v17/leanback/widget/Presenter.html>
14. **Factory**:
<https://developer.android.com/reference/android/arch/lifecycle/ViewModeIProvider.Factory.html>
15. **Cloud**: Cloud computing, és un paradigma que permet oferir serveis de computació a través d'una xarxa, que usualment és Internet.

7. Bibliografia

1. 3&Punt Solucions Informàtiques: <http://www.tresipunt.com> – 30/09/2017
2. Laravel 5.4 - Requeriments: <https://laravel.com/docs/5.4> – 3/10/2017
3. Ubuntu: <https://www.ubuntu.com> – 3/10/2017
4. Estructura Kotlin
<https://github.com/HelmMobile/KotlinCleanArchitecture> – 12/10/2017
5. Centos: <https://www.centos.org> – 3/10/2017
6. Netbeans (Oracle): <https://netbeans.org> – 3/10/2017
7. Android Studio: <https://developer.android.com/studio> – 3/10/2017
8. Diagrama de Gantt Online: <https://www.tomsplanner.es> – 30/09/2017
9. Logo aplicació: <https://www.crearlogogratisonline.com> – 7/10/2017
10. Creació de mockups <https://balsamiq.com> – 7/10/2017
11. MagicDraw Personal Edition 18.4 - No Magic – 7/10/2017
12. Raons per utilitzar Kotlin
<https://devexperto.com/12-razones-usar-kotlin-android/> – 12/10/2017
13. Helm Web <http://www.helm.cat/es/> – 12/10/2017
14. Laravel passport <https://laravel.com/docs/5.4/passport> – 12/10/2017
15. Mailtrap.io <https://mailtrap.io> – 12/10/2017
16. Admin panel
<https://startlaravel.com/themes/sb-admin-laravel-5/> – 12/10/2017
17. Cicle vida app Android <http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad> – 25/10/2017
18. Android Activity Lifecycle
19. <https://www.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png> – 25/10/2017
20. Anko <https://github.com/Kotlin/anko> – 29/10/2017
21. Cambi nom del paquet
<https://stackoverflow.com/questions/16804093/android-studio-rename-package> – 10/11/2017
22. Google Maps
<https://www.youtube.com/watch?v=lchyOhPREh4> – 13/12/2017
23. Estructura d'un projecte Android <http://www.sgoliver.net/blog/estructura-de-un-proyecto-android-android-studio/> – 20/12/2017
24. Architecting Android <https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/> - 20/12/2017

8. Annexos

8.1 Llistat de preguntes realitzades en les entrevistes

1. Què treball exerceix?
2. Quin és el seu procediment?
3. Quina tasca és la que més recursos consum?
4. De quines eines disposa actualment per exercir el seu treball?
5. Què eina li facilita més el treball?

8.2 Resum de les conclusions de les entrevistes

1. Gairebé tots han contestat les mateixes funcionalitats bàsiques.
 - Veure els enviaments pendents.
 - Llegir el codi de l'enviament.
 - Signar l'enviament quan s'està fent l'entrega.
2. Tots fan el mateix procediment, alguns tenen una política d'empresa diferent, però tots coincideixen en el fet que han d'anar a l'oficina central o similar per emplenar el vehicle (cotxe, moto, camió, bicicleta, carro de la compra, etc.) i fer entrega de tots els enviaments
3. La tasca que més recursos consum per un 75% dels entrevistats és fer el camí fins al lloc d'entrega, mentre que un 25% és fer l'entrega en si (Identificar el paquet de l'enviament i que el signi la persona que el recull).
4. Tots tenen un vehicle i un dispositiu per enregistrar l'entrega.
5. En aquest cas, les respostes han sigut en un 90% que l'eina que les facilita més el treball és el lector QR de la PDA.

Amb aquests resultats podem treure diverses idees.

- Quines funcionalitats bàsiques hem d'implementar.
- Els repartidors siguin del tipus que siguin les seves funcions bàsiques són les mateixes.
- Una funcionalitat interessant seria implementar un mapa per indicar el lloc on fer l'entrega.
- Estan habituats a utilitzar eines tipus PDA o mòbil.
- El lector QR és una eina que s'ha d'implementar per facilitar l'accés a les dades de l'enviament.

8.3 Perfils d'usuaris

Com s'indica anteriorment a part de l'aplicació mòbil s'ha plantejat una part Servidor que dona diversos serveis. Aquests serveis estan dividits en dos: Backend i Frontend. És per això que apareixen dos perfils d'usuaris que faran ús d'aquests serveis.

Client Final

Aquest perfil d'usuari és el receptor de l'enviament. Aquest usuari serà qui rebrà en cada canvi d'estat un correu electrònic i podrà utilitzar la part Frontend del servidor web per consultar l'estat de l'enviament en qualsevol moment. Les seves característiques de perfil poden ser molt diverses.

- **Característiques demogràfiques:** Qualsevol part del món on es faci ús aquesta aplicació.
- **Interessos:** Vol poder saber l'estat del seu enviament i rebre'l.
- **Motivacions:** Poder recollir el seu enviament.
- **Experiència amb l'ús de la tecnologia mòbil:** Pot tenir diverses experiències, ja que depenen de l'usuari i aquest perfil de treball conté un ampli rang d'edat i estudis.
- **Contexts d'ús:** Farà ús de l'aplicació durant el temps d'espera de l'enviament.
- **Anàlisi de tasques:** Les tasques a realitzar per assolir els seus objectius seran:
 - Rebre Correus electrònics.
 - Anar a la pàgina web.
 - Omplir correctament el formulari de consulta de l'estat de l'enviament.
- **Llistat de característiques:**
 - Una part visible per qualsevol usuari al servidor web.
 - Un formulari de consulta de l'estat de l'enviament.
 - Una taula amb l'històric d'estats de l'enviament.

Gestor Panell Administratiu

Aquest perfil d'usuari és qui crea els enviaments simulant que un client a comprat els serveis de missatgeria per poder enviar un o mes paquets al Client Final. Aquest usuari existeix pel fet de que tal i com s'ha plantejat i dissenyat el servidor, conte una part Backend amb un panel administratiu, aquest panel administratiu conte les dades de l'aplicació que després a través d'una API s'envien a l'aplicació mòbil.

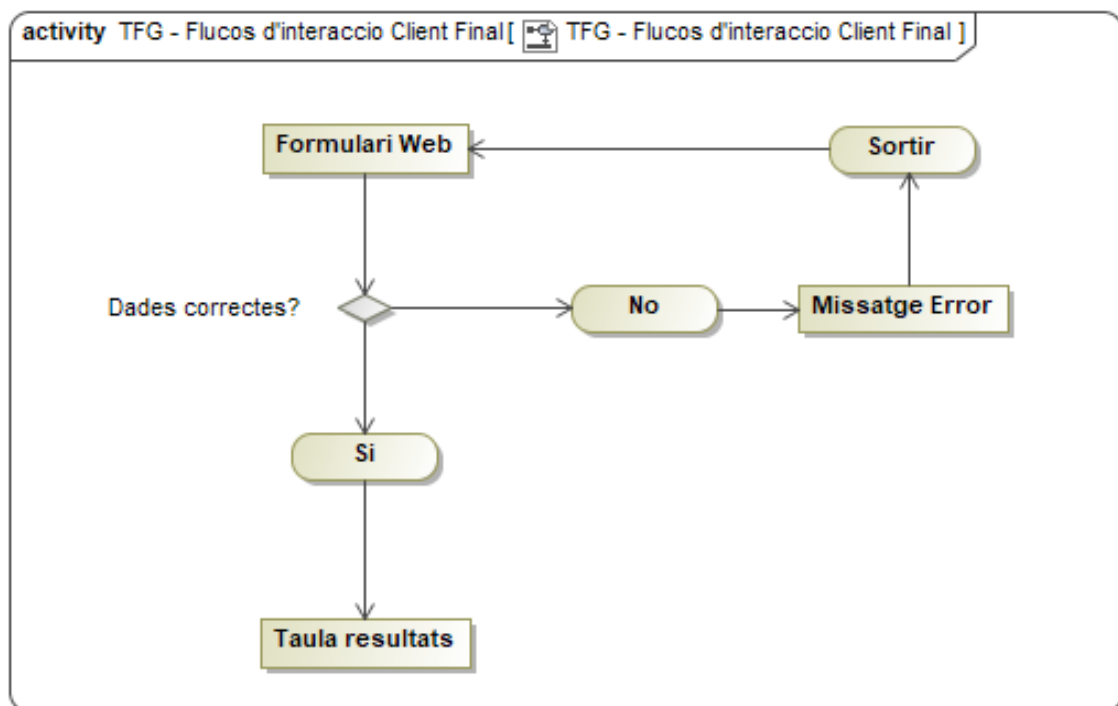
- **Contexts d'ús:** Farà ús de l'aplicació per poder tenir dades simulades per poder utilitzar l'aplicació.
- **Anàlisi de tasques:** Les tasques a realitzar per assolir els seus objectius seran:
 - Crear o modificar usuaris del tipus repartidor.
 - Crear o modificar estats de l'enviament.
 - Crear enviaments.
 - Modificar l'estat de l'enviament.

- **Llistat de característiques:**
 - Un mòdul per gestionar els usuaris.
 - Un mòdul per gestionar els estats dels enviaments
 - Un mòdul per gestionar els enviaments

8.4 Escenaris d'us

Client Final

El client final pot consultar l'estat de l'enviament en qualsevol moment d'ençà que obté el codi de l'enviament.

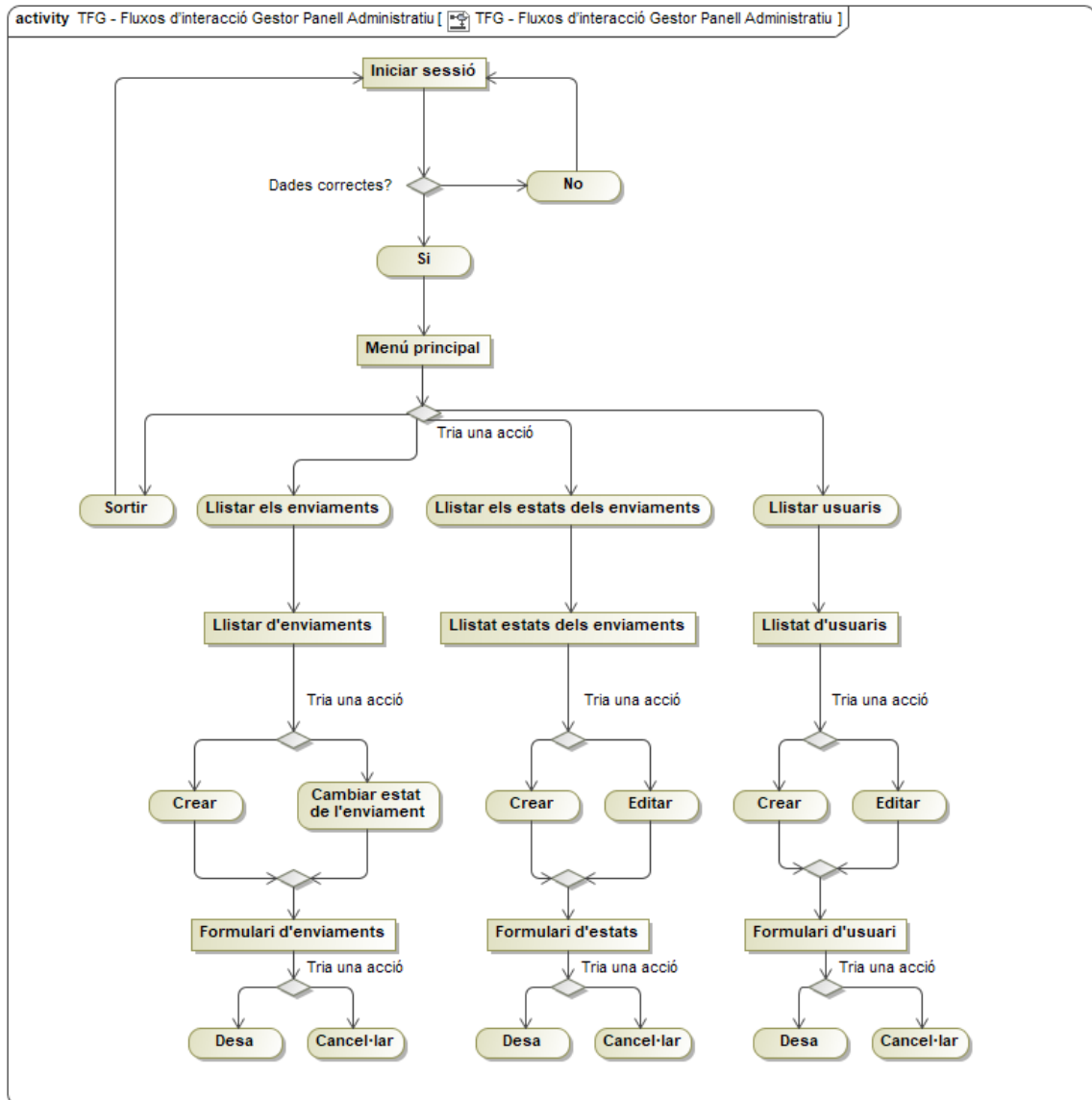


II-lustració 30 - Flux d'interacció Frontent Web

Gestor Panell Administratiu

El gestor del panell administratiu pot gestionar totes les dades de l'aplicació amb el Backend.

De moment no s'implementarà l'opció de borra en cap gestió del Backend.



II-lustració 31 - Flux d'interacció Panell Administració

8.5 Manual instal·lació servidor web.

- Instalar Composer

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

- Instalar Laravel

Per crear la base del projecte Web instalem el "Core" de Laravel

```
php composer.phar create-project --prefer-dist laravel/laravel transpdroid
"5.4.*"
chown -R www-data:www-data storage/
chown -R www-data:www-data bootstrap/cache/
php composer.phar require laravel/passport
```

- Crear VirtualHost del Apache

```
<VirtualHost *:80>
    ServerName transpdroid.tresipunt.com
    ServerAlias transpdroid.tresipunt.com
    DocumentRoot /var/www/html/cbonache-tfg/transpdroid/public
    SetEnv APPLICATION_ENV "development"
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    <Directory /var/www/html/cbonache-tfg/transpdroid/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Per que els canvis tinguin efecte hem de reiniciar l'Apache.

Per defecte no esta configurat el .env

```
cp .env.example .env
```

8.6 Manual compilar codi Android i crear APK.

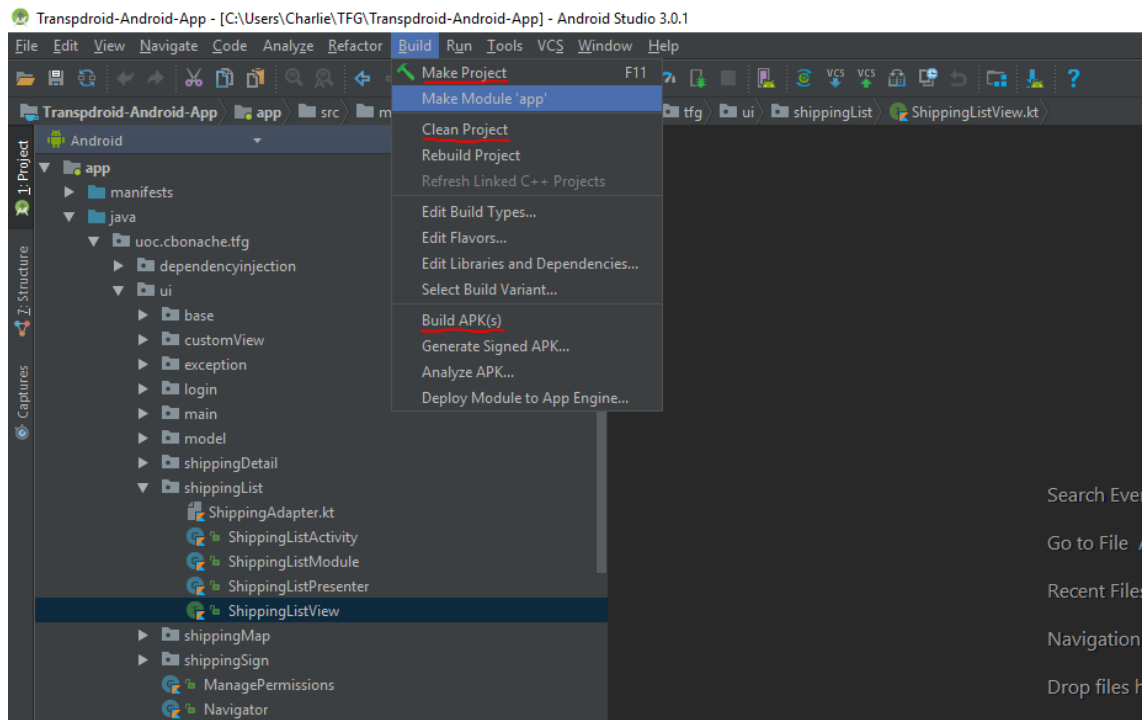
Per compilar s'utilitza el Android estudio.

Primer farem "Build" → "Clean Project" per netejar el projecte

Després farem "Build" → "Make Project" amb això ja s'ha compilat el codi.

Per crear la APK també s'utilitza el Android estudio.

Es fa en "Build" → "Build APK(s)"

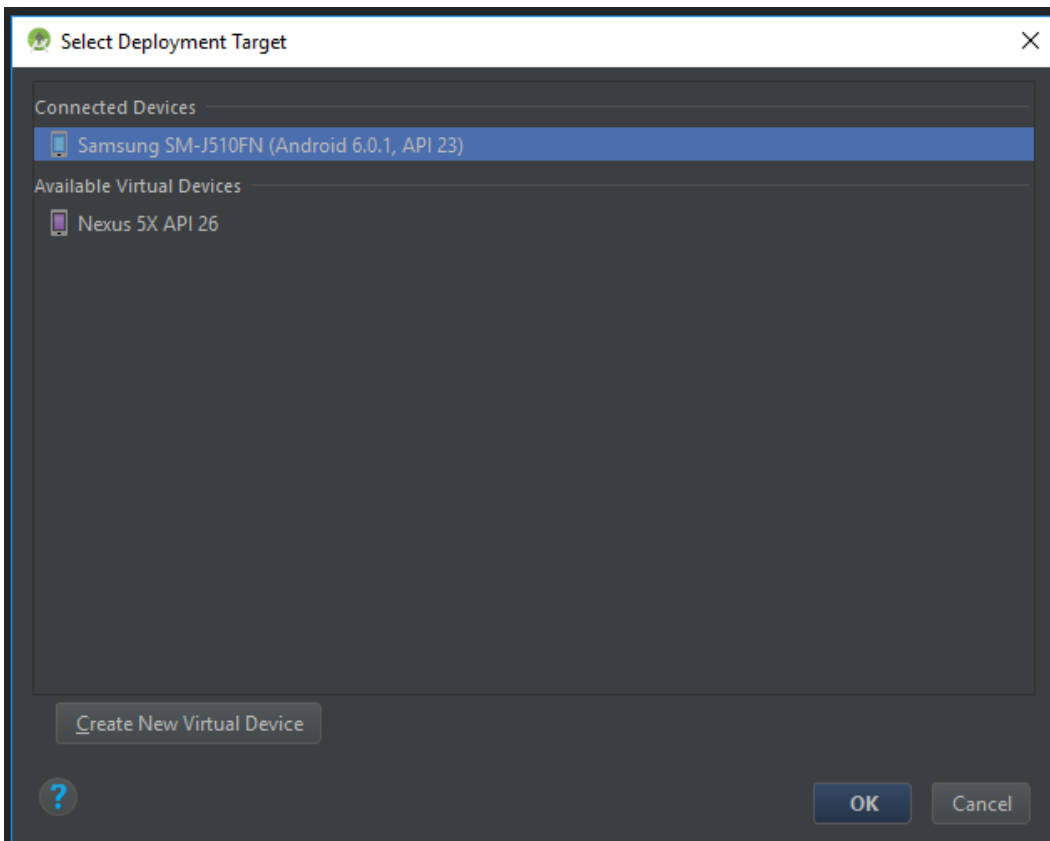


Il·lustració 32 – Android Studio

8.7 Manual instal·lar apk.

Es pot fer de dues maneres:

1. Un cop està compilat el codi es connecta mitjançant USB en mòbil a l'ordinador i fem clic en "Run" i et donarà l'opció d'utilitzar l'emulador o el mòbil connectat. Si seleccionen l'opció del mòbil connectat, s'instal·larà automàticament.



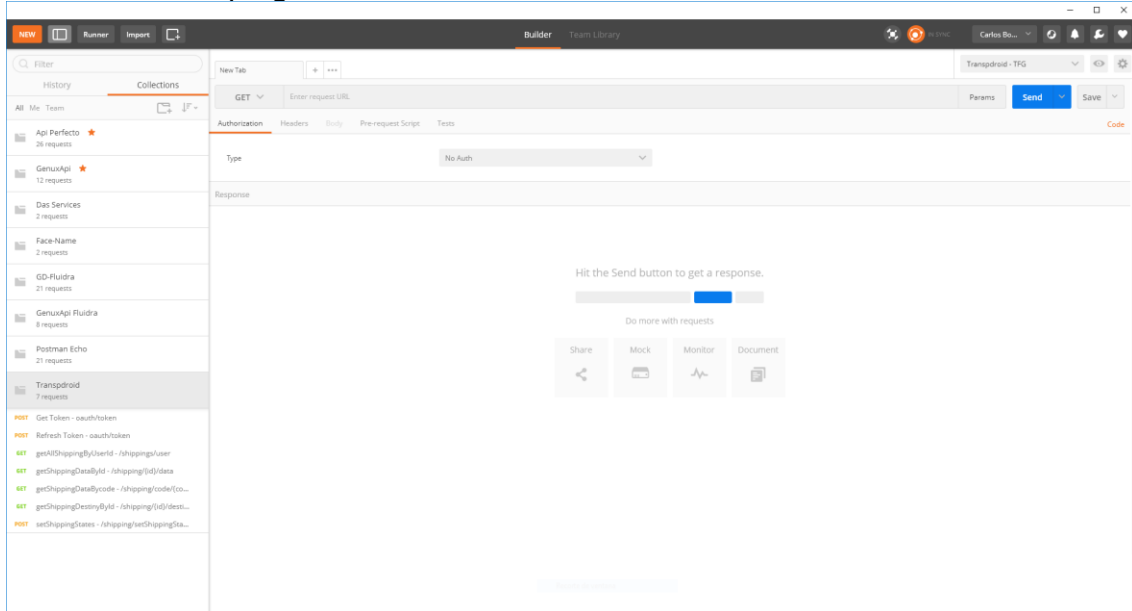
Il·lustració 33 – Android Studio carregar APK

2. Generant l'APK i habilitar "Orígens desconeguts" al mòbil es pot passar l' APK al mòbil i instal·larà directament des del mòbil.

8.8 Manual Postman.

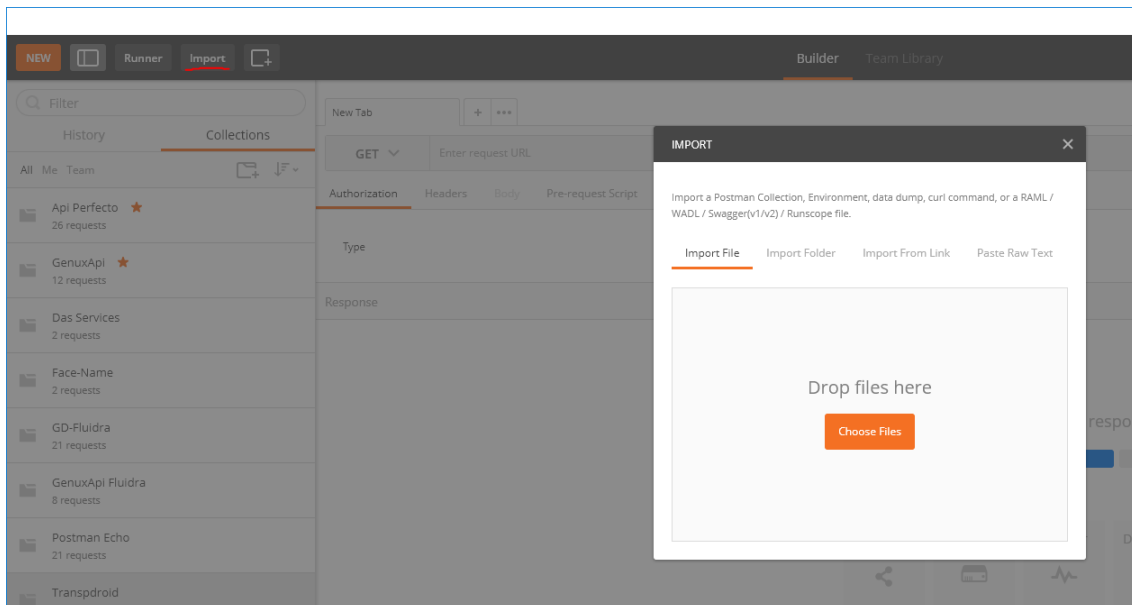
Aquest petit manual mostra com importar el Postman facilitat i fer crides a l'api per provar el correcte funcionament.

Postman és un plugin de Chrome.



Il·lustració 34 - Postman plugin de Chrome.

S'ha de fer clic en "Import", seleccionar la pestanya "Import File" i finalment al botó "Choose Files".



Il·lustració 35 - Importar col·lecció Postmant

Un cop tenim ja la col·lecció importada podem començar a fer proves. Les dades bàsiques ja vénen configurades en aquesta col·lecció.