

Trueckee: Plataforma d'intercanvi

Miquel Bosch Cortés

Resum— Aquest projecte consisteix en el desenvolupament d'una aplicació per a dispositius mòbils a partir de la generació d'una idea que s'ha batejat com a *Trueckee*. *Trueckee* és una plataforma per a l'intercanvi de productes i serveis, utilitzant la metodologia *swipe* tal i com fan aplicacions com *Tinder*. D'aquesta manera pretenem que el nostre usuari trobi, a la nostra plataforma, alguna cosa que no tenia en ment i l'aconsegueixi sense la necessitat de pagar amb diners, retornant al "*trueque*" com a alternativa. Es tracta d'una aplicació on l'usuari ha de pujar un producte o servei per poder ser intercanviat amb els altres, mitjançant un breu formulari. L'usuari en tot moment podrà donar de baixar el contingut que ha pujat. Per al desenvolupament de l'aplicació s'ha utilitzat el llenguatge de programació Java amb el *framework Android Studio*. Per a la part dels servidors s'utilitza una plataforma externa que està en el núvol anomenada *Backendless*, la qual ens permet guardar dades i autenticar als usuaris. El treball inclou, pas a pas, com aquesta idea ha anat agafant forma fins a obtenir una aplicació operativa i intuïtiva.

Paraules clau— Aplicació, Android, Android Studio, Java, *swipe*, intercanvi, bescanvi, banc de temps, productes, serveis, sorpresa.

Abstract— This project consists in the development of application software for mobile devices resulting from an idea that has been named "Trueckee". Trueckee is a goods and services exchange platform using the "swipe" method as in other application software, such as Tinder. In doing so, we aim at providing our users with a platform where they can find something random and get it without paying money, but reverting back to bartering as an alternative. To use this application software the user must upload a good or service to be exchanged for others, by means of filling a brief form. At all times, the user will be able to take down any uploaded material. This application software has been developed using Java programming language under Android Studio framework. As a server, we have relied on an external, cloud-based platform called Backendless, which provides support for data storage and user authentication. This paper includes step by step how this idea came into being until it became an intuitive application software.

Index Terms— Application, App, Android, Android Studio, Java, *swipe*, Exchange, product, service

1 INTRODUCCIÓ

Des de els inicis de l'història humana hem tingut de buscar la manera d'obtenir béns i serveis que no teníem i que necessitàvem per sobreviure. Una de les formes era cercar a una persona que tingués allò que necessitem i proposar-li un pacte, per exemple: "*Et canvio aquests pollastres si tu m'ajudes a caçar un mamut*". En aquestes circumstàncies, l'individu que té allò que necessitem, és on valora si el tracte és equitatiu o no, si creu que no, la persona interessada tindrà que proposar canvis fins a que les dos parts estiguin satisfetes. Això pot ser un procés lent, ja que fins que no trobis alguna cosa ideal per a l'intercanvi pot passar un temps considerable.

Per millorar aquest temps d'espera, es va inventar un

nou sistema de pagament, la moneda. En el que un objecte ja sigui de paper, pedra, or, etc. comença a cobrar un valor considerable, el qual reflexa el cost d'aconseguir un determinat producte o servei. Però això va portar uns efectes secundaris a la societat: la desigualtat econòmica.

Actualment, hi ha una tendència a l'alça en reaprofitar productes o coneixement (serveis), ja sigui a través de la compra de segona mà o amb els intercanvis[1].

El que busquem amb aquest projecte és donar facilitats per a que es produeixin els intercanvis i la gent pugui trobar d'una manera inesperada allò que no busca. Per tant, fem un canvi de mentalitat: ara ja no busquem allò que necessitem sinó que ens interessarem en allò que té un altre i que, possiblement, no teníem en ment. Aquest intercanvi només és produirà quan l'altra persona també estigui interessada en alguna cosa que nosaltres oferim.

Amb aquesta idea d'intercanvi inesperat, neix el projecte

E-mail de contacte: miquel.bosch@e-campus.uab.cat

Menció realitzada: *Enginyeria del Software*

Treball tutoritzat per: *Alicia Fornes (Dept. Ciències de la Computació)*
Curs 2016/17

final de grau: *Trueckee* i que queda documentat en aquest article que segueix la següent estructura:

En primer lloc analitzarem l'entorn actual de les aplicacions i mercat de segona mà. Més endavant, veurem els objectius que s'han volgut aconseguir i les diferents eines i metodologies que s'han emprat per aconseguir-los. Amb aquesta visió global, ja podrem aprofundir en les diferents etapes: de captació de requisits, de disseny, d'implementació i de test. Finalment veurem els resultats i les conclusions que s'han obtingut.

2 ESTAT DE LA NATURA

A dia d'avui, hi ha varies empreses que es dediquen al mercat de segona mà. La filosofia és simple, proporcionen una plataforma social on els usuaris poden introduir productes i els venen als usuaris interessats a canvi de diners. S'ha analitzat cada una d'aquestes empreses, tot i que algunes permeten intercanviar productes no ho tenen com a filosofia principal i no ho exploten.

Per altra banda, tenim portals webs que permeten intercanvis però es centren en canviar un determinat objecte X per un determinat objecte Y, cosa que limita molt l'intercanvi i pot fer avorrir als usuaris registrats.

2.1 Wallapop

És una plataforma que ha aconseguit posicionar-se a la cresta de la compra-venta en productes de segona mà i és una de les que permet fer intercanvis però no ho acaben de tenir ben plantejat:

- Quan hi ha la icona d'intercanvi marcat, a vegades, també hi ha la icona de preu no negociable marcada.
- Si veiem un producte que admet intercanvi i intentem fer una oferta, un dels primers camps que en surt en el diàleg és el d'introduir una quantitat en euros. Si no hi poses res, surt un diàleg d'error.

Funcionalitats d'èxit
Organitzar i filtrar els productes (per distància)
Seguir, contactar amb els usuaris és fàcil i ràpid per xat
Formulari per pujar productes és fàcil i ràpid
Valorar als usuaris
Guardar productes com a favorits i tenir un seguiment
Veure les visualitzacions totals d'un determinat anunci
Perfil usuari: vendes, valoració i articles actuals

Taula 1: Wallapop

2.2 Vibbo

Neix per a renovar el portal web *segundamano.es*. Té la mateixa doctrina que Wallapop, però categoritza els productes per tres grans grups: vehicles, immobles o coses que tenim per casa. D'on els dos primers si que tenen unes subcategories estandarditzades per la plataforma, el tercer donen més llibertat a l'usuari.

Funcionalitats d'èxit
En tot moment veus els filtres activats

Taula 2: Vibbo

Bàsicament té els que té Wallapop i aquest. Un aspecte negatiu que podem destacar és la revisió del contingut quan es penja un l'anunci. Pot donar més seguretat, però la pujada no es instantània com ho fa Wallapop. Són bastant semblants, aquest hi podem trobar més productes de cost elevat: cases, cotxes i a l'altra hi podem trobar serveis.

2.3 Time Republik

Aquesta plataforma web ja aplica una filosofia diferent. Ja no és un portal de compra-venta, sinó que ens ofereix un banc de temps on la gent registrada ofereix els seus serveis a canvi de temps. Traslladem el valor monetari, al valor de temps. Qui tingui més temps acumulat, més serveis podrà contractar als altres usuaris.

Funcionalitats d'èxit
Veure possibilitats de fer el servei a distància (ex: Skype)
Definir i buscar per habilitats
Premiar amb insígnies segons els serveis realitzats
Veure historial de temps guanyat i com s'ha gastat
Connectar amb LinkedIn

Taula 3: Time Republik







2.4 Altres


Tenim plataformes que es centren en un nínxol de mercat, com per exemple l'intercanvi de llibres o cotxes. És el cas de: BookMooh, Relibrea, etc. Però tenen unes funcionalitats similars a les que proporcionen les empreses comentades en els apartats anteriors.

3 OBJECTIUS I METODOLOGIA

Per poder aconseguir una plataforma amb les funcionalitats desitjades, s'han definit una sèrie d'objectius: uns de caràcter general i d'altres més específics.

3.1 Objectius generals

Objectiu	Prioritat
Desenvolupar una aplicació que permeti l'intercanvi de productes i/o serveis.	
Assolir el màxim possible dels requisits plantejats en el principi i els modificats o afegits ens el transcurs del projecte.	
Cuidar el disseny i la navegació de les diferents pantalles.	
Dissenyar una base de dades ordenada i eficient.	
Obtenir un codi comprensible, reutilitzable i fàcil de mantenir.	
Testejar cada mòdul per veure cada funcionalitat obté el resultat esperat.	

Llençar l'aplicació al mercat: <i>Google Play</i>	
--	---

Taula 4: Objectius generals

3.2 Objectius específics

	Crítics	Prioritaris	Secundaris
Disseny i lògica del taulell que ens mostrarà els anuncis		X	
Disseny i lògica que mostri els anuncis en la metodologia <i>Swipe</i>	X		
Formulari producte ràpid de fer	X		
Disseny i lògica del xat	X		
Establir un sistema de notificacions			X
Crear un registre històric dels anuncis ja tancats		X	
Crear una lògica per a l'usuari admin			X
Fer una aplicació que funcioni ens els dispositius mòbils: Android i iOS			X
Augmentar l'eficiència en la navegació de les diferents vistes	X		

Taula 5: Objectius específics

3.3 Metodologia

Si aquest treball fos en equip és seguiria SCRUM. Aquest treball és individual però també ha seguit una metodologia àgil basada en un procés incremental i iteratiu.

L'idea és dividir els objectius o requeriments més generals en objectius més específics (blocs) obtenint, així, una sèrie de tasques. La unió d'aquests ens permetrà aconseguir l'èxit del nostre treball.

En aquest tipus de metodologies, és té molt en compte les entregues i les opinions que en poden sortir de les diferents parts implicades. De cada etapa, iteració, s'ha obtingut un *feedback*, ja que s'ha mostrat l'aplicació a possibles usuaris finals, que ha permès valorar altres opcions no analitzades i valorar el canvi que implicaria en el projecte per implementar-ho o no.

Per una banda, tenim la part incremental on es busca la

producció de cada funcionalitat per separat. Així aconseguim que a cada iteració el nostre projecte vagi afegint funcionalitats.

Per altra banda, tenim la part d'iteració on s'han mirat que totes les funcionalitats fetes fins al moment satisfan el major nombre de requeriments establerts. També ens permet veure quines funcionalitats podem afegir o treure.

Per al projecte s'han escollit unes iteracions de 15 dies. A cada iteració s'ha necessitat escollir una o varies funcionalitats de la llista total de requisits a assolir (simulant el *Product Backlog* del SCRUM) ordenats per prioritat. Entre cada iteració s'ha dut a terme una reunió per parlar de les fortaleses i els possibles cavis de la feina realitzada. Al final de cada iteració s'ha verificat que tot estigués correcte i que el flux tingués el comportament desitjat.

4 PLANIFICACIÓ DEL PROJECTE

Aquest projecte va començar el 4 de febrer del 2017 amb la proposta ja acceptada i es finalitza el 27 de juny del 2017. Això són 4 mesos i 27 dies, un total de 143 dies de treball. En l'apartat 5 de l'annex hi ha el diagrama de Gantt.

El treball fi de grau suposa unes 300 hores de treball personal aportant un total de 12 crèdits, on $300/12 = 25$ hora/crèdit. Al estar realitzat per una sola persona, suposarà una carrega de 23 hores setmanals.

FASES	ACTIVITATS
Planificació del projecte	Contextualitzar el projecte
	Veure les necessitats
	Definir objectius del projecte
	Analitzar el sector
	Analitzar la documentació de sistemes similars
	Identificar actors
	Primera reunió
	Assignar un calendari
Captació de requisits	Entrevista amb possibles usuaris per captar requisits
	Reunir tots els requisits funcionals i no funcionals
	Definir objectius més específics
	Avaluar riscos i restriccions
Disseny	Reunió amb la tutora
	Disseny de l'arquitectura
	Disseny de la base de dades
	Avaluar possibles patrons

	Reunió
Implementació i test	Implementació de la base de dades
	Implementació de la codificació de les diferents funcionalitats
	Verificar el correcte funcionament de cada mòdul
	Aplicar modificacions
Presentació del sistema	Presentació del projecte a la tutora
	Realització de canvis si són necessaris
	Mostrar el sistema
Lliurament i exposició	Generar documentació
	Fer l'article
	Presentació del sistema davant el tribunal
	Entrega final del sistema i de la documentació

Taula 6: Fases del projecte













5 REQUISITS, ACTORS, CASOS D'ÚS

Tot projecte del software comença amb aquesta etapa. És important definir bé aquelles necessitats que s'han de cobrir amb el projecte, aquestes podran canviar però s'ha de construir una base sòlida per què el treball tingui sentit.










5.1 Requisits funcionals



Els requisits funcionals són aquells que ens defineixen què ha de fer la nostra aplicació. Es troben resumits i classificats per importància a la següent taula, indicant si s'han complert o no.

Requisits funcionals	
Essencials	Mostrar els productes i serveis disponibles actualment 
	Mostrar informació detallada quan l'usuari triï un determinat anunci 
	Augmentar una visita del anunci, si ja s'havia vist anteriorment no 
	Sumar 1 quan l'usuari premi el botó de favorit en un determinat anunci 
	Permetre la valoració del usuari 
	El detall del anunci ha d'incloure: una petita descripció del anunci, en quina categoria es troba, veure les visualitzacions totals i el nombre de favorits 
	Un botó que permeti inicialitzar un xat 
	Fer un sistema de visualització d'ofertes de tipus <i>Swipe</i> . 







La fitxa de pujada del producte ha de ser ràpida i intuïtiva. Ha de permetre pujar com a màxim 3 imatges i com a mínim 1. 
S'ha de mostrar una llista amb els xats inicialitzats 
El perfil d'usuari permetrà editar la informació personal 
Poder veure la llista d'anuncis que en algun moment has pujat a l'aplicació (estiguin o no actius) 
Poder veure la llista d'anuncis que t'han agradat (estiguin o no actius) 
Poder veure la llista de <i>matches</i> 
Retornar a la pantalla principal prement el logotip de la <i>toolbar</i> 
Permetre que l'usuari es registri 
Seleccionar una data de naixement a través de la visualització d'un calendari 
Només mostrar el taulell principal si no s'està registrat a l'aplicació 
Permetre autenticar-se i guardar aquest inici de sessió 
Registrar a la base de dades: els anuncis, usuaris i les relacions per complir les funcionalitats 

Taula 7: Requisits funcionals - essencial

Condicionals	Diferenciar entre producte o servei 
	Notificacions en el xat 
	Notificacions quan fem <i>match</i> 
	Editar l'anunci 
	Eliminar l'anunci 
	Editar l'usuari 
	Eliminar l'usuari 
	Canviar la contrasenya 
	Cercar productes o serveis concrets 

Animacions per cada pantalla	
Mostrar foto i valoració de l'usuari en l'anunci pujat	

Taula 8: Requisits funcionals - condicional

Opcionals	Canviar data de naixement	
	Consultar històric d'anuncis	
	Interfície d'usuari admin	
	Mostrar pantalles d'error: "no estàs connectat a la red Wi-fi", etc. Amb un disseny elegant i acord amb la resta de l'aplicació	
	Aplicar tècniques de machine learning que permetin mostrar per gustos dels usuaris	
	Eliminar xat o conversa	

Taula 9: Requisits funcionals - opcional

5.2 Requisits no funcionals

En els no funcionals estan relacionats amb les restriccions del problema, entorn i client. En aquest projecte s'han complert tots.

Aplicació ha de ser intuïtiva i fàcil d'utilitzar, per arribar a un major públic.
L'accés a base de dades i servidors no han de representar una gran càrrega per al dispositiu. Amb temps de resposta ràpids: 4 seg/màxim.
Les dades dels usuaris seran guardades de forma segura seguint la llei de protecció de dades.
Sistema que s'adapti als canvis.
S'ha de visualitzar el contingut de les vistes a una distància de 30-40 cm
L'aplicació només funcionarà en orientació vertical
L'aplicació funcionarà per els sistemes Android
Per instal·lar Android Studio: RAM de 2GB, espai de disc disponible de 1.5 GB, tenir instal·lat SDK i JDK

Taula 10: Requisits no funcionals

5.3 Actors

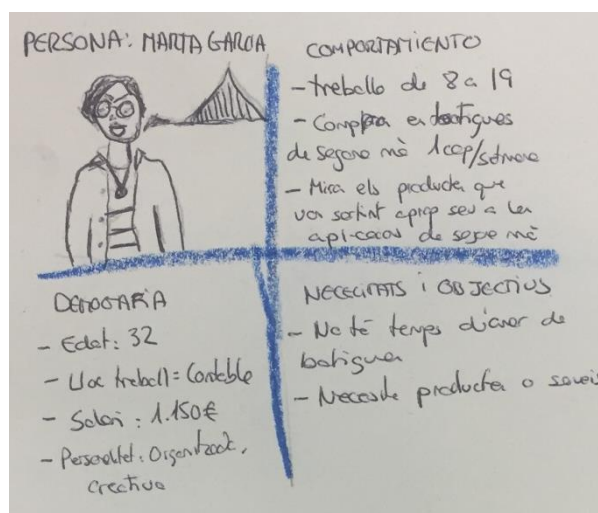
Són entitats externes que participen en algun escenari

dels casos d'ús plantejats en l'annex. Per aquest projecte s'ha considerat que hi ha els usuaris de la pròpia aplicació i l'usuari admin que ha de permetre gestionar les diferents irregularitats que puguin sorgir quan l'aplicació surti al mercat. Per aquest tram del projecte ens centrarem només en l'usuari principal. En un futur permetrem una sessió administrador que farà estadístiques dels usuaris registrats, dels anuncis més vistos, les categories on més accedeix la gent, etc. També gestionarà les denúncies dels anuncis, això ja es pot fer però ho hem de fer a través de la base de dades i del servidor. També podrà eliminar usuaris, anuncis, etc. En canvi, els usuaris, sense cap privilegi, són aquelles persones que es descarregaran l'aplicació i, estiguin registrades o no, interactuaran amb el sistema buscant la funcionalitat desitjada.

Per poder veure les necessitats, dissenyar, etc. s'ha seguit una tècnica en la que hem creat un personatge virtual amb un perfil concret i unes necessitats. A partir d'aquí hem anat construint el projecte.

6 DISSENY

Per al disseny s'ha tingut en compte varis factors: que sigui elegant però eficient, que permeti una navegació ràpida i intuïtiva. Per altra banda, s'ha dissenyat la base de dades permeten les relacions que els requeriments demanaven.



Il·lustració 1: Persona

6.1 Disseny base de dades

Pel que fa al disseny de la base de dades es va estar debatent que seria millor: si contractar i programar els nostres propis servidors, obtenint una completa llibertat per fer el que vulguem o cercar alguna empresa externa que proporciona serveis de hosting, aquestes tenen els seus propis servidors i els gestionen sota la seva filosofia (Mobile Backend as a Service). On és perd llibertat i control. Per aquest projecte, i per les seves condicions: 1 programador, poc temps per a tot el cicle de vida del projecte (naixement de l'idea, captació de requisits, dis-

seny, codificació i test) en uns escassos 4 mesos, s'ha decidit contractar el servei d'una empresa *de back-end* que ens proporcioni la manera de comunicar la nostra aplicació amb els seus serveis. En cas contrari, tindriem que fer el codi de la pròpia aplicació i el codi que permeti comunicar-nos amb el nostre servidor.

Escollida la metodologia que emprarem per a la base de dades, s'ha avaluat les diferents plataformes que ens prestin aquets serveis. Les més importants:

- a) Firebase: és una de les plataformes més populars, tenen milions d'usuaris. Va ser comprada per Google i ara és gestionada per ells. Ens proporciona:
 - Base de dades a temps real
 - Sistemes d'autenticació (Login)
 - Hosting d'arxius
 - Enviar notificacions a les aplicacions
 - Sistema de comprovació de correus
 - Estadístiques
- b) Backendless: és menys popular que Firebase, però proporciona uns serveis similars. Es diferencia per la forma en que gestionen el contingut. Per exemple: en aquesta plataforma la base de dades es mostra en forma de taules, que és més visual. A Firebase és mostra per un arxiu JSON.

Però el factor decisiu per acabar decidint *Backendless* és que tenim més llibertat a l'hora d'escollir els servidors. S'ha acordat que és començaria per un servidor totalment gratuït on les nostres dades s'allotjarien al servidor de Backendless (amb limitacions d'arxius, etc.). En el cas d'un gran acolliment d'usuaris i de dades, emigrariem als nostres propis servidors, comprant-los prèviament. A Firebase es guarden les dades a Google Cloud Storage. A l'annex es pot veure una comparativa de prestacions i serveis.

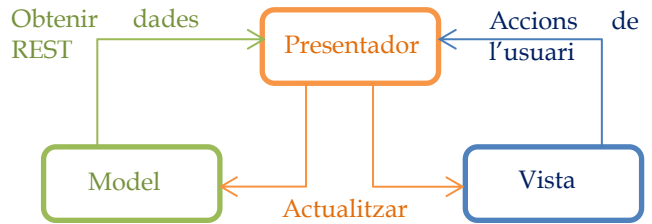
Tot i contractar un servei extern, no s'ha limitat en utilitzar els mètodes i serveis que proporciona la seva API. Sinó que, gràcies a *Retrofit*, hem pogut fer peticions de tipus REST obtenint llibertat per poder complir amb les necessitats de cada tasca.

Per acabar, s'ha dissenyat la base de dades. Tenint en compte totes les relacions que preservin les dades i la seva integritat. A l'annex queda reflexat el diagrama de relacions.

7 IMPLEMENTACIÓ

Un dels riscos més important d'aquest projecte és la falta d'experiència en programar amb JAVA per Android. Per assolir l'objectiu d'aconseguir un codi comprensible, fàcil de mantenir i escalable, s'ha buscat i analitzat diferents arquitectures. Finalment, s'ha seguit un estil arquitectònic model vista presentador, que s'assembla molt al model vista controlador de programació web.

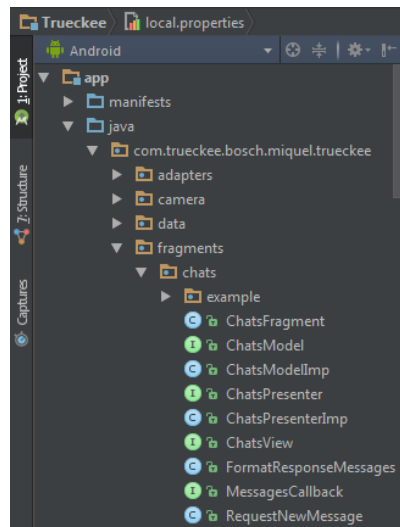
7.1 Arquitectura MVP



Il·lustració 2: MVP

Com tots els patrons i arquitectures de software, serveixen com a guia i es poden modificar segons les necessitats de cada moment.

Per començar a aplicar l'arquitectura MVP[2] s'ha separat la vista (tot el que veu l'usuari a través de la interfície: llistes, botons, etc.) del codi (tota la part lògica que permet el funcionament desitjat de l'aplicació). Per altra banda, s'ha obtingut les dades registrades a la base de dades mitjançant peticions REST que fa la part del Model i les envia al presentador, que és el que gestiona la part lògica. Podem veure com s'ha aplicat a la pantalla que ens permet xatejar.



Il·lustració 3: Exemple estructura MVP

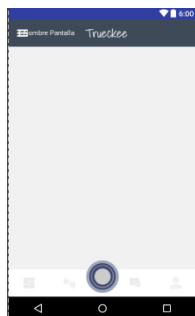
S'han seguit unes regles que són les següents:

- L'única responsabilitat de la vista és dibuixar l'interfície d'usuari sota les instruccions del presentador.
- La vista detecta totes les interaccions que fa l'usuari i ho envia al presentador.
- La vista mai és comunica amb el model.
- Es comuniquen per interfície.

7.2 App

Referent a la codificació, s'ha decidit utilitzar el *framework* d'Android Studio, amb el llenguatge de progra-

mació Java. El funcionament és el següent: quan s'obre l'aplicació s'inicia una *activity* que té varis contenidors: el principal, el que conté la *toolbar* i el que conté el menú ubicat al peu de la pantalla. El contenidor principal és un espai buit d'aquesta *activity* i permet introduir dins d'ell el contingut de les diferents pantalles a mostrar. Aquest contingut és mostra a través de *fragments*. Tota l'aplicació funciona amb una sola *activity* i varis *fragments* i s'anirà mostrant o amagant la *toolbar* o el *footer* en funció del disseny de cada pantalla.



II Il·lustració 4: Main Activity

A la il·lustració 5, podem veure com queda seccionada l'activitat principal. Per sobre de tot tenim la *toolbar*, el requadre gris del mig és el contenidor principal i per últim, al peu, tenim el menú.

Per no carregar molt el fil principal s'han creat fils secundaris que permeten executar taques en segon pla. Aquí s'ha controlat que cada fil retornés els resultats abans, en el moment precís, ja que si un fil conté un resultat que necessitem i, per alguna determinada condició, s'executa abans el fil que necessita d'aquestes dades causarà un error en l'aplicació. Per tant, s'ha pensat que a l'obrir l'activitat principal simultàniament executi les tasques adients per obtenir la informació de la base de dades per mostrar en segon pla. Aquestes consultes són: obtenir les dades dels anuncis i dels usuaris registrats.

Utilitzar patrons ha ajudat i simplificat la feina. El més destacable ha estat l'ús del Singleton[3], patró que ens permet crear una sola instància de l'objecte desitjat. En aquesta aplicació s'ha creat una instància del usuari *loggejat*, per així poder accedir a ell a totes les classes de l'aplicació i s'ha creat un altre per a conèixer la ubicació actual de l'usuari *loggejat*. Un dels altres patrons és l'*Adapter*, que permet mostrar el contingut en unes determinades vistes.

7.3 Punts crítics en el desenvolupament

En l'apartat número 8 d'aquest article, és podrà veure un resum de de les pantalles amb una petita explicació de les seves funcionalitats. Aquí es mostra les parts més conflictives en quant a codificació.

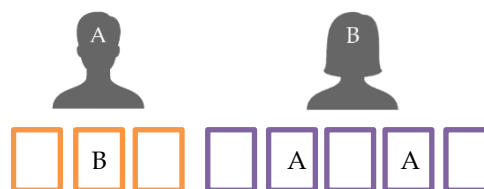
a) En la lògica del *swipe*:



II Il·lustració 5: Pantalla Match

El disseny ha exigut que de la llista total d'anuncis registrats a la base de dades es mostrin aquells que: estiguin actius, no siguin els del propi usuari i no s'hagin vist anteriorment. Un cop tenim aquesta llista, s'han d'anar mostrant els anuncis.

Per cada anunci, si es rebutja no es fa res. En canvi, si es marca que si (*like*) l'aplicació ha de mirar si existeix una entrada a la taula relació (usuari1, usuari2, true) on el true ens indica que ens ha agradat algun anunci de l'altre usuari.. Per exemple: l'usuari A té 3 anuncis penjats i li agraden 2 dels 5 anuncis de l'usuari B. Aquí l'aplicació buscarà una entrada a la taula *match* amb el format anterior. L'usuari B encara no ha entrat ni a mirat els possibles anuncis per a fer *match*. Per tant, no hi ha cap entrada a la taula *match*. Suposem que entra i visualitza els 3 anuncis de l'usuari A i posa *like* només a un d'ells. En aquesta iteració si que trobarà dos entrades a la taula amb la relació (A,B,true) , i que són els dos anuncis que li ha agradat anteriorment a l'usuari A.



II Il·lustració 6: Exemple usuaris

En aquest punt, s'afegeix una nova conversa a la llista que permetrà inicialitzar o continuar el xat. La dificultat ha estat en treure de la llista dels anuncis ja vistos per un usuari, que s'ha resolt com es mostra en l'annex. I també quan hi ha *match*, s'han buscat tots els anuncis que s'han fet *match* per els dos usuaris i es mostren en el xat en forma de llista. No és una relació 1 a 1.

- b) En el número de visualitzacions dels anuncis: Tal i com s'ha definit, sumarem 1 al nombre total de visualitzacions d'aquell anunci. Però si el tornem a mirar des de la mateixa sessió d'usuari, no és contarà. Només és sumarà si la sessió està inicialitzada. Per poder portar aquesta funcionalitat, s'ha creat una taula amb el format (usuari, anunci).

8 TEST

Paral·lelament, amb la codificació, s'ha anat testejant l'aplicació amb l'objectiu de trobar errors de disseny o d'execució.

S'han dut a terme tècniques de caixa negra que ens han permès veure quina sortida s'obté si introduïm un determinat input. No es tenen en compte els mecanismes interns de l'aplicació. Com a exemple, s'ha utilitzat un valor límit, 0, a l'hora de sumar o restar un anunci favorit. I s'ha verificat que mai pugui ser un nombre negatiu, comprovant els valors frontera, en aquest cas seria el -1. I comprovant valors de flux normal. L'usuari pot marcar com a favorit el anunci que li agradi, i desmarcar-lo en un futur. A les proves de test, va donar algun cop negatiu si es polsava molts cops seguits el botó. Això és degut a que: quan es prem el botó de *like*, automàticament es fa una crida a la base de dades i es mira si ja existeix una entrada a la relació anunci - usuari que li agrada (taula favorits). Si existeix la relació, al prémer de nou, el que farà és eliminar l'entrada. Si no existeix la crearà. Això comporta un temps d'assimilació. Per evitar l'error s'han posat límits, on per sota de 0 no sigui possible estar-hi.

A fi de realitzar un test més exhaustiu, per cada mòdul realitzat s'ha plantejat tots els seus escenaris possibles i s'ha analitzat el comportament introduint varis valors com a entrada. Per exemple:

Els anuncis del taulell es mostren segons la distància que hi ha entre la localització actual de l'usuari i la ubicació del anunci. L'escenari és clar: càlcul de les distàncies entre la coordenada del usuari i la coordenada del anunci. S'han plantejat diferents escenaris i s'ha decidit com ha de comportar-se l'aplicació en cas de que un d'aquets escenaris es produeixi. Podem veure a l'annex la taula que recull les dades d'aquest test. Es parteix d'un escenari ideal, en el que lluitarem perquè passi el major nombre de cops, aquest és el d'obtenir les coordenades del usuari i les del anunci sense cap problema i correctament. Això permetrà calcular una distància. En el cas de que no hi hagi una coordenada: si és la coordenada del anunci, se se n'assignarà una estàndard (hi ha gent que no vol localitzar el anunci, no ha de ser un error). Però si el problema està en la coordenada de l'usuari tindrem que recordar-li si te els permisos de localització activats i si continua persistint el problema, també se n'assignarà una. Si per alguna raó el sistema cau en una coordenada errònia (que no sigui del tipus double, etc.) el sistema mostrarà un error de GPS o de connexió.

9 EINES UTILITZADES

Ja se n'ha anat parlat al llarg d'aquest article, però en aquest apartat s'han resumit les eines utilitzades durant el transcurs projecte.

9.1 Android Studio

Android Studio és l'entorn utilitzat per desenvolupar aplicacions en Android sota el llenguatge de programació Java[4]. És una eina gratuïta que ens ajuda a l'hora de programar, hi ha moltes llibreries a la xarxa que ens faciliten la feina. Algunes de les que s'ha utilitzat: *Retrofit* (per poder fer peticions REST), *Glide* (per carregar imatges de d'una manera eficient i aplicant transformacions, si cal), la de *Google Play Services* (per poder mostrar i interactuar amb mapes), etc. Aquest *framework*, ha anat reemplaçant el *pluggin* ADT (*Android Developer Tools*) que es feia servir abans amb Eclipse per poder desenvolupar programes *Android*. És un IDE oficial de Google amb la col·laboració de JetBrains [7].

9.2 Backendless

És una eina de backend que ens proporciona múltiples serveis per poder comunicar la nostra aplicació amb la base de dades. També ens gestiona els usuaris registrats, localitzacions, emmagatzema els arxius multimèdia i les dades.

9.3 Git

Per al control de versions s'ha utilitzat *Git*. *Android Studio* ens facilita un *pluggin* que permet fer les funcions *Git* sense sortir del programa, introduint les comandes en una consola.



Il·lustració 7: Git

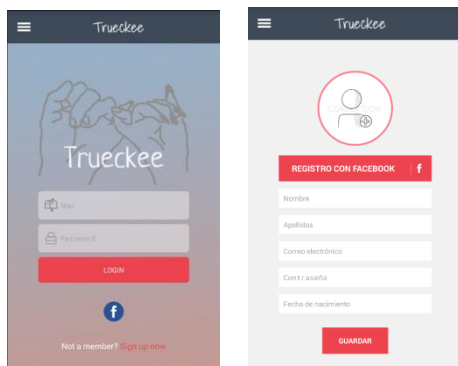
Al no ser un projecte en equip es redueixen les funcionalitats i avantatges d'utilitzar *Git*, ja que no hi ha col·lisions (conflictes d'actualització del document) i les duplicitats o errors que es poden produir amb la simultaneïtat d'arxius en execució.

No obstant s'ha utilitzat com a eina de control de versions que ens ha garantit tenir còpies de seguretat per si teníem que tornar a una versió anterior. S'ha creat una branca principal (Màster) que és la que conté la versió més estable i més recent del projecte. Per cada iteració o funcionalitat rellevant s'ha obert una nova branca per treballar paral·lelament. Un cop s'ha verificat que té el comportament ideal s'ha acoblat amb la resta del projecte, i s'ha tancat la branca.

10 RESULTATS

Trueckee ha permès posar en pràctica els coneixements que s'han assolit al llarg de la carrera d'enginyeria informàtica aconseguint desenvolupar una aplicació per a dispositius Android totalment operativa. A continuació és mostren algunes pantalles:

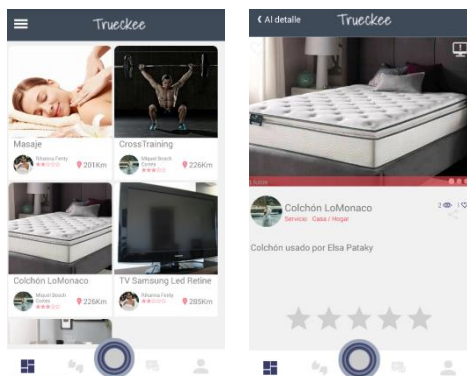
10.1 Login i registre



Il·lustració 9: Pantalla Login i registre

Tal i com marquen els requeriments, l'usuari si no està registrat, podrà veure el taulell d'anuncis però no se li permetrà interactuar. Si es vol tenir aquest privilegi tindrà que iniciar sessió o registrar-se. Per al registre cal que tots els camps estiguin complerts, la data de naixement sigui més petita o igual a la data d'avui, i s'ha de penjar una foto de perfil obligatòriament.

10.2 Pantalla principal i detall de l'anunci

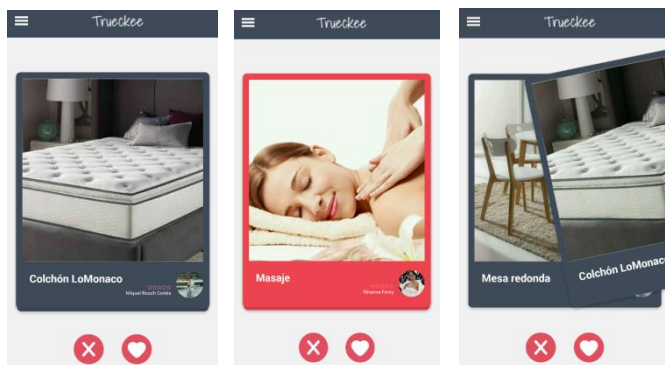


Il·lustració 10: Pantalla principal

Aquestes pantalles són les que poden veure els usuaris sense registrar. La pantalla principal és la de l'esquerra, on es mostren, en un contenidor, tots els anuncis disponibles. Així a simple vista, ens dóna la informació del nom del anunci, com està valorat l'usuari i la distància. Un cop premem a sobre un d'ells, se'ns obre la pantalla de la dreta amb informació més detallada del anunci. En aquesta pantalla podem guardar l'anunci com a favorit, veure la seva localització al mapa, denunciar-lo i veure el mapa. Es pot canviar de foto arrastrant-la cap a la dreta o

esquerra (si n'hi ha més d'una).

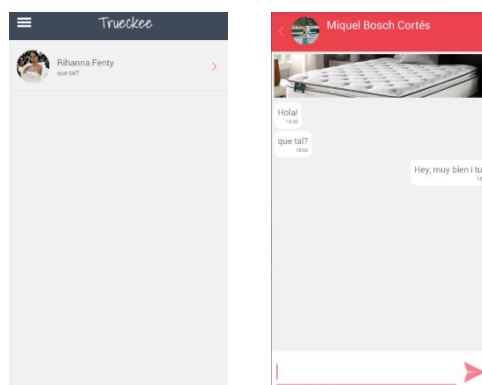
10.3 Pantalles Match



Il·lustració 8: Pantalla Swipe

Tal i com es pot observar en la il·lustració 9, el disseny de les targetes diferencia de si es producte o de si es servei aplicant un color diferent. El funcionament és fàcil, cal anar canviant de targeta movent-la cap a la dreta si s'està interessat en aquell producte/servei o cap a l'esquerra si no se'n està. En el cas de que a l'altre usuari li agradi algun dels nostres anuncis se'ns obrirà una conversa.

10.4 Pantalla Xat

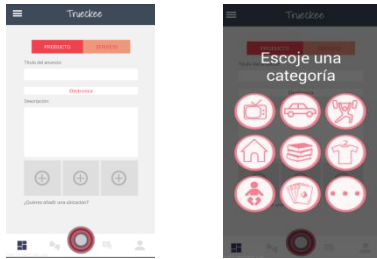


Il·lustració 11: Xat

Si premem la pestanya de conversacions se'ns mostra la pantalla de l'esquerra que és una llista dels xats disponibles. En aquesta imatge només hi ha un xat disponible. La pantalla ens mostra una petita imatge circular de l'usuari amb el que parles, el seu nom i cognom i l'últim missatge enviat o rebut.

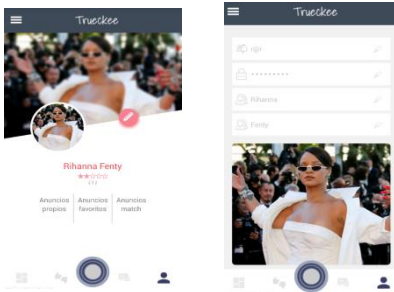
Un cop premem alguna de les conversacions s'obra automàticament la pantalla de xat, la de l'esquerra. Aquesta ens mostra a dalt una llista d'imatges rectangulars dels productes o serveis que t'han agradat de l'altra persona (que has arrastrat cap a la dreta en la pantalla de Match). A sota, tenim el container que permet el xat.

10.5 Pantalla formulari pujada producte



Il·lustració 12: Pantalla formulari

L'idea principal d'aquesta pantalla és fer un petit formulari que demani la informació bàsica: Classificar si es producte o servei, títol del anunci, una breu descripció, escollir categoria, posicionar en el mapa i penjar com a mínim una imatge i com a màxim tres.



Il·lustració 13: Pantalla perfil

Per últim, podem veure la pantalla de perfil d'usuari. Aquí és on pots modificar el teu perfil i mirar llistes de: anuncis favorits, anuncis propis (eliminar-ne algun) o llista de *match*.

11 CONCLUSIONS I LÍNIES FUTURES

11.1 Conclusions

Desenvolupar una aplicació per a dispositius Android que sigui capaç d'interconnectar a dos parts mitjançant un anunci que els interessi, ha estat el principal objectiu. Des de el naixement de la idea he estat molt motivat i il·lusionat amb aquest projecte, cosa que m'ha fet involucrar i invertir-hi molt del meu temps. Aquesta motivació ha estat positiva en molt sentits, he disfrutat del treball, però també a comportat aspectes no tant positius. Un d'ells és que sempre he tingut en ment que l'aplicació fos elegant, amb colors atractius als ulls dels potencials usuaris, una estètica cuidada, una navegació ràpida, eficient, que no doni errors, etc. cosa que ha fet que hagi cuidat molt tots aquets aspectes i m'hagin alentit el projecte en global. És a dir, he perdut la noció del temps intentant millorar aspectes que no estan relacionats en l'enginyeria informàtica (dissenyant botons, tipografia, etc) Fent que quedi una aplicació agradable però implicat a que hagi de invertir més temps per fer les tasques que realment són d'enginyeria. Aquí és on la meua tutora m'ha ajudat a reposicionar-me i a centrar-me en els objectius informàtics.

En termes de metodologia o de creació d'un projecte, he

pogut veure els elements que s'han d'analitzar i tenir en compte per dur-lo a terme. He après a organitzar-me, a preveure un calendari de les coses que s'han de fer. Fer un projecte no es només aportar una idea i desenvolupar-la. Implica molta feina d'investigació, un cop tens tots els pilars sòlids i veus les necessitats que pots cobrir has de pensar un bon disseny, no només estètic, sinó de l'arquitectura, de les eines que utilitzaràs, de la base de dades, etc. Cal remarcar que cal tenir molt ben definit el projecte abans de començar a codificar, això m'ha ajudat molt. He pogut codificar aplicant una lògica analitzada amb anterioritat i, com ja he comentat al article, he buscat obtenir un codi eficient, fàcil d'interpretar i de mantenir.

Durant el projecte m'han sorgit varis problemes, un d'ells va ser a la pantalla de *Match*. Hi havia moltes condicions a complir i moltes decisions a prendre. Una d'elles la comentaré en el següent apartat. Per fer aquesta presa de decisions, m'he tingut de posar en la pell de l'usuari i veure que seria lo més adient i lo més còmode per a ell, contrastant-ho amb analitzar com ho fa la competència.

La satisfacció de la feina feta és enorme, s'han complert tots els objectius crítics i dels no crítics s'han deixat els que menys afecten a l'idea en si.

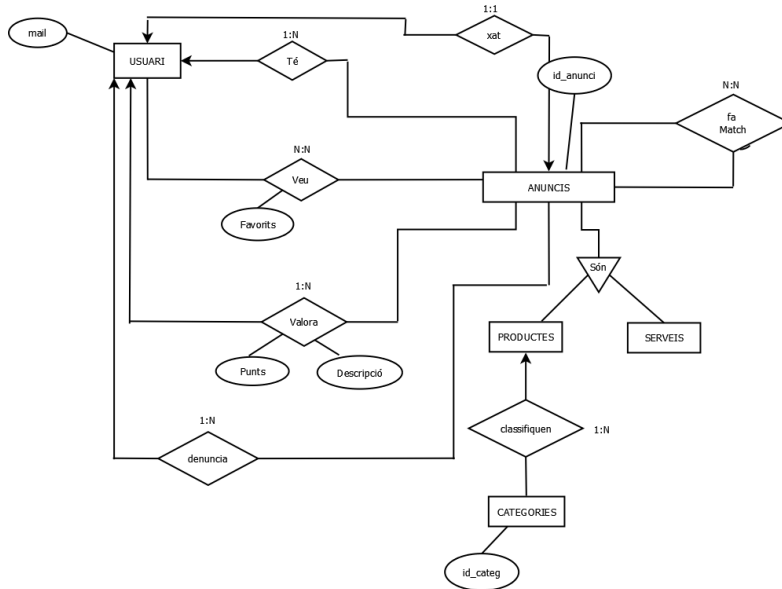
11.2 Linies futures

A dia d'avui, afirmo que la versió està operativa. Però per el poc temps que teníem (hem dissenyat i implementat tot un projecte des de zero i en tres mesos) ha fet que hagués de prendre decisions ràpides. Una de les decisions que estic menys content: quan a un usuari desplaça cap a la dreta la targeta del anunci, és a dir li agrada, a la pantalla de *Match* i a l'altre usuari, posteriorment, li agrada també algun dels anuncis de l'usuari, s'obre automàticament la pantalla de conversacions. Aquest model el caviaria i jugaria amb alguna pantalla d'animació que tingués un enllaç directe a la pantalla de converses (l'usuari tries si marxar de la pantalla actual de *Match* o no) o posar algun tipus de notificació.

Per altra banda, trobo que el mercat de segona mà està de moda i que hi ha poques companyies que incentivin l'intercanvi. Nosaltres ho fem amb l'intercanvi sorpresa. Abans de fer el projecte i durant, he anat preguntant a la gent que els hi sembla l'idea i sempre he tingut comentaris positius i motivadors: "si ho fas jo invertiré", "és molt bona idea aplicar aquesta metodologia del *Tinder* per a l'intercanvi", etc. Amb això vull dir que sembla una bona idea per la resta.

Però al no poder llençar-la al mercat m'ha quedat una mica el sabor agrí-dolç. Crec que he de corregir aquestes decisions abans de llençar-la al mercat, ja que podrien crear una mala imatge de l'aplicació o del desenvolupador si un usuari experimenta.

A2. DIAGRAMA ENTITAT- RELACIÓ



II Iustració 16: Diagrama Entitat relació

A3. COMPARATIVA BACKENDLESS AMB FIREBASE

PRODUCT	DEVELOPERS	BLOG	PRICING	SUPPORT	CUSTOMERS	SERVICES	COMPANY	Log in	Sign up
CLOUD PRICING	FREE	CLOUD 9	CLOUD 99	ENTERPRISE					
	\$0.00 /MONTH	\$25.00 /MONTH	\$99.00 /MONTH	CONTACT US					
API calls/minute	300	600	1200	unlimited					
Development team size	1 developer	unlimited	unlimited	unlimited					
Versions	1	5	5	unlimited					
Custom Security Roles	1	3	5	unlimited					
Data tables	20	100	200	unlimited					
Data objects in a table	20,000	100,000	400,000	unlimited					
Geopoints	20,000	100,000	400,000	unlimited					
Geofences	1	3	3	unlimited					
File storage	5 GB	10 GB	20 GB	unlimited					
Objects in cache	30	50	100	unlimited					
Pub/sub messages	100,000	500,000	1,000,000	unlimited					
Push notifications	100,000	500,000	1,000,000	unlimited					
Cloud Code scripts	3	5	10	unlimited					
Cloud Code deployment size	1 MB	2 MB	3 MB	unlimited					
Cloud Code execution time	5 seconds	5 seconds	10 seconds	unlimited					
Cloud Code timer frequency	60 seconds	60 seconds	5 seconds	unlimited					
Limits can be increased	✖	✔	✔	✖					
	REGISTER	REGISTER	REGISTER	CONTACT US					

	SPARK Free <small>Generous limits for hobbyists</small>	FLAME \$25 per month <small>Predictable pricing for growing apps</small>	BLAZE Play as you go <small>Commodity pricing for apps at scale</small>
Included Free Analytics, App Indexing, Authentication, Cloud Messaging, Crash Reporting, Dynamic Links, In-app, Notifications & Remote Config	✔	✔	✔
Realtime Database	100 Simultaneous connections	Unlimited ¹	Unlimited ¹
	1 GB stored	2.5 GB	\$5/GB/month
	10 GB/month	20 GB/month	\$1/GB
	Automated backups	✖	✔
Storage	5 GB stored	50 GB	\$0.026/GB ²
	1 GB/day	50 GB/month	\$0.12/GB ²
	20,000/day	100,000/day	\$0.10/thousand ²
	Download operations	50,000/day	250,000/day
	50,000/day	250,000/day	\$0.01/thousand ²
Functions	125,000/month	2,000,000/month	\$0.40/million ³
	40,000/month	400,000/month	\$0.0025/thousand ³
	40,000/month	200,000/month	\$0.01/thousand ³
	Google-only ³	5GB/month	\$0.12/GB ³
Hosting	1 GB stored	10 GB	\$0.026/GB
	10 GB/month	50 GB/month	\$0.15/GB
	Custom domain hosting & SSL	✔	✔
Test Lab	Daily quota or hourly rate	15 tests/day ⁴	15 tests/day ⁴
Google Cloud Platform	Use BigQuery & other tools ⁵	✖	✔
	START NOW	SELECT PLAN	SELECT PLAN

II Iustració 17: Comparativa Backendless i Firebase

A4. PROVA DE CAIXA NEGRA UTILITZANT CASOS DÚS

Escenari: Calcular distàncies								
ID	Condicció	Coordenades		Coordenades		Distància		Resultat esperat
		x	y	x	y	x	y	
1	Coordenades vàlides	41,61142	2,01386	41,40044	2,15014	0,21099	-0,13628	Actualització de la distància realitzat amb èxit
2	No s'obté la coordenada de l'ubicació actual de l'usuari	-	-	41,40044	2,15014	-	-	S'assigna una ubicació estàndar per tothom, la coordenada (0,0) o es preguntarà per els permisos de localització
3	No s'ha entrat la localització del anunci	41,61142	2,01386	-	-	-	-	També utilitzem una coordenada estàndar
4	No es detecta cap coordenada	-	-	-	-	-	-	S'utilitzen les coordenades estandarts comentades anteriorment
5	Alguna de les coordenades es erronia	-	-	-	-	-	-	Mostrar error de GPS o connexió

Taula 11: Prova de caixa negra. Escenari: coordenades vàlides

A4. CODIFICACIÓ

```

@Override
public void getRelationsMatch() {
    mMatchModel.getMatchRelations((formatResponseRelationMatch) -> {
        if(formatResponseRelationMatch.getTotalObjects() != 0){
            mAdsWithoutWatched = new ArrayList<Ads>();
            Log.d("onRelationLoad", "" + "Initialsize: " + mAdsWithoutUser.size());
            Log.d("onRelationLoad", " ");

            for (int j=0; j<formatResponseRelationMatch.getData().size(); j++){
                for (int i=0; i<mAdsWithoutUser.size(); i++){
                    if(mAdsWithoutUser.get(i).getObjectId().equals(formatResponseRelationMatch.getData().get(j).getAddId().getObjectId())){
                        //mAdsWithoutUser.remove(i) = null;
                        mAdsWithoutUser.remove(i);
                        Log.d("onRelationLoad", "" + "ListActualitzat: " + mAdsWithoutUser.size());
                    }
                }
            }

            mAdsWithoutWatched = new ArrayList<Ads>(mAdsWithoutUser);
        }
    });
}
else {
    mAdsWithoutWatched = new ArrayList<Ads>(mAdsWithoutUser);
}
}

```

Il·lustració 18: Codi llista anuncis a mostrar

```

@Override
public void addOneWatch() {
    if(isLoggedIn()){
        //obtenir la llista
        //https://api.backendless.com/v1/data/adsWatched?relationsDepth=1&where=userId.objectId%3D%27C3797D22-7BC1-
        mDetailDashboardModel.getIsAdVisitedForUser(mAd.getObjectId(), UserSingleton.Instance().getUser().getObjectId());
        int totalObjects = formatResponseRelationadUser.getTotalObjects();
        Log.d("addOneWatch - onSuccess", "Success" + totalObjects);
        if(totalObjects == 0){
            int numViews = mAd.getAdViews() + 1;
            int numFav = mAd.getAdFavorites();
            mDetailDashboardModel.updateTableAds(mAd.getObjectId(), numViews, numFav, (ad) -> {
                Log.d("addOneWatch - onSuccess", "Success" );
                mAd = ad;
                mDetailDashboardModel.registerRelation(mAd, UserSingleton.Instance().getUser(), true);
                mDetailDashboardFragment.sumOneInViewsAd();
            });
        }
    }
}

```

Il·lustració 19: Afegir una vista més a l'anunci

