



WearBank: la nova banca adaptada per smartwatches

Nom Estudiant: Josep M^a Viladegut Pelegrí

Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils

Nom Consultor/a: Eduard Martin Lineros

Professor/a responsable de l'assignatura: Carles Garrigues Olivella

08/06/2016

© Josep Maria Viladegut Pelegrí
Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>WearBank: La nova banca adaptada per SmartWatch</i>
Nom de l'autor:	<i>Josep M^a Viladegut Pelegrí</i>
Nom del consultor/a:	<i>Eduard Martin Lineros</i>
Nom del PRA:	<i>Carles Garrigues Olivella</i>
Data de lliurament (mm/aaaa):	<i>06/2016</i>
Titulació o programa:	<i>Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Android, SmartWatch, Banca digital</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>El TFM consisteix en desenvolupar una aplicació Android per a smartwatch, simulant la que podria ser una aplicació bancària en la qual, un usuari pot consultar tota la seva informació del banc.</p> <p>Aquesta informació contindrà:</p> <ul style="list-style-type: none">• Els diferents comptes actius• Els moviments i el detall d'aquests fets en cada compte bancari• El perfil de l'usuari en el banc (nom, cognom, DNI, correu, etc.) <p>També es permetrà simular accions com, per exemple, deixar un compte amb descobert (números negatius), el qual activarà la següent tasca que serà la d'enviar una notificació push des del servidor al client per indicar-li el problema en aquell compte.</p> <p>Les accions comentades solament es duren a terme quan l'usuari hagi superat el logueig inicial de l'aplicació que serà el que s'encarregarà d'identificar el client i saber quines són les seves dades.</p> <p>Mentre no es superi la identificació de l'usuari, les accions disponibles es reduiran deixant solament l'acció de consultar les oficines més pròximes a la posició de l'usuari, poden triar una per rebre el guiat fins aquesta, i també la de poder trucar a un servei d'atenció al client.</p>	

Abstract (in English, 250 words or less):

This FMP (Final Master Project) is about the development of an Android application for smartwatches, simulating what could be a banking application which any user will be able to check all their own information of the bank.

These information will contain:

- The different active accounts
- The movements and detail of these facts in each account
- User profile in the bank (name, surname, ID number, email, etc.)

It will also allow to simulate actions such as letting a bank account to be overdrawn (negative numbers), which will activate the next task that will be sending a push notification from the server to the client to indicating him the problem in the account.

The mentioned actions could only be carried out when the user get through the initial login of the application that will be the responsible for identifying the client and know their own information.

While the identification of the user is not exceeded, the available actions will be reduced to consult the user's nearest offices, having the option to choose one of them and getting the guidance to there. The user will be able to call a customer service number too.

Índex

1. Introducció	8
1.1. Context i justificació del Treball.....	8
1.2. Objectius del Treball.....	9
1.3. Enfocament i mètode seguit.....	10
1.4. Planificació del Treball	11
1.5. Breu sumari de productes obtinguts.....	13
1.6. Breu descripció dels altres capítols de la memòria	13
2. Disseny	14
2.1. Disseny centrat en l'usuari	14
2.1.1. Usuaris i context d'ús.....	14
2.1.2. Disseny conceptual.....	17
2.1.3. Prototipat	18
2.1.3.1. Dispositiu mòbil	18
2.1.3.2. Smartwatch	23
2.1.4. Avaluació	26
2.2. Diagrama de flux	27
2.3. Definició dels casos d'ús	30
2.3.1. Especificació dels cassos d'ús.....	32
2.3.1.1. Consultar oficines properes.....	32
2.3.1.2. Trucar al telèfon d'assistència	33
2.3.1.3. Autenticació d'un usuari	34
2.3.1.4. Consultar els comptes i les targetes	35
2.3.1.5. Consultar els moviments d'un compte o d'una targeta	36
2.3.1.6. Consultar els detalls d'un moviment.....	37
2.4. Disseny de l'arquitectura	38
2.4.1. Diagrama UML del disseny de la base de dades.....	38
2.4.2. Diagrama UML del disseny d'entitats i classes	40
2.4.3. Diagrama de l'arquitectura del sistema	42
3. Android	43
3.1. Què és Android?	43
3.2. Historial de versions	44
3.3. Arquitectura.....	45
3.3.1. Aplicacions	45
3.3.2. Entorn de l'aplicació.....	45
3.3.3. Biblioteques	45
3.3.4. Runtime d'Android	46
3.3.5. Nucli Linux	46
3.4. Android Wear	46

4. Implementació	47
4.1. Desenvolupament de l'aplicació	47
4.1.1. Ús de <i>SharedPreferences</i>	47
4.1.2. Google API Client i Wearable API	50
4.1.2.1. <i>LocationService.API</i>	53
4.1.3. Llibreria <i>EventBus</i>	55
4.1.4. Backend <i>Backendless</i>	56
4.2. Proves de l'aplicació	57
5. Aplicació final	58
5.1. Aplicació "Mobile"	58
5.1.1. Pantalla inicial	58
5.1.2. Pantalla d'oficines	59
5.1.3. Pantalla d'identificació	62
5.1.4. Pantalla de la zona privada	62
5.1.5. Pantalla de targetes	65
5.1.6. Pantalla de moviments	66
5.1.7. Pantalla de detall d'un moviment	67
5.1.8. Pantalla d'accions del rellotge	69
5.2. Aplicació "Wear"	71
5.2.1. Pantalla inicial	71
5.2.2. Pantalla principal	74
5.2.3. Pantalla d'oficines	74
5.2.4. Pantalla de targetes	76
5.2.5. Pantalla de moviments	76
5.2.6. Pantalla de detall d'un moviment	77
5.2.7. Notificacions	77
6. Manuals de l'aplicació	79
6.1. Importar el projecte a Android Studio	79
6.1.1. Instal·lació d'Android Studio	79
6.1.2. Importar el projecte	80
6.2. Executar l'aplicació	82
6.3. Credencials d'usuari	82
7. Conclusions	83
8. Glossari	84
9. Bibliografia	85
10. Annexos	87
10.1. Creació i execució de l'aplicació en un emulador	87

Llista de figures

Figura 1: Pantalla inicial	18
Figura 2: Trucada d'atenció al client.....	18
Figura 3: Pantalla amb la llista d'oficines.....	19
Figura 4: Pantalla de GPS fins l'oficina	19
Figura 5: Pantalla d'identificació	20
Figura 6: Pantalla principal de zona clients	20
Figura 7: Llista de targetes.....	21
Figura 8: Llista de moviments d'una targeta.....	21
Figura 9: Detall d'un moviment.....	22
Figura 10: Pantalla d'error de connexió.....	23
Figura 11: Pantalla d'error de "login"	23
Figura 12: Alerta "login".....	23
Figura 13: Menú sense credencials.....	24
Figura 14: Llista d'oficines	24
Figura 15: Menú principal	24
Figura 16: Llista de targetes	24
Figura 17: Llista de moviments.....	25
Figura 18: Detall d'un moviment.....	25
Figura 19: Diagrama de flux de l'aplicació mòbil	28
Figura 20: Diagrama de flux de l'aplicació del rellotge	29
Figura 21: Diagrama de casos d'ús de l'aplicació mòbil	31
Figura 22: Diagrama de casos d'ús de l'aplicació smartwatch	31
Figura 23: Diagrama de classes.....	41
Figura 24: Diagrama de l'arquitectura del sistema	42
Figura 25: Versions d'Android	44
Figura 26: Arquitectura Android.....	45
Figura 27: Estructura "clau-valor"	48
Figura 28: Arxiu de <i>SharedPreferences</i> creat per l'aplicació.....	49
Figura 29: Contingut del fitxer creat per les <i>SharedPreferences</i>	50
Figura 30: <i>Wearable</i> API	53
Figura 31: Funcionament de la llibreria <i>EventBus</i>	55
Figura 32: Pantalla inicial	58
Figura 33: Pantalla de trucada	58
Figura 34: Sol·licitud de permisos de GPS.....	60
Figura 35: Sol·licitud d'activació del GPS.....	60
Figura 36: Diàleg de progrés	61
Figura 37: Llista d'oficines	61
Figura 38: Pantalla d'identificació	62
Figura 39: Alerta d'enllaç entre els dispositius	64
Figura 40: Alerta de canvi d'usuari amb un SmartWatch vinculat	64
Figura 41: Pantalla principal de la zona de clients	65
Figura 42: Llista de targetes d'un usuari.....	66
Figura 43: Llista de moviments d'una targeta.....	67
Figura 44: Detalls d'un moviment	68
Figura 45: Pantalla d'accions del rellotge	69
Figura 46: Desvinculació del SmartWatch.....	69
Figura 47: Selecció de notificacions	70

Figura 48: Pantalla d'inici	72
Figura 49: Pantalla d'error de connexió.....	72
Figura 50: Pantalla d'error de comunicació	73
Figura 51: Alerta d'execució sense identificació.....	73
Figura 52: Pantalla principal sense vincle	74
Figura 53: Pantalla principal amb vincle	74
Figura 54: Llista d'oficines	75
Figura 55: Missatge de GPS no existent	75
Figura 56: Missatge d'error d'accés al GPS	75
Figura 57: Pantalla amb la llista de targetes.....	76
Figura 58: Llista de moviments.....	77
Figura 59: Pantalla de detalls d'un moviment.....	77
Figura 60: Targeta de notificació al rellotge	78
Figura 61: Accions de la notificació	78
Figura 62: Pantalla inicial de la web de "Developers" d'Android	79
Figura 63: Exportació d'un projecte en Android Studio.....	80
Figura 64: Selector de projecte	81
Figura 65: Menú d'opcions d'Android Wear.....	81
Figura 66: Web de "Genymotion"	87
Figura 67: Finestra de "Genymotion"	88
Figura 68: Llista de dispositius pendent d'identificació	88
Figura 69: Dispositiu "Nexus5"	89
Figura 70: Activació de la localització de "GenyMotion"	90

Llista de taules

Taula 1: Aplicacions de banca adaptades per smartwatch	10
Taula 2: Classificació de versions d'Android	44

1.Introducció

1.1. Context i justificació del Treball

El sector financer i la digitalització està lligat des de fa molts anys, afegint, modificant o creant noves aplicacions, tecnologies o eines amb les quals poder donar suport a les necessitats dels bancs i als seus clients.

Amb l'aparició del Internet, ordinadors, dispositius mòbil intel·ligents i ara *wearables* intel·ligents, aquest sector s'ha anat unint cada vegada més amb la tecnologia fins al punt de què els bancs permeten realitzar accions des dels teus dispositius o fins i tot s'han creat bancs que solament funcionen online.

El ventall de funcionalitats que es pot fer amb una aplicació bancària cada vegada és més gran. Un exemple en l'actualitat és l'intent de substituir la típica targeta de crèdit/dèbit per una simple aplicació mòbil la qual conté targetes virtuals. Amb aquestes targetes i certs components dels dispositius mòbils es pot inclús realitzar pagaments en comerços.

En resum, cada vegada hi ha més tecnologia i el banc intenta aprofitar-la perquè els seus clients puguin tenir més comoditat en el moment de realitzar accions bancàries sense necessitat d'acudir a una entitat bancària.

Això no vol dir que les oficines vagin a desaparèixer. Aquestes es mantindran actives, ja que hi ha gran part de la població que no és capaç d'afrontar el canvi cap a la digitalització (per exemple, pels pocs coneixements tecnològics) i també perquè la gent prefereix el suport d'una persona en realitzar algunes tasques.

Fa relativament poc que han començat a sortir els anomenats *wearables*. Aquests són els nous dispositius que una persona porta posats, com una peça de roba o un complement. Els més evolucionats fins al moment són els rellotges i polseres digitals. Aquests, a través de sensors, permeten recollir tota la informació de la persona que els porta ficats, per exemple, controlar el son i les hores dormides. Els rellotges, respecte de les polseres, tenen més funcionalitats gràcies a què els forma més tecnologia, per exemple, es poden veure les notificacions del mòbil sense necessitat de treure'l de la butxaca o inclús contestar trucades.

Alguns bancs estan començant a fer aplicacions per aquests nous dispositius, però aquestes no estan molt desenvolupades, és a dir, solament permeten fer una petita quantitat d'accions.

El que es vol fer en aquest projecte és, a partir de les funcionalitats que ja conté aquestes aplicacions, crear-ne de noves per, en un futur, poder acabar afegint-les a les altres aplicacions dels bancs.

El motiu pel qual s'ha escollit el sistema Android per desenvolupar aquest projecte és la gran afició a aquest sistema operatiu. Es va començar a estudiar el sistema operatiu durant els estudis en el grau d'enginyeria informàtica, on es va fer el primer contacte com a desenvolupador Android.

Seguidament, es va seguir estudiant el llenguatge i, a més a més, desenvolupar petites aplicacions per al dispositiu propi.

Ara que han aparegut els smartwatch i juntament amb el TFM és un bon moment per investigar sobre aquesta nova tecnologia i preparar alguna aplicació optimitzada per aquests dispositius.

1.2. Objectius del Treball

Els objectius del treball són els de poder crear un projecte en el qual es desenvolupi una aplicació per dispositius Android optimitzada per als smartwatch que tinguin Android Wear[1]. El tema del projecte estarà enfocat en la banca digital, bàsicament en poder realitzar certes accions d'un banc a través del dispositiu mòbil.

El projecte tindrà com a objectiu generar una aplicació per dispositius mòbils i una per a smartwatch. Les dues podran realitzar les següents tasques:

- Consultar les comptes d'un client identificat
- Veure els moviments fets en cada compte d'un client identificat
- Consultar les oficines properes a la posició de l'usuari
- Trucar a un telèfon d'assistència del banc

Mentre l'usuari no estigui "loguejat", l'aplicació solament permetrà realitzar les dues últimes tasques (consultar oficines i tenir assistència)

L'aplicació mòbil serà l'encarregada de tenir la pantalla de "login" per a identificar un client i, seguidament, obtenir la seva informació. També tindrà una funcionalitat per enviar notificacions al mateix dispositiu, les quals es mostraran al rellotge i es podran gestionar des d'allí.

1.3. Enfocament i mètode seguit

L'objectiu del projecte és desenvolupar una aplicació ja existent al mercat, tot i que, actualment encara no està molt extensa amb el sistema operatiu d'Android. A partir de l'estudi de les altres aplicacions i de l'anàlisi de les seves funcionalitats, s'ha plantejat un projecte amb alguns requeriments similars i altres de nous, com, per exemple, poder realitzar certes accions si l'usuari no està "loguejat" a l'aplicació.

En la Taula 1 es pot veure l'anàlisi d'algunes de les aplicacions que hi ha actualment al mercat. Aquest estudi s'ha fet utilitzant informació d'Internet (vídeos promocionals o demostratius, blogs, fotos, etc.) a causa del fet que l'accés a l'aplicació estava restringit per a clients del banc.

Aplicació	Sistema Operatiu	Característiques
BBVA	<ul style="list-style-type: none">• iOS• Android	<ul style="list-style-type: none">• Posició global• Saldo de les comptes• Productes contractats per l'usuari• Ingresos/Gastos del dia
Banco Santander (Santander Watch)	<ul style="list-style-type: none">• iOS• Android	<ul style="list-style-type: none">• Consulta de saldo de les comptes i targetes de crèdit• Consulta en detall els moviments de comptes i targetes• Consulta el saldo disponible de les targetes de crèdit
Bankinter	<ul style="list-style-type: none">• iOS• Android	<ul style="list-style-type: none">• Rebre notificacions a temps real dels moviments en les seves targetes
LaCaixa	<ul style="list-style-type: none">• Android (solament per als Sony Smartwatch 2)	<ul style="list-style-type: none">• Seguir l'evolució de la Borsa

Taula 1: Aplicacions de banca adaptades per smartwatch

1.4. Planificació del Treball

Per crear la planificació s'ha generat un diagrama de Gantt, el qual es pot veure a la següent pàgina. Aquest està dividit en quatre parts que són les quatre pràctiques de l'assignatura, i cadascuna conté les seves tasques a realitzar.

En el diagrama es pot veure la data d'inici, la de fi i la seva durada. També es pot veure la relació entre les tasques, de forma que es veu quina funció depèn de l'anterior o no.

En aquest Gantt es mostra, separat per tasques i/o pràctiques, la durada tant en dies com en hores de cadascuna amb la seva data d'inici i la seva data de fi.

El càlcul de les hores s'ha fet adaptat a una persona la qual treballa més de 8 hores diàries amb els caps de setmana com a dies festius, i a més a més, estar cursant certes assignatures dels seus estudis. Tenint en compte aquestes hores, s'ha arribat a la conclusió que el temps que es podrà dedicar al TFM serà una mitjana de 3hores durant els dies laborals i 6hores en els dies festius.

Per tant, si una tasca comença el 24/02/2016 i acaba el 09/03/2016, la seva durada serà de 15dies. D'aquests 15 dies, 4 són cap de setmana, i 11 són dies laborables, de forma que: $11\text{dies} * 3\text{hores} = 33\text{hores}$ durant la jornada laboral, i $4\text{dies} * 6\text{hores} = 24\text{hores}$ durant els festius i/o caps de setmana. La durada en hores en realitzar la tasca serà $33+24 = 57\text{hores}$.

Id.	Nom de la tasca	Inici	Fi	Durada	Durada (hores)	feb 2016						mar 2016					
						24	25	26	27	28	29	1	2	3	4	5	6
1	PEC 1 (Pla de treball)	24/02/2016	09/03/2016	15d	57h												

Id.	Nom de la tasca	Inici	Fi	Durada	Durada (hores)	mar 2016																abr 2016														
						10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6			
1	PEC 2 (Disseny)	10/03/2016	06/04/2016	28d	108h																															
2	Anàlisi d'usuaris	10/03/2016	16/03/2016	7d	27h																															
3	Disseny conceptual	17/03/2016	23/03/2016	7d	30h																															
4	Prototipat	24/03/2016	30/03/2016	7d	27h																															
5	Evaluació	31/03/2016	06/04/2016	7d	27h																															

Id.	Nom de la tasca	Inici	Fi	Durada	Durada (hores)	abr 2016														may 2016																					
						7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12
1	PEC 3 (Implementació)	07/04/2016	18/05/2016	42d	165h																																				
2	Aplicació mòbil	07/04/2016	30/04/2016	24d	93h																																				
3	Identificació d'usuaris	07/04/2016	10/04/2016	4d	18h																																				
4	Obtenció i mostra de dades del backend	11/04/2016	20/04/2016	10d	36h																																				
5	Notificacions entre dispositius	21/04/2016	24/04/2016	4d	18h																																				
6	Localització i obtenció d'oficines properes	25/04/2016	30/04/2016	6d	21h																																				
7	Aplicació smartwatch	01/05/2016	18/05/2016	18d	69h																																				
8	Obtenció de dades d'un usuari	01/05/2016	09/05/2016	9d	36h																																				
9	Obtenció d'oficines properes	10/05/2016	18/05/2016	9d	33h																																				

Id.	Nom de la tasca	Inici	Fi	Durada	Durada (hores)	may 2016											jun 2016						
						19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5
1	PEC 4 (Entrega final)	19/05/2016	08/06/2016	21d	84h																		
2	Presentació	01/06/2016	08/06/2016	8d	33h																		

1.5. Breu sumari de productes obtinguts

Es presentarà un projecte amb una aplicació Android nativa optimitzada per Android Wear això vol dir que, dins de la mateixa aplicació hi haurà dos sub-aplicacions, una que serà instal·lada al dispositiu mòbil i l'altra que serà instal·lada al smartwatch. Les dues tindran les mateixes funcionalitats, excepte les que necessitin teclat per poder seguir el seu curs, aquestes solament estaran al smartphone.

Aquesta aplicació permetrà consultar la informació bancària d'un client, i també tenir informació sobre les oficines del banc més properes.

Juntament amb l'aplicació hi haurà una memòria amb l'estudi previ de l'aplicació, els detalls de la implementació i decisions preses durant el desenvolupament d'aquest.

Al final del projecte es presentarà una demostració de l'aplicació, amb captures de pantalla i/o vídeos mostrant l'aplicació resultant.

1.6. Breu descripció dels altres capítols de la memòria

El document que conté la memòria del treball final de màster s'estructura de la següent forma:

- En el capítol 1 s'explica tota la introducció al treball contextualitzant l'aplicació, explicant els motius pels quals s'ha desenvolupat aquest projecte i descrivint els objectius als quals es vol arribar
- El capítol 2 conté tota la part de disseny d'estudi de l'aplicació, tant visual com de dades. També es mostra el seu estudi i els requeriments i funcionalitats que contindrà.
- El capítol 3 es centra en Android i una mica de la seva història pel fet que és el sistema operatiu pel qual s'està desenvolupant el projecte.
- En el capítol 4 es tracten els punts més importants, crítics i decisius que s'han trobat en el desenvolupament d'aquesta aplicació.
- En el capítol 5 es descriu en detall l'aplicació resultant juntament amb captures de pantalla.
- El capítol 6 tracta sobre com visualitzar i testejar el codi font adjuntat al projecte.
- Finalment en el capítol 7 s'expressen les conclusions que s'han extret després de dur a terme aquest treball i possibles línies de futur sobre l'aplicació.

2. Disseny

2.1. Disseny centrat en l'usuari

El disseny centrat en l'usuari és una forma de dissenyar i avaluar un producte amb informació de les persones que l'utilitzaran.

L'avaluació ha de ser feta en tots els estats del procés de disseny, tan aviat com sigui possible. Des de la creació de prototips fins a la versió final.

L'estàndard ISO 13407 "Human centred design for interactive systems", proveeix d'un marc teòric pel desenvolupament d'activitats centrades en l'usuari. Es desenvolupen prototips i aquests són provats per usuaris reals amb la finalitat d'obtenir un feedback. Amb això, es substitueixen o modifiquen els prototips en funció del feedback.

DCU permet produir software que:

- Més fàcil d'usar i d'entendre
- Millora la qualitat de vida dels usuaris i incrementa la satisfacció
- Incrementar la productivitat i eficàcia operacional dels usuaris i per tant els de l'organització
- El procés permet identificar problemes en un estat en el qual encara és barat i possible fer canvis

2.1.1. Usuaris i context d'ús

L'aplicació que es desenvolupa amb el projecte està enfocada als *wearables* i al sector bancari, la qual cosa disminueix el rang d'escenaris en el qual es podrà utilitzar l'aplicació. El que es vol dir és que, una persona menor de 18 anys aprox., segurament no disposarà de comptes bancaris propis i tampoc serà una persona que tindrà molt interès en controlar els seus comptes. En l'altre extrem, una persona de la tercera edat serà molt estrany que disposi d'un smartwatch, i en cas de tenir-lo, serà perquè té coneixements de tecnologia.

En resum, els usuaris de l'aplicació podran ser persones amb coneixements informàtics, o persones de mitjana edat amb pocs coneixements. Per definir aquest tipus d'usuaris s'han creat les següents "Personas"

Nom: Jordi Gutiérrez

Sexe: Home

Edat: 33 anys

Estat social: Casat

Estat laboral: Actiu



Estudis Acadèmics: Grau Superior en Comerç Internacional

Idiomes que domina: Castellà, Català, Anglès, Francès

Experiència laboral: 20 anys de comercial en la indústria paperera.

Habilitat en tecnologia: Bona

Descripció de l'usuari:

En Jordi és un home que treballa de comercial i es passa cada dia de la setmana viatjant entre diferents poblacions i ciutats per intentar captar clients pel seu producte.

A la seva empresa li paguen les dietes, el quilometratge amb el cotxe i l'estança a un hotel en cas que tingui de quedar-se més d'un dia, però ho paguen a final de mes juntament amb la nòmina, la qual cosa implica que primer ho ha de pagar tot ell. A causa d'això, li agrada mantenir al dia els moviments fets amb les seves targetes, d'aquesta forma pot portar un control dels diners que té de pagar-li l'empresa i també vigila per no deixar cap compte al descobert en cas de quedar-se molts dies en una localitat.

Descripció del "scenario":

En Jordi està reunit amb un client i rep una notificació avisant que un dels seus comptes està a punt de quedar-se a 0, però com que té el telèfon amb silenci no s'adona de la notificació. Com a conseqüència, se li passarà una factura que té pendent de pagar la qual farà que el compte passi a estar al descobert i llavors el banc li cobrarà un recàrrec de 50€.

Nom: Carolina Matías
Sexe: Dona
Edat: 21 anys
Estat social: Soltera
Estat laboral: Estudiant



Estudis Acadèmics: Batxillerat humanístic
Idiomes que domina: Castellà, Català

Experiència laboral: Cap

Habilitat en tecnologia: Mitjana

Descripció de l'usuari:

La Carolina és una universitària que ha tingut de marxar de la casa dels seus pares per poder anar a la universitat on cursen la carrera que volia estudiar. Ara viu al mig de la ciutat en un pis d'estudiants el qual el comparteix amb 3 amigues seves.

El fet de viure separada dels pares fa que s'hagi de pagar el pis, el seu manteniment, la seva roba, el transport, les festes, etc., el que acaba sent una gran quantitat de diners.

Per poder assolir aquestes despeses, la Carolina treballa on pot durant l'estiu, però la resta de l'any solament el dedica a estudiar.

Tot això fa que sempre controlï els moviments fets amb les seves targetes bancàries, mirant que en un mes no es gastï més diners dels previstos i no pugui arribar a l'estiu amb la resta.

Descripció del "scenario":

Cada nit després de la universitat, la Carolina es fica a fer treballs i pràctiques dels seus estudis, i també revisa les seves targetes a través de la web del banc. L'ordinador que té és molt lent i antic, i a ella no l'hi agrada gens, però no disposa de cap altra alternativa i s'adapta amb el que té.

Fa uns dies va ser l'aniversari de la Carolina i la seva parella va regalar-li un smartwatch. Tot i no tenir massa coneixements sobre com funciona, ella el porta ficat durant tot el dia.

Amb l'aplicació del banc optimitzada per rellotges intel·ligents es pot estalviar el temps d'engegar i para l'ordinador, ja que pot mirar-ho ràpidament al seu rellotge amb un parell de clics.

2.1.2. Disseny conceptual

Veient els "scenarios" de les "Personas" creades es pot crear l'estructura i les funcionalitats de l'aplicació.

El principal requeriment de l'aplicació és que un client que tingui usuari i contrasenya pugui veure l'estat de les seves targetes i comptes. A més a més, també pugui consultar els moviments fets en cadascuna i veure el detall d'aquest.

A aquesta funcionalitat se li suma la necessitat de poder identificar un usuari, el que vindria ser una validació de credencials. Això vol dir que un altre dels requeriments de l'aplicació serà el de tenir un "login" per poder accedir a les dades d'un usuari concret.

Per tal de no limitar l'aplicació solament a usuaris que tinguin credencials es permetrà realitzar funcionalitats que no necessitaran identificació. Una d'aquestes serà la de poder obtenir una llista d'oficines properes a través de la ubicació del dispositiu. L'altra, per tal d'ajudar a un usuari que tingui problemes o un futur client, s'oferirà la possibilitat de realitzar una trucada a un telèfon d'assistència del banc.

Finalment, l'aplicació ha de ser capaç de mostrar les notificacions enviades des del servidor en cas que hi hagi algun avís. Aquestes notificacions han de permetre realitzar accions, com per exemple, en cas que arribi una notificació indicant que s'ha realitzat un moviment de més de 100€, poder pitjar sobre la notificació per accedir a aquest moviment.

2.1.3. Prototipat

El prototipat es basa en l'elaboració de models o prototips de la interfície de l'aplicació que es desenvoluparà. Amb aquest model es podran avaluar les decisions del disseny, permetent modificar-les d'una forma senzilla, ràpida i econòmica.

El disseny de l'aplicació desenvolupada s'ha dividit en dues parts, una per la part que es veurà al dispositiu mòbil i l'altra pel rellotge intel·ligent.

2.1.3.1. Dispositiu mòbil

L'aplicació del smartphone començarà amb una pantalla inicial, la qual es pot veure a la Figura 1. En aquesta es podran realitzar tres accions: la primera i principal és la d'entrar a la zona privada en la qual es podran consultar les targetes, comptes i els moviments de cadascuna, tot això després de validar l'usuari al "login". La segona és per buscar les oficines més properes. Per acabar, l'última opció és per trucar a un telèfon d'atenció al client, la qual, quan es pitja sobre el botó, truca directament a les oficines del banc, tal com es veu a la Figura 2.



Figura 1: Pantalla inicial

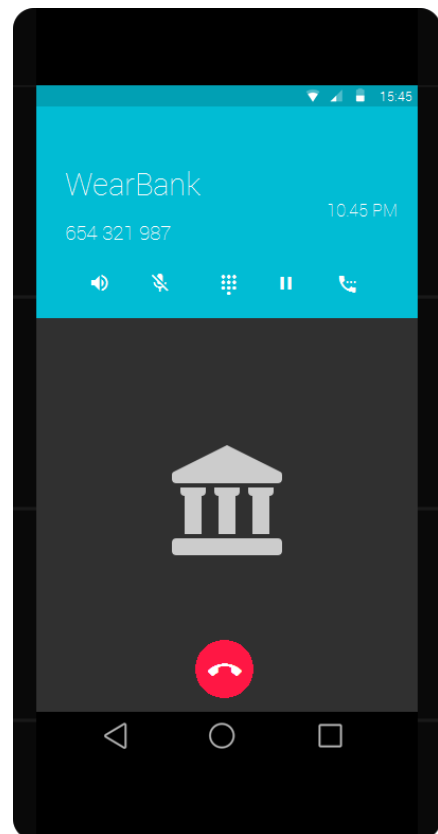


Figura 2: Trucada d'atenció al client

Pel que fa al localitzador d'oficines, aquest obtindrà la posició de l'usuari i mostrarà la llista d'oficines en ordre de distància, de més propera a més llunyana, igual que la Figura 3.

Quan es seleccioni algun element de la llista, s'executarà la navegació per GPS per indicar la ruta fins a l'oficina triada. Aquesta navegació la durà a terme el mapa natiu d'Android (Figura 4).

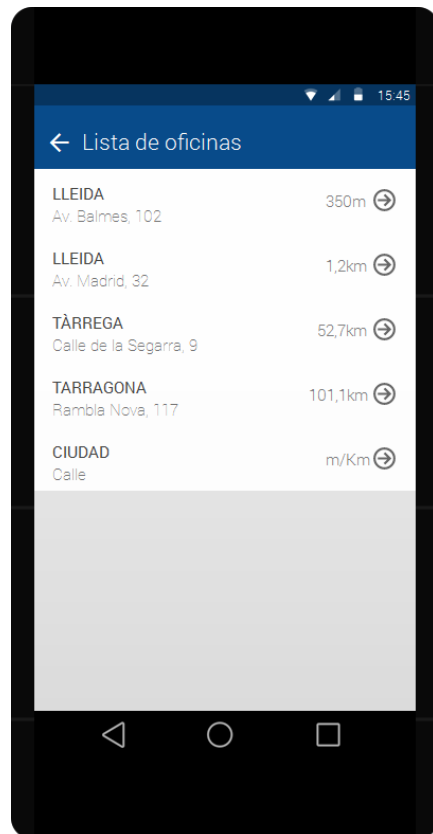


Figura 3: Pantalla amb la llista d'oficines

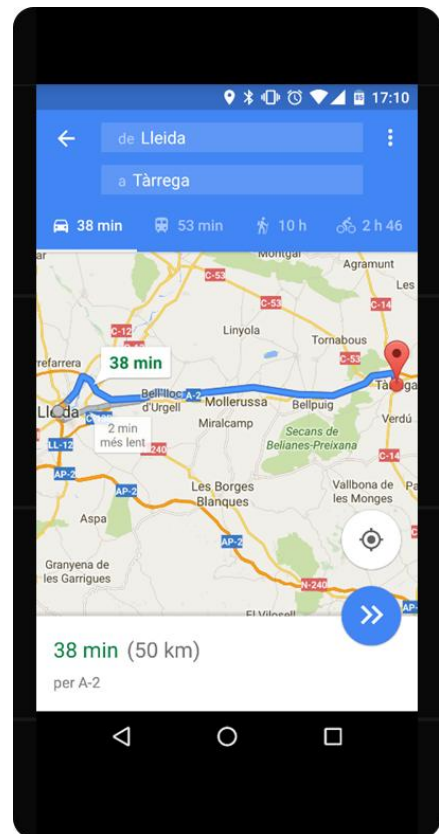


Figura 4: Pantalla de GPS fins l'oficina

Per últim, falta la part de la zona privada d'un client, és a dir, la part on un usuari pot entrar i veure els seus comptes. Per accedir a aquesta zona s'utilitzarà el botó de "Acceso a la zona privada" de la Figura 1. Aquest obrirà la Figura 5 en la qual s'introduiran les credencials d'accés de l'usuari. La pantalla serà molt simple, ja que solament disposarà de dos camps de text i un botó. Qualsevol error amb les dades introduïdes per l'usuari seran mostrades en un missatge per pantalla.

Una vegada superat aquest pas, es mostrarà la pantalla principal de la zona privada que serà com la Figura 6. Aquesta estarà formada per un text de benvinguda i una llista de botons que s'utilitzaran per accedir a les dades de l'usuari. En un futur, per fer que aquesta pantalla fos més atractiva visualment, es podria modificar i afegir/mostrar les dades de posició global utilitzant gràfics o altres eines visuals.

Les accions que es poden realitzar en aquesta pantalla són les de poder consultar les targetes i comptes de l'usuari i també, igual que abans, poder obtenir la llista d'oficines properes. La part de posició global és solament decorativa, ja que no s'implementarà en aquest projecte a causa del fet que es prefereix aprofundir en l'aplicació instal·lada en el rellotge (es podria implementar en un cas futur, o tal com s'ha comentat abans, es podria modificar la pantalla principal i que directament mostrés la posició global).



Figura 5: Pantalla d'identificació

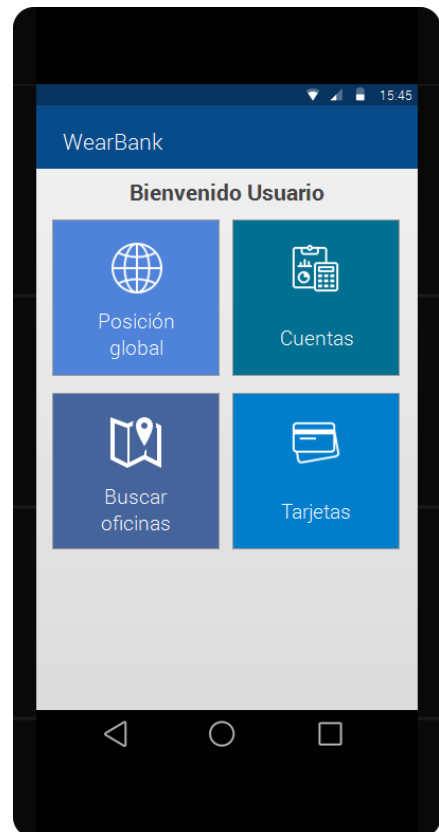


Figura 6: Pantalla principal de zona clients

L'última part que falta per veure és la llista de targetes/comptes, els seus moviments i el detall d'aquests. Per començar, si es selecciona l'acció de "Tarjetas" o la de "Cuentas" s'obrirà una pantalla que mostrarà la llista de targetes de l'usuari, com la Figura 7 (per simplificar l'aplicació, l'opció de comptes no s'implementarà, ja que seria molt semblant a la de targetes, per tant, quan es faci clic a un dels dos botons es dirigiran a la pantalla de targetes).

Una vegada triada la targeta, es mostrarà la llista de moviments que s'hi ha realitzat (Figura 8). Els moviments es mostraran ordenats per data, de forma que el primer serà el més actual i l'últim el més antic. Cada element tindrà una fletxa que marcarà si el moviment és positiu o negatiu. Igual que la quantitat de diners, es pintarà de color verd en cas de ser positiu i roig en cas de ser negatiu. També es veurà el concepte del moviment i la seva categorització.

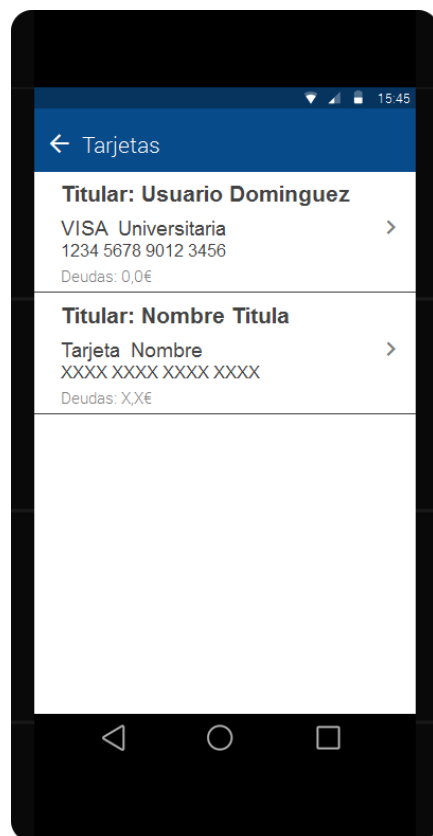
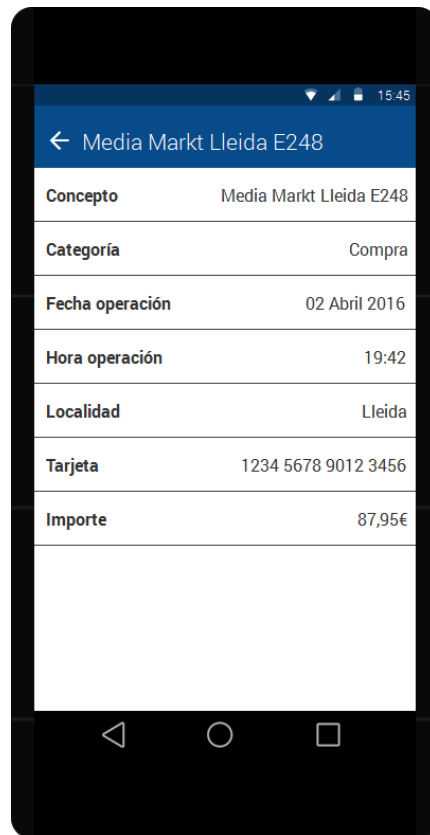


Figura 7: Llista de targetes



Figura 8: Llista de moviments d'una targeta

Finalment, si es pitja sobre un moviment, es mostrarà el seu detall en una pantalla com la Figura 9, en la qual es podrà veure tota la informació del moviment.



The image shows a smartphone screen with a dark blue header bar containing a back arrow and the text 'Media Markt Lleida E248'. Below the header is a table with transaction details. The table has two columns: labels on the left and values on the right. The labels are 'Concepto', 'Categoría', 'Fecha operación', 'Hora operación', 'Localidad', 'Tarjeta', and 'Importe'. The values are 'Media Markt Lleida E248', 'Compra', '02 Abril 2016', '19:42', 'Lleida', '1234 5678 9012 3456', and '87,95€' respectively. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Concepto	Media Markt Lleida E248
Categoría	Compra
Fecha operación	02 Abril 2016
Hora operación	19:42
Localidad	Lleida
Tarjeta	1234 5678 9012 3456
Importe	87,95€

Figura 9: Detall d'un moviment

2.1.3.2. Smartwatch

L'aplicació dedicada al rellotge començarà per comprovar si hi ha connexió entre el dispositiu mòbil i el smartwatch. En cas negatiu, es mostrarà una pantalla com la Figura 10. Tot seguit es comprovarà si l'usuari ha iniciat sessió a l'aplicació mòbil. Per fer-ho, enviarà un missatge al mòbil per tal que aquest comprovi si el rellotge té accés o no. Si el telèfon no respon al missatge, s'entendrà que hi ha algun error de connexió o que el dispositiu mòbil no té instal·lada l'aplicació i per això no contesta, de forma que es mostrarà la Figura 11 permetent tancar l'aplicació. Si el telèfon contesta, s'analitzarà la resposta i, en cas de no tenir accés es mostrarà una alerta preguntant si es vol utilitzar l'aplicació sense estar identificat (l'alerta es pot veure a la Figura 12).



Figura 10: Pantalla d'error de connexió



Figura 11: Pantalla d'error de "login"

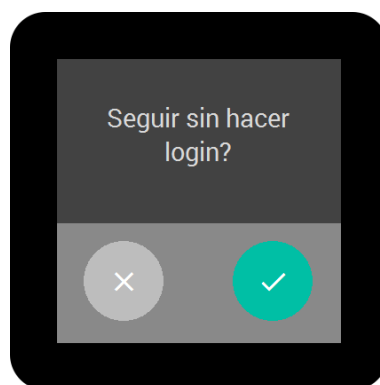


Figura 12: Alerta "login"

Si se selecciona la creu, es tancarà l'aplicació, però si es tria el "tick" verd s'accedirà a la Figura 13 la qual mostra el menú de l'aplicació. En aquesta pantalla es podrà veure la llista de funcionalitats que es podran realitzar: buscar les oficines més properes i trucar al telèfon d'atenció al client.

Les dues duran a terme les mateixes tasques que en l'aplicació del mòbil.

Pel que fa al llistat d'oficines, serà el de la Figura 14, i si es pitja sobre qualsevol element, aquest activarà el GPS per començar el guiat fins a l'oficina.



Figura 13: Menú sense credencials



Figura 14: Llista d'oficines

En cas que l'usuari estigui "loguejat" quan s'executi l'aplicació, es veurà un menú amb les mateixes opcions que quan s'entra a l'aplicació del dispositiu mòbil, tal com es pot veure a la Figura 15.

Si es selecciona "Tarjetas" es mostraran la llista de targetes que té l'usuari (Figura 16) i en pitjar una qualsevol s'accedirà a la llista de moviments d'aquesta.



Figura 15: Menú principal

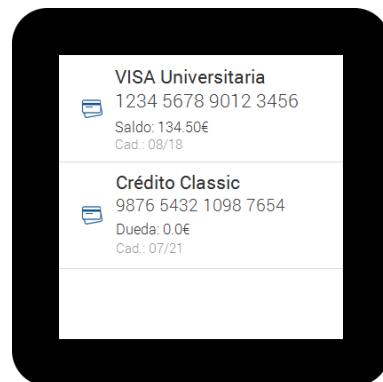


Figura 16: Llista de targetes

La llista de moviments d'una targeta serà més o menys igual que la llista de l'aplicació mòbil, però tenint en compte que la pantalla del rellotge és molt més petita que la del telèfon, es mostraran menys dades a cada element. Això es pot veure a la Figura 17.

Per veure el detall del moviment s'ha de pitjar sobre l'ítem de la llista i apareixerà la pantalla de la Figura 18 amb tota la informació del moviment.



Figura 17: Llista de moviments



Figura 18: Detall d'un moviment

Totes les figures del prototip es poden trobar dins la carpeta "[Adjunts\Prototips](#)".

2.1.4. Avaluació

L'avaluació del prototip serveix per anar revistant els dissenys de l'aplicació podent modificar-los i adaptar-los segons les necessitats.

Per fer l'avaluació hi ha diferents mètodes. En aquest projecte s'ha utilitzat la tècnica de l'avaluació heurística la qual consisteix a realitzar una valoració dels elements del disseny per comprovar que es compleixen els principis d'usabilitat establerts. Aquests principis no estan molt especificats per a les aplicacions orientades a dispositius mòbils, però s'utilitzen els que va publicar Molich i Nielsen el 1990[10] juntament amb nous principis adaptats per les pantalles tàctils:

1. Visibilitat de l'estat del sistema
2. Similitud entre el sistema i el món real
3. Control i llibertat de l'usuari
4. Consistència i compliment d'estàndards
5. Prevenció d'errors
6. Preferència al reconeixement que a la memorització
7. Flexibilitat i eficiència d'ús
8. Estètica i disseny minimalista
9. Ajuda davant d'errors
10. Ajuda i documentació
11. Disseny d'interacció consistent
12. Mides i espais per a dits imprecisos o grossos
13. Simplificar la navegació amb ajudes (una icona visible, per exemple, d'una casa per accedir a la pantalla inicial)

Aquests punts s'han avaluat dins del prototip que s'ha creat (apartat 2.1.3). Per fer-ho s'ha utilitzat una taula d'avaluació heurística que es pot trobar a "[Adjunts\EvaluacionHeuristica.xls](#)" i en la qual es pot veure el resum de les avaluacions heurístiques realitzades amb un nombre reduït de persones especialitzades amb el camp de la tecnologia, la informàtica i les aplicacions mòbils.

Amb el resultat d'aquesta avaluació heurística s'ha pogut extreure els problemes/errors que afectaven més a l'usuari. Amb això s'han fet alguns canvis al disseny, per exemple, s'ha afegit el botó de tirar endarrere a la barra superior juntament amb el nom de la pantalla en la qual es troba. D'aquesta forma, a part del botó de "back" que ja tenen els mòbils, també hi ha l'opció de tirar un pas endarrere a través de la icona superior.

2.2. Diagrama de flux

La Figura 19 representa el diagrama de flux de l'aplicació mòbil i la Figura 20 la del rellotge. Aquest diagrama serveix per mostrar les activitats que es realitzen en un procés mostrant la relació entre elles. Fent-ho amb un gràfic és més fàcil de veure aquestes relacions i aquestes activitats.

La diferència principal que hi ha entre l'aplicació mòbil i l'aplicació del rellotge és en el moment d'iniciar l'app, en la qual el mòbil mostra una pantalla d'inici i permet accedir a la pantalla de "login" o d'identificació, en canvi, en smartwatch, solament obrir l'aplicació ja comprova si hi ha connexió amb el dispositiu i, en cas afirmatiu, si s'ha identificat un usuari a l'aplicació del smartphone.

L'altra diferència es pot veure en el moment de buscar les oficines. Aprofitant que els rellotges porten GPS, s'intenta utilitzar aquest, i en cas que no estigui actiu s'accedeix al del dispositiu mòbil.

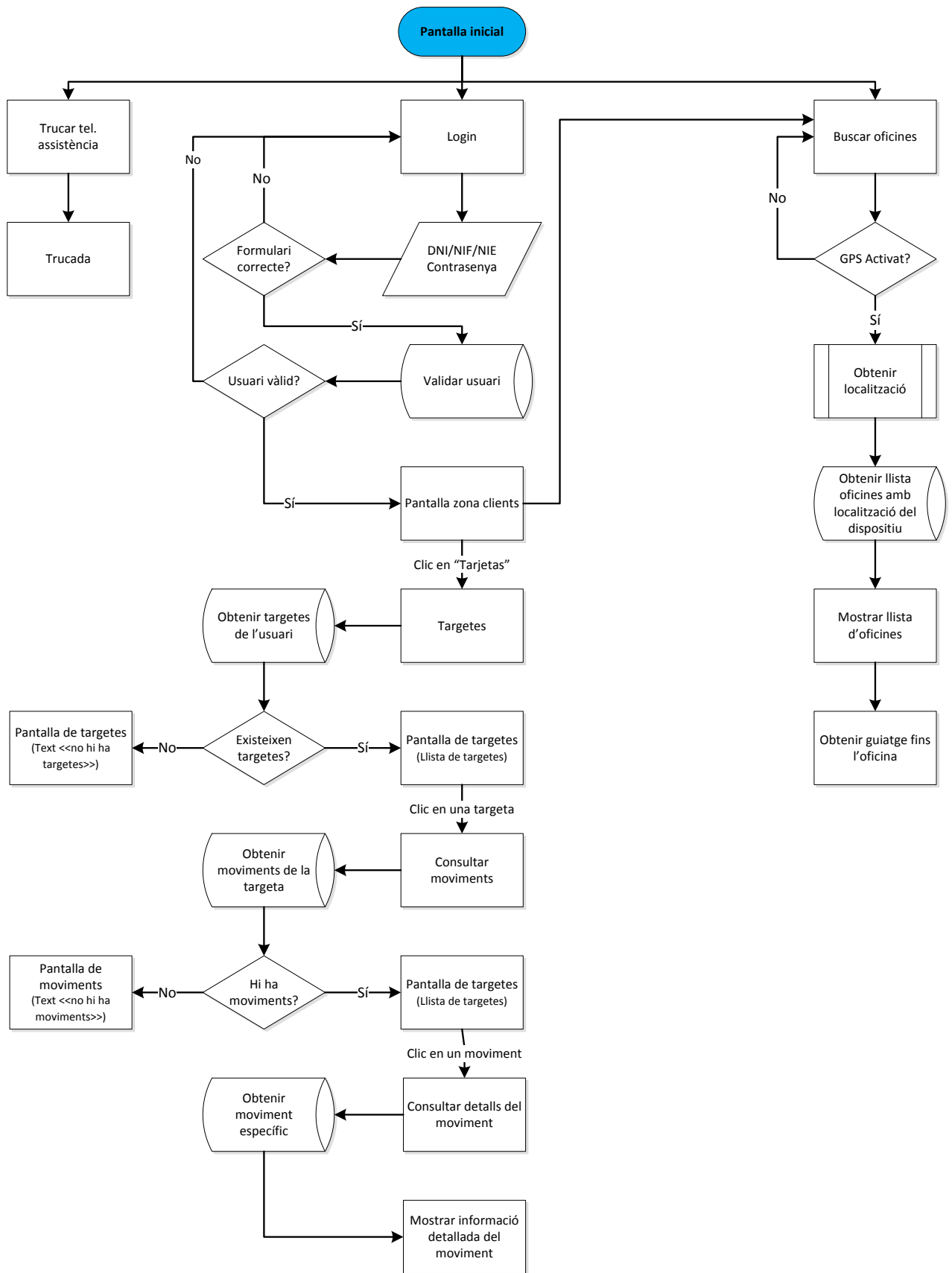


Figura 19: Diagrama de flux de l'aplicació mòbil

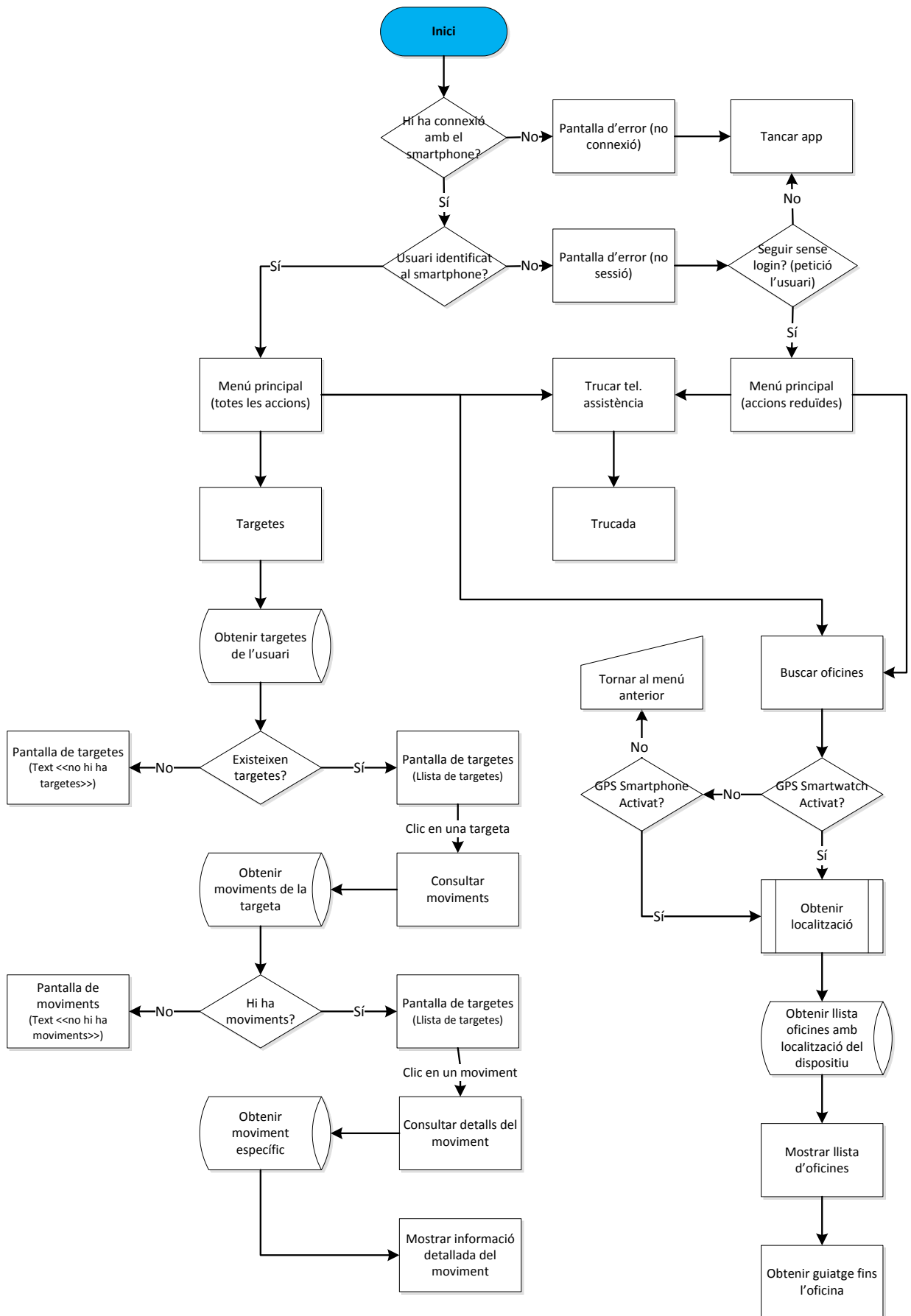


Figura 20: Diagrama de flux de l'aplicació del rellotge

2.3. Definició dels casos d'ús

Els casos d'ús serveixen per identificar el comportament de l'aplicació des del punt de vista de l'usuari.

En aquest projecte s'han realitzat dos diagrames de casos d'ús, un per cada aplicació. Això s'ha fet així perquè el dispositiu mòbil serà el que mostrarà la pantalla per poder identificar un usuari, ja que disposa de teclat. En canvi, el rellotge no té teclat i per tant no es podrien introduir les dades d'un usuari.

La Figura 21 i representa els casos d'ús de l'aplicació mòbil i la Figura 22 la de l'aplicació del rellotge. Tal com s'ha comentat abans, l'única diferència que es pot veure és el cas d'ús de "login", el qual en el primer diagrama apareix, i en el segon no. Pel que fa a la resta, tot és igual.

En els diagrames es pot veure un usuari sense estar identificat el qual pot realitzar dues funcionalitats: consultar les oficines properes i trucar al telèfon d'assistència. En el cas del mòbil també pot accedir al "login".

Aquest primer usuari té una relació de generalització amb l'usuari identificat. Això vol dir que el segon usuari hereta els casos d'ús del primer i, a més a més, en té de propis. Aquests casos són bàsicament els que obtenen i mostren dades privades a l'usuari.

A partir d'aquí, tots els casos d'ús van relacionats entre si, ja que no es poden dur a terme sense haver passat per l'anterior.

Per començar es pot veure el de consultar els comptes i les targetes, el qual va relacionat amb el "login", ja que sense haver passat per aquest no es podran consultar les dades de l'usuari.

Tot seguit hi ha el de consultar els moviments. Aquest es relaciona amb els comptes i targetes perquè serà necessari saber per quina targeta/compte s'han de buscar i mostrar els moviments.

Finalment hi ha el detall d'un moviment, el qual va lligat amb els moviments, ja que s'ha de seleccionar un per poder veure la seva informació.

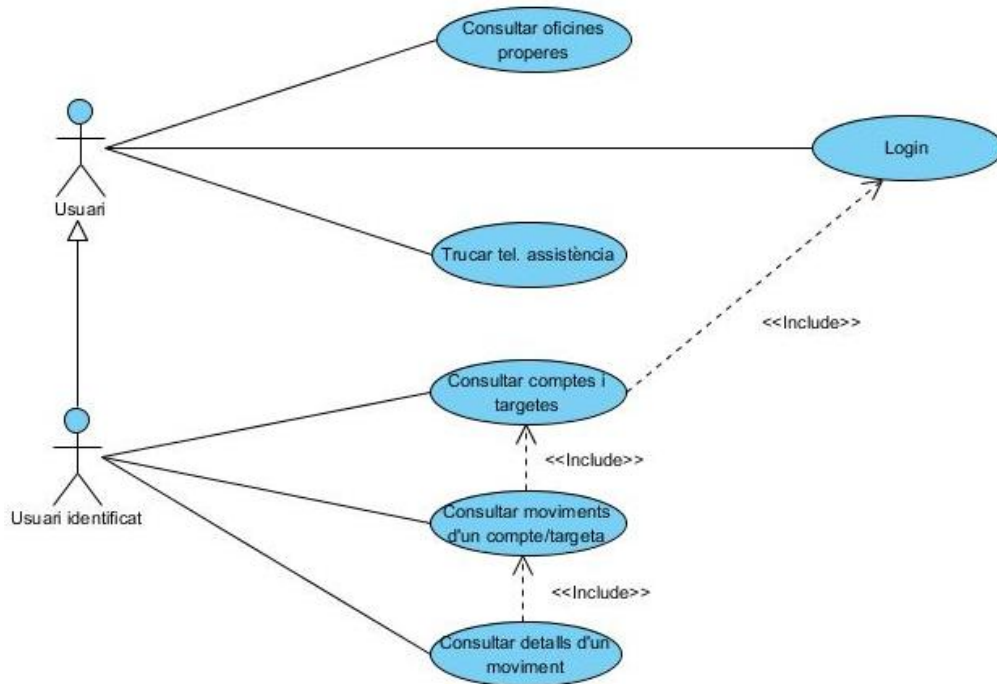


Figura 21: Diagrama de casos d'ús de l'aplicació mòbil

En el smartwatch, el cas d'ús de consultar comptes no dependrà del "login". El que farà l'aplicació serà comprovar si la part del mòbil ha identificat un usuari, però això no serà un cas d'ús, ja que serà una acció interna de l'aplicació en la qual un usuari no podrà interactuar.

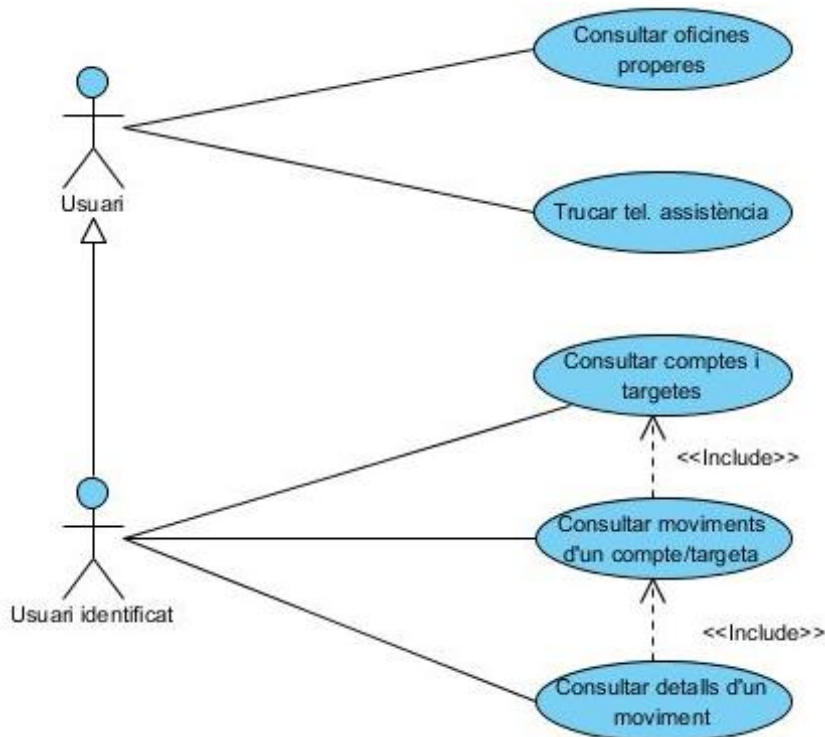


Figura 22: Diagrama de casos d'ús de l'aplicació smartwatch

2.3.1. Especificació dels cassos d'ús

2.3.1.1. Consultar oficines properes

Cas d'ús: Consultar oficines properes
Descripció: Permet veure la llista d'oficines més properes a la posició de l'usuari
Actors: Usuari
Precondicions: -
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre el botó de buscar oficines2. El dispositiu mostra una barra de progrés mentre obté la seva ubicació3. Amb la localització, es consulta la llista d'oficines a la base de dades4. Desapareix la barra de progrés i es mostra la llista d'oficines per pantalla
Flux alternatiu: <ol style="list-style-type: none">1. El GPS no està activat:<ol style="list-style-type: none">a. Desapareix la barra de progrés i es mostra un missatge indicant que no està actiu i si l'usuari vol activar-lo2. El sistema no està actiu:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema
Post condició: Es mostra la llista d'oficines ordenada per proximitat.

2.3.1.2. Trucar al telèfon d'assistència

Cas d'ús: Trucar tel. d'assistència
Descripció: Permet trucar al servei d'atenció al client del banc
Actors: Qualsevol
Precondicions: -
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre el botó de trucar al servei d'atenció al client2. El telèfon mòbil inicia la trucada
Flux alternatiu: -
Post condició: Es truca a atenció al client.

2.3.1.3. Autenticació d'un usuari

Aquest cas d'ús solament es durà a terme dins de l'aplicació mòbil, no a la del rellotge.

Cas d'ús: Login
Descripció: Permet identificar un usuari al sistema
Actors: Qualsevol
Precondicions: -
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre el botó d'entrar a la zona privada de l'aplicació2. El dispositiu mostra una pantalla amb dos camps de text per introduir les credencials i un botó per validar-les3. L'usuari introdueix el seu usuari i contrasenya i pitja sobre el botó de validar4. El dispositiu mostra una barra de progrés mentre comprova que els camps de text no estan buits i envia al sistema les dades introduïdes.5. El sistema valida les dades i respon amb el resultat de la comprovació.6. El dispositiu interpreta la resposta:<ol style="list-style-type: none">a. Si el resultat és correcte, deixa entrar a l'usuari a la zona privada i indica al rellotge que s'ha identificat un usuarib. Si el resultat és incorrecte, mostra un missatge per pantalla i no deixa entrar a l'usuari
Flux alternatiu: <ol style="list-style-type: none">1. No es disposa de connexió a la xarxa:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema2. Els camps de text de la pantalla de "login" estan buits<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema3. El sistema no està actiu:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema
Post condició: L'usuari és identificat per poder entrar a la zona de clients

2.3.1.4. Consultar els comptes i les targetes

Cas d'ús: Consultar comptes i targetes
Descripció: Permet veure la llista de comptes i targetes associades a un client
Actors: Usuari identificat
Precondicions: L'usuari ja ha estat identificat
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre el botó de comptes o de targetes2. El dispositiu mostra una barra de progrés mentre es consulta a la base de dades la informació del client3. El sistema retorna la informació al dispositiu4. El dispositiu mostra la informació per pantalla i desactiva la barra de progrés
Flux alternatiu: <ol style="list-style-type: none">1. No es disposa de connexió a la xarxa:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema2. L'usuari no té comptes ni targetes<ol style="list-style-type: none">a. La llista de la pantalla desapareix deixant a la vista un text on s'indica que no hi ha cap targeta/compte3. El sistema no està actiu:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema
Post condició: Es mostra la llista de targetes o comptes de l'usuari

2.3.1.5. Consultar els moviments d'un compte o d'una targeta

Cas d'ús: Consultar moviments d'un compte/targeta
Descripció: Permet veure la llista de moviments realitzats en un compte o targeta
Actors: Usuari identificat
Precondicions: L'usuari ja ha estat identificat i ha accedit a la pantalla de comptes o targetes
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre una targeta o un compte de la llista2. El dispositiu mostra una barra de progrés mentre consulta a la base de dades els moviments realitzats en l'element seleccionat3. El sistema retorna la informació al dispositiu4. El dispositiu crea una llista amb els valors obtinguts, la mostra per pantalla i desactiva la barra de progrés
Flux alternatiu: <ol style="list-style-type: none">1. No es disposa de connexió a la xarxa:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema2. No s'ha realitzat cap moviment en el compte o targeta seleccionada:<ol style="list-style-type: none">a. La llista de la pantalla on estarien els moviments desapareix deixant a la vista un text on s'indica que no hi ha cap moviment3. El sistema no està actiu:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema
Post condició: Es mostra la llista de moviments realitzats amb una targeta o un compte

2.3.1.6. Consultar els detalls d'un moviment

Cas d'ús: Consultar detalls d'un moviment
Descripció: Permet veure tota la informació d'un moviment realitzat en un compte o una targeta
Actors: Usuari identificat
Precondicions: L'usuari ja ha estat identificat, ha accedit a la pantalla de comptes o targetes i ha mostrat els moviments d'una d'aquestes.
Flux: <ol style="list-style-type: none">1. L'usuari pitja sobre un moviment de la llista2. El dispositiu mostra una barra de progrés mentre consulta a la base de dades la informació del moviment seleccionat3. El sistema retorna la informació al dispositiu4. El dispositiu mostra una pantalla amb la informació obtinguda d'aquell moviment detallada.
Flux alternatiu: <ol style="list-style-type: none">1. No es disposa de connexió a la xarxa:<ol style="list-style-type: none">a. Es mostra un missatge per pantalla indicant el problema2. El sistema no està actiu:<ol style="list-style-type: none">b. Es mostra un missatge per pantalla indicant el problema
Post condició: Es mostra tota la informació relacionada amb un moviment

2.4. Disseny de l'arquitectura

El disseny de l'arquitectura serveix per identificar les entitats que es representaran a la base de dades, les classes i objectes que s'utilitzaran per gestionar els diferents processos i l'estructura de l'API que servirà per realitzar les peticions al servidor.

2.4.1. Diagrama UML del disseny de la base de dades

La base de dades forma part de *Backendless*[7], un MBaaS (Mobile Backend as a Service). Aquest servei conté un projecte privat per l'aplicació que es desenvolupa al projecte, el qual conté una base de dades on es pot emmagatzemar tota la informació.

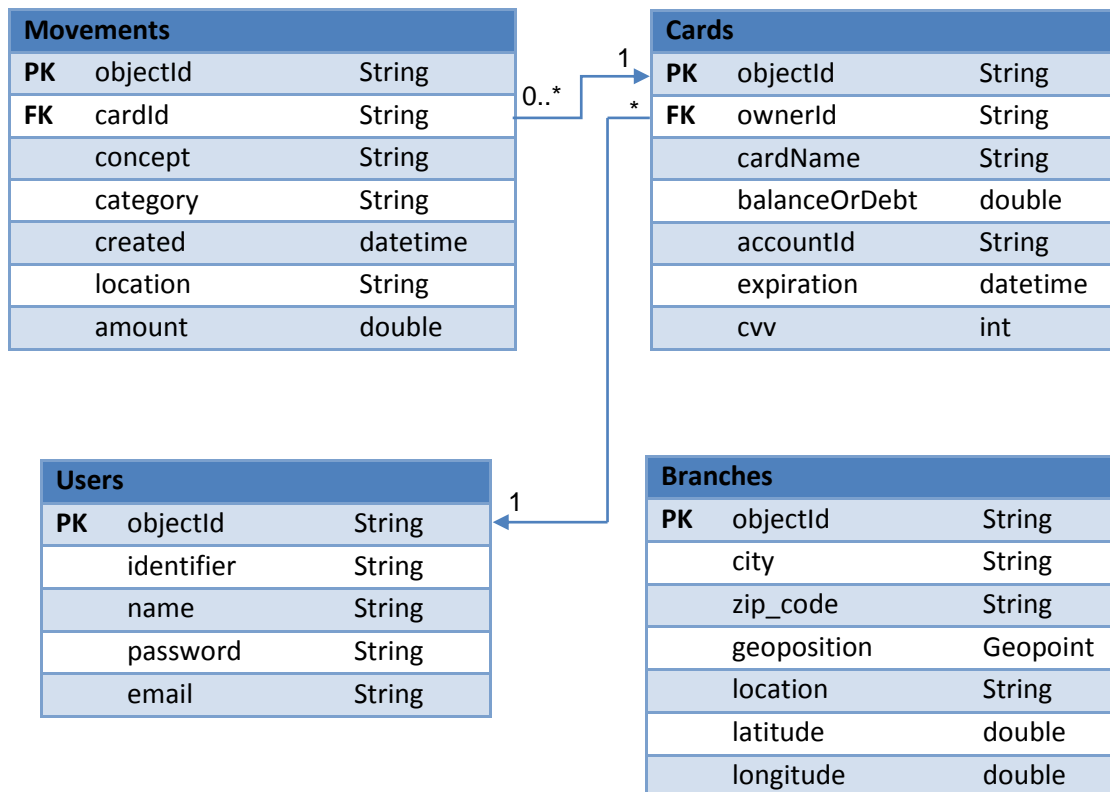
Aquesta guardarà totes les dades necessàries per poder fer que l'aplicació sempre tingui algun valor per mostrar. Les dades s'introduiran manualment en cada taula i des de l'aplicació es mostraran. En un futur, si des de l'aplicació es poguessin fer transferències, seria necessari crear un nou moviment per l'acció realitzada. El mateix passaria en el cas que un usuari contractés una nova targeta o obrís un nou compte, però de moment aquests casos no existeixen i es poden ignorar.

La base de dades d'aquest projecte està formada per quatre taules:

- Movements: emmagatzema tota la informació d'un moviment realitzat amb una targeta:
 - objectId: identificador de l'element a la taula
 - cardId: identificador de la targeta amb la qual s'ha realitzat el moviment
 - concept: Concepte del moviment
 - categoria: Categoria del moviment. Aquesta categoria la podria dur a terme un categoritzador com "Fintonic"[8]
 - created: data (dia i hora) en la qual s'ha realitzat el moviment
 - location: ciutat o població en la qual s'ha realitzat el moviment. Pot ser un camp buit, ja que no es pot obtenir localització de tots els moviments que un usuari realitza
 - amount: quantitat de diners implicats al moviment

- Cards: informació d'una targeta d'un usuari
 - objectId: identificador de l'element a la taula
 - ownerId: identificador del propietari de la targeta
 - cardName: nom de la targeta (per exemple: VISA Universitària)
 - balanceOrDebt: deute de la targeta. Si no en té, és 0.
 - accountId: número de la targeta
 - expiration: data de caducitat de la targeta

- cvv: (Card Verification Value) codi de seguretat de la targeta
- Users: usuaris de l'aplicació
 - objectId: identificador de l'element a la taula
 - identifier: DNI o NIF amb el qual es farà el validrà un usuari
 - name: nom i cognom de l'usuari
 - password: contrasenya de l'usuari per identificar-se
 - email: correu electrònic de l'usuari
- Banches: guarda una llista d'oficines de certs punts de Catalunya
 - objectId: identificador de l'element a la taula
 - city: Ciutat o població en el qual està l'oficina del banc
 - zip_code: Codi postal de la localitat
 - geoposition: Coordenades de l'oficina.
 - location: Direcció de l'oficina
 - latitude: Latitud de l'oficina
 - longitude: Longitud de l'oficina



2.4.2. Diagrama UML del disseny d'entitats i classes

El model de dades de l'aplicació serveix per veure les classes del projecte i les relacions entre aquestes, en aquest cas, les relacions de les classes que representen les taules de la base de dades. En la Figura 23 es pot veure el diagrama de classes creat per aquest projecte. Bàsicament, taules de la base de dades passen a ser classes Java i les columnes d'una taula representen els atributs de la classe. En aquest diagrama també es mostren les operacions que tindrà el model, en aquest cas solament hi ha els constructors amb paràmetres i sense, i els "getters" i "setters" dels atributs.

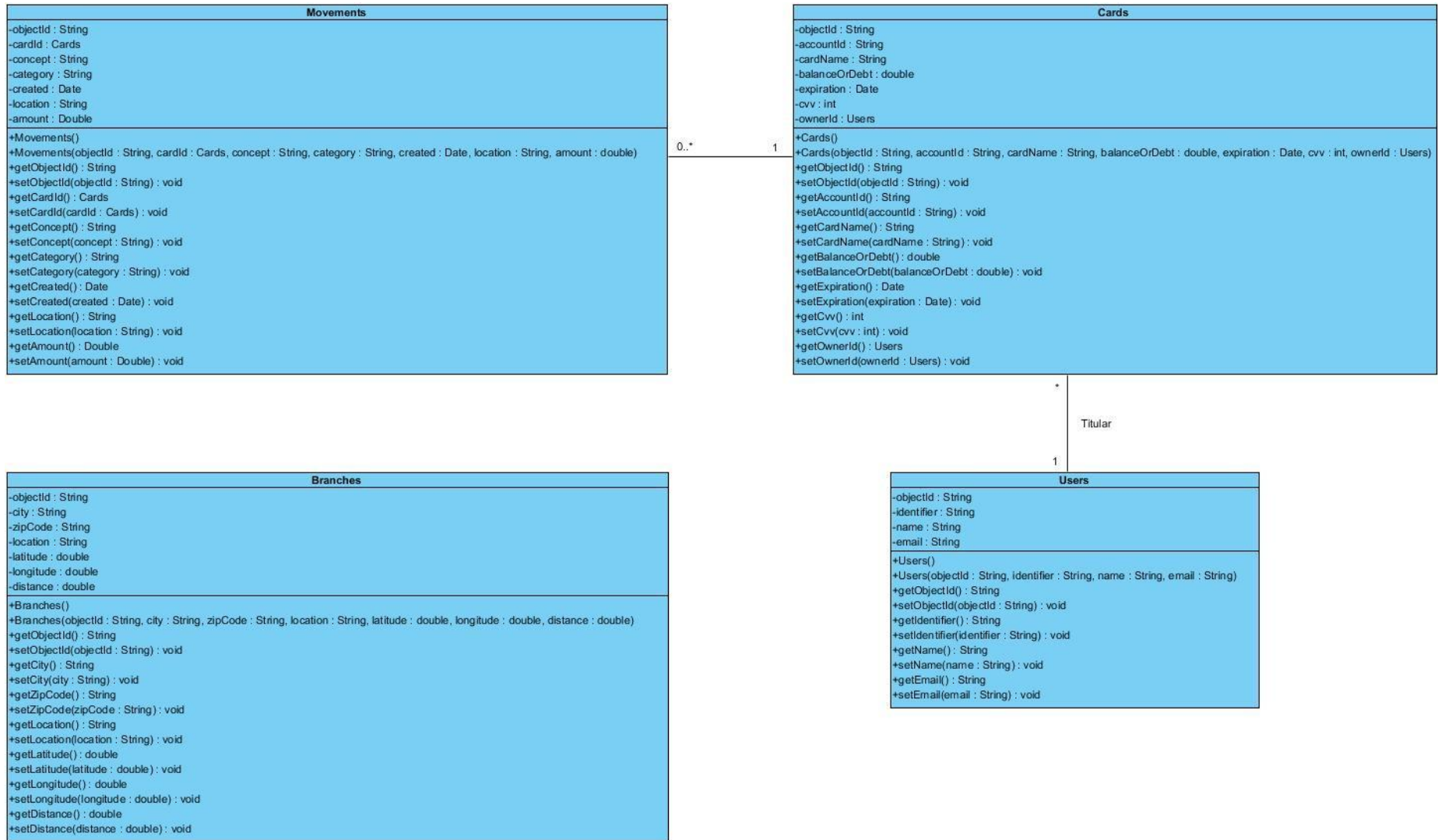


Figura 23: Diagrama de classes

2.4.3. Diagrama de l'arquitectura del sistema

El diagrama de l'arquitectura del sistema serveix per oferir una visió simplificada del sistema. En aquest cas es troba un patró MVC (model-vista-controlador) que es pot veure a la Figura 24.

La capa superior la forma la vista, la qual és l'encarregada de mostrar les interfícies d'usuari amb o sense dades i deixar que l'usuari interactuï amb elles.

El controlador és la capa central i controlar tot el que es pot realitzar a l'aplicació. Respon a les accions dels usuaris en la vista, i també a accions programades (amb condicions o temporitzadors). A més a més, és l'enllaç entre la vista i els models de dades, podent accedir a la part de dades i modificant-les segons la seva necessitat.

Aquestes dues capes estan dins de l'aplicació instal·lada al dispositiu.

Finalment, la capa del model és la capa que s'encarrega de treballar amb les dades. D'aquesta part s'encarregarà el servidor (*Backend*) i serà el que anirà a buscar les dades a la base de dades i les tornarà al dispositiu.

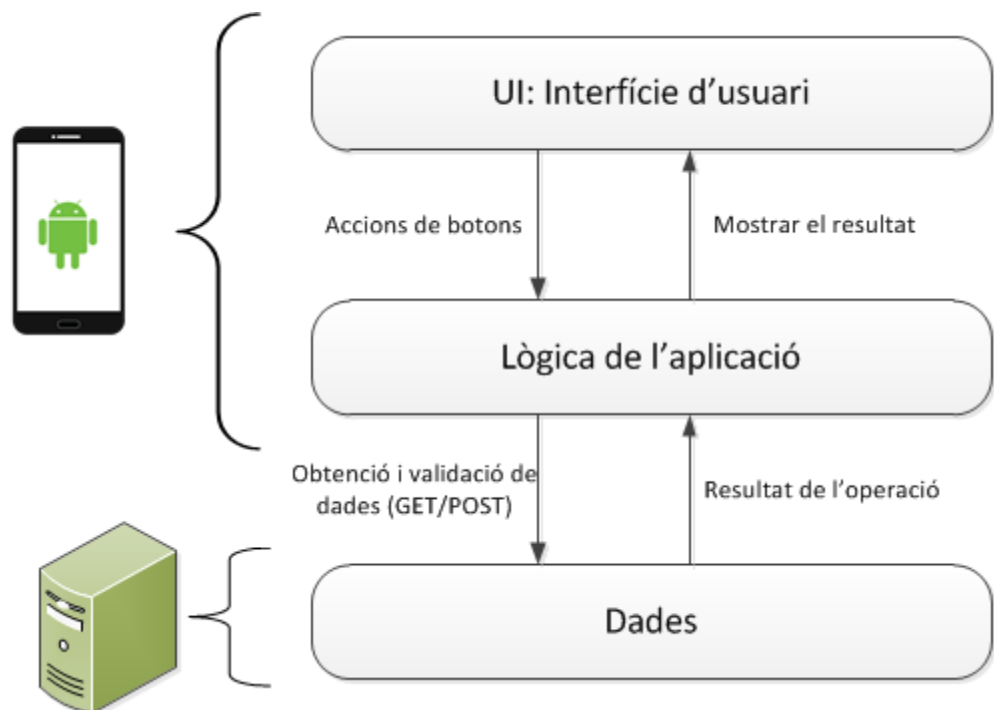


Figura 24: Diagrama de l'arquitectura del sistema

3. Android

3.1. Què és Android?

Android és un sistema operatiu basat amb el nucli de Linux dissenyat per dispositius mòbils amb pantalla tàctil, com telèfons intel·ligents (smartphone) o tabletas, desenvolupat per Android i seguidament comprat per Google.

Avui en dia, aquest sistema operatiu s'utilitza en gran varietat de dispositius, a part dels comentats anteriorment, també hi ha ordinadors, netbooks, rellotges de polsera i altres.

Està dissenyat per un baix consum i, principalment, per la integració de la xarxa implicant poder estar connectat a Internet sempre que es vulgui, juntament amb grans capacitats de memòria i de potència de processament.

El fet d'estar basat en Linux ajuda a obtenir un sistema operatiu lliure, gratuït i multi plataforma, i juntament amb l'existència d'eines de programació gratuïtes, fan que hi hagi una gran quantitat d'aplicacions disponibles. És a dir, qualsevol usuari pot programar en el sistema i introduir els programes al seu dispositiu sense necessitat de pagar res. Aquest fet és un gran avantatge pels fabricants i els desenvolupadors, ja que d'aquesta forma, el cost per llençar una aplicació o un dispositiu és molt baix.

3.2. Historial de versions

Des del seu llançament el 2008, Android ha anat arreglant el seu Sistema Operatiu agregant noves funcionalitats, corregint errors o actualitzant funcionalitats anteriors.

Una dada curiosa és que cada actualització rep, en Anglès, nom a diferents postres, i estan ordenats alfabèticament. Aquestes actualitzacions són:

Plataforma	Nom Original	Traducció	API Level
Android 1.0	Apple Pie	Pastel de poma	1
Android 1.1	Banana Bread	Pa de plàtan	2
Android 1.5	Cupcake	Magdalena (muffin)	3
Android 1.6	Donut	Rosquilla	4
Android 2.0/2.1	Éclair	Pastel francès	7
Android 2.2	Froyo	logurt gelat	8
Android 2.3	Gingerbread	Pa de gingebre	10
Android 3.0/3.1/3.2	Honeycomb	Panell de mel	13
Android 4.0	Ice Cream Sandwich	Gelat de sandvitx	15
Android 4.1/4.2/4.3	Jelly Bean	Gominola	16/17/18
Android 4.4	KitKat	KitKat	19
Android 5.0	Lollipop	Piruleta	21
Android 5.1	Lollipop	Piruleta	22
Android 6.0	Marshmallow	Núvol	23
Android 7.0 o 6.X	N	-	-

Taula 2: Classificació de versions d'Android

Per veure en detall les modificacions de cada versió, Android té una web dedicada ha explicar la història de cadascuna[14].



Figura 25: Versions d'Android

3.3. Arquitectura

L'arquitectura d'Android es pot dividir en cinc capes[16]:

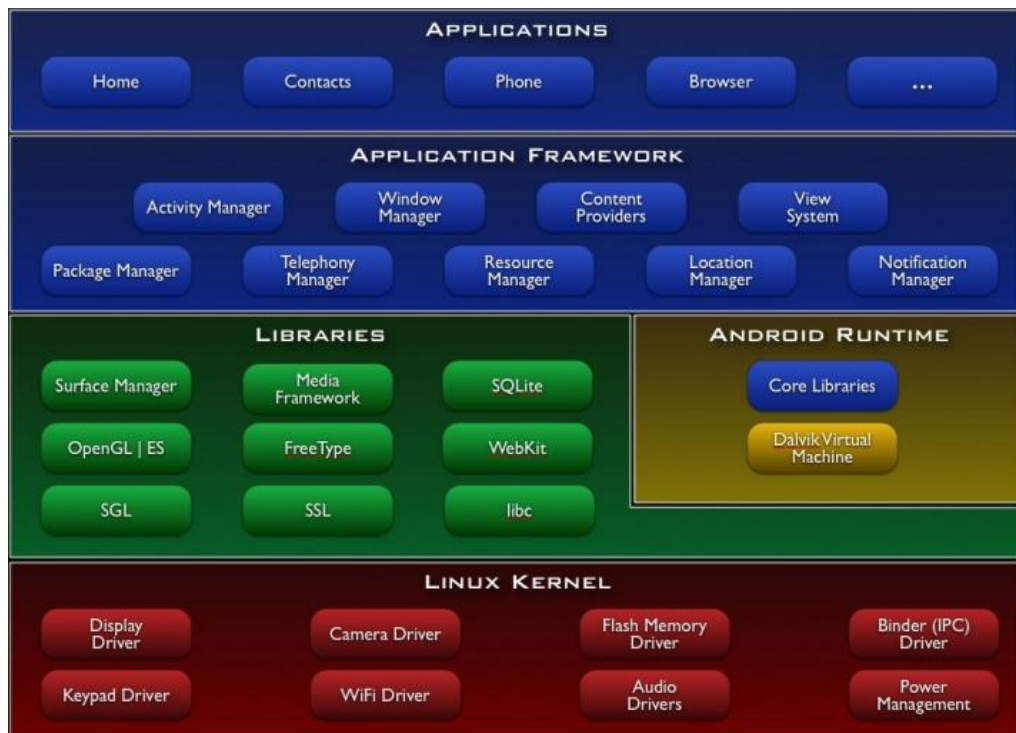


Figura 26: Arquitectura Android

3.3.1. Aplicacions

Aquesta capa inclou totes les aplicacions del dispositiu, ja siguin natives com instal·lades de fàbrica i de l'usuari, com per exemple: programes de SMS, calendari, mapes, client de correu electrònic, etc.

3.3.2. Entorn de l'aplicació

Proporciona una plataforma de desenvolupament lliure per aplicacions, i els desenvolupadors tenen total accés a ells.

Aquesta capa per simplificar la reutilització dels components; qualsevol aplicació pot publicar les seves capacitats i una altra aplicació pot utilitzar-les (subjecte a unes normes de seguretat).

3.3.3. Biblioteques

Android inclou un conjunt de biblioteques en C/C++ utilitzades en diferents components del sistema. Moltes d'aquestes utilitzen projectes de codi obert. Algunes d'elles són: biblioteques de medis, biblioteques de gràfics, 3D, SQLite, i moltes altres.

3.3.4. Runtime d'Android

Està basat amb el concepte de màquina virtual utilitzant Java. Donades les limitacions dels dispositius on ha de córrer Android (poca memòria i processador limitat) no era possible utilitzar una màquina virtual Java estàndard. Google va crear una nova màquina anomenada *Dalvik*, que podria ser millor i evitar aquestes limitacions.

Android permet utilitzar múltiples màquines virtuals per cada aplicació de forma eficient.

Runtime també incorpora un conjunt de llibreries base les quals contenen la majoria de les funcions disponibles de les llibreries de Java.

3.3.5. Nucli Linux

El nucli d'Android està format pel sistema operatiu Linux versió 2.6. Aquesta capa proporciona serveis com la seguretat, el control de memòria, el multiprocés, la pila de protocols i el suport de "drivers" pels dispositius.

Aquest nivell actua com a capa d'abstracció entre el hardware i la resta de la pila de software.

3.4. Android Wear

Android Wear[15] és el sistema operatiu que va crear Google i que va presentar el març de 2014 pels dispositius *Wearables*. Aquests poden ser rellotges, polseres, anells, en definitiva, qualsevol complement intel·ligent que una persona pot portar a sobre.

Aquest sistema, simplement és una optimització del sistema operatiu Android adaptant-se a aquests tipus de dispositius i les seves característiques, com per exemple, la mida de la pantalla.

Algunes de les seves funcionalitats és:

- Poder veure i gestionar les notificacions del telèfon Android
- Registrar l'activitat diària
- Accés a les aplicacions de Google (*Gmail*, *Google Keeper*, *Hangouts*, *Google Maps*, etc.)
- Gestionar el correu
- Poder realitzar accions mitjançant la veu. Dient "Ok Google" s'activa el reconeixement per veu i es pot començar a demanar informació a *Google Now*.

4. Implementació

4.1. Desenvolupament de l'aplicació

4.1.1. Ús de *SharedPreferences*

Les *SharedPreferences*[17] tenen la funcionalitat d'emmagatzemar un valor dins la memòria del dispositiu i tot seguit, poder llegir-lo des de qualsevol part de l'aplicació.

En aquest projecte s'utilitzen per poder guardar les dades relacionades amb el vincle entre el telèfon i el rellotge, exactament per poder tenir un únic dispositiu enllaçat.

El problema era que si hi havia més d'un dispositiu connectat al mateix mòbil, podia enviar la informació a un dels dos o als dos, podent enviar dades a un dispositiu que no fos de confiança (per exemple, a un SmartWatch de proves de l'empresa en la qual treballa l'usuari).

Per evitar aquest problema, quan l'usuari s'identifica a l'aplicació, aquesta comprova si hi ha cap dispositiu connectat. En cas afirmatiu, comprova si hi ha algun dispositiu ja enllaçat. En cas negatiu es mostra una alerta preguntant a l'usuari si vol permetre el vincle amb el rellotge connectat actualment. Si l'usuari indica que sí, en aquell moment s'obra la memòria de les *SharedPreferences* i s'insereixen quatre camps.

Per fer-ho solament fa falta instanciar la interfície *SharedPreferences* amb un nom (com si fos un nom d'una base de dades) i ja es pot accedir al contingut:

```
SharedPreferences prefs =  
getSharedPreferences(getString(R.string.app_prefs),  
Context.MODE_PRIVATE);
```

Amb aquesta nova instància es pot comprovar si té valors emmagatzemats de forma que es podrà saber si l'usuari ja ha enllaçat l'aplicació amb algun rellotge.

La informació que s'emmagatzema dins d'aquesta memòria té una estructura "clau-valor", tal com es veu a la Figura 27, i per obtenir les dades guardades solament cal buscar la clau.

key	value
firstName	Bugs
lastName	Bunny
location	Earth

Figura 27: Estructura "clau-valor"

Per fer-ho es pot utilitzar algun dels mètodes "get" dels que disposa la classe, passant-li el text o valor de la clau, i també marcant-li un valor per defecte en cas que no es trobi cap clau igual a la indicada:

```
boolean connectionPermittedToWatch =  
prefs.getBoolean("key_connect_watch", false);
```

Tal com es pot veure a la línia de codi, es vol obtenir el valor que té per clau "key_connect_watch" (primer paràmetre) i en cas de no trobar-lo, el valor per defecte serà fals (segon paràmetre).

El valor emmagatzemat indica si hi ha un rellotge vinculat, de forma que, fent aquests passos, solament falta comprovar si el valor és verdader o fals per saber si s'han utilitzat les *SharedPreferences* o no, i saber si s'ha enllaçat un SmartWatch o no.

Si no hi ha cap rellotge vinculat i l'usuari ha d'acceptar la unió dels dos dispositius actuals s'emmagatzemen les següents dades:

- Un valor per indicar que hi ha enllaç: valor que servirà per fer les comprovacions de si existeix un enllaç o no.
- L'identificador del node amb el qual s'ha enllaçat: node al qual s'enviaran els missatges i l'únic dispositiu que podrà accedir a les dades privades de l'usuari en la seva aplicació.
- El nom del dispositiu enllaçat: per mostrar el nom en la pantalla d'accions del SmartWatch.
- L'identificador de l'usuari identificat: útil per quan el rellotge demani dades del servidor i aquest tingui d'obtenir-les limitant-se a aquell usuari (per exemple, les seves targetes). Aquest valor, en una aplicació real, no s'hauria de guardar o s'hauria de xifrar, ja que identifica un usuari i es pot accedir a les seves dades amb aquella informació. En aquest projecte, al ser solament una prova, tenir totes les dades falses i tenir el temps limitat es pot permetre emmagatzemar-lo d'aquesta forma.

Per guardar-les dins de les *SharedPreferences* s'ha de fer utilitzant una altra interfície anomenada *SharedPreferences.Editor*[18]. Aquesta consta de mètodes "put" per poder emmagatzemar la

informació en forma de "clau-valor". En el següent codi es pot veure com s'emmagatzema l'identificador del node connectat.

```
//Instància de la interfície
SharedPreferences.Editor editor = prefs.edit();

//Afegeix amb clau "key_connected_node_id" l'identificador del
node connectat
editor.putString("key_connected_node_id", mNode.getId());

//Aplica els canvis dins de les SharedPreferences
editor.apply();
```

El problema que té aquesta memòria és que el fitxer on s'emmagatzema aquesta informació és accessible, sempre i quan el dispositiu sigui un emulador o un amb permisos d'administrador.

Per veure'l s'ha de mirar dins de la carpeta de l'aplicació generada al dispositiu, normalment seguint l'estructura de carpetes:

data → data → <paquet_de_aplicació> → shared_pref

Dins s'hi troba un fitxer amb extensió *xml* amb el nom que s'ha marcat quan s'han creat les "preferences", tal com es pot veure a la Figura 28.

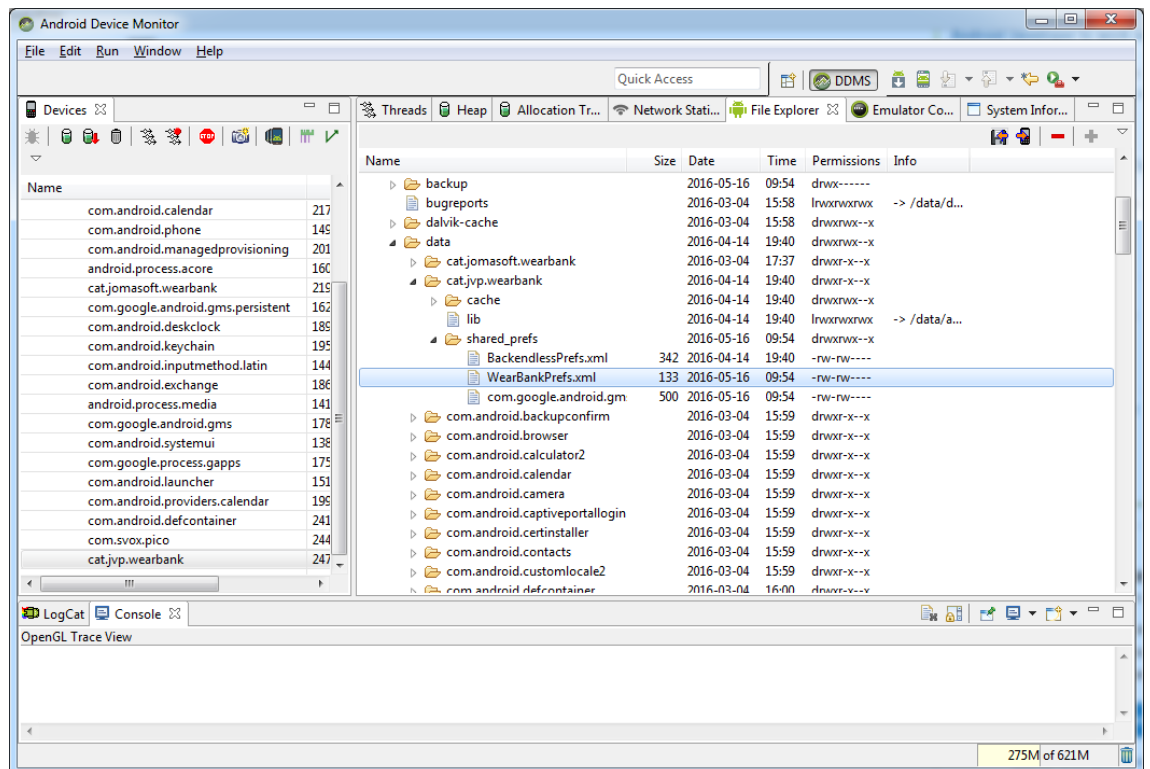


Figura 28: Arxiu de *SharedPreferences* creat per l'aplicació

Si s'examina el contingut del fitxer, es pot veure com s'ha guardat la informació introduïda en l'anterior pas (Figura 29).

```
<map>
  <string name="key_connected_node_display_name">SmartWatch 3 4185</string>
  <boolean name="key_can_connect_with_watch" value="true" />
</map>
```

Figura 29: Contingut del fitxer creat per les *SharedPreferences*

Tal com es pot veure, tota la informació guardada es pot llegir sense cap problema, per això s'ha comentat abans que alguna informació seria millor no guardar-la en aquest fitxer.

4.1.2. Google API Client i Wearable API

Per poder establir comunicació entre el telèfon mòbil i el rellotge s'utilitza la mateixa API de Google, l'anomenada *Google API Client*[19]. Aquesta permet accedir a tots els serveis que ofereix Google Play i gestionar la connexió entre l'usuari i qualsevol servei de Google.

En aquest projecte, l'API utilitzada és la *Wearable*[21], la qual ha estat creada per treballar amb Android Wear i permetre la comunicació entre els dos dispositius.

Per establir connexió i indicar quina API s'utilitzarà es fa de la següent forma:

```
GoogleApiClient googleApiClient = new GoogleApiClient.Builder(this)
    .addConnectionCallbacks(
        new GoogleApiClient.ConnectionCallbacks() {
            @Override
            public void onConnected(Bundle connectionHint) {
                Log.d(TAG, "onConnected: " + connectionHint);
                // Now you can use the Data Layer API
            }
            @Override
            public void onConnectionSuspended(int cause) {
                Log.d(TAG, "onConnectionSuspended: " + cause);
            }
        })
    .addOnConnectionFailedListener(
        new GoogleApiClient.OnConnectionFailedListener() {
            @Override
            public void onConnectionFailed(ConnectionResult result) {
                Log.d(TAG, "onConnectionFailed: " + result);
            }
        })
    // Request access only to the Wearable API
    .addApi(Wearable.API)
    .build();

googleApiClient.connect();
```

Google s'ha encarregat d'oferir un manual de com treballar amb els seus SmartWatch el qual es pot trobar en [22], i mostra, entre altres coses, com unir el rellotge i el telèfon per poder-se comunicar amb missatges.

Bàsicament, l'aplicació del SmartWatch està basada amb aquest tipus de comunicació. Quan es volen obtenir unes dades, per exemple, les targetes d'un usuari, s'envia un missatge demanant-les i el telèfon és l'encarregat de fer la petició a servidor i retornar les dades obtingudes. El problema dels missatges és que solament es poden enviar textos (*String*), per això, pel retorn de les dades s'utilitza la sincronització de "Data Items".

Un "Data Item" defineix una interfície de dades que s'utilitzen per sincronitzar informació entre el telèfon i el rellotge. Aquest està format per:

- *Payload*: Una array de bytes en la que es pot ficar qualsevol tipus d'informació. La memòria d'aquest està limitada a 100KB.
- *Path*: Nom amb el qual s'emmagatzemarà la informació. Aquest és únic i ha de començar amb una barra diagonal (per exemple, "/direcció").

Quan és modificat un objecte d'aquests, s'envia una notificació a qualsevol node connectat, pel qual s'ha d'implementar un "Listener" d'aquests esdeveniments.

Totes les notificacions enviades per canvi de dades o missatges es poden escoltar de dues formes. La primera és dins de la mateixa "Activity" de l'aplicació, en la qual es tindrà d'indicar que estarà escoltant aquests esdeveniments, però fent-ho d'aquesta forma, quan es tanqui la pantalla o l'aplicació, aquest "listener" també deixarà d'estar actiu, per tant, tots els missatges que puguin anar allí, no seran interpretats.

L'altra opció és a través d'un servei creat per estar escoltant aquests esdeveniments. Aquest s'executa en segon pla, de forma que no fa falta tenir l'aplicació activa perquè estigui funcionant. És útil per si l'aplicació no ha de realitzar cap acció a la interfície de l'usuari una vegada rep un canvi, o per si es necessita que en qualsevol moment estigui disponible per realitzar certes accions.

En l'aplicació realitzada al projecte, la part del mòbil sempre manté actiu aquest escoltador utilitzant el segon mètode comentat. Ho fa així, ja que és necessari que sempre estigui disponible per quan el rellotge demani qualsevol informació. De l'altra forma, el rellotge podria demanar dades però el telèfon solament respondria si l'aplicació estigués activa, això implicaria que el rellotge sempre estaria pendent de que l'aplicació del mòbil estigués activa, i no seria útil.

En el cas del SmartWatch, els "listeners" estan dins de les "Activities" corresponents, ja que el mòbil solament envia informació

al rellotge quan aquest ho demana, i les dades que rep les mostra per pantalla.

Per exemple, quan el *Wearable* accedeix a la pantalla de targetes, el que fa aquests és, una vegada establida la connexió entre els dos dispositius, envia un missatge al telèfon demanant la llista de targetes d'un usuari i, a més a més, indica que aquella "Activity" estarà escoltant els esdeveniments quan es modifiquin les dades del "Data Item". En les següents línies de codi es pot veure com s'afegeix el "listener" i com s'envia el missatge:

```
@Override
public void onConnected(Bundle bundle) {
    Wearable.DataApi.addListener(mGoogleApiClient, this);
    Wearable.MessageApi.sendMessage(mGoogleApiClient,
    nodeId, "getCards", null);
}
```

El telèfon rebrà aquest missatge i el gestionarà analitzant la direcció d'aquest. En aquest cas és "getCards", per tant, indica que vol la llista de targetes, la qual cosa farà que el dispositiu obtingui la llista del servidor i la sincronitzi al "Data Item".

Una vegada fet això, el rellotge rebrà la notificació del canvi i la mateixa "Activity" que havia enviat el missatge, rebrà la resposta amb la llista de targetes. Solament faltaria mostrar-les per pantalla.

El problema del "Data Item" és que si no detecta cap canvi a les seves dades, no notifica als dispositius connectats. Per exemple, si s'ha afegit una llista de moviments, i al cap d'una estona es torna a afegir la mateixa, no ho notifica. Això fa que, si l'aplicació està esperant la sincronització d'aquestes dades per poder mostrar-les, però realment no rep aquesta actualització, la "Activity" es queda penjada esperant la resposta.

Aquest fet es pot solucionar de dues formes. Si es detecta que no hi ha resposta, es pot intentar obtenir les dades que hi ha en aquell moment dins del "Data Item", el qual vol dir que són les mateixes que ja hi havia.

L'altra opció és fer que cada vegada que es sincronitzin dades dins d'aquesta capa d'informació, hi hagi un valor que sempre estigui modificat, de forma que farà executar l'esdeveniment. Per exemple, un valor que sigui la data actual. Aquest paràmetre sempre serà diferent, la qual cosa farà que hi hagi la notificació del canvi.

En la Figura 30 es pot veure d'una forma il·lustrada com funciona la API destinada a Android Wear.

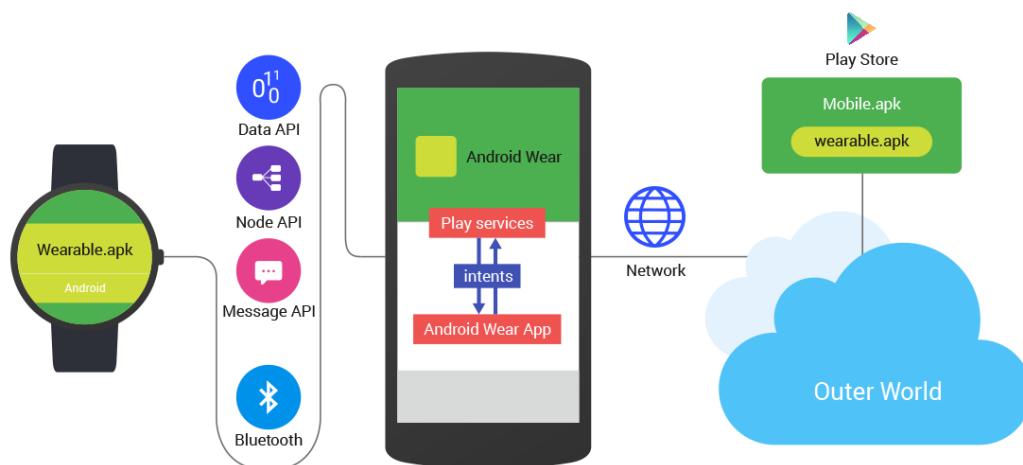


Figura 30: *Wearable API*

4.1.2.1. LocationService.API

Els SmartWatches estan dotats amb sensors per poder recollir dades i millorar les seves capacitats, però, igual que els telèfons mòbils, alguns tenen més sensors i altres menys.

Amb el GPS dels rellotges passa el mateix, alguns ho tenen i altres no, per això s'ha de controlar si s'utilitza dins d'una aplicació, per no fer que s'aturi inesperadament intentant accedir a un hardware que realment no té.

Per comprovar-ho des del codi, solament fa falta mirar les característiques del dispositiu i comprovar si té el sensor de localització:

```
boolean hasGPS = getPackageManager().hasSystemFeature(
    PackageManager.FEATURE_LOCATION_GPS);
```

Si el rellotge té GPS, llavors es podrà obtenir la localització del dispositiu. Per fer-ho, la Google API Client ofereix una API anomenada "LocationServices.API"[20] la qual gestiona la localització de la posició dels Android Wear. Solament fa falta afegir-la dins de la llista d'APIs que utilitzarà el Google Client quan es connecti.

```
GoogleApiClient mGoogleApiClient =
    new GoogleApiClient.Builder(this)
        .addApi(Wearable.API)
        .addApi(LocationServices.API)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build();
mGoogleApiClient.connect();
```

Per estar escoltant els esdeveniments del canvi de posició, la "Activity" ha d'implementar la classe "LocationListener" la qual rep les notificacions quan la localització canvia i es pot obtenir aquesta nova ubicació amb el seu mètode abstracte "onLocationChanged".

Finalment sol falta iniciar inicial la petició de localització, que això es farà quan el client de Google s'hagi connectat.

```
@Override
public void onConnected(Bundle bundle) {
    LocationRequest locationRequest = LocationRequest.create()
        .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
        .setInterval(UPDATE_INTERVAL)
        .setFastestInterval(FATEST_INTERVAL);

    LocationServices.FusedLocationApi
        .requestLocationUpdates(mGoogleApiClient, locationRequest,
            this)
        .setResultCallback(new ResultCallback<Status>() {
            @Override
            public void onResult(Status status) {
                if (status.getStatus().isSuccess()) {
                    Log.d("REQUEST LOCATION", "Successfully requested
                        location updates");
                } else {
                    Log.e("REQUEST LOCATION", "Failed in requesting --
                        location updates, "
                            + "status code: "
                            + status.getStatusCode()
                            + ", message: "
                            + status.getStatusMessage());
                }
            }
        });
}
```

L'aplicació creada a aquest projecte s'ha pogut provar amb un rellotge amb GPS (el Sony SmartWatch3), però no amb un que no en tingues (solament amb el simulador de dispositius que incorpora el SDK d'Android).

4.1.3. Llibreria *EventBus*

EventBus és una llibreria "open-source" per Android que ajuda a simplificar la comunicació entre diferents parts de l'aplicació, per exemple, entre diferents fils d'execució. Això ho fa utilitzant el patró "publisher/subscriber" o també anomenat patró observador, el qual fa que un o més objectes (observadors) es subscriguin a les publicacions d'un altre objecte, de forma que, quan el segon canvia d'estat, ho notifica a tots els objectes dependents.

La Figura 31 és la que utilitza la mateixa llibreria en la seva web per explicar el funcionament, i mostra d'una forma molt clara com un publicador pot enviar un esdeveniment i aquest arribarà a tots els seus subscriptors.

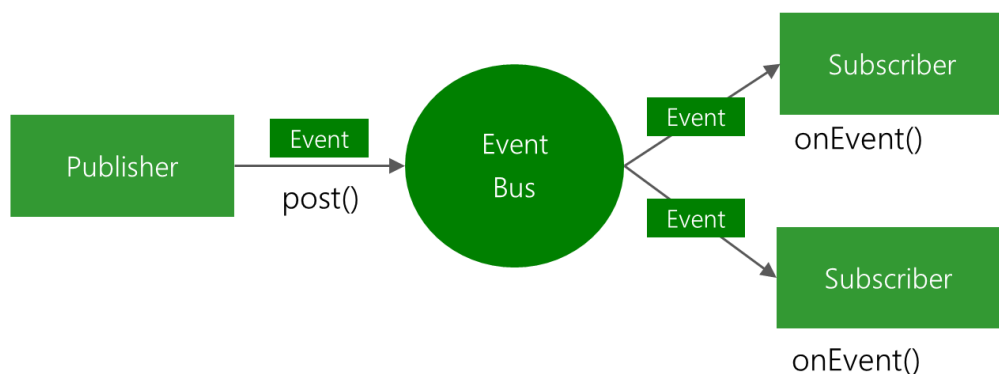


Figura 31: Funcionament de la llibreria *EventBus*

Dins del projecte hi ha un publicador que és un servei que el que fa és obtenir la posició del telèfon i, una vegada obtinguda, es busquen les oficines més properes a aquella ubicació. Una vegada obtinguda una llista de bancs, el servei ho notifica als seus subscriptors, afegint dins dels esdeveniments la llista obtinguda:

```
EventBus.getDefault().post(  
    new FindOfficesEvent(listOffices));
```

FindOfficesEvent és una classe model que conté la llista d'oficines.

Com a subscriptors hi ha dues classes. La primera és la "Activity" encarregada de la pantalla d'oficines de l'aplicació mòbil. Aquesta, primer de tot comprova si el dispositiu té la versió d'Android 6, i en cas afirmatiu, s'ha de comprovar si l'aplicació té permisos d'ús del GPS. Si no és Android 6 i/o té autorització per utilitzar el GPS, el següent pas és comprovar que el GPS estigui actiu. Si està actiu, s'executa el servei comentat i espera a rebre la notificació del publicador. Una vegada rebut el canvi, mostra la llista per pantalla.

```
@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent (FindOfficesEvent
findOfficesEvent) {
    createListViewOffices (findOfficesEvent.getListOffices ());
}
```

L'altre subscriptor és el servei que implementa la classe "WearableListenerService" la qual forma part de la API de *Wearable*. Aquest rep els missatges enviats des del rellotge, els intercepta i, mirant el seu contingut, realitza una sèrie de tasques. Una d'aquestes és la d'obtenir la llista d'oficines utilitzant el GPS del telèfon.

El missatge que demana aquesta tasca s'envia des del rellotge quan l'usuari accedeix a la pantalla d'oficines. Aquesta "Activity" realitza les mateixes comprovacions que les comentades abans en l'aplicació mòbil, però n'afegeix una més, la qual comprova si el dispositiu té GPS o no, pel fet que hi ha dispositius SmartWatch que no en tenen. En cas de no tenir-ne, el que fa és demanar al telèfon que busqui les oficines utilitzant la seva posició.

D'aquesta forma l'aplicació mòbil rep el missatge i executa el servei indicat com a publicador i, igual que abans, quan acaba la seva execució, envia el resultat als seus subscriptors, en aquest cas, el servei que implementa la classe "WearableListenerService". Aquest, una vegada rep l'actualització, l'afegeix al "Data Item" perquè se sincronitzi amb les dades del rellotge i aquest últim pugui mostrar la llista d'oficines que s'han trobat.

4.1.4. Backend *Backendless*

Backendless[24] és un "mBaaS (mobile backend as a service)" el qual ofereix un núvol d'emmagatzematge juntament amb un conjunt d'eines per poder comunicar-se amb aquest, sigui obtenint, guardant, esborrant o modificant dades. També ofereix notificacions "push", escalabilitat i altres característiques que fan que el desenvolupador solament hagi de centrar en l'aplicació que està creant, ja que la part "backend" de l'aplicació la dona feta aquesta aplicació.

Hi ha molts altres programes que ofereixen les mateixes funcionalitats o semblants. Un dels coneguts era *Parse*[25] però l'empresa que el portava (*Facebook*) ha decidit tancar-lo a finals d'aquest any.

S'ha decidit utilitzar *Backendless* perquè és molt similar a *Parse*, i ja es tenia coneixement previ sobre aquest, la qual cosa donava una mica d'avantatge al desenvolupament.

La seva inicialització és molt simple. Solament fa falta seguir els passos indicats a la seva web els quals expliquen com crear un projecte, obtenir la seva identificació i la clau privada i com inicialitzar l'aplicació Android amb aquest "backend".

Una vegada fet això ja solament falta utilitzar els mètodes necessaris per realitzar les tasques definides, com per exemple, identificar-se (fer "login") o obtenir dades filtrades.

4.2. Proves de l'aplicació

Durant el desenvolupament d'aquest projecte totes les proves s'han realitzat amb els següents dispositius:

- SmartPhones
 - LG Nexus 5 amb la versió 6 d'Android (API 23)
 - Emulador: LG Nexus 5 amb la versió 5 d'Android (API 21)
- SmartWatch
 - Emulador:
 - Android Wear Square
 - Versió 6 d'Android (API 23)
 - Sense GPS
 - Pantalla quadrada
 - Dispositiu físic:
 - Sony Smartwatch 3
 - Versió 6 d'Android (API 23)
 - Amb GPS
 - Pantalla quadrada

No s'ha pogut provar amb més dispositius ni amb més versions d'Android a causa de la limitació del temps (aquest fet queda com a plans de futur)

5. Aplicació final

5.1. Aplicació "Mobile"

5.1.1. Pantalla inicial

La pantalla inicial es pot veure a la Figura 32 i es podria dir que està formada per tres seccions. A la part inferior la publicitat, en aquest cas sobre assegurances de la llar. A la part esquerra l'accés a la pantalla d'identificació per accedir a les dades privades d'un usuari (explicat a l'apartat 5.1.3). I a la part dreta les accions que es poden realitzar sense necessitat d'identificació.

Aquesta es divideix en dos botons. Començant per la part de baix, el primer botó que hi ha és el de trucar al telèfon d'assistència del banc. Si es pitja, automàticament comença una trucada a un telèfon (Figura 33) (el telèfon marcat és inventat).

El següent botó és el que accedeix a la Pantalla d'oficines, el qual s'explica al següent apartat.

Finalment, a la barra superior on hi ha el títol (anomenada "ActionBar"), a la part de la dreta hi ha un botó que permet accedir a la Pantalla d'accions del rellotge.

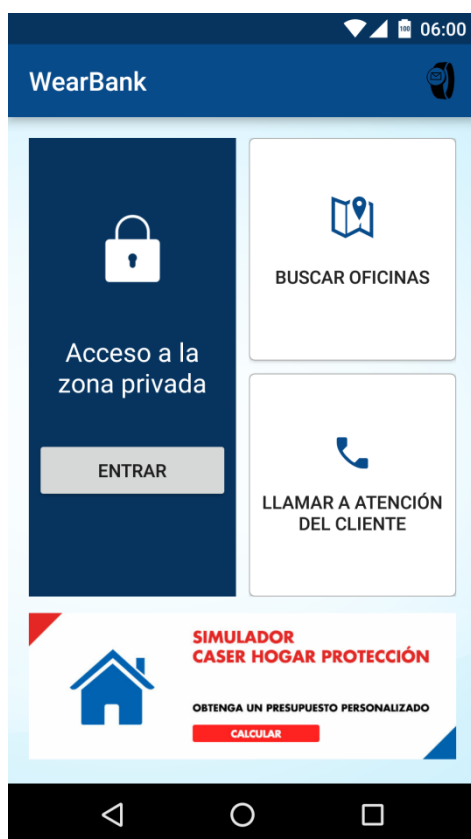


Figura 32: Pantalla inicial

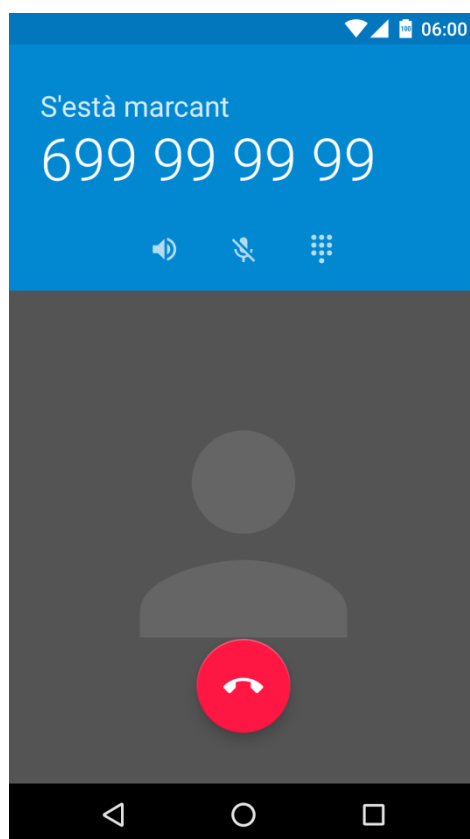


Figura 33: Pantalla de trucada

5.1.2. Pantalla d'oficines

Aquesta pantalla és l'encarregada d'obtenir la llista d'oficines més properes a la posició de l'usuari. Per fer-ho, primer de tot necessita obtenir la ubicació del dispositiu i amb aquesta realitzarà la petició al servidor.

Per obtenir-la, es necessita tenir el GPS del mòbil activat, i en cas d'utilitzar un dispositiu amb la versió 6 d'Android, també és necessari tenir permisos d'ús, per això, les primeres tasques que realitza la classe són les de comprovar si l'aplicació disposa de permisos d'execució del GPS i seguidament, si es té accés a aquest, comprova que estigui activat.

Si el GPS està bloquejat o és la primera vegada que s'executa aquella classe, es mostrarà una alerta com la de la Figura 34. Aquesta alerta es mostra quan s'executa la següent línia de codi:

```
requestPermissions(  
    new String[]{  
        Manifest.permission.ACCESS_FINE_LOCATION,  
        Manifest.permission.ACCESS_COARSE_LOCATION},  
    Constants.MY_PERMISSIONS_REQUEST_LOCATION);
```

Aquesta sol·licita permisos pels valors que s'indiquen dins del vector de "String".

A continuació, cal comprovar si el GPS està actiu. En aquest cas, si no ho està, el desenvolupador pot escollir què fer. Al projecte s'ha creat una alerta indicant el problema i deixant que l'usuari esculli l'opció que cregui convenient (clarament, si tria el "no", l'execució s'atura i no es mostren oficines).

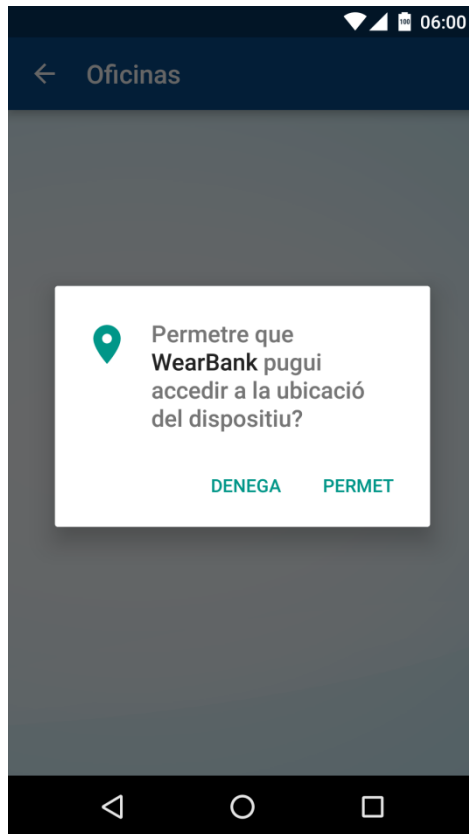


Figura 34: Sol·licitud de permisos de GPS

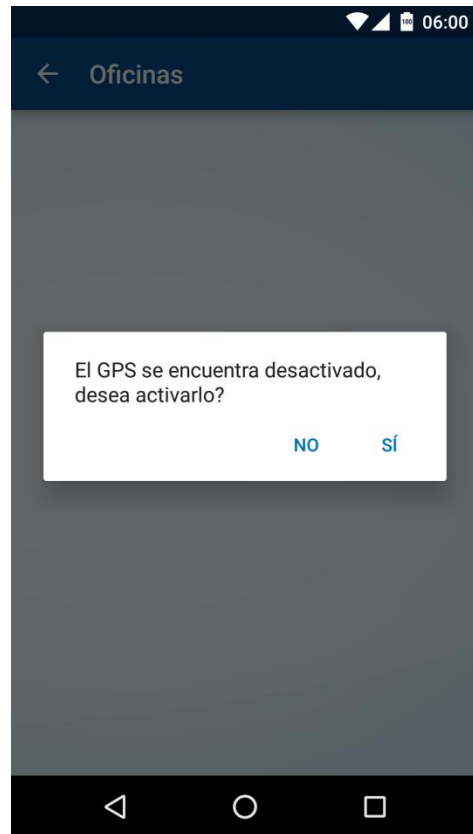


Figura 35: Sol·licitud d'activació del GPS

Una vegada superades les validacions, s'executa el servei encarregat d'obtenir la posició i la llista d'oficines. Mentre aquest està treballant es mostra un diàleg de progrés com el de la Figura 36 per mostrar a l'usuari que l'aplicació està treballant (aquest diàleg apareix a totes les pantalles que requereixin dades del servidor, ja que la resposta d'aquest pot tardar una estona).

Per obtenir el llistat d'oficines amb un radi de distància, el "backend" ofereix un mètode que ho fa. Es tracta de crear una condició a la petició indicant que solament es vol obtenir les oficines a X distància d'una posició.

```
String whereClause = "distance( " +
    actualLocation.getLatitude() + ", " +
    actualLocation.getLongitude() + ", geolocation.latitude,
    geolocation.longitude) < km(" + radius + ")";
```

En cas d'obtenir una llista buida, s'augmenta el radi i es torna a executar el mètode fins que es trobi almenys una oficina.

En una aplicació de producció, segurament es faria d'una altra forma, per exemple, solament enviant la localització del dispositiu, el mateix servidor ja analitzaria quines són les oficines més properes i les enviaria com a resposta de la petició juntament amb la distància de cadascuna.

Per acabar, una vegada obtingudes les oficines, es mostren per pantalla dins d'una llista (Figura 37) en la qual cada ítem té un format personalitzat per mostrar la màxima informació: la ciutat o població en la qual està situada l'oficina, el carrer, i la distància des del punt de l'usuari fins a arribar-hi.

El fet de pitjar un dels elements farà que s'obri *Google Maps* i fiqui un marcador a la localització de l'oficina.

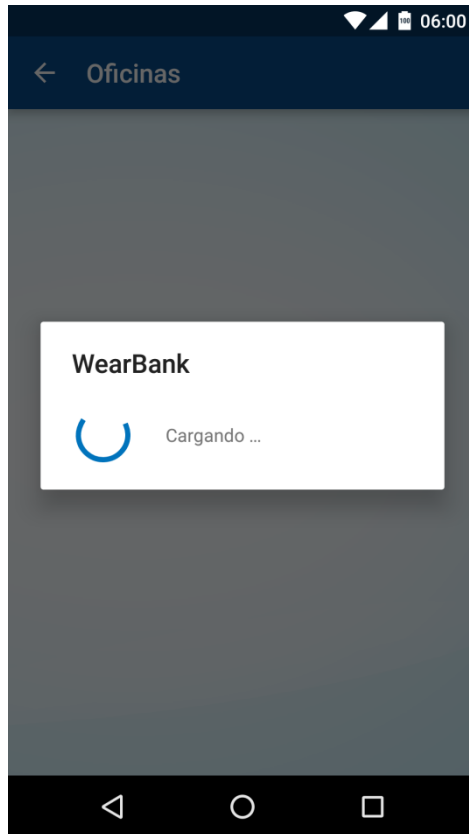


Figura 36: Diàleg de progrés

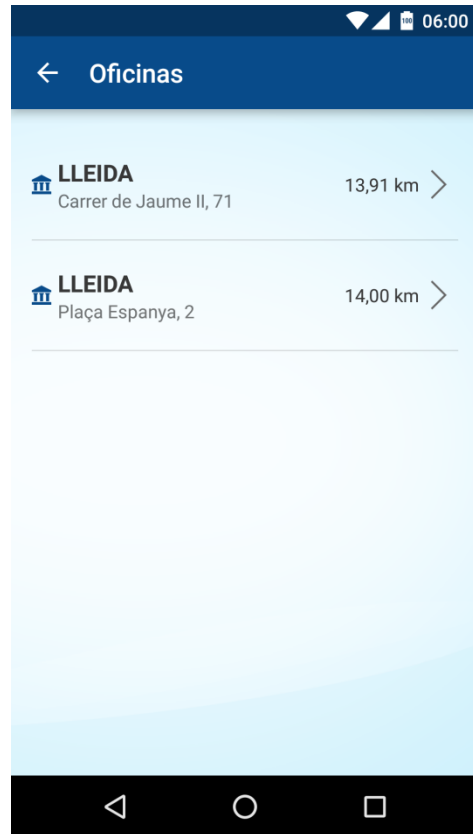


Figura 37: Llista d'oficines

5.1.3. Pantalla d'identificació

La pantalla d'identificació es pot veure a la Figura 38 i, bàsicament, està formada per un formulari que consta de dos camps de text i per un botó.

Si l'usuari no omple els camps i pitja el botó o fica les credencials malament, es mostrarà un missatge indicant el problema.

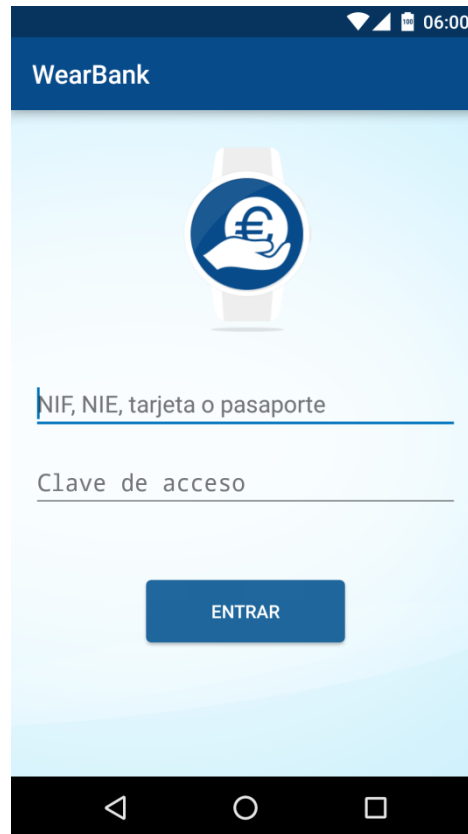


Figura 38: Pantalla d'identificació

5.1.4. Pantalla de la zona privada

Una vegada identificat l'usuari, s'accedeix a la zona privada en la qual solament poden accedir clients amb credencials.

Quan s'obre aquesta pantalla, es realitza un seguit de comprovacions per poder detectar si hi ha un rellotge connectat i si aquest té permisos per poder utilitzar les credencials de l'usuari.

Primer de tot activa la *Wearable API* per detectar si es pot connectar a algun dispositiu i, si ho fa, vol dir que hi ha un SmartWatch enllaçat amb aquell mòbil. A continuació obté l'identificador d'aquest rellotge i comprova si l'aplicació té algun dispositiu vinculat. Aquesta comprovació la fa mirant els valors emmagatzemats dins de les *SharedPreferences* (4.1.1). Si no hi ha cap, valor vol dir que no hi ha hagut cap vincle, i per tant, que es pot realitzar amb el dispositiu

connectat. En aquest moment es mostra una alerta per pantalla preguntant si es permet la unió dels dos dispositius de forma que permetrà que el rellotge pugui accedir a les seves dades privades (Figura 39).

Si l'usuari ho accepta, es fiquen els següents valors dins la memòria del dispositiu per indicar que hi ha un vincle i s'envia un missatge al rellotge indicant que s'ha creat l'enllaç i per tant que ja té accés a les dades privades de l'usuari:

- Un valor que indica que hi ha hagut un vincle
- L'identificador de l'usuari que ha permès l'accés a les seves dades
- L'identificador del rellotge al qual se li ha permès l'accés
- Nom del rellotge per mostrar-lo en la Pantalla d'accions del rellotge.

Tota aquesta informació és útil per fer validacions en futures connexions, ja que una aplicació bancària requereix seguretat. Per exemple, per comprovar que l'usuari que s'ha identificat és el mateix que el que va permetre que el rellotge pogués accedir a les seves dades. Si no es compleix aquesta condició, es mostra una alerta a l'usuari indicant que hi ha un rellotge connectat que té accés a les dades i preguntant si vol mantenir la connexió i permetre que utilitzi les seves dades, o vol anul·lar-la (Figura 40). En cas d'anul·lar-la, s'elimina el contingut de la memòria i s'envia un missatge al rellotge indicant la fi del vincle.

En cas contrari, s'actualitza l'identificador de l'usuari de la memòria amb el nou usuari.

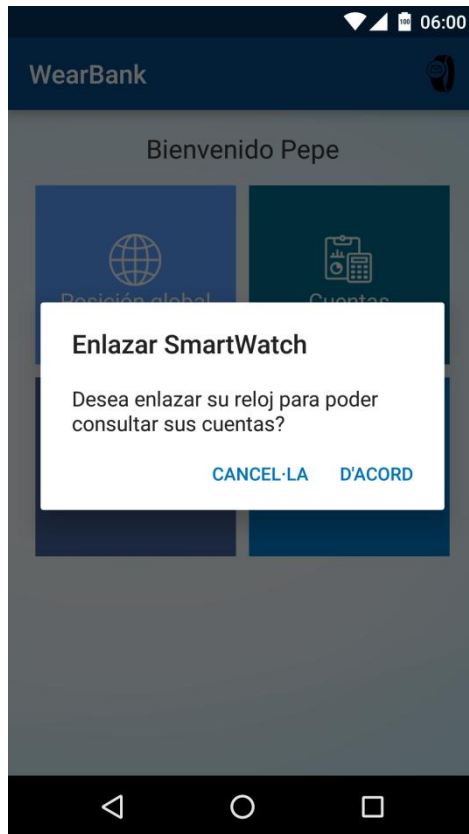


Figura 39: Alerta d'enllaç entre els dispositius

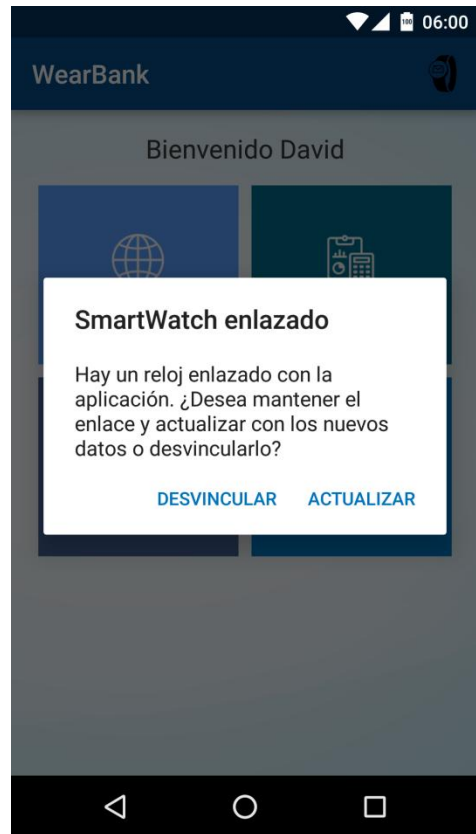


Figura 40: Alerta de canvi d'usuari amb un SmartWatch vinculat

Una vegada superades aquestes comprovacions es mostra la Figura 41 la qual mostra la pantalla principal de la zona de clients.

Aquesta la formarien tots els botons per poder accedir a totes les possibles funcionalitats de l'aplicació, en aquest cas, accedir a les targetes i comptes.

La funcionalitat de "Posició global" no està implementada, però dins d'aquest es podria utilitzar gràfics per mostrar la informació. El botó d'oficines porta a la mateixa pantalla que la que s'ha comentat en una de les anteriors seccions.

Finalment el botó de targetes i el de comptes obren la pantalla que mostra la llista de targetes de l'usuari identificat (el botó de comptes també obre la pantalla de targetes, ja que tindrien el mateix estil).

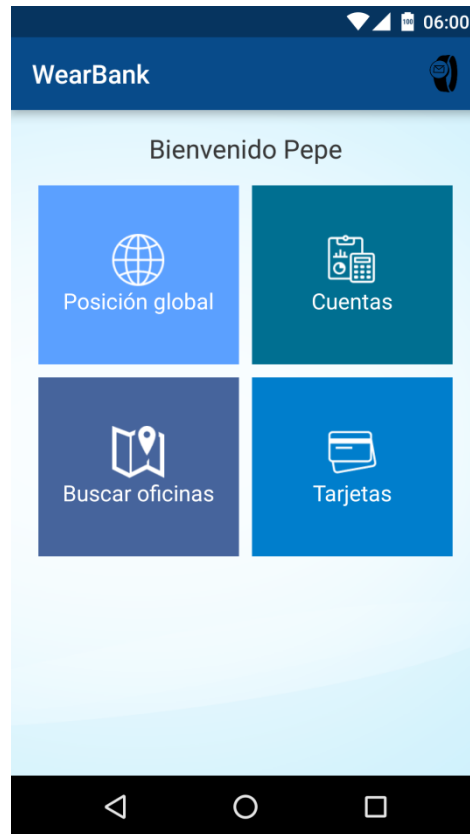


Figura 41: Pantalla principal de la zona de clients

Aquesta pantalla també disposa del botó que permet accedir a la Pantalla d'accions del rellotge.

5.1.5. Pantalla de targetes

La pantalla de targetes, tal com diu el seu nom, simplement obté la llista de targetes de l'usuari identificat i les mostra per pantalla, tal com es pot veure a la Figura 42.

Si la llista de targetes està buida, es mostra un text per defecte a la pantalla indicant que no n'existeix.

Aquestes targetes es diferencien si són de crèdit o de dèbit. Si és el primer cas, es mostra el deute pendent amb aquella targeta (en cas de tenir-ne). Si és el segon cas, es mostra el saldo de la targeta.

Si es pitja a qualsevol de les targetes, s'accedirà a una pantalla on es podran veure tots els moviments que s'han realitzat amb aquella targeta.

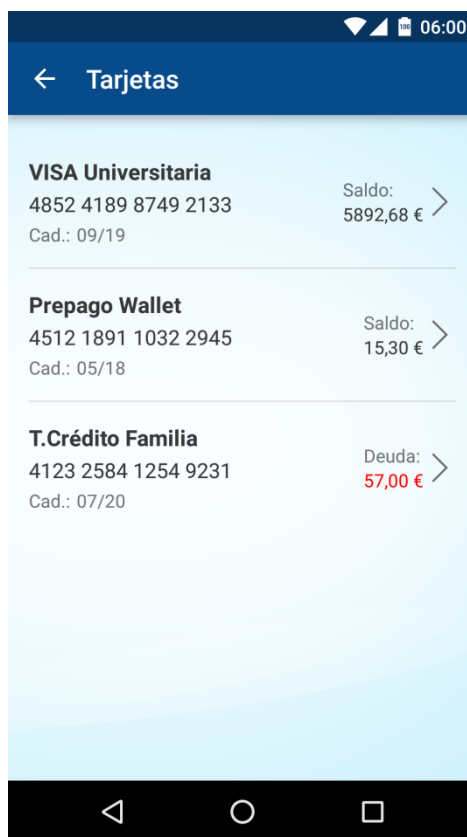


Figura 42: Llista de targetes d'un usuari

5.1.6. Pantalla de moviments

Igual que abans, la pantalla de moviments solament mostra la llista de moviments realitzats per una targeta (Figura 43).

A la part superior de la pantalla es pot veure el número i el nom de la targeta amb la qual s'han realitzat aquests moviments. Cadascun té una fletxa verda o roja indicant si el moviment és positiu o negatiu.

Finalment, per veure més informació d'un moviment, solament fa falta pitjar a sobre d'aquest.

Si una targeta no té moviments, es mostra un text per defecte indicant-ho.

VISA Universitaria 4852 4189 8749 2133		
↓	Invicat Transporte	-2,40 € 14/05/2016
↓	Recargo Transpaso	-40,00 € 13/05/2016
↓	Sushi Restaurant Restaurantes	-51,74 € 13/05/2016
↓	Repsol Transporte	-42,00 € 21/04/2016
↓	Fondo inversión Transpaso	-150,00 € 21/04/2016
↑	Ingreso nómina Nómina	1121,54 € 21/04/2016

Figura 43: Llista de moviments d'una targeta

5.1.7. Pantalla de detall d'un moviment

Tal com es pot veure a la Figura 44, aquesta pantalla simplement mostra tota la informació que es coneix sobre un moviment. Si algun dels camps de la llista no existeix, s'amaga automàticament. Per exemple, solament alguns moviments tindran localització, la qual cosa fa que en uns aparegui la fila i en altres desaparegui.

The screenshot shows a mobile application interface with a dark blue header containing a back arrow and the text 'Invicat'. Below the header is a light blue area with a list of transaction details, each separated by a horizontal line. The details are: Concepto (Invicat), Categoría (Transporte), Fecha operación (14/05/2016), Hora operación (18:19), Localidad (Martorell), Tarjeta (4852 4189 8749 2133), and Importe (-2,40 €). The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Concepto	Invicat
Categoría	Transporte
Fecha operación	14/05/2016
Hora operación	18:19
Localidad	Martorell
Tarjeta	4852 4189 8749 2133
Importe	-2,40 €

Figura 44: Detalls d'un moviment

5.1.8. Pantalla d'accions del rellotge

Durant el desenvolupament de l'aplicació, hi havia accions que no es podien simular o que s'havien de fer d'una forma molt costosa, la qual cosa va fer que es creés una pantalla destinada a poder controlar aquestes accions. Aquesta es pot veure a la Figura 45 (per accedir-hi s'ha d'utilitzar el botó en forma de rellotge situat a la part superior a la dreta de la pantalla inicial i la pantalla principal de la zona privada).

Està formada per dues seccions, una per conèixer el rellotge que hi ha vinculat i l'altra per simular o realitzar accions.

La primera mostra el nom del dispositiu vinculat (per això es guardava el nom del rellotge dins de les *SharedPreferences*) i si es pitja a sobre, es mostra una alerta que permet desvincular el rellotge de l'aplicació (Figura 46). Si s'accepta, es neteja la memòria del dispositiu i s'envia un missatge al *Wearable* indicant el canvi.

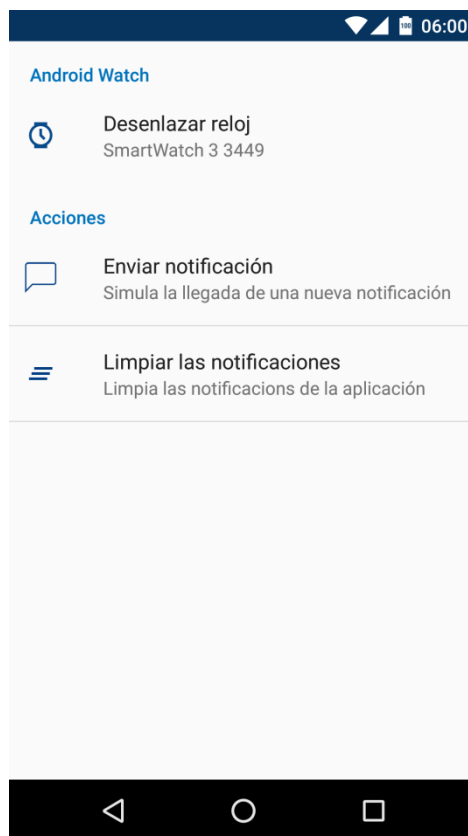


Figura 45: Pantalla d'accions del rellotge

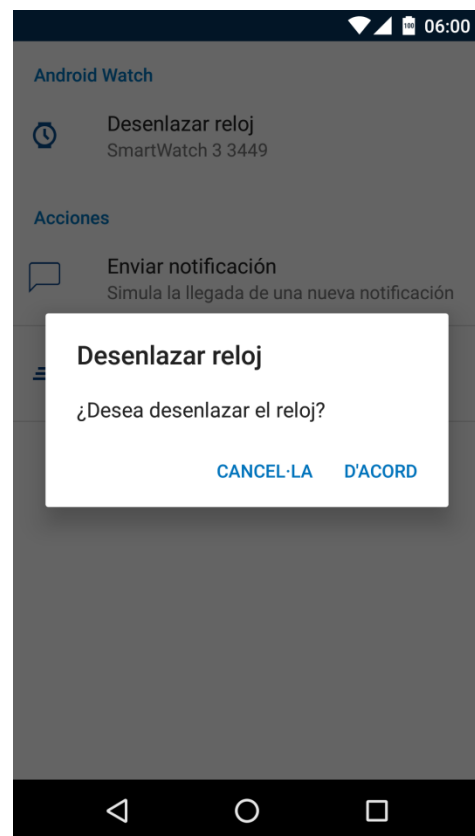


Figura 46: Desvinculació del SmartWatch

L'altra secció la formen dues parts. Una per crear notificacions i l'altra per esborrar-les.

Un dels requeriments principals del projecte era que l'aplicació i el rellotge fossin capaços de rebre notificacions i gestionar-les. Per simular-les, es creen i s'envien des del mateix mòbil (en una aplicació real, aquestes notificacions s'enviarien a través de "Push Notifications" des del servidor). Si es pitja sobre l'ítem d'enviar notificacions, s'obra un menú que permet seleccionar entre tres missatges diferents per enviar.

Cada missatge afegeix unes accions diferents, les quals, quan es pitgin, obriran l'aplicació en una pantalla diferent. Per exemple, el primer missatge disponible de la llista parla sobre que l'usuari ha d'anar a l'oficina, per tant, una de les accions que incorporarà la notificació serà la d'obrir la pantalla amb el llistat d'oficines per poder veure quina és la més propera. (el llistat d'accions està limitat a les notificacions que apareguin al rellotge).

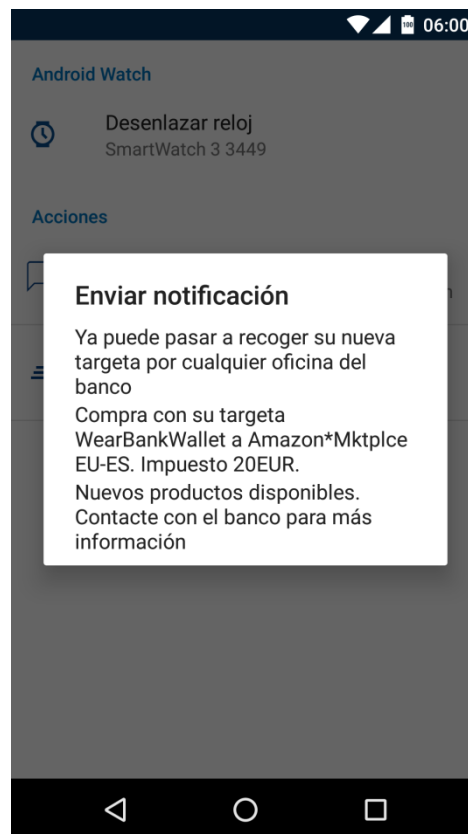


Figura 47: Selecció de notificacions

5.2. Aplicació "Wear"

5.2.1. Pantalla inicial

Quan s'executa l'aplicació al rellotge, es mostra la pantalla de la Figura 48 mentre s'estableix la connexió entre el telèfon i el SmartWatch per a poder-se comunicar.

Aquesta es fa utilitzant la API de *Wearable*, comentada a l'apartat 4.1.2, i que consisteix a crear una instància de l'objecte *GoogleApiClient*, connectar-lo, i esperar el resultat d'aquest.

En aquest punt poden passar dues coses: que s'estableixi la connexió o que no. Si no ho fa, es mostra la imatge de la Figura 49 la qual indica el problema i permet tancar l'aplicació.

Si ho fa, segueix amb l'execució obtenint l'identificador del node (dispositiu) al qual està connectat i guardant-lo per poder enviar els futurs missatges.

El primer d'aquests s'envia a continuació, el qual pregunta al dispositiu si aquell rellotge té autorització per accedir a les dades privades de l'usuari i també per comprovar que la connexió està bé i que el dispositiu mòbil té l'aplicació del banc instal·lada, ja que, si no l'ha té, no realitzarà cap acció que pugui demanar el rellotge. Per comprovar que tot això està bé, es crea un *timeout* el qual, una vegada passats 15 segons des de l'enviament del missatge, comprova si s'ha rebut resposta.

```
new android.os.Handler().postDelayed(  
    new Runnable() {  
        public void run() {  
            if(!receivedResponseDevice) {  
                //No se recibió ninguna respuesta  
            }  
        }  
    },  
    15000); //15 segundos
```



Figura 48: Pantalla d'inici

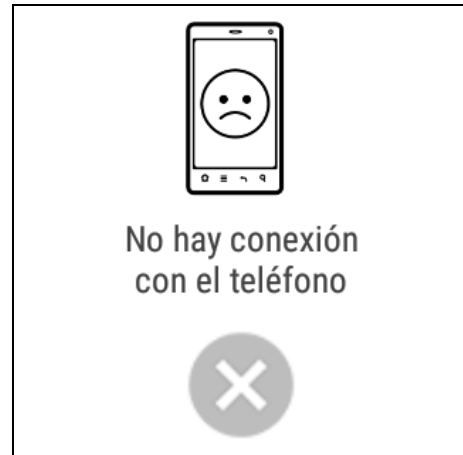


Figura 49: Pantalla d'error de connexió

Si no hi ha hagut cap contesta, es mostra la Figura 50 indicant el problema.

Si tot funciona bé, el telèfon mòbil rebrà el missatge i comprovarà si l'usuari ha creat un vincle, i en cas afirmatiu, si aquest enllaç s'ha fet amb aquell SmartWatch concret. A continuació el telèfon enviarà un missatge amb la resposta d'aquestes comprovacions.

Una vegada rebuda la resposta se sabrà si es pot donar a aquell rellotge els permisos per accedir a qualsevol de les dades de l'usuari.

A partir d'aquest moment, el rellotge ja pot funcionar amb o sense aquests permisos, però la diferència la marcarà el menú principal, el qual mostrarà més o menys accions depenent de la resposta del telèfon.

Si aquesta contesta és negativa, és a dir, que no es pot accedir a la zona privada, es mostra l'alerta de la Figura 51, la qual indica que es pot utilitzar l'aplicació però amb una llista de funcionalitats reduïda. La pantalla té dos botons de forma que l'usuari pot triar si vol continuar o no amb l'execució. Aquesta alerta està creada amb una vista personal dins del projecte, ja que Andriod Wear no ofereix una pantalla per mostra alertes, sinó que s'han de crear manualment.

Si es pitja el botó amb forma de creu, igual que les anteriors pantalles, l'aplicació es tancarà. Si es prem el botó verd, s'accedeix a la pantalla amb el menú principal.



Figura 50: Pantalla d'error de comunicació

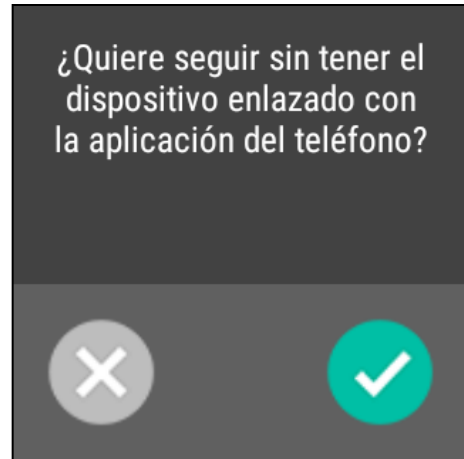


Figura 51: Alerta d'execució sense identificació

5.2.2. Pantalla principal

La pantalla principal la forma una llista d'opcions que es crea dependent de l'accés o no a les dades privades de l'usuari. Si no es pot accedir, es mostra la llista de la Figura 52 la qual conté solament dues accions.

Si es pot accedir, es mostra la Figura 53 la qual augmenta la llista de les seves accions permetent accedir a les targetes i comptes de l'usuari.

L'estil de la llista s'ha hagut de crear manualment, ja que Android Wear no ofereix cap llista amb el mateix estil que s'utilitza al sistema operatiu en, per exemple, el menú d'aplicacions o de contactes.

Mentre la pantalla està activa, la comunicació amb el telèfon es manté oberta, d'aquesta forma, en cas que el mòbil envii un missatge al rellotge indicant que l'usuari ha acceptat el vincle, el menú es pot actualitzar i es poden mostrar totes les funcionalitats. També serveix al revés, si l'usuari decideix desvincular el rellotge, aquest és avisat i s'amaguen les opcions de targetes i comptes.



Figura 52: Pantalla principal sense vincle

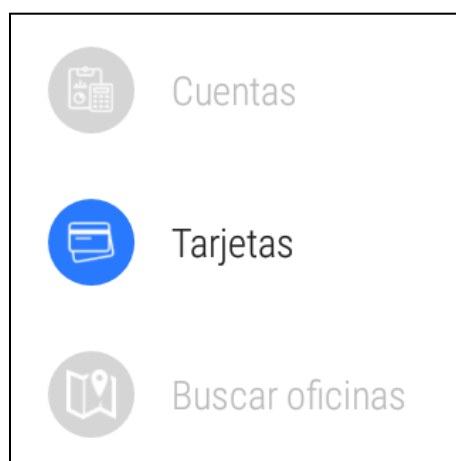


Figura 53: Pantalla principal amb vincle

5.2.3. Pantalla d'oficines

L'obtenció de les oficines més properes, igual que en l'aplicació mòbil, ha estat de les parts més complicades de desenvolupar.

La llista d'oficines resultant és la que es pot veure a la Figura 54 i mostra les mateixes dades que l'aplicació mòbil: la ciutat o població en la qual es troba l'oficina, la direcció i finalment, la distància des de la posició del GPS fins al banc.

El problema dels rellotges és que n'hi ha que tenen GPS i altres no, la qual cosa fa que s'hagin de controlar els dos casos.

Aquesta és la primera comprovació que fa la classe una vegada s'accedeix a ella.

Si el dispositiu pot obtenir la ubicació (té GPS), s'utilitzarà el mateix. En cas contrari, s'informarà al telèfon que s'ha d'utilitzar la seva posició per obtenir el llistat d'oficines. En aquest mateix moment també es mostra un missatge per pantalla indicant a l'usuari que s'utilitzarà la ubicació del telèfon (Figura 55). Si per algun motiu el mòbil no fos capaç d'accedir al GPS (ubicació desactivada o falta de permisos d'execució en la versió 6 d'Android), aquest indica el problema al rellotge i aquest mostra el missatge de Figura 56 per què l'usuari revisi que pot estar causant el problema. Si tot funciona correctament, el mòbil sincronitzarà la llista d'oficines al "Data Item" i el rellotge ja podrà mostrar-la per pantalla.



Figura 54: Llista d'oficines

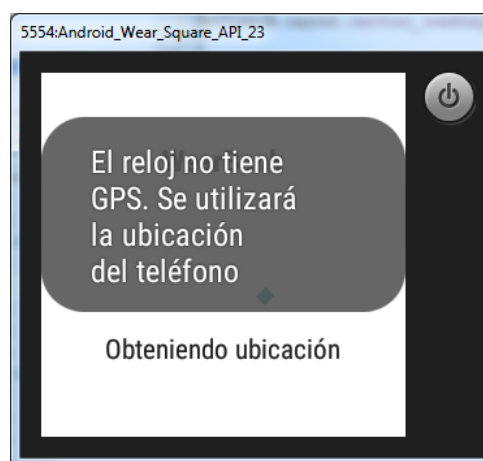


Figura 55: Missatge de GPS no existent

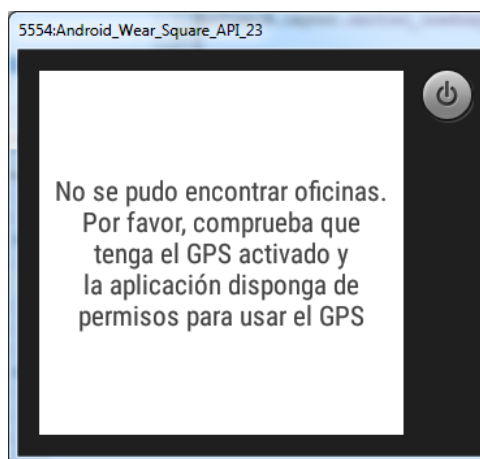


Figura 56: Missatge d'error d'accés al GPS

Si el rellotge disposa del propi GPS, el flux canvia. Ell mateix obté la seva posició i l'envia al telèfon, i aquest solament ha d'obtenir la llista d'oficines amb la localització rebuda. A continuació la sincronitza amb el *wearable* i aquest la mostra per pantalla

Però per poder fer això, s'ha de comprovar si el dispositiu té instal·lada la versió 6 d'Android i, en cas de tenir-la, validar si es té

permisos per accedir al GPS del dispositiu. En cas de no tenir-ne, es mostra una alerta del mateix sistema sol·licitant-lo.

5.2.4. Pantalla de targetes

Quan s'accedeix a aquesta "Activity", es realitzen dues tasques. La primera és la de sol·licitar a l'aplicació del telèfon la llista de targetes de l'usuari que ha permès el vincle.

La segona s'executa quan rep la llista demanada, i és la que s'encarrega de mostrar les targetes per pantalla (Figura 57).

Si la llista de targetes està buida, es mostra un text per defecte a la pantalla indicant que no n'existeix.

Igual que en el mòbil, si la targeta és de crèdit, es mostrarà el seu deute, en cas de ser de dèbit, es mostra el seu saldo.

Si es pitja a qualsevol dels elements de la pantalla, s'obrirà una altra pantalla on es podran veure tots els moviments que s'han realitzat amb aquella targeta.



Figura 57: Pantalla amb la llista de targetes

5.2.5. Pantalla de moviments

La pantalla de moviments realitza les mateixes tasques que l'anterior, però en aquest cas, amb moviments.

La llista de la Figura 58 mostra el concepte de cada moviment, la data d'aquest i el cost. Aquest últim el pinta de color roig o verd juntament amb una fletxa que indica si el moviment ha estat positiu o negatiu.

Si una targeta no té moviments, es mostra un text per defecte indicant-ho.

En pitjar a sobre de qualsevol ítem, s'obrirà l'última pantalla que conté els detalls del moviment seleccionat.

	-42.00 €	21/04/2016
	Repsol	
	-150.00 €	21/04/2016
	Fondo inversión	
	1121.54 €	21/04/2016
	Ingreso nómina	

Figura 58: Llista de moviments

5.2.6. Pantalla de detall d'un moviment

L'última pantalla de l'aplicació la forma la Figura 59, i conté una vista amb *scroll* amb tot de text en els quals es mostra tota la informació possible d'un moviment.

21 April 2016  -87.95 €
Concepto Media Markt Lleida E248
Categoría Compra
Fecha operación

Figura 59: Pantalla de detalls d'un moviment

5.2.7. Notificacions

Una de les funcionalitats dels SmartWatch és la de poder rebre i gestionar les notificacions que rep el telèfon mòbil. En l'aplicació mòbil s'ha creat una pantalla per poder simular notificacions enviades des del banc (Pantalla d'accions del rellotge) i, aquestes contenen accions que es poden completar des del rellotge. Un exemple és el de la Figura 60 que avisa que el client ja pot passar a

recollir la seva targeta nova per la seva oficina, i ja que aquesta notificació parla sobre anar a l'oficina, es mostra l'acció de la Figura 61 la qual permet veure la llista d'oficines.

Com que la notificació ha estat creada des del mòbil, en pitjar qualsevol botó d'aquesta al rellotge, s'executarà l'aplicació del telèfon, ja que el "Intent's" de les "Activities" creats pertanyen al mateix dispositiu.

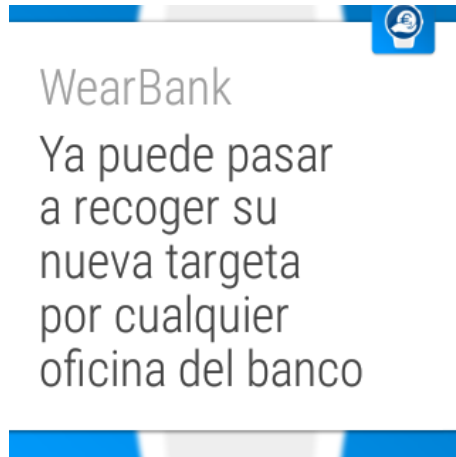


Figura 60: Targeta de notificació al rellotge



Figura 61: Accions de la notificació

6. Manuals de l'aplicació

6.1. Importar el projecte a Android Studio

El fitxer que es troba a "[App\Codi font\WearBank.zip](#)" conté el codi font del projecte generat amb l'entorn de desenvolupament oficial d'Android, Android Studio.

6.1.1. Instal·lació d'Android Studio

Abans d'importar el projecte es necessita tenir el *IDE* preparat a la màquina en la qual es vol provar.

Per instal·lar-lo solament fa falta dirigir-se a la web oficial d'Android[26] (Figura 62), desplegar l'acordió de "Desenvolupar" de la part esquerra de la pantalla i pitjar sobre l'enllaç que posa "Android Studio".

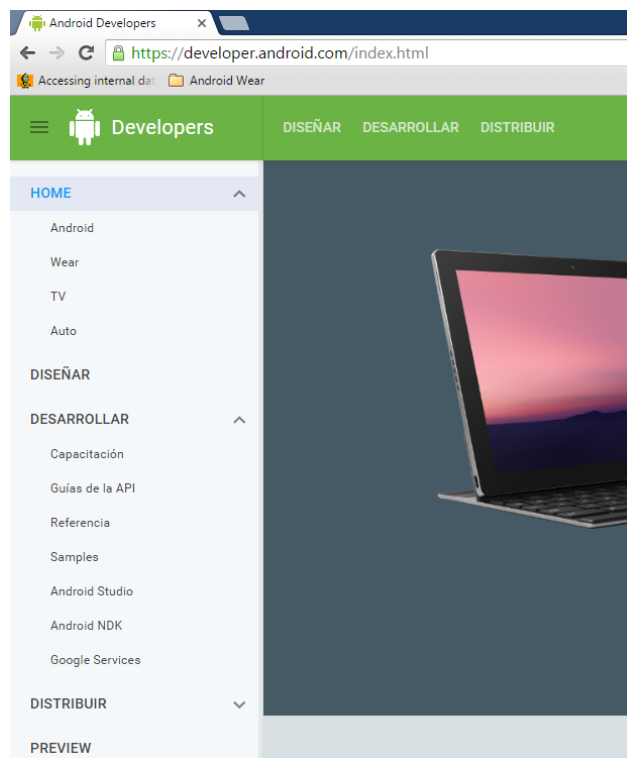


Figura 62: Pantalla inicial de la web de "Developers" d'Android

Una vegada oberta la nova pantalla[27] ja es pot veure un botó verd que fica "Download Android Studio X.X" (on X.X és la versió del programa). Si es pitja, s'inicia la descàrrega d'un executable el qual instal·larà l'aplicació. Si es té un sistema operatiu diferent de Windows o es vol baixar en un altre format l'entorn de desenvolupament, a la part de baix de la web hi ha un apartat anomenat "Select a different platform" que conté la llista d'opcions. També hi ha un altre apartat destinat solament al SDK d'Android, de

forma que es pot baixar per una banda el IDE i per l'altra les eines de desenvolupament i després unir-ho. (tot i que és més senzill instal·lar-ho tot junt).

Un dels requeriments de l'entorn és el de tenir el "Java Development Kit (JDK)" instal·lat i afegit al "PATH" de la màquina (en cas de ser Windows). Per descarregar-lo es pot fer a través de la web de "Oracle"[28] en la qual solament fa falta seleccionar el sistema operatiu en el qual s'instal·larà.

6.1.2. Importar el projecte

Per importar-lo i executar-lo solament fa falta descomprimir el fitxer d'abans i obrir-lo amb el programa comentat.

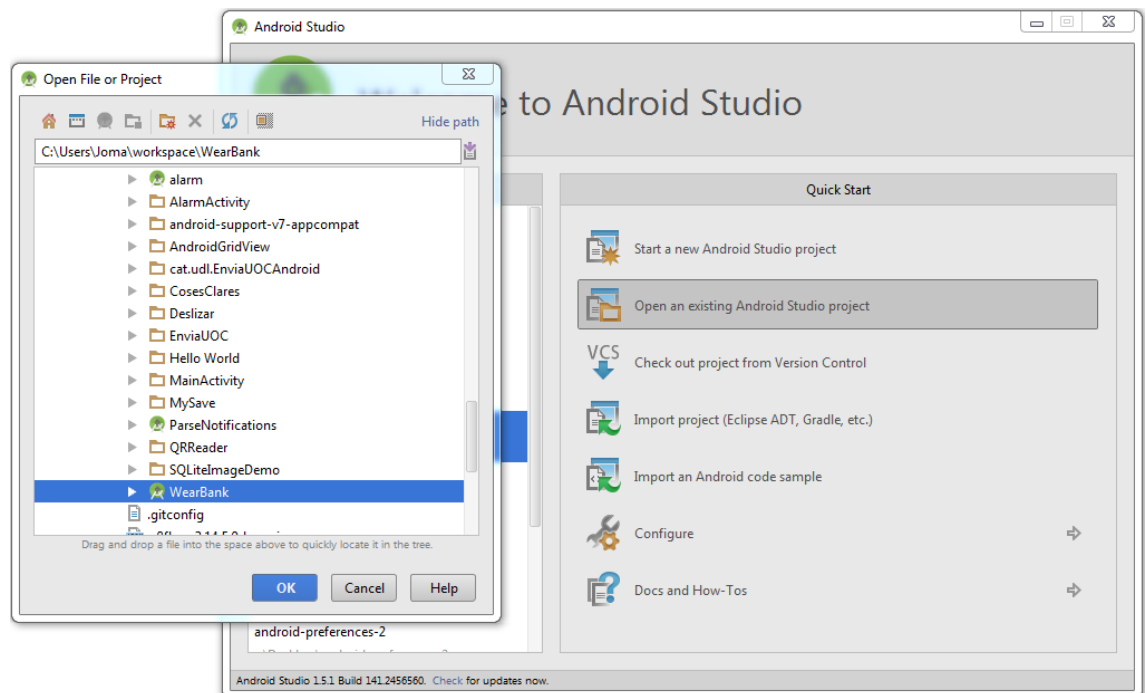


Figura 63: Exportació d'un projecte en Android Studio

Suposant que ja s'ha instal·lat tot el SDK d'Android, solament queda llençar el projecte a un dispositiu (per crear un emulador, consultar l'annex 10.1).

A la part superior hi ha un desplegable en el qual es pot escollir si es vol executar el projecte de mòbil o el de rellotge (es pot veure a la Figura 64).

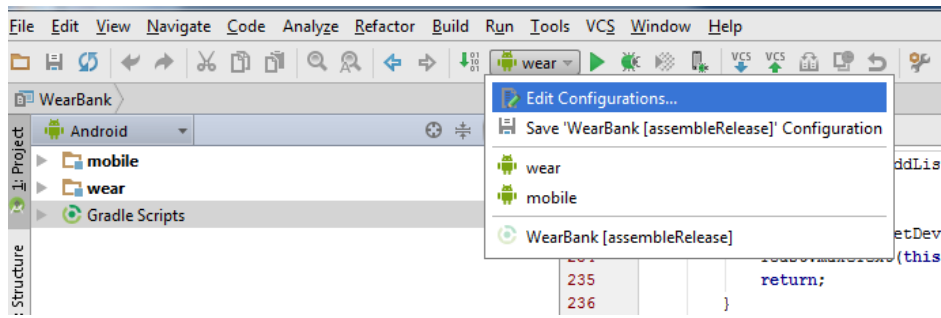


Figura 64: Selector de projecte

Una vegada triat un, solament fa falta pitjar sobre el botó verd en forma de triangle que es troba justament al costat d'aquest selector i escollir el dispositiu en el qual es vol executar.

Si es disposa d'un telèfon mòbil amb Android, però no d'un rellotge, es pot utilitzar un emulador i connectar-los entre si[29]. Per fer-ho, el telèfon ha de tenir l'aplicació de "Android Wear"[30] instal·lada i ha d'estar connectat per USB al ordinador.

A continuació es crearà o s'obrirà un emulador de tipus "Wear". Per crear-lo es pot fer a través del "ADV Manager" que ofereix el SDK d'Android.

Una vegada estigui el rellotge amb execució, solament queda crear la connexió entre els dos dispositius que es farà per línia de comandes de la següent forma:

```
$ adb -d forward tcp:5601 tcp:5601
```

A continuació, per comprovar que tot està funcionant, es pot simular una notificació amb l'aplicació de "Android Wear". Per fer-ho s'ha de pitjar al menú de l'aplicació i seguidament pitjar sobre l'opció de provar una notificació (en la Figura 65, és la segona opció que es veu). Si s'envia la notificació i apareix a la pantalla del rellotge, vol dir que tot està funcionant.

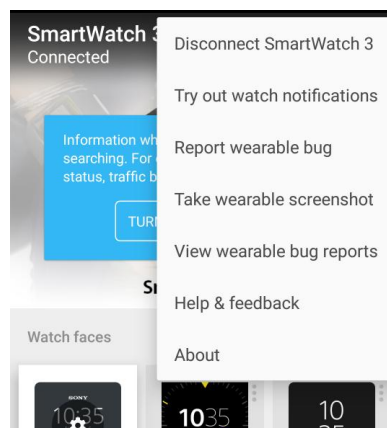


Figura 65: Menú d'opcions d'Android Wear

6.2. Executar l'aplicació

Dins de la carpeta "[App\Executables](#)" hi ha els arxius en format "apk" els quals serveixen per ser instal·lats directament a un dispositiu.

Hi ha dos fitxers, un és l'aplicació que es ficarà al mòbil i l'altre és la que es ficarà al rellotge.

Per instal·lar-les es pot fer de diferents formes. La primera és utilitzant el projecte i executant-lo directament, tal com s'ha comentat en l'apartat "6.1 Importar el projecte a Android Studio".

La segona és a través de línia de comandes, però serà necessari tenir el SDK d'Android instal·lat. Per fer-ho s'ha d'obrir la carpeta que contenen els instal·lables i executar la següent comanda:

```
$ adb install /path/<nom_aplicacio>.apk
```

En cas de provar-ho a un dispositiu físic, aquest haurà de tenir l'opció de depuració per USB activada.

Finalment, l'última opció és la d'obrir directament el fitxer des del dispositiu. Per fer-ho es pot copiar el mateix arxiu i ficar-lo a la memòria del telèfon a través d'un cable USB i connexió a l'ordinador. També es pot fer utilitzant algun "Cloud" i accedint directament a ell (per exemple, Dropbox).

6.3. Credencials d'usuari

L'aplicació disposa d'una pantalla d'identificació, i per superar-la es poden utilitzar les següents credencials:

- Usuari amb informació (targetes i moviments)
 - Identificador: 123456789P
 - Contrasenya: 1234

- Usuari sense informació:
 - Identificació: 111111111B
 - Contrasenya: 1111

7. Conclusions

El desenvolupament d'aquest projecte ha aportat molts coneixements i motivacions de cara a la programació de futures aplicacions mòbils, sobretot, en llenguatge natiu.

Ha ajudat a conèixer i aplicar les noves funcionalitats que incorporen les noves versions d'Android, a utilitzar mètodes i classes per crear components que estiguin atents en tot moment per qualsevol canvi en l'aplicació i, sobretot, a comunicar-se i treballar amb els nous *Wearables*.

Aquests últims han demostrat que encara són molt nous en el món de la tecnologia i la programació, i que hi ha molt poca comunitat que es dediqui al desenvolupament d'aplicacions per aquests dispositius. Unint aquests problemes amb la desconexió de la creació d'aplicacions per relloges, ha fet que les hores implicades en la creació d'aquest projecte hagi estat més del previst.

Al llarg del desenvolupament de les aplicacions s'han ocorregut noves funcionalitats que, en un futur, es poden desenvolupar i integrar a l'aplicació. Algunes d'aquestes es basen en augmentar la seguretat, per exemple, fer una pantalla de desbloqueig per a l'aplicació del relloge. La idea és que quan l'usuari faci el vincle dels dos dispositius, hagi de crear un PIN de quatre números el qual servirà per poder executar l'aplicació del relloge. Fent-ho així, solament l'usuari que ha creat el vincle sabrà quin és el codi de desbloqueig i, per molt que un altre usuari utilitzi el relloge, no podrà accedir a l'aplicació.

Aquesta nova pantalla podria aparèixer una vegada s'ha comprovat que hi ha l'enllaç creat, o l'altra opció és crear un nou botó al menú i fer que l'usuari s'identifiqui en aquell punt, d'aquesta forma es permet que qualsevol usuari pugui seguir utilitzant l'aplicació amb les funcionalitats reduïdes.

Per concloure, aquest projecte es presentarà dins de l'empresa GFT (empresa orientada al sector financer) com una prova de concepte oferint la possibilitat d'augmentar les seves funcionalitats i acabar donant-li una finalitat comercial.

8. Glossari

- **Wearable:** dispositiu que s'incorpora en alguna part del cos i que interactua contínuament amb l'usuari i altres dispositius amb la finalitat de realitzar alguna funció específica.
- **MBaaS:** "Mobile Backend as a Service" és un model que permet vincular aplicacions amb emmagatzematge al núvol i altres serveis d'anàlisi i emmagatzematge de dades de dades, notificacions push, gestió d'usuaris i roles, entre altres coses.
- **IDE:** "Integrated Development Environment" o entorn de desenvolupament integrat és el programa que utilitzen els programadors per crear les aplicacions.

9. Bibliografía

- [1] **Android.** *Android Wear.* [En línea]
https://www.android.com/intl/es_es/wear/
- [2] **BBVA.** *BBVA en smartwatches: ya está aquí la app para Android Wear.* [En línea] [Consulta: 04/03/2016]
<https://www.blogbbva.es/bbva-android-wear/>
- [3] **ComputerHoy.** *La Caixa estrena apps para Google Glass y smartwatch de Sony.* [En línea] [Consulta: 04/03/2016]
<http://computerhoy.com/noticias/apps/caixa-estrena-apps-google-glass-smartwatch-sony-9469>
- [4] **iTunes Apple.** *Santander Watch.* [En línea] [Consulta: 04/03/2016]
<https://itunes.apple.com/es/app/santander-watch/id1007514901?mt=8#>
- [5] **Play Google.** *Santander Watch.* [En línea] [Consulta: 04/03/2016]
<https://play.google.com/store/apps/details?id=es.bancosantander.android.sanwear&hl=es>
- [6] **Bankinter.** *Bankinter desarrolla un sistema instantáneo de avisos y alertas a través de Apple Watch.* [En línea] [Consulta: 04/03/2016]
<http://blog.bankinter.com/blogs/bankinter/archive/2015/05/27/bankinter-apple-watch-alertas-reloj.aspx>
- [7] **Backendless.** [En línea]
<https://backendless.com>
- [8] **Fintonic.** [En línea]
<https://www.fintonic.com/app/landing/home.do>
- [9] **Dispersium.** *Evaluación heurística.* [En línea] [Consulta: 03/04/2016]
<http://dispersium.es/evaluacion-heuristica/>
- [10] **NN/g Nielsen Norman Group.** *10 usability heuristics for User Interface Design.* [En línea] [Consulta: 03/04/2016]
<https://www.nngroup.com/articles/ten-usability-heuristics/>
- [11] **UXmatters.** *Empirical Development of Heuristics for Touch Interfaces.* [En línea] [Consulta: 03/04/2016]
<http://www.uxmatters.com/mt/archives/2014/06/empirical-development-of-heuristics-for-touch-interfaces.php>
- [12] **No solo usabilidad.** *En busca del Diseño Centrado en el Usuario (DCU): definiciones, técnicas y una propuesta.* [En línea] [Consulta: 25/03/2016]
<http://www.nosolousabilidad.com/articulos/dcu.htm>
- [13] **Construcción de Software.** *Modelo de casos de uso.* [En línea] [Consulta: 25/03/2016]
<http://dis.um.es/~jnicolas/CSW/casoPracticoParaClaseVideoclub/CDUCasoPracticoVideoclub.pdf>
- [14] **Android.** [En línea]
<https://www.android.com/history/>
- [15] **Android.** *Android Wear.* [En línea]
https://www.android.com/intl/es_es/wear/
- [16] **Universitat politècnica de València.** *Arquitectura de Android.* [En línea] [Consulta: 30/03/2016]
<http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>

- [17] **Google Developers.** *SharedPreferences*. [En línia]
<http://developer.android.com/intl/es/reference/android/content/SharedPreferences.html>
- [18] **Google Developers.** *SharedPreferences.Editor*. [En línia]
<http://developer.android.com/intl/es/reference/android/content/SharedPreferences.Editor.html>
- [19] **Google Developers.** *Accessing Google APIs*. [En línia]
<https://developers.google.com/android/guides/api-client>
- [20] **Google APIs for Android.** *LocationServices*. [En línia]
<https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices>
- [21] **Google Developers.** *Google APIs for Android*. [En línia]
<https://developers.google.com/android/reference/com/google/android/gms/wearable/Wearable#nested-class-summary>
- [22] **Google Developers.** *Building Apps for Wearables*. [En línia]
<http://developer.android.com/intl/es/training/building-wearables.html>
- [23] **Greenrobot.** *EventBus: Events for Android*. [En línia]
<http://greenrobot.org/eventbus/>
- [24] **Backendless.** [En línia]
<https://backendless.com/>
- [25] **Parse.** [En línia]
<https://parse.com/>
- [26] **Google Developers.** [En línia]
<https://developer.android.com/index.html>
- [27] **Android Studio.** *Android Studio The Official IDE for Android*. [En línia]
<https://developer.android.com/studio/index.html#downloads>
- [28] **Oracle.** *Java SE Development Kit 8 Downloads*. [En línia]
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [29] **Google Developers.** *Creating and Running a Wearable App*. [En línia]
<http://developer.android.com/intl/es/training/wearables/apps/creating.html#SetupEmulator>
- [30] **Google Play.** *Android Wear – Smartwatch*. [En línia]
<https://play.google.com/store/apps/details?id=com.google.android.wearable.app>
- [31] **Genymotion.** [En línia]
<https://www.genymotion.com/>
- [32] **GitHub.** *Genymotion 2.0 Emulators with Google Play support*. [En línia]
https://github.com/codepath/android_guides/wiki/Genymotion-2.0-Emulators-with-Google-Play-support#setup-google-play-services

10. Annexos

10.1. Creació i execució de l'aplicació en un emulador

En aquest apartat s'explicarà com crear un emulador amb on poder executar l'aplicació i poder-li treure el màxim rendiment.

Aquest dispositiu virtual es crearà utilitzant "Genymotion", una aplicació que permet descarregar i emular un dispositiu Android d'una forma fàcil i, a més a més, aquests emuladors funcionen d'una forma molt lleugera, és a dir, no s'enganxen ni van molt lents en cas que la màquina en la qual s'executi, no sigui molt potent.

Els passos a seguir per instal·lar l'aplicació i obtenir el primer dispositiu són:

1. Crear una compta a la web de "Genymotion"[31] amb la qual es podrà accedir a l'apartat per poder descarregar l'aplicació.

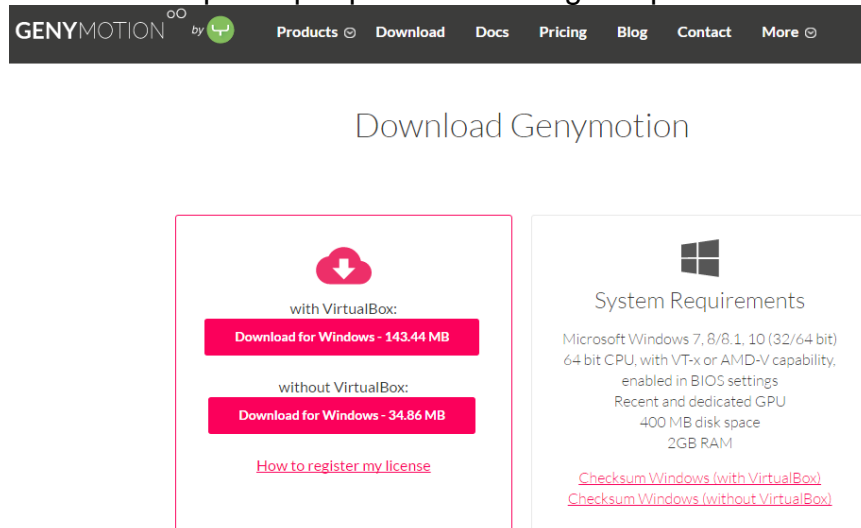


Figura 66: Web de "Genymotion"

2. Una vegada descarregada i instal·lada l'aplicació, es pot obrir el programa, el qual presentarà la finestra de la Figura 67.

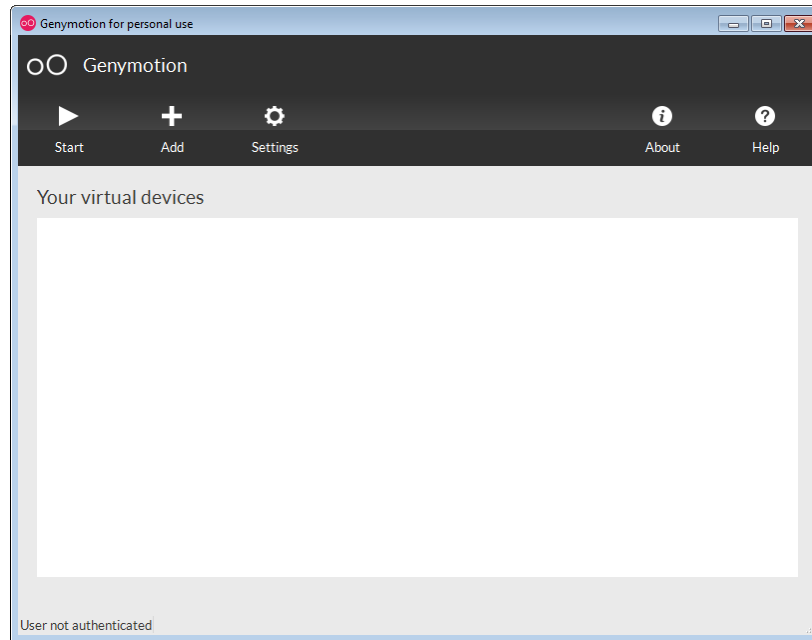


Figura 67: Finestra de "Genymotion"

3. Per afegir un dispositiu nou, solament fa falta pitjar sobre el botó de "Add" de la part superior de la finestra. Una vegada s'obri la nova pestanya, el primer que es tindrà de fer és introduir les credencials d'usuari que s'han utilitzat per descarregar el programa (la mateixa aplicació ja indica que no es pot accedir a la llista de dispositius si no hi ha un usuari identificat).

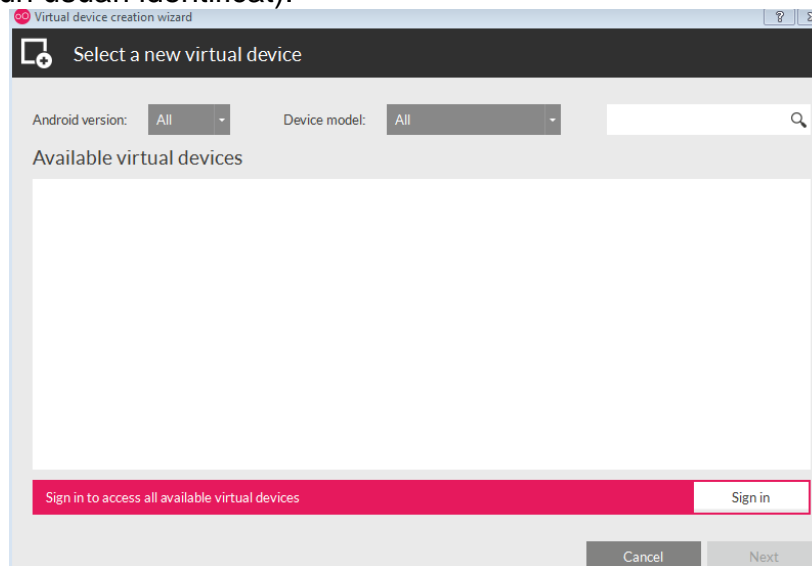


Figura 68: Llista de dispositius pendent d'identificació

4. Una vegada identificat, ja apareixerà la llista de dispositius disponibles. Per provar l'aplicació es recomana buscar un "Nexus5" amb la "API22".

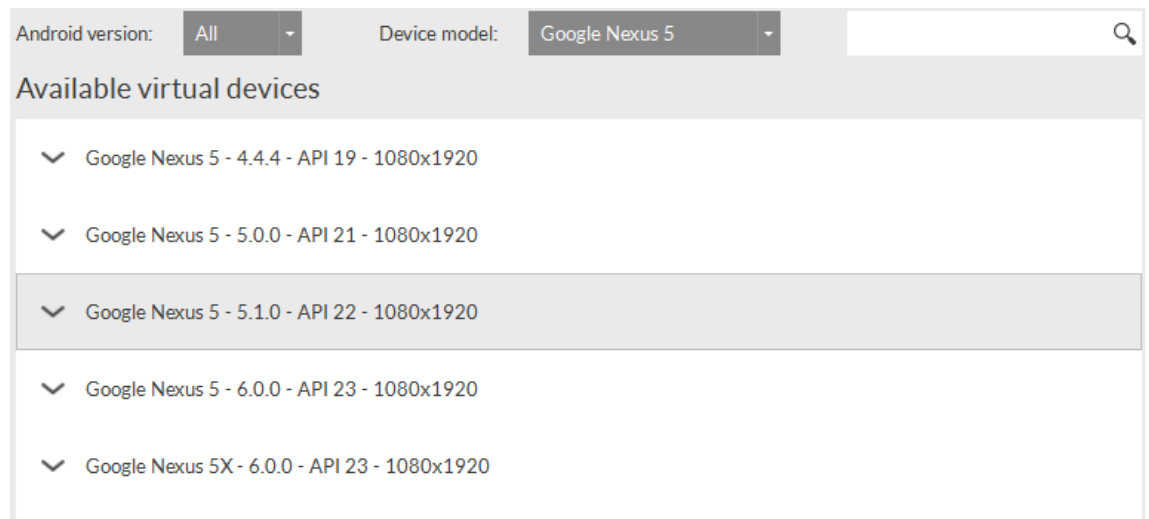


Figura 69: Dispositiu "Nexus5"

5. Una vegada fet aquest procediment, es començarà a descarregar el nou emulador.

Quan hagi finalitzat, es pot executar el dispositiu i instal·lar-li l'aplicació (es pot fer tal com s'ha explicat a l'apartat 6.2). Un dels avantatges de "Genymotion" és que es pot instal·lar una aplicació arrossegant el fitxer instal·lable d'Android (".apk"), d'aquesta forma, si es disposa de l'instal·lable, es pot fer directament. Es pot veure un exemple a la web[32].

Per finalitzar, si es vol consultar la llista d'oficines, s'ha d'activar el GPS de "Genymotion" i també el del sistema Android, tot i que per defecte ja està actiu. Si s'entra a l'apartat de "Buscar oficines" sense que la localització de l'aplicació estigui activat i amb unes coordenades, la pantalla de l'aplicació del banc es quedarà carregant esperant obtenir unes coordenades.

Això és pel fet que l'aplicació del mòbil detecta que el GPS està activat i per tant comença a obtenir la ubicació, però com que "Genymotion" no ho té activat, no li envia cap posició, de forma que, l'aplicació desenvolupada es queda esperant.

Per activar el GPS es fa a través del mateix emulador que s'ha obert. A la part de la dreta hi ha una barra d'accions de color negre, i el segon d'aquests botons és el que permet modificar el GPS. Si es pitja, s'obrirà una nova finestra en la qual es podrà activar i desactivar i també simular la ubicació (Figura 70), per exemple, si es posen les coordenades de Lleida (latitud: 41.6171098 i longitud: 0.6175525) es mostraran dues oficines properes. Si es prefereix, també es pot marcar la posició al mapa mitjançant el botó de "Map".

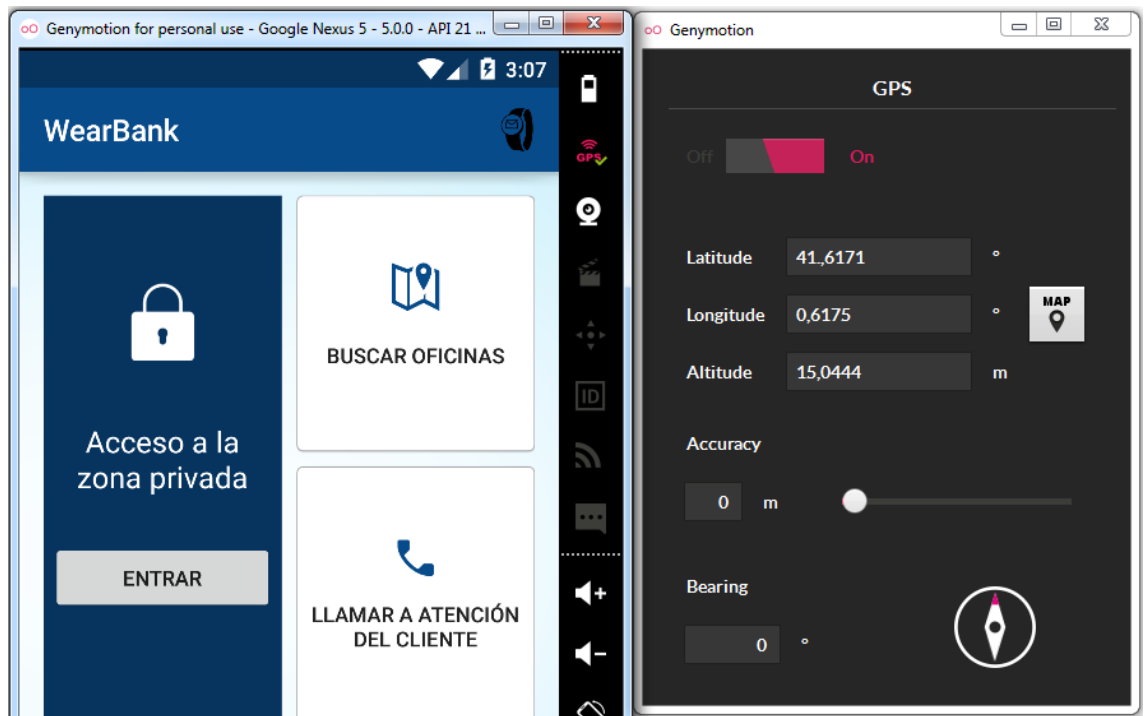


Figura 70: Activació de la localització de "GenyMotion"