

Guarant-it.

Gestor de garanties personal.

Albert Giró Quer

Grau d'enginyeria informàtica

Aplicacions mòbils en Android

David Escuer Latorre

Jordi Almirall López

Carles Garrigues Olivella

30/05/2022



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Guarant-it. Gestor de garanties personal.</i>
Nom de l'autor:	<i>Albert Giró Quer</i>
Nom del consultor/a:	<i>David Escuer Latorre Jordi Almirall López</i>
Nom del PRA:	<i>Carles Garrigues Olivella</i>
Data de lliurament (mm/aaaa):	<i>05/2022</i>
Titulació o programa:	<i>Grau d'enginyeria informàtica</i>
Àrea del Treball Final:	<i>Aplicacions mòbils en Android</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Gestió, garanties, incidències</i>
Resum del Treball	
<p>Guarant-it és un projecte que neix amb l'objectiu de generar un aplicatiu capaç de gestionar els productes que comprem. La gestió implica no tan sols mantenir un inventari de productes sinó també els documents i incidències relacionats amb cadascun d'ells.</p> <p>El producte objectiu és una aplicació Android capaç de sincronitzar-se en diversos dispositius i ser capaç de persistir i recuperar les dades en cas de ser desinstal·lat o instal·lat en dispositius nous.</p> <p>El procés de definició del producte s'ha basat en el disseny centrat en l'usuari. Això ha permès que els usuaris prenguessin part del projecte i adaptar certes funcionalitats als seus interessos. Per a l'arquitectura s'han usat tecnologies i paradigmes que siguin eficients i fàcilment escalables (MVVM, Realm ...).</p> <p>Pel desenvolupament s'ha seguit la metodologia Kanban on s'han anat solucionant els casos d'ús, des dels més generals als més específics.</p> <p>Com a resultat del procés s'ha obtingut una aplicació funcional que satisfà els objectius inicials.</p>	

Abstract

Guarant-it is a project borned with a target of produce an app able to maintain the product that we bought. This management has to cover the product inventory, and tho documents and issues related to them.

The resulting product is an Android app that store all neede data and sincronize it between devices. Also, has to be able to acces the data even the aplication was deleted or installed in new devices.

The product definitioon proces was based in the User Centered Design, that allowed to put the users in the proces and use them to take some decisions. The solution use for the project architecture has been selected in order to promote eficienci and scalability.

The development proces was run over Kanban methodology, and the use cases has been developed from more generals to more concrete ones.

As a result, we obtain a functional app that satisfies initial objectives.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	4
1.2.1 Objectius generals:.....	4
1.2.2 Objectius funcionals:.....	4
1.2.3 Objectius no funcionals:.....	4
1.3 Enfocament i mètode seguit.....	5
1.4 Planificació del Treball.....	6
1.5 Breu sumari de productes obtinguts.....	8
1.6 Breu descripció dels altres capítols de la memòria.....	8
2. Definició del projecte.....	9
2.1 Disseny Centrat en l'Usuari.....	9
2.1.1 Investigació i recerca.....	11
2.1.2 Disseny conceptual.....	20
2.1.3 Prototipat.....	22
2.1.4 Avaluació.....	25
2.2 Disseny de l'arquitectura.....	26
2.2.1 Cas d'ús.....	30
2.2.2 Model de dades.....	33
3. Implementació.....	34
3.1 Implementació i casos d'ús.....	34
3.2 Tests.....	41
Cas d'ús CU-001 – Inici de sessió/Registre.....	41
Cas d'ús CU-002 – Afegir productes.....	42
CU-003 – Afegir productes.....	43
CU-004 – Afegir incidència.....	44
CU-006 – editar productes.....	45
3.3 - Planificació i seguiment.....	46
4. 4. Conclusions.....	47

5. Glossari.....	49
6. Bibliografia.....	51
7. Annexos.....	52
7.1 Preparació entorn de desenvolupament.....	52
7.2 Preparació de l'aplicació al núvol de Realm.io/MongoDB.....	53

Taula de figures

Figura 1: Distribució i planificació de les tasques.....	6
Figura 2: Diagrama de Gantt del projecte.....	6
Figura 3: Relació entre edat i experiència tecnològica.....	18
Figura 4: Tipus de compra.....	19
Figura 5: Moment d'afegir el producte.....	19
Figura 6: Visualització de productes.....	19
Figura 7: Diagrama de flux de la navegació.....	22
Figura 8: Flux des de la pantalla d'inici de l'aplicació.....	23
Figura 9: Flux a partir de la pantalla de producte.....	24
Figura 10: Infraestructura de l'aplicació.....	27
Figura 11: «Componentes de arquitectura de Android, de MVC a MVVM», Izertis.....	28
Figura 12: aplicació de l'arquitectura hexagonal.....	29
Figura 13: Botons d'editar i eliminar producte.....	38
Figura 14: UML casos d'ús producte.....	38
Figura 15: Casos d'ús de documents.....	39
Figura 16: Nou model de dades d'Incidències.....	40
Figura 17: Uml casos d'ús incidència.....	40
Figura 18: "HelloWorld" execució.....	52
Figura 19: "HelloWorld" Entorn de desenvolupament.....	52

Índex de taules

Taula 1: Comparativa aplicacions alternatives.....	2
Taula 2: Objectius funcionals.....	4
Taula 3: Objectius no funcionals.....	4

1. Introducció

1.1 Context i justificació del Treball

Avui dia moltes de les compres que fem de productes es fan per mitjà del comerç electrònic, però en moltes ocasions comprem productes en comerços locals o supermercats. No és estrany que, en aquestes ocasions, passat un temps es pugui tenir algun problema amb aquests productes i, per tal de poder reclamar, necessitem el justificant de compra. És probable que en la compra d'un producte car es tingui especial cura amb els rebuts i les garanties. És en altres ocasions que, arribat el moment d'anar a reclamar la nostra garantia, és fàcil que no trobar el rebut de compra o, fins i tot, no estar segurs de si encara estem en el període de garantia.

És en aquestes ocasions que, per un oblit, no podem exercir el nostre dret com a consumidors i reclamar la garantia del nostre producte. En el millor dels casos es tenen els justificants guardats en algun arxivador i es poden consultar de pressa sempre que s'hagi desat de manera organitzada. Però, fins i tot en aquests casos, hi poden haver alguns rebuts o justificants de compra que estiguin impresos amb paper tèrmic i, quan es recupera per utilitzar-lo, el contingut sigui ja intel·ligible.

En l'ecosistema d'aplicacions per a Android s'hi poden trobar algunes aplicacions que permeten desar la informació dels tiquets de compra i, inclús, avisar de la finalització períodes de garantia o de devolució. Per exemple, [Phototicket](#)¹ que se centra en la gestió, la justificació, i el manteniment de les despeses. O d'altres més centrades en la gestió de la de garanties com poden ser [Warranty Sured](#)² o [Warranty manager](#)³. Aquestes aplicacions compleixen el requisit principal de mantenir un registre dels nostres rebuts. A la taula comparativa d'aplicacions alternatives Taula 1, podem veure les característiques detallades de cadascuna d'elles.

1«PhotoTicket.App», consulta 26 febrer 2022, <https://www.phototicket.app/>.

2«Warranty Sured - Your Warranty Manager - Apps on Google Play», consulta 26 febrer 2022, https://play.google.com/store/apps/details?id=com.jncwebs.warrantysured&hl=en_US&gl=US.

3«Warranty Manager - Apps on Google Play», consulta 26 febrer 2022, https://play.google.com/store/apps/details?id=io.epix.warranty_manager&hl=en_US&gl=US.

Característiques	Warranty Manager	Warranty Sured	PhotoTicket
Data de compra	✓	✓	✓
Establiment de compra	✓	✓	✓
Període de garantia	✓	✓	✓
Preu	✓	✓	✓
Marca	✓	✓	✗
Model	✗	✓	✗
Número de sèrie	✗	✓	✗
Categoria del producte	✓	✓	✓
Notes/Observacions	✓	✓	✓
Cerca	✗	✗	✓
Documents adjunts	4 imatges (producte, tiquet de compra, garantia, imatge adicional)	Imatge del tiquet de compra	Imatge del tiquet de compra
Llistes	Per validesa de garantia (actives, proximes a caducar, caducades)	Llista única sense filtres	Llista amb filtres: - data - preu - tipus de tiquet - categoria
Notificacions	✗	✗	✓
Incidències	✗	✗	✗

Taula 1: Comparativa aplicacions alternatives

En el cas de Warranty Sured i Warranty Manager és palès la seva especialització cap a la gestió de garanties, donant la possibilitat de desar fins i tot el número de sèrie. Si bé la informació general del producte és bastant completa, tant la navegació com la cerca són molt pobres o inexistentes en el cas de Warranty Sured.

L'aplicació PhotoTicket, és més completa pel que fa a la gestió dels tiquets desats, la cerca, el filtratge o desar informació sobre el tipus de despesa (producte tangible, intangible, servei...). Però, pel que fa a les dades de garantia, queda relegat a un camp opcional del producte, i tampoc permet adjuntar més imatges que no siguin el justificant de compra.

Una aplicació on poder desar els rebuts dels productes en el mateix moment de la compra i d'aquesta manera tenir un registre de productes, garanties manera centralitzada, permanent i assegurar-ne la persistència en són els objectius principals de Garant-it

L'aplicació Garant-it aporta característiques que no es troben en altres de les aplicacions trobades al PlayStore. Ofereix, per exemple, la possibilitat de desar documents relacionats com ara instruccions, manuals o informació relacionada o rellevant que es vulgui mantenir. Addicionalment, permet gestionar les reclamacions, incidències o reparacions relacionades emmagatzemant la informació dels expedients oberts.

1.2 Objectius del Treball

1.2.1 Objectius generals:

- L'aplicació ha de facilitar a l'usuari la gestió de tota la informació relativa als productes que va adquirint.
- Ha de ser intuïtiva de manera que la corba d'aprenentatge sigui curta.
- Donar les eines necessàries a l'usuari per a permetre la justificació dels productes en futures reclamacions en cas avaria o robatori.

1.2.2 Objectius funcionals:

Codi	Descripció
OF-001	L'usuari s'ha de poder registrar a l'app per poder tenir un perfil
OF-002	L'usuari ha de poder crear i editar nous productes a la llista d'adquisicions.
OF-003	L'usuari ha de poder desar els justificants de compra/garanties
OF-004	L'usuari ha de poder adjuntar, documents i imatges relacionades amb el producte com ara manuals o fotografies de l'aparell.
OF-005	L'usuari ha de poder registrar informació sobre tràmits i incidències relacionades al producte com reparacions o reclamacions.

Taula 2: Objectius funcionals

1.2.3 Objectius no funcionals:

Codi	Descripció
ONF-001	L'aplicació ha de ser multilingüe.
ONF-002	L'aplicació ha de desar la informació al núvol de manera que es pugui consultar de qualsevol aparell.
ONF-003	L'aplicació ha de navegar entre pantalles amb un temps màxim de resposta de 2 segons.
ONF-004	L'aplicació ha de poder visualitzar-se encara que no hi hagi connexió a la xarxa.

Taula 3: Objectius no funcionals

1.3 Enfocament i mètode seguit

L'aplicació Guarant-it es crea amb l'objectiu de satisfer una necessitat no coberta per altres aplicacions. Unes centrades més en la gestió de despeses i les altres, més centrades en la gestió de garanties, massa senzilles i amb falta de funcionalitats. A més, durant la recerca de programari similar, no s'ha trobat una aplicació similar de codi obert que pugui ser usada com a punt de partida. De manera que l'estratègia que es durà a terme és la de creació d'un producte nou.

Per dur a terme el projecte s'ha optat per usar el paradigma de metodologia àgil. Aquest paradigma permetrà poder tenir lliurables completament funcionals des del primer moment i afegir funcionalitats en cada nova entrega. Tot i que, aquesta metodologia s'ha hagut d'adaptar a la naturalesa del treball de final de grau de la UOC i organitzat en quatre blocs. Els primers lliurables se centren en el disseny centrat a l'usuari, per extraure quines són les funcionalitats requerides per l'usuari. Serà durant el tercer lliurable, el desenvolupament del producte, on es treballaran segons les funcionalitats trobades en l'estudi previ i les prioritats trobades.

1.4 Planificació del Treball

A la Figura 1, i al diagrama de Gantt situat a sota, s’hi ha representat la càrrega de treball i la distribució de les diferents tasques repartides durant la realització.

▼	PAC 1	45	...	+	
	Pla de treball		February 16, 2022 → February 21, 2022	18	
	Documentació de la memòria part 1		February 22, 2022 → February 26, 2022	15	
	"Hello world"		February 27, 2022 → March 2, 2022	12	
	+ New				
▼	PAC 2	87	...	+	
	DCU - contextos d'ús		March 2, 2022 → March 11, 2022	30	
	DCU - Definició dels casos d'ús		March 12, 2022 → March 19, 2022	24	
	Disseny de l'arquitectura		March 20, 2022 → March 30, 2022	33	
	+ New				
▼	PAC 3	132	...	+	
	Impl - Casus d'us Producte		March 31, 2022 → April 16, 2022	51	
	Imp - Casos d'us Documentació		April 16, 2022 → April 23, 2022	24	
	Casos d'us Incidències		April 23, 2022 → May 3, 2022	33	
	Testes de l'aplicació		May 4, 2022 → May 11, 2022	24	
	+ New				
▼	Liurament	57	...	+	
	Finalització memòria		May 12, 2022 → May 21, 2022	30	Liurar
	Presentació		May 22, 2022 → May 30, 2022	27	Liurar

Figura 1: Distribució i planificació de les tasques

El projecte es desenvolupa de manera individual, i per això només consta d'un recurs per poder-lo dur a terme. La dedicació que es preveu és de 2 hores durant els dies laborals i 6 els caps de setmana i festius. Aquestes hores són orientatives i poden variar en funció de la càrrega de treball que necessiti el projecte.

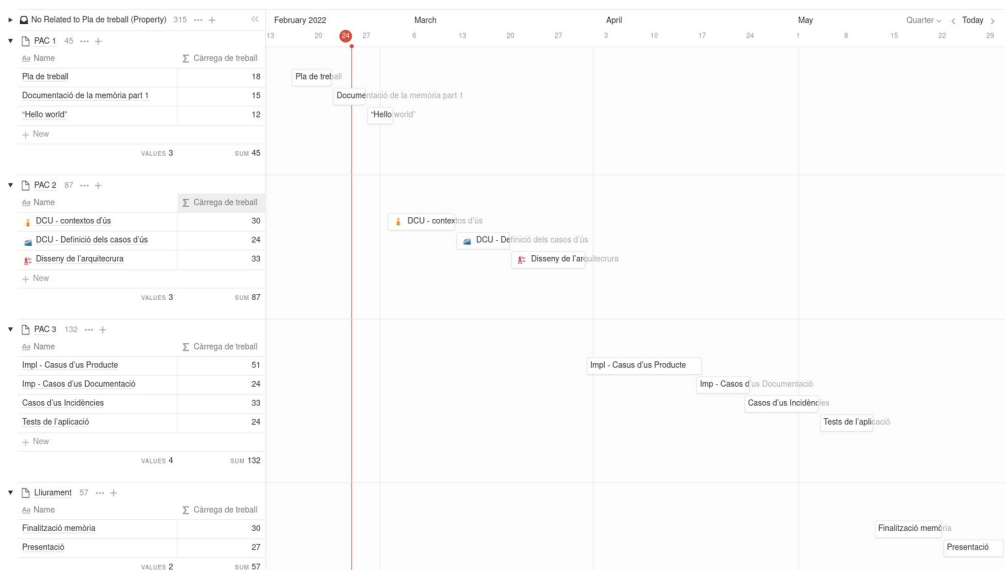


Figura 2: Diagrama de Gantt del projecte

L'execució s'ha estructurat amb quatre fites corresponents als treballs corresponents a cada PAC.

En la primera entrega es treballarà l'apartat inicial del projecte així com una aplicació en la qual es pugui comprovar que l'entorn de treball ja és operatiu.

La segona entrega servirà per definir el projecte tant en l'àmbit d'usabilitat com d'arquitectura. Aquests dos apartats són els que marquen aquesta entrega i són fonamentals, ja que definiran l'aplicació, de manera que en el següent pas es pugui començar amb el desenvolupament.

La tercera entrega és on s'implementarà l'aplicació i per tant és la que més càrrega de treball implica. La metodologia triada és la metodologia àgil Kanban de manera que les tasques s'han estimat en el temps, però són susceptibles de poder reduir o augmentar-se segons les decisions preses en la fase de definició.

La darrera entrega estarà dedicada a finalitzar aquesta memòria i la realització de la presentació.

1.5 Breu sumari de productes obtinguts

En finalitzar el projecte, s'hauran generat tres artefactes:

- Memòria del projecte. Memòria que recull el detall del procés d'execució del projecte. S'hi detalla la planificació, l'arquitectura usada, les decisions preses i els processos per extreure-les.
- Aplicatiu. El codi font de l'aplicatiu resultant de la realització del projecte i l'artefacte instal·lable per a Android Guarant-it.apk
- Presentació. Vídeo amb una presentació que sintetitzi el projecte.

1.6 Breu descripció dels altres capítols de la memòria

- 2. Definició del projecte

Apartat on es detallarà tot el procés de definició del projecte.

- 2.1 Disseny Centrat en l'Usuari

L'estudi de necessitats dels usuaris i definició de la lògica de negoci necessària per poder-les satisfer.

- 2.2 Disseny de l'arquitectura

Definició i justificació de les decisions arquitectòniques i tecnologies adoptades.

- 3. Implementació

En aquest apartat es detallarà el procés d'implementació que s'ha dut a terme durant aquesta fase

- Conclusions

Capítol on es poden veure les conclusions extretes de l'execució del projecte, lliçons apreses, errors detectats i millores per a propostes futures.

2. Definició del projecte

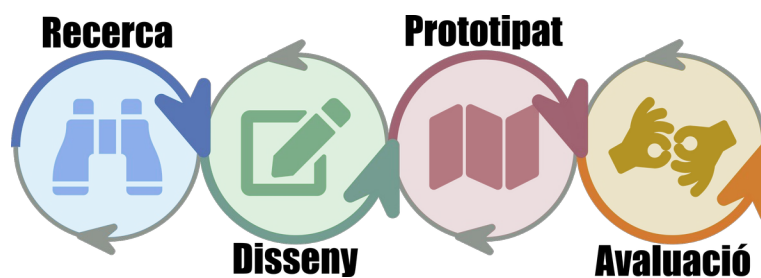
2.1 Disseny Centrat en l'Usuari

Aquesta fase del projecte és la de més importància del projecte, ja que, és a partir del resultat que n'obtinguem, en resultarà una aplicació o una altra. Duran aquesta fase és on es busquen, perfilen i defineixen els detalls que faran que l'aplicació resultant sigui atractiu a l'usuari.

Per aconseguir aquesta meta, es necessita donar a l'usuari una posició una rellevant durant el procés de disseny de l'aplicació. Posicionar l'usuari i el seu punt de vista ens permet dissenyar l'aplicació tenint en compte el seu punt de vista i poder-la adaptar a les seves necessitats.

Aquest procés consta de diverses fases en les quals es va definint les necessitats dels usuaris i els seus contextos. Tot i que aquestes fases s'executen de manera seqüencial, un cop assolida una definició, es poden repetir i iterar els passos anteriors per anar definint nous contextos d'ús que puguin aparèixer o redefinir els que s'han fixat per millorar-ne la usabilitat. És aquesta naturalesa orgànica del procés de definició que fa que el disseny centrat en l'usuari sigui una eina imprescindible en la definició d'aplicacions i, amb més èmfasi si cal, en el desenvolupament d'aplicacions mòbils.

Les tasques que es duran a terme durant el procés es poden categoritzar en quatre fases definides com a recerca, disseny, prototipatge i avaluació.



Cada una de les fases té un objectiu a satisfer. Veurem amb més detall cada fase en l'explicació de cada procés de la memòria. Però es poden sintetitzar de la següent manera:

- Recerca i investigació. La fase contempla l'obtenció dels usuaris potencials de l'aplicació, els seus contextos d'ús, les seves aspiracions i preferències.
- Disseny. Per tal de poder extreure'n conclusions de les necessitats de l'aplicació, amb el resultat de la fase anterior, es dissenyen els perfils d'usuari, personatges i contextos d'ús.

- Prototipat. El disseny es plasma en un prototip que on es pugui veure les pantalles i les transicions entre elles, la disposició, els botons, etc. Aquest prototip serà una eina important per utilitzar en l'última fase.
- Avaluació. L'usuari torna a aparèixer en aquesta fase com a protagonista, s'encarregarà d'avaluar i proposar les seves observacions i millores possibles. A partir d'aquest punt, es pot iterar sobre les fases anteriors segons els resultats que s'obtingui d'aquesta fase per tal de refer o ajustar el disseny.

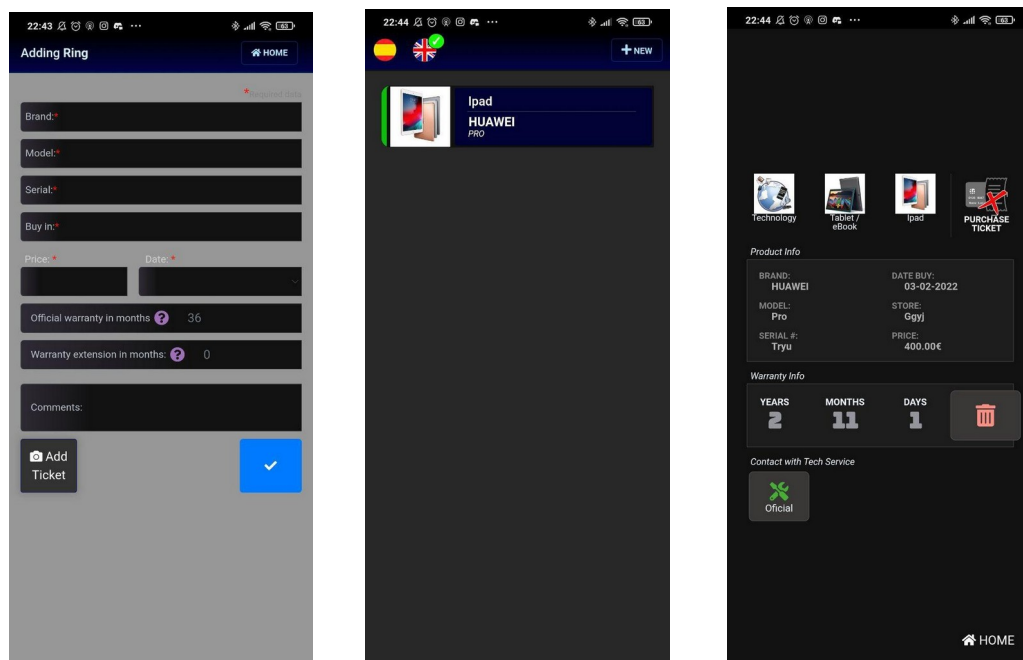
2.1.1 Investigació i recerca

Per la recerca d'aquest projecte es faran servir tres aproximacions diferents. L'anàlisi competitiva, entrevistes en profunditat i enquestes.

Anàlisi competitiva.

El primer que s'ha dut a terme en la fase de recerca, és un estudi de les aplicacions alternatives que podem trobar en l'ecosistema d'Android. En l'apartat 1.1 Context i justificació del Treball ja s'han introduït algunes alternatives i en la Taula 1 s'hi han llistat les característiques de cadascuna. Tot seguit es detallen cadascuna:

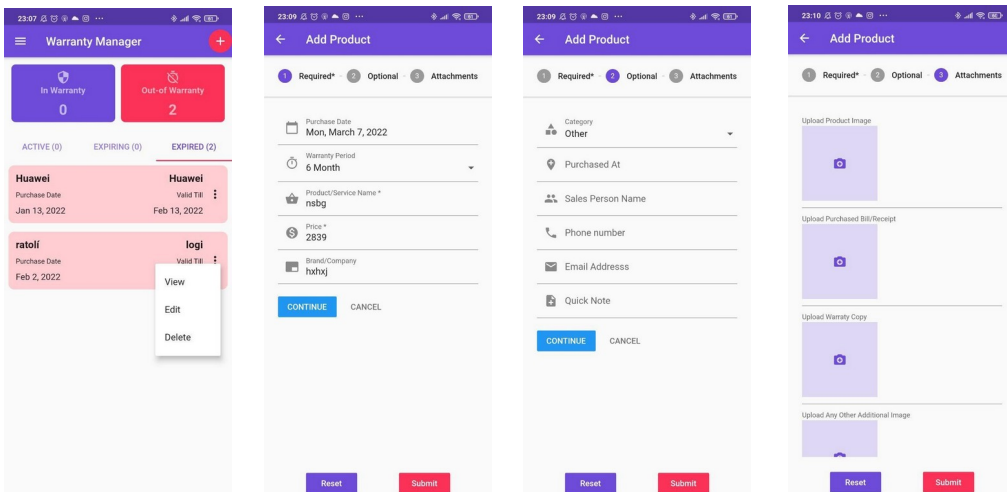
Warranty Sured



Aquesta aplicació no té una estètica massa agradable, i és confús afegir un nou producte, ja que has de navegar primer per categories i subcategories fins a poder arribar a la pantalla d'afegir producte. Com a detall quan s'afegeix apareix un error de clau duplicada tot i que el producte s'enregistra bé.

Com a punt positiu es pot indicar les dades que desa com ara la marca i el model.

Warranty Manager



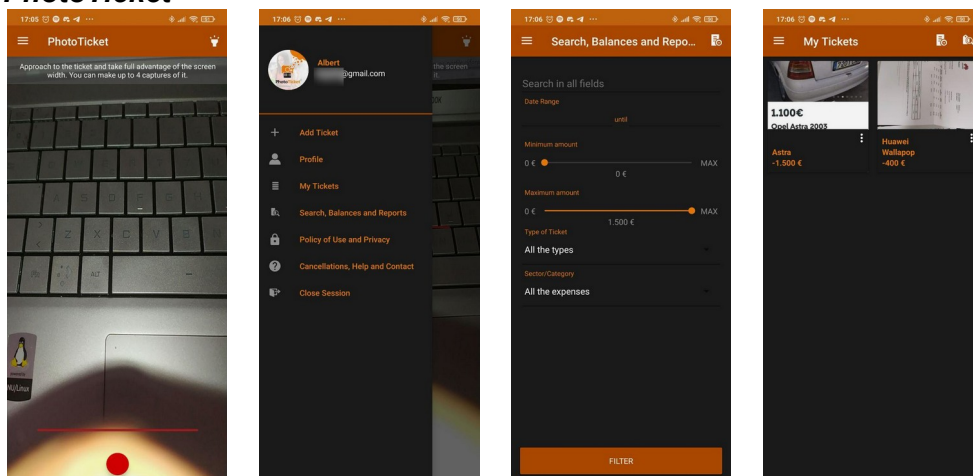
En aquest cas, la usabilitat de l'aplicació és millor que en el cas anterior.

Els menús i botons són autodescriptius i és fàcil afegir un nou producte. Per afegir-lo, són necessaris tres passos, el primer on es defineixen les dades obligatòries, el segon amb les dades opcionals i, al final, les imatges que s'hi vulguin adjuntar.

Les imatges queden associades a un tipus concret. Així doncs, només podrem assignar 5 imatges a cinc categories: de documentació, producte, factura, garantia o altra. Però no permet desar cap més tipus de document o imatge.

Com a punt bo de l'aplicació, el fet que et mostri els productes organitzats per garantia vigent a punt d'expirar o caducada.

PhotoTicket



Aquesta alternativa tot i ser més enfocada a la gestió de tiquets i les compres, no tant a les garanties, és un bon exemple de funcions més avançades que no es troben en les anàlisis anteriors, com el filtratge segons diversos camps. Un bon detall és que només començar s'activa la càmera per fer una fotografia i començar a afegir un producte. Tot i

que pot ser funcional segons quin sigui el context d'ús, hauria de ser una funcionalitat opcional que pugui elegir l'usuari.

Com a conclusió de les aplicacions analitzades, podem extreure que, si bé la part de gestió de productes pot estar més o menys coberta, però cap de les aplicacions analitzades se centra en la part documental del producte, com els documents relacionats o les incidències.

Entrevistes en profunditat

Un cop analitzades les aplicacions alternatives al projecte, es realitzen algunes entrevistes en profunditat. S'han elegit perfils d'usuari diferents per tal de poder extreure respostes de punts de vista diferenciats.

Les entrevistes s'han mantingut remotament. Tot i que s'ha preparat un petit guió, per tal de poder extreure'n el màxim partit a les entrevistes s'ha deixat als usuaris que prenguin partit amb noves idees o suggeriments.

El guió és el següent:

- Preguntes personals per tal de poder definir el perfil (edat, professió, aficions...)
- Indagacions sobre com gestionen les garanties dels productes i en desen els rebuts.
- Una petita introducció de com és el projecte. Finalitat i funcionalitats bàsiques.
- Indagacions sobre dades a desar per als productes comprats.
- Indagacions sobre la gestió d'incidències.

Els usuaris elegits són:

- **Anton:** seixanta-nou anys i jubilat. Tot i estar jubilat, té interès en la mecànica i es dedica a crear prototips i maquinària agrícola. Normalment compra en botigues físiques tot i que també compra material en comerç electrònic.

L'entrevista es manté per videoconferència. De bon començament se sent atret per la idea de l'aplicació, ja que diu que és molt despistat i que no és estrany que perdi els rebuts un cop adquirits nous productes.

Quan es pregunta pel sistema que utilitza per mantenir el control, respon que no en té cap de fix, o bé des de els rebuts de compra dins la caixa del producte que a vegades des de, o bé els deixa en un calaix en el millor dels casos.

Quan es profunditza en els detalls que l'hi interessaria des de dels productes, l'hi sembla interessant des de data de compra, durada de la garantia, model i marca del producte. Mostra especial interès a poder des de el número de sèrie. Comenta que d'aquesta

manera no tan sols es podrà usar per a mantenir el registre de garanties sinó que, a més, es podria usar en cas de reclamar a una assegurança o en cas de robatori.

En centrar el tema en el registre d'incidències, ho troba interessant i creu que amb poder introduir el número d'incidència, adjuntar un fitxer, i poder afegir-hi notes n'hi ha prou.

- **Gerard:** quaranta-dos anys, enginyer industrial. Treballa en el sector de medi-ambient en un parc natural. És aficionat al món "maker" i acostuma a comprar per internet.

Es contacta amb en Gerard amb videoconferència, la idea l'hi sembla interessant tot i que no acaba de fer-se la idea de com ha de ser l'aplicació.

El primer que es pregunta és pel sistema que té de mantenir el control. En Gerard comenta que quan compra productes més cars té una carpeta on desa els rebuts, però que els productes més barats o no tan importants moltes vegades no acaben a la carpeta i es perden.

Quan es parla de com desar el registre de productes, troba útil poder afegir un nou producte a partir de la fotografia del rebut, les dades que comenta que l'hi interessaria desar són la data de compra, el lloc, la durada de la garantia, el preu i el nom.

Quan es comença a parlar sobre el registre d'incidències l'hi sembla correcte desar número de referència document i notes relacionades amb el producte. Però en aquest moment troba que potser seria interessant poder desar un altre tipus de producte que no fos tangible, com ara contractes de telefonia assegurances de cotxe, etc. Comenta que ha hagut de fer reclamacions en diferents serveis i que una aplicació així l'hi hagués resultat interessant per poder mantenir el registre i l'historial de reclamacions.

- **Marina:** trenta-dos anys, funcionària tècnica en Topografia. Compra productes de moda i de salut (alimentació, esport...) en botigues físiques i comerç electrònic indistintament.

La Marina, de primer moment no l'hi sembla una aplicació que faria servir, però la idea no l'hi desagrada.

Com en les altres entrevistes, es pregunta pel sistema de gestió actual. Comenta que quan una cosa l'hi sembla important de no perdre en desa una còpia al núvol de manera que la té accessible quan la vol tornar a tenir.

En consultar quines dades resulten interessants per desar en l'aplicació, comenta que, a part de les ja esmentades, l'hi sembla interessant poder desar la data de devolució que té el producte.

Pel que fa al registre d'incidències no sembla donar-l'hi massa importància i respon que si l'hi pot resultar interessant desar-hi una nota i la data de la incidència.

Quan s'estava finalitzant l'entrevista, a la Marina proposa que podria ser interessant incorporar una secció on es poguessin desar les targetes de fidelització dels establiments en els quals compra.

Enquestes

A partir de l'anàlisi de les alternatives i de les entrevistes en profunditat ja s'ha aconseguit un bon punt de partida. Arribat aquest punt, una aproximació quantitativa com són les enquestes ens ajudarà a decidir quins perfils i contextos d'usuari tindrà la nostra aplicació.

S'ha preparat una enquesta, que s'ha ofert mitjançant una pàgina oberta per tal de poder arribar a més gent, mitjançant el programari ohMyForm⁴.

L'enquesta que s'ha allotjat en un domini propi era d'accés obert i comptava de pantalla d'introducció, enquesta i agraïment. El cos de l'enquesta s'ha deixat amb poques preguntes per tal que respondre fos ràpid i no fos tediós. El guió és el següent:

1. Quin és el teu gènere?
 - Femení
 - Masculí
 - Indefinit
2. Quina és la teva edat?
3. Quin és el teu sector professional?
4. Com puntues la teva experiència amb aplicacions mòbils?
Per tal de poder classificar els perfils d'usuari és d'ajuda conèixer la seva habilitat amb les noves tecnologies, sobretot amb els telèfons intel·ligents. Puntua d'un a cinc.
5. Quin tipus de producte afegiries a l'aplicació?
 - Productes comprats físicament
 - Productes comprats En línia
 - Altres
6. Hi afegiries altres tipus de producte quins?
Subscripcions a serveis, Internet...
7. En quin moment afegiries un nou producte?
 - Al mateix moment de comprar-ho
 - Un cop a casa, al moment de treure'l de la caixa
 - En un altre moment
8. Quines dades trobes útils en desar un producte?
Per a cada producte s'hi associen característiques per tal de poder identificar-lo millor i tenir-ne millor control.
 - Data de la compra

⁴«OhMyForm | OhMyForm is the best open source form solution for the web», consulta 12 març 2022, <https://ohmyform.com/>.

- Duració de la garantia
- Preu
- Número de referència
- Model
- Marca
- Lloc de la compra

9. Consideres que falta algun camp addicional al llistat anterior?

10. A part de la foto del rebut, quantes imatges creus necessàries per producte?

11. Et sembla útil poder desar altres tipus de fitxer que no siguin imatges?

12. Com prefereixes organitzar els productes?

- Categories
- Filtres
- Vigència de la garantia

13. Quines altres dades desaries en el registre de reclamacions?

Per a cada producte es manté un registre d'incidències o reclamacions. Per a cada registre es desarà:

- Número de registre
- Document
- Descripció

14. Trobes a faltar alguna funcionalitat que pogués ajudar-te en el teu registre de garanties i que no s'hagi tingut en compte?

Anàlisi de l'enquesta

L'enquesta l'ha respost unes 40 persones amb diferents perfils, tot i que tots els enquestats són majors de vint-i-cinc anys.

Una primera dada per extreure els perfils d'usuari que han respost és òbviament la relació entre l'edat i l'expertesa amb la tecnologia. En el gràfic de la Figura 3, observem com hi ha una relació directa entre l'edat i l'experiència tecnològica diferenciant dos grups segons siguin majors o menors de cinquanta anys aproximadament.

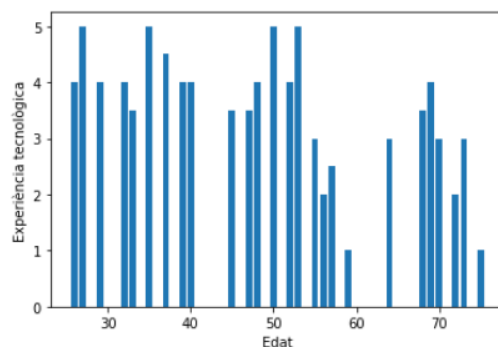


Figura 3: Relació entre edat i experiència tecnològica.

Quan avaluem els tipus de producte es veu com la majoria usarien l'aplicació per mantenir el registre d'ambdós tipus de compres En línia i física. Tot i que hi ha algun cas que només els interessa un dels dos tipus, això no és decisiu per a poder enfocar l'aplicació a un dels tipus de compra. A més hi ha alguns casos que hi afegirien altres tipus de producte com ara assegurances de cotxe, contractes de telefonia, compres per altres apps com Wallapop o Todocoleccion.

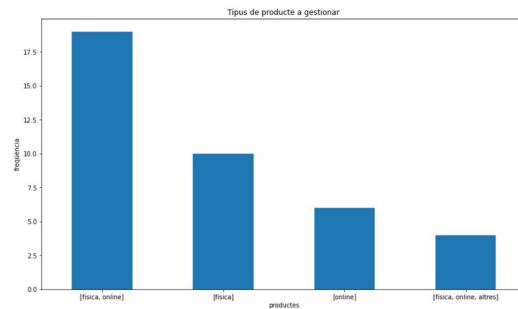


Figura 4: Tipus de compra

Les dades a desar del producte són bastant concloents, no es mostra el gràfic, ja que gairebé tothom ha seleccionat totes les opcions. A més, alguns usuaris han indicat que també els interessaria:

- Data límit per canviar/retornar.
- Notes i comentaris.
- Mètode de pagament
- Si és un regal

Quan s'ha preguntat per quin moment afegir el producte, la majoria ha contestat que és en el moment de desembalar la caixa, l'*unboxing*. Tot i que molta gent també ha elegit el moment de la compra. L'opció d'altres ocasions és més residual, però el cas d'ús és molt similar al de l'*unboxing*.

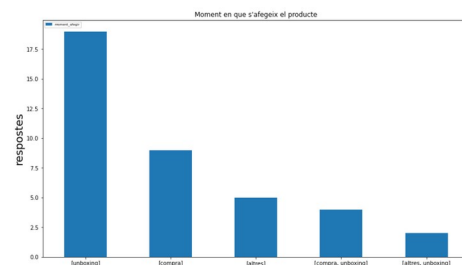


Figura 5: Moment d'afegir el producte

Un altre punt important és el de la visualització dels productes. En aquest cas no hi ha una clara opció que predomini per sobre de les altres. Per tant, una bona solució crec que pot ser la d'implementar un filtre en el qual es pugui filtrar per l'estat dels productes i afegir una opció de categories.

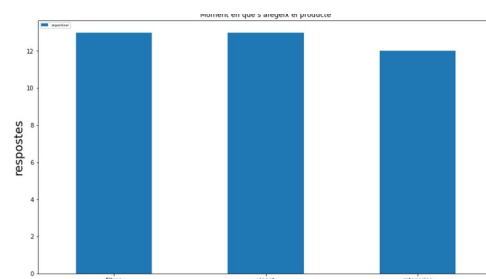


Figura 6: Visualització de productes

2.1.2 Disseny conceptual

A partir de les dades que tenim de la indagació feta en la recerca es pot començar el disseny conceptual de l'aplicació. Aquest procés es farà en dues fases:

- Definició de personatges.
- Definició d'escenaris d'ús.

Definició de personatges i escenaris d'ús

A partir de les enquestes que s'han fet, es veu una relació entre edat i coneixement tecnològic. Partint d'aquesta premissa, es pot definir perfils segons l'edat i el tipus de compres que realitzen, productes amb garanties i per serveis i compres sense garantia com ara compres de moda.

Amb aquestes dades s'han creat els següents personatges amb els respectius contextos d'ús.

Perfil



Nom: Àngela
Edat: 32
Professió: Dissenyadora d'interiors
Nivell tecnològic: 4,5/5

Biografia

L'Àngela viu en parella en a la ciutat on treball. La seva feina l'hi permet una flexibilitat horària que fa que pugui fer activitats durant el dia. El seu temps lliure l'hi agrada practicar esport, les activitats a de muntanya i anar "de compres".

És una usuària habitual de telèfon mòbil i és molt activa en les xarxes socials ja que les utilitza professionalment. Acostuma a comprar articles de moda, tecnologia i gadgets per l'esport.

Context d'ús

L'Àngela te prevista una ruta per el Pirineu com que estem en pla hivern ha decidit comprar una jaqueta d'una marca coneguda de muntanya, com que encara falta per començar la ruta decideix emportar-se la jaqueta tot i que no està convençuda depèn de com ho vegi amb la seva parella la tornarà o se la quedarà.

En comprar la jaqueta fa la foto al tiquet de compra per desar-lo a la aplicació introdueix les dates, la imatge i la data màxima de retorn.

Perfil



Nom: Ramón
Edat: 66
Professió: Jubilat
Nivell tecnològic: 3/5

Biografia

En Ramon fa 2 anys que està jubilat, és pare de 2 fills i viu amb la seva parella. Dedicava el temps a pintar i dibuixar, el yoga i la muntanya, però des de que s'ha jubilat, que s'ha dedicat amb més fervor a la seva afició de la fotografia.

Usa el telèfon intel·ligent cada dia, amb un ús d'aplicacions bàsiques com el correu, aplicacions de xat, Youtube i navegar. No acostuma a comprar molt sovint, però sí que l'hi agrada comprar-se productes

Context d'ús

En Ramón s'ha decidit a comprar un nou objectiu per la càmera aprofitant el "black friday". Un cop arriba a casa el desempaqueta la caixa, fa la foto a la factura, introdueix totes les dades del producte i crea una nova entrada a l'aplicació.

Passats uns mesos, troba que l'objectiu que ha comprat fa un soroll estrany al enfocar. Decideix posar-se en contacte amb el SAT i enviar l'objectiu a revisar. Entra a Garant-it i hi afegeix un registre d'incidència amb el document que l'hi han donat a l'entrega de l'objectiu. A més, hi posa com a nota el l'hi han dit "Això podria ser un problema del motor d'enfocament o de l'engrenatge... revisarem que les juntes d'estanquitat estiguin bé."

2.1.3 Prototipat

Flux de navegació

El disseny del prototipat va estretament lligat al flux d'utilització de l'aplicació. Aquests fluxos s'han definit tenint en compte els contextos d'ús que s'han extret en l'apartat del disseny conceptual.

La navegació comença amb les pantalles d'inici de sessió, a partir d'aquí, la navegació depèn de cada cas d'ús la navegació.

En el diagrama de la Figura 7 queden reflectides les pantalles i la relació entre elles. S'han diferenciat segons son pantalles de llistat, de detall o d'edició.

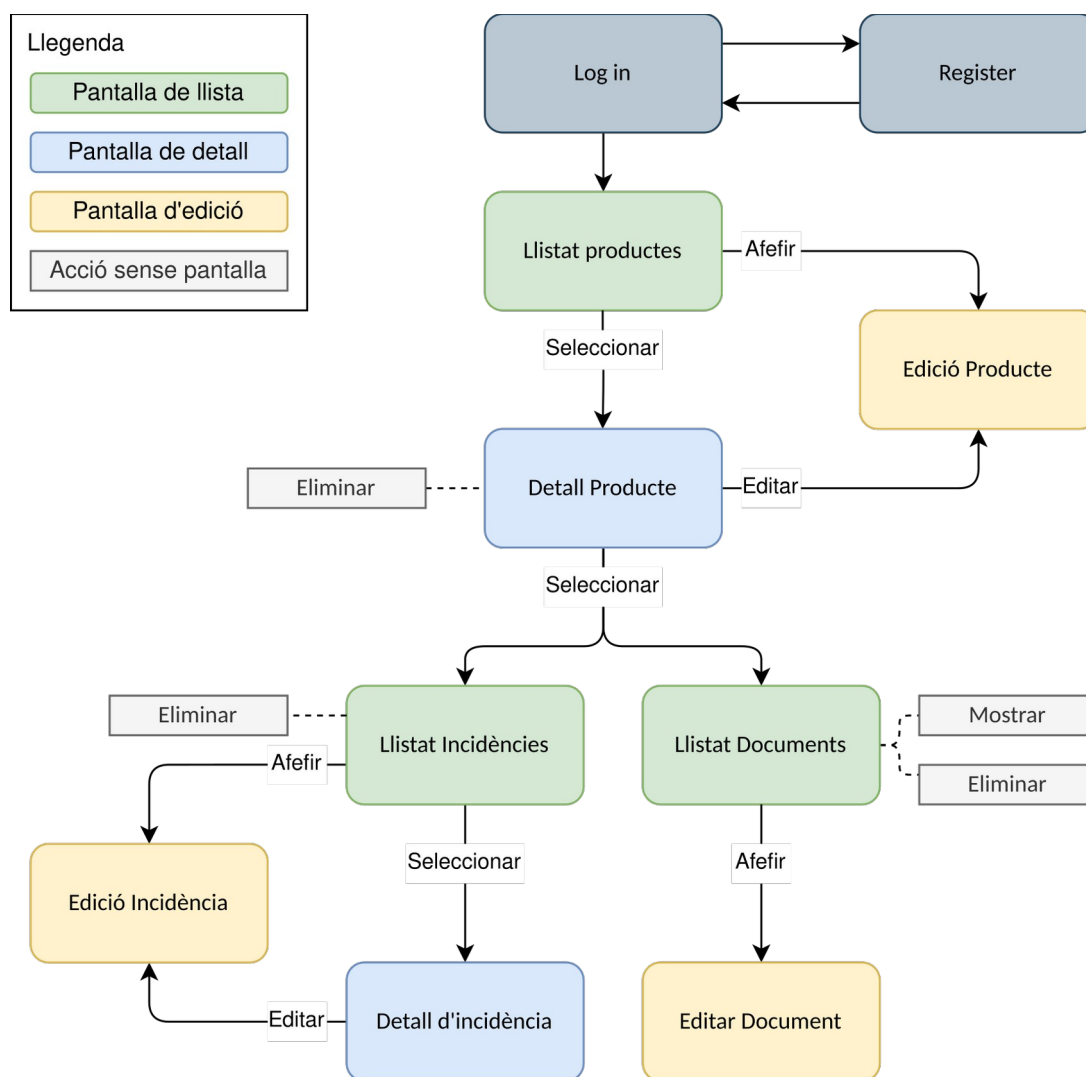


Figura 7: Diagrama de flux de la navegació

Prototip

Un cop definit el flux de navegació s'han prototipat les pantalles de cada tipus. El prototipat es mostra dividint la navegació en diferents fluxos per simular la navegació interactiva del prototip.

El primer diagrama de pantalles Figura 8 es mostra el flux des de l'entrada a l'aplicació fins a seleccionar o afegir un producte:

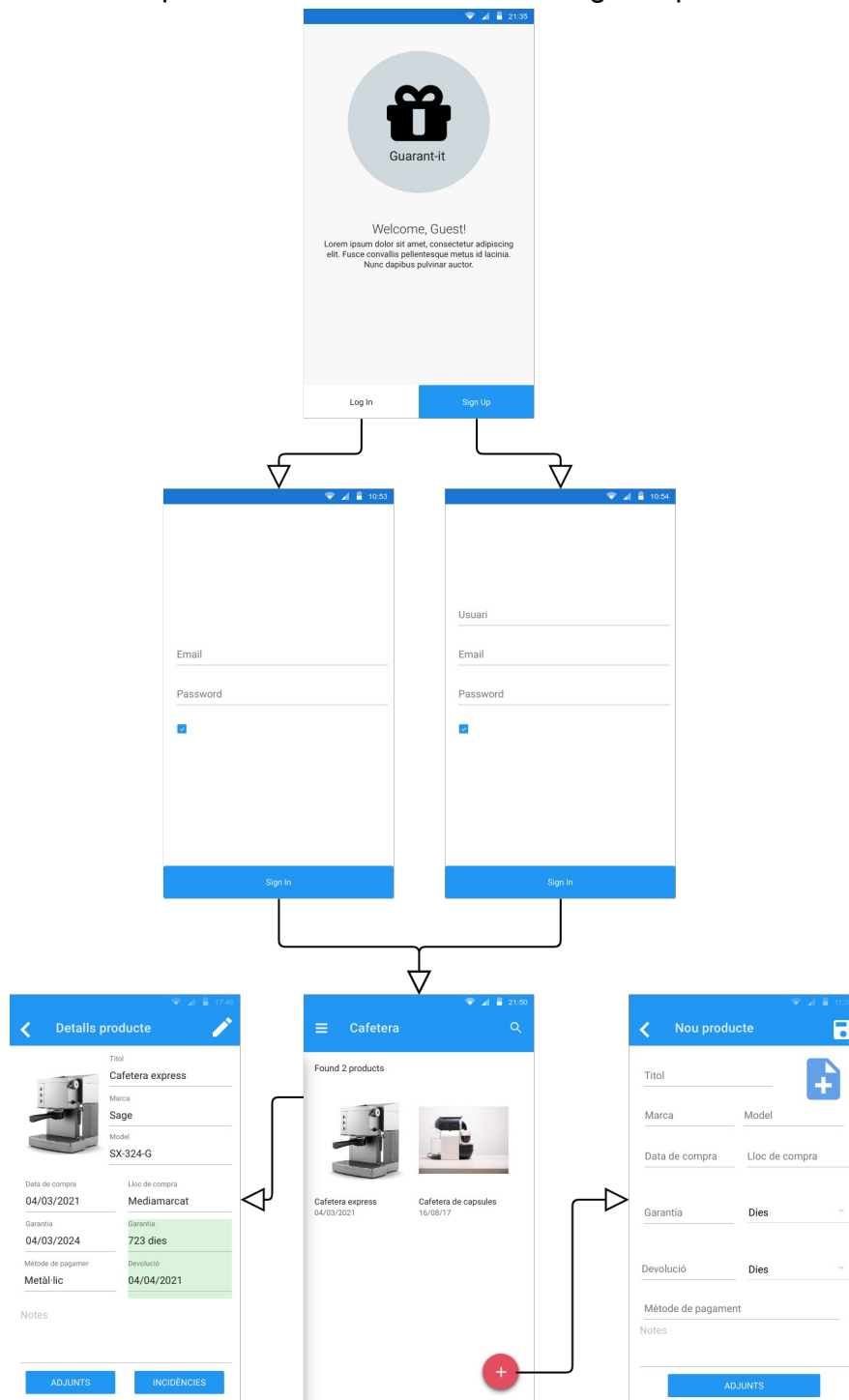


Figura 8: Flux des de la pantalla d'inici de l'aplicació

A la Figura 9 es poden veure les transicions de pantalla a partir de la vista de detall d'un producte.

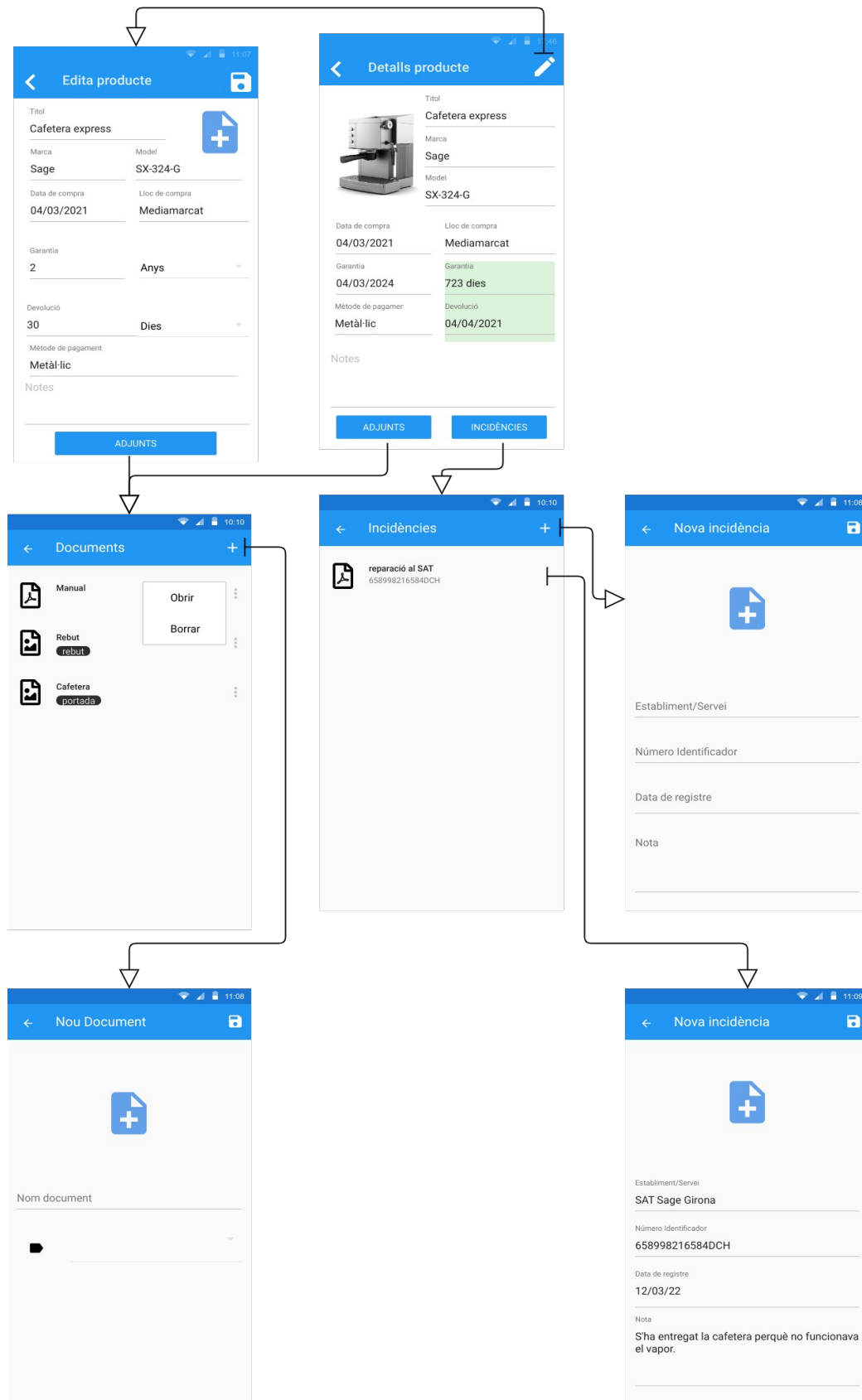


Figura 9: Flux a partir de la pantalla de producte

2.1.4 Avaluació

L'objectiu d'aquesta fase és poder contrastar amb els usuaris que realment el disseny al qual s'ha arribat és funcional i s'adapta a les necessitats. Per tal de poder avaluar el prototip, s'ha generat un model d'interacció amb el qual es poden provar les pantalles i botons amb dades predefinides.

Per tal d'avaluar la usabilitat i es definiran una sèrie de tasques a realitzar i es proposarà als usuaris que els realitzin mentre van explicant el que els van pensant en cada moment. Les reflexions, les emocions, les dificultats que van trobant mentre intenten fer les tasques assignades.

El guió que s'estableix és el següent:

Tasca 1. Registre

1. Arranca l'aplicació
2. Registra les teves credencials
3. Accedeix a la pantalla principal

Tasca 2. Afegir producte

1. Vés a la pantalla principal
2. Afegeix un producte
3. Omple les dades del producte
4. Afegeix

Tasca 3. Afegir documents

1. Selecciona un producte
2. Afegeix un document

Tasca 4. Afegir incidència

1. Selecciona un producte
2. Afegeix una incidència

Tasca 5. Buscar producte

1. Vés a la pantalla principal
2. Fes una cerca de text

2.2 Disseny de l'arquitectura

Si el disseny centrat en l'usuari ens aporta un benefici pel que fa a la usabilitat i funcionalitat de l'aplicació, l'apartat del disseny de l'arquitectura és clau per al correcte desenvolupament i posterior manteniment del producte.

Un disseny de l'arquitectura pensat des de bon començament centrat en la modularitat i el manteniment, permetrà que en posteriors iteracions i, a mesura que evolucioni el producte, aquests canvis no suposin un entrebanc en el desenvolupament.

Selecció de tecnologies

L'aplicació es desenvoluparà amb Android Studio, i es faran servir biblioteques de codi obert que ens permeten utilitzar certes funcionalitats. Per exemple, es farà ús de DaggerHilt⁵ com a gestor d'injecció de dependències.

Pel que fa a la gestió i manteniment de les dades s'han analitzat dues solucions basades en el núvol, Firebase eina desenvolupada per Google i Realm, eina desenvolupada per MongoDB.

Firestore

Firestore ofereix una solució *transversal* per a totes les funcionalitats que necessita l'aplicació. L'eina permet l'autenticació d'usuaris, desar les dades en una base de dades "*en temps real*" i desar fitxers amb Firestore Cloud Storage. A més, consta d'un model gratuït fins que no se superen unes quotes mínimes. Des d'aquest punt de vista és una eina perfecta per a començar el desenvolupament d'aquesta o qualsevol aplicació.

Però Firestore està pensat amb un model *Online First*⁶. Això implica que si bé permet una desconexió temporal sense més problemàtica, es torna inconsistent quan es facin operacions de modificació fora de línia i no permetria el seu ús sense connexió requisit necessari per poder satisfer l'ONF-004.

Realm.io

L'altra opció que s'ha pres en consideració és Realm adquirida i gestionada per MongoDB. Aquesta opció està enfocada des del punt de vista *Offline first*. Això significa que es manté la base de dades en el dispositiu client i se sincronitza amb el núvol quan es té oportunitat. Té

⁵«Hilt», consulta 27 març 2022, <https://dagger.dev/hilt/>.

⁶«Firestore vs Realm - MongoDB Realm», MongoDB Developer Community Forums, 3 octubre 2020, <https://www.mongodb.com/community/forums/community/forums/t/firebase-vs-realm/9929>.

una bona gestió de diferències i, per tant, és capaç de resoldre els conflictes que puguin sorgir de la modificació d'un mateix objecte de manera simultània.

Com a punt positiu, és una tecnologia *codi obert* i, per tant, queda fora de la utilització de tecnologies privatives que personalment és per a mi una preferència.

Per contra, Realm permet desar fitxers de mida petita, com imatges "thumbnails", com a fitxer en binari, però no és aconsellable desar-hi fitxers grans. Això fa que s'hagi buscat una alternativa per poder desar els documents de cada usuari.

La solució proposada és que la base de dades només emmagatzemi la ruta on s'està desant el document, però aquest es desarà en el sistema de fitxers del dispositiu de cada client.

Per tal que el client pugui tenir sempre disponible una còpia dels documents s'habilita l'opció de fer servir el compte personal de Dropbox. D'aquesta manera, en cas d'aparèixer algun problema amb el dispositiu o en desinstal·lar l'aplicatiu, la carpeta de l'aplicació estarà sincronitzada en el núvol i en reinstal·lar l'aplicació tornarien a estar disponibles.

En el diagrama Figura 10 es representa el model de persistència de les dades mitjançant Realm i utilitzant el servei de Realm.io per la seva sincronització amb altres dispositius.

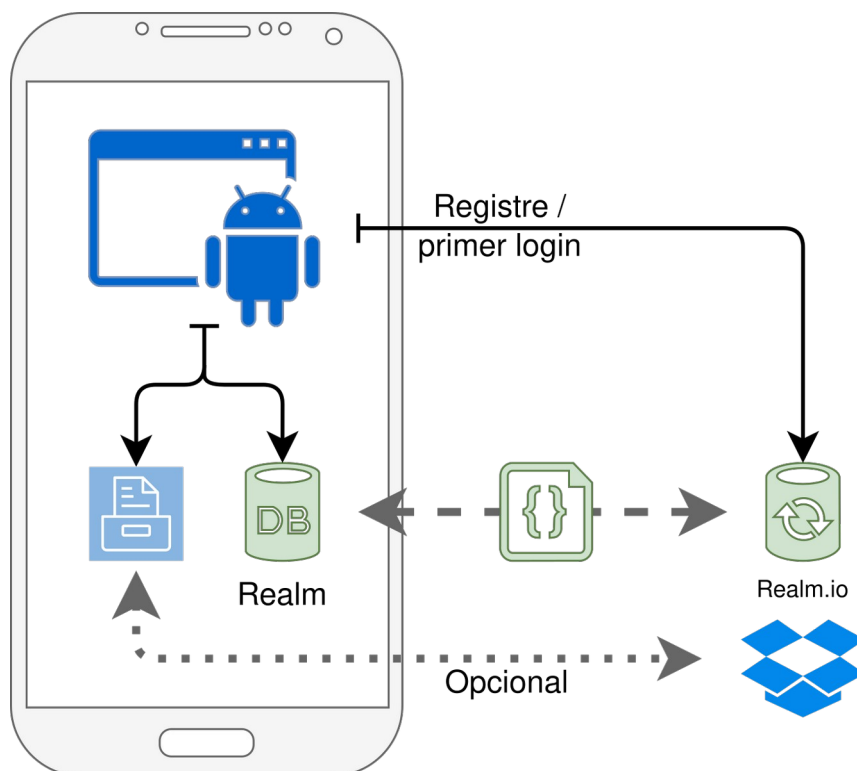


Figura 10: Infraestructura de l'aplicació

Arquitectura

Com s'ha comentat a la introducció, l'elecció de l'arquitectura afectarà el manteniment i a l'evolució del nostre aplicatiu de manera directa.

És per això, que pel que fa al model d'aplicació d'Android es farà servir un model MVVM que ens aporta la separació entre la part visual i la part encarregada de gestionar les dades. D'aquesta manera desacoblem la implementació de les pantalles de les lògiques encarregades de gestionar les dades.

Les vistes només s'encarreguen de mostrar les dades i enviar els esdeveniments que es produeixen durant la interacció de l'usuari. És el ViewModel l'encarregat de rebre els esdeveniments i gestionar-ne la lògica amb els repositoris. També s'encarrega de rebre els canvis provinents del repositori de dades per enviar-lo a la vista.

El repositori és el que s'anomena l'única font de veritat i s'encarrega de persistir les dades i recuperar-ne els canvis o consultes.

A l'article d'Izertis⁷ es detallen els beneficis que aporta el model MVVM al MVC o MVP. Aquest model ens aportarà l'aïllament de les vistes de manera que només es mantinguin en espera de les notificacions a les quals estiguin subscriïdes, no s'encarreguen de cap mena de lògica. Serà la capa ViewModel qui s'encarregui de mantenir les dades de la vista i llançar els esdeveniments pertinents per actualitzar la vista.

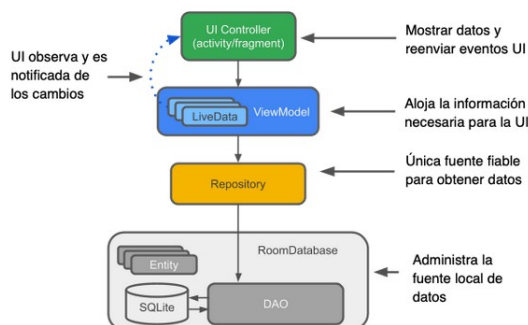


Figura 11: «Componentes de arquitectura de Android, de MVC a MVVM», Izertis

Però a Garant-it, la capa definida a Figura 11 com a *Repository* estarà constituïda a partir d'unes capes extres per tal que l'aplicació sigui més modular i extensible. Com a base es farà servir l'arquitectura hexagonal⁸ o una de les variants de les anomenades *Clean architecture*⁹. Aquests models permeten separar els objectes i lògiques de negoci dels models i

⁷Por Izertis, «Componentes de arquitectura de Android, de MVC a MVVM», Solid GEAR, 20 juny 2019, <https://ahorasomos.izertis.com/solidgear/componentes-de-arquitectura-de-android-de-mvc-a-mvvm/>.

⁸Alistair Cockburn, «Hexagonal Architecture», Alistair Cockburn (blog), consulta 26 març 2022, <https://alistair.cockburn.us/hexagonal-architecture/>.

repositoris de dades de manera que no acoblem el nostre codi a una solució d'infraestructura.

Així doncs, com a resultat d'aplicar aquesta arquitectura, la capa *Repository* passa a contenir les capes *Application*, *Domain* i *Infraestructura* com es mostra al diagrama de la Figura 12.

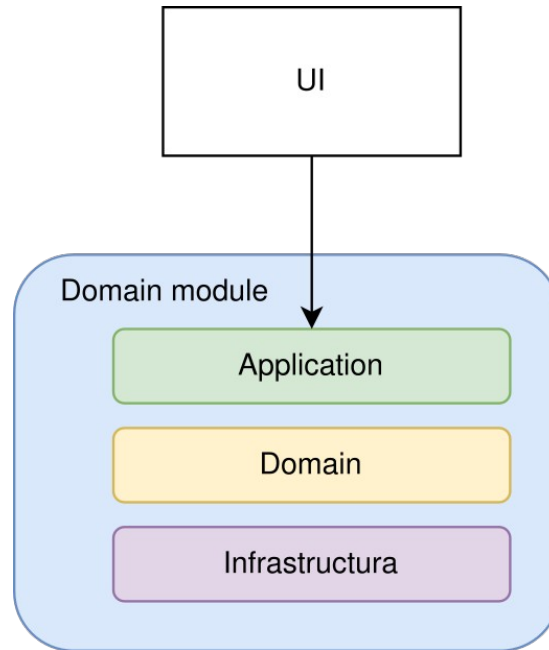


Figura 12: aplicació de l'arquitectura hexagonal

La capa d'*Application* que tindrà les lògiques dels casos d'ús, la *Domain* que conté els objectes de negoci i la *Infraestructura* que conté tot el codi condicionat a la solució de persistència elegida per administrar les dades de l'aplicació.

Un dels beneficis que ens pot aportar aquesta arquitectura és poder fer canvis de tecnologies i serveis externs de manera eficient i sense haver de modificar les classes que s'encarreguen de la lògica de negoci. Per exemple, en aquest cas, s'usarà una solució de persistència amb Realm, però un canvi a Firebase no suposaria una modificació del nostre codi de negoci.

2.2.1 Cas d'ús

Per tal de dur a terme i definir les tasques necessàries en el procés d'implementació de l'aplicació, a partir dels requisits i contextos d'ús recollits en la fase de disseny, es defineixen els principals casos d'ús de l'aplicació.

Amb els casos d'ús aportats a continuació, es tenen els requisits funcionals de la Taula 2 representats i, per tant, una aplicació funcional i amb les necessitats essencials cobertes.

Adicionalment, en el moment de la implementació s'ha de tenir en compte els requisits esmentats a les taules de requisits no funcionals de la Taula 3.

CU-001	Inici de sessió/Registre	
L'usuari entra a l'aplicació y necessita autenticarse per poder recuperar la seva informació		
Flux bàsic	L'usuari ja està registrat a l'aplicatiu	
	1	Introdueix les seves credencials
	2	Prem el botó de login
Flux alternatiu	L'usuari no està registrat a l'aplicatiu	
	1	L'usuari introdueix les seves dades
	4	Prem el botó de registre
	2	S'informa del correu de confirmació
	3	L'usuari confirma el correu
Condicions posteriors	Es redirigeix a la pàgina inicial	

CU-002	Afegir producte
Com a usuari necessito poder afegir productes nous a l'aplicació	
Flux bàsic	L'usuari té una imatge del rebut i vol desar el nou producte
	1 Des de la pantalla principal prem el botó d'afegir
	2 Afegeix les dades al formulari
	3 Prem la icona d'imatge i afegeix la imatge
	4 Desa el producte
Flux alternatiu	L'usuari no té la imatge
	1 Des de la pantalla principal prem el botó d'afegir
	2 Afegeix les dades al formulari
	3 Prem la icona d'imatge i fa la foto al rebut
	4 Desa el producte
Condicions posteriors	El producte està desat a la base de dades i apareix al llistat principal.

CU-003	Afegir document
Com a usuari necessito poder afegir productes nous a l'aplicació	
Flux bàsic	L'usuari té una imatge del rebut i vol desar el nou producte
	1 Se selecciona un producte
	2 Se selecciona els documents del producte
	3 Clic al botó d'afegir
	4 Afegeix les dades al formulari i se selecciona un fitxer
Condicions posteriors	El document s'ha desat i ha quedat associat al document.

CU-004	Afegir incidència
Com a usuari necessito poder afegir incidències a un producte de l'aplicació	
Flux bàsic	L'usuari tè una imatge del rebut i vol desar el nou producte
	1 Se selecciona un producte
	2 Se seleccionen les incidències.
	3 Click al botò afegir
	4 S'introdueixen les dades
	5 Inrodueix un fitxer
	6 Desa la incidència
Condicions posteriors	La incidència s'ha desat i ha quedat associatal document.

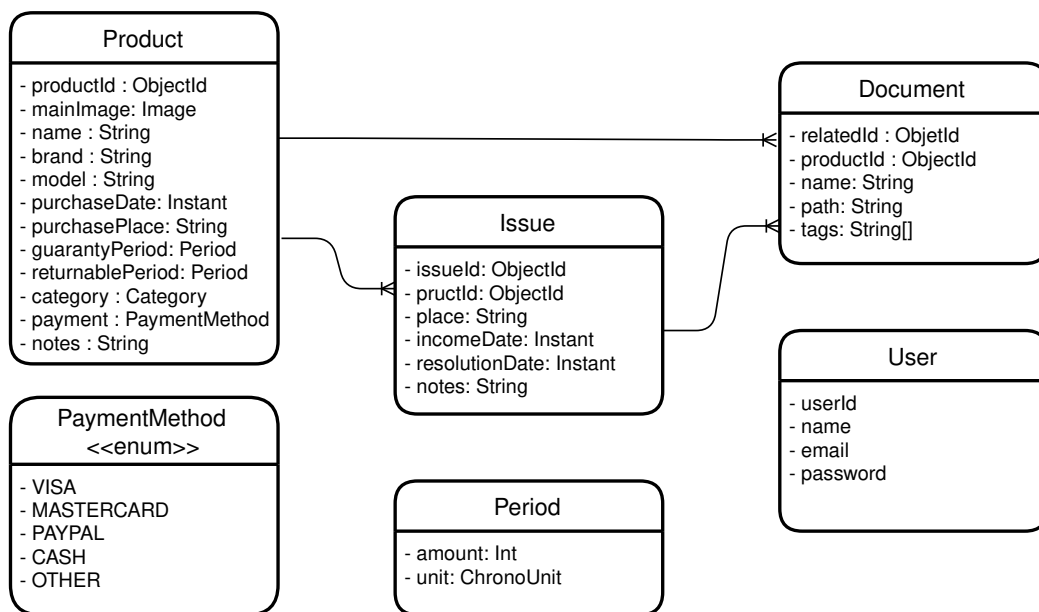
CU-005	Busca producte
Com a usuari necessito poder buscar entre els documents desats	
Flux bàsic	L'usuari necessita buscar un producte per el seu nom
	1 Des de la llista de productes es selecciona la lupa
	2 Escriu el nom d'un producte
Condicions posteriors	En el llistat de productes hi apareix el producte

CU-006	Editar producte
Com a usuari necessito poder editar les dades dels productes existents	
Flux bàsic	L'usuari vol afegir més dades al producte existent
	1 Des de la pantalla principal busca el producte
	2 Selecciona el producte
	3 Edita les dades que necessita

2.2.2 Model de dades

A partir dels casos d'us i les principals funcionalitats trobades durant la definició de Garant-it, s'ha definit un model de dades que pogui satisfer les necessitats establertes.

La base de dades de Realm necessita el que s'anomena una *partitionKey* per tal de poder preservar les dades i definir-ne restriccions d'accés. En el cas de l'aplicació Garant-it les dades estan associades a un usuari i és aquest l'únic que ha de tenir accés. Per això, la clau de partició serà l'id d'usuari que permetrà recuperar només les dades associades al seu userId.



3. Implementació

3.1 Implementació i casos d'ús

Backend

Tal com s'ha definit a l'apartat 2.2 Disseny de l'arquitectura, la part de backend que es fa servir és la solució de mongoDB a la sincronització Realm.

Es pot definir breument aquesta solució, definint-la com la conjunció de dos artefactes Realm i Atlas. Realm és l'encarregat de gestionar la sincronització entre dispositius i els models de dades i, Atlas, és el clúster que allotja la base de dades MongoDB on es persisteixen les dades. Per a l'ajuda al desenvolupament, existeix el mode DEV en el que els clients són els que marquen com és l'esquema de dades, això aporta rapidesa i versatilitat en el desenvolupament.

En l'annex 7.2 Preparació de l'aplicació al núvol de Realm.io/MongoDB es pot veure detalladament la posada a punt de l'entorn.

Aplicació

Per a l'aplicació s'ha decidit usar com a versió base la plataforma la 9.0-**Pie**, ja que, els beneficis que ofereix en el tractament de fitxers compensen la pèrdua de mercat en comparació a 8.0-**Oreo** i, és previsible que cada vegada més dispositius siguin compatibles amb aquesta versió o posteriors.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION	Pie
4.1 Jelly Bean	16		System Indoor positioning with Wi-Fi RTT Multi-camera support Display cutout support
4.2 Jelly Bean	17	99,8%	Security and privacy Android Protected Confirmation Biometric authentication dialogs Hardware security module
4.3 Jelly Bean	18	99,5%	User Interface Secure key import Client-side encryption backups
4.4 KitKat	19	99,4%	Accessibility Improved notifications Improved text support ImageDecoder and new animation classes
5.0 Lollipop	21	98,0%	Media Navigation semantics Convenience actions Magnifier
5.1 Lollipop	22	97,3%	HDR VP9 video HEIF image compression Improved media APIs
6.0 Marshmallow	23	94,1%	
7.0 Nougat	24	89,0%	
7.1 Nougat	25	85,6%	
8.0 Oreo	26	82,7%	
8.1 Oreo	27	78,7%	
9.0 Pie	28	69,0%	
10. Q	29	50,8%	
11. R	30	24,3%	

Com a entorn de desenvolupament s'ha usat kotlin i la compilació mitjançant Gradle que avui dia són els estàndards en el desenvolupament Android.

Per a la implementació s'ha fet ús de llibreries externes i de tercers com a ajuda per al desenvolupament.

Les principals llibreries usades fins al moment en la implementació són:

- SDK Realm.io
→ <https://www.mongodb.com/docs/realm/sdk/kotlin/>
- realm android adapters: Per la implementació dels adaptadors del RecyclerView
→ <https://github.com/realm/realm-android-adapters>
- kotlinx-corrutines: gestió de fils dins l'aplicació.
→ <https://github.com/Kotlin/kotlinx.coroutines>
- dagger-hilt: Injecció de dependències
→ <https://dagger.dev/hilt/>
- Glide: Gestió dels recursos d'imatge
→ <https://bumptech.github.io/glide/>
- EventBus: Gestió per a la publicació i consum d'esdeveniments dins l'aplicació.
→ <https://greenrobot.org/eventbus/>

La implementació de casos d'ús s'ha executat a partir dels mòduls definits Producte, Document, i Incidència.

Per a cadascun dels mòduls s'han anat afegint les pantalles vistes i viewModels per la part de UI i els models de dades, serveis i repositoris per la part d'aplicació.

La gestió entre model i pantalla s'ha fet a partir de databinding que ens permet gestionar els valors dels formularis a partir d'un objecte o model. Per fer-ho s'ha hagut d'afegir a la configuració de gradle el paràmetre 'databinding = true'.

D'aquesta manera hem pogut associar un objecte a partir del model de dades i associar els camps del formulari a les propietats de l'objecte permetent l'actualització bidireccional entre formulari i objecte.

```
<data>
  <import type="dev.agiro.guarant_it.core.product.domain.value_objects.PaymentMethods" />
  <variable
    name="product"
    type="dev.agiro.guarant_it.core.product.domain.Product" />
</data>
```

Figura 13: Variable product en l'àmbit del layout

Un exemple es troba en el formulari d'edició de producte, on es defineix una variable en el fitxer de layout.xml S'ha produït un error: no s'ha trobat la font de referència i un cop associat es pot accedir i actualitzar els valors utilitzant les propietats de l'objecte definit.

En cas de necessitar només l'actualització del formulari amb els valors del model es fa servir la notació '@{model.propietat}', però si el que es necessita és el que s'anomena "two way databinding" s'ha d'afegir un '=' després de l'arrova. A to il·lustratiu mostrem la definició de l'input del títol en la pantalla d'edició.

```
android:text="@={product.title}"
```

Per tal de facilitar el procés de traducció tots els texts que s'han usat estan els fitxers de recursos de cadenes de text, definits en el fitxer 'values/strings.xml'. D'aquesta manera, només s'ha hagut d'afegir els fitxers amb les respectives traduccions en la carpeta value_ca o amb el sufix corresponent a la llengua dels textos.

Tot seguit es mostren els detalls d'implementació de l'aplicació agrupats pels casos d'ús a mesura que s'han anat implementant:

Cas d'ús de registre

El cas d'ús de registre és el primer cas d'ús amb el qual es troba l'usuari. De moment s'ha decidit restringir el mètode de registre a correu electrònic i mot de pas.

Com que cal estar registrat per tal que el backend pugui desar les associades a l'usuari. Només entrar a l'aplicació l'activitat principal i redirigeix a l'activitat LoginActivity si detecta que no t'has registrat en el dispositiu.

Un cop dins de l'activitat de login hi han dues possibilitats, registre i login, en el cas de registre si tot va correctament es mostra un avís per tal que es confirmi el correu. Un cop es clica a l'enllaç de confirmació l'app deixa l'usuari completament registrat i es pot procedir a entrar mitjançant el login. La implementació d'aquest procés queda explicada en detall en l'annex 7.2 Preparació de l'aplicació al núvol de Realm.io/MongoDB.

En el següent diagrama de classes es pot veure com s'han separat les capes de UI i lògica de negoci.

En la part de la UI s'hi troba MainActivity que fa ús del LoginActivity, nutrida per LoginViewModel que conté els LiveData i mètodes login/registre per tal de poder interactuar amb els casos d'ús.

Per la part de la lògica de negoci, hi veiem els casos d'ús de registre/login que fan ús del repositori RealmLoginRepository per interactuar amb el *backend*.

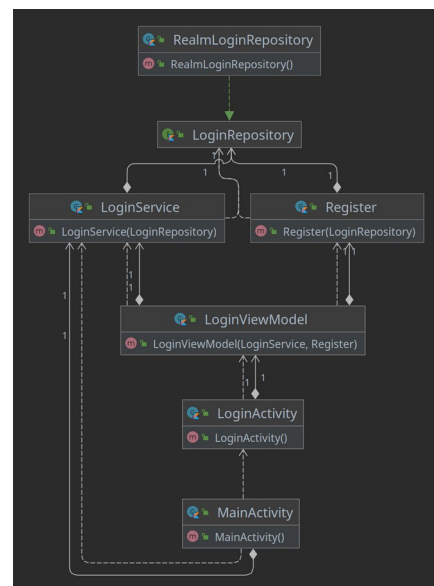


Figura 13: UML del cas d'ús Inici de sessió/registre

Casos d'ús de Producte

Aquest grup de casos d'ús és el més complex, ja que realment és el cos de l'aplicació. Consta de tres casos que ens permeten el llistar, mostrar detalls, editar i crear productes. S'ha decidit modificar la vista de detall de producte seguint els consells rebuts durant la fase de disseny (moure els botons per canviar a la vista d'adjunts i incidències a la part de dalt). Ara, el detall del producte es visualitza com un conjunt de pestanyes que

contenen detalls del producte, documents i incidències, aquests dos últims explicats en els casos d'ús corresponents.

Així tenim una primera vista amb el llistat i la funcionalitat de cerca formada per un RecyclerView que s'alimenta de dades a partir d'una classe ProductAdapter que s'encarrega d'actualitzar el llistat en temps real segons hi hagi modificacions a l'aplicació o al *backend*. Cada item de la llista té associat un esdeveniment que en clicar al damunt ens portarà a l'activitat de detall.

El botó d'afegir producte activa l'activitat d'edició amb un nou producte acabat d'inicialitzar en el LiveData. En el formulari d'edició es poden afegir les dades bàsiques així com els períodes de garantia i retorn.

A més a més, es pot afegir una imatge principal, ja sigui a partir d'un fitxer existent, o una fotografia agafada des de la càmera. Per poder decidir s'obra un Menú on es tria una de les dues opcions. En el cas de la fotografia es desa un fitxer temporal mentre no es desa el definitiu en la carpeta corresponent del producte.

L'activitat de detall és una activitat formada per tres fragments, de moment ens centrem en el primer que és el de detall. En aquest fragment hi podem veure els detalls que hem anotat durant l'edició o creació del producte així com el càlcul dels dies que queden per finalitzar els períodes de garantia i retorn. També des d'aquesta activitat és des d'on es pot eliminar el producte i accedir a l'edició mitjançant els botons de la part superior mostrats a la Figura 13.

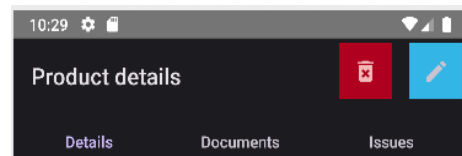


Figura 13: Botons d'editar i eliminar producte

El botó d'edició obrirà la mateixa activitat que en crear un nou producte, però aquesta vegada el model del databinding es nodreix amb el producte recuperat des de la base de dades a partir del seu Id.

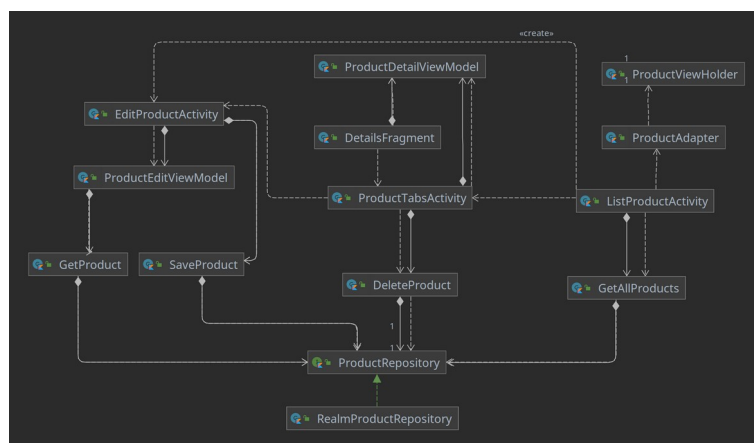


Figura 14: UML casos d'ús producte

El diagrama UML de la Figura 14 s'hi pot veure la separació que hi ha entre les capes i els tres casos d'ús de producte.

Casos d'ús de Documents

El segon fragment que compon els detalls d'un producte és el fragment de Documents. Aquest fragment, si bé és simple quant a la visualització, la lògica és més complicada, ja que és l'encarregat de gestionar l'emmagatzematge dels documents al dispositiu.

Per tal de poder accedir als documents emmagatzemats i desar-ne de nous s'ha hagut de declarar permisos especials a l'aplicació.

```
<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
```

Els noms dels fitxers es creen amb l'ID del nou Document creat i l'extensió del fitxer seleccionat i, es desen a l'emmagatzematge intern de l'aplicació. Per a cada producte es crea una carpeta amb l'identificador de producte, així, tots els documents relacionats amb un producte queden desats de manera agrupada. Aquest procés es fa en el cas d'ús de saveDocument.

Per al llistat s'ha utilitzat altra vegada l'opció d'un recyclerView per mitjà d'un DocumentAdapter i el corresponent DocumentViewHolder per a cada registre de la llista. Cada registre de document conté un menú que en clicar obre un desplegable on, de moment, només apareix l'opció d'eliminar.

L'acció d'eliminar llança un esdeveniment que recull i processa l'activitat pare i s'encarrega d'executar l'acció pertinent mitjançant el cas d'ús de deleteDocument.

Per altra banda, des del llistat podem afegir nous documents mitjançant el formulari que ofereix l'activitat AddDocumentActivity. El formulari de nou document ens permet donar un nom al document, per defecte el del fitxer seleccionat, una llista d'etiquetes separades per comes, i la selecció d'un fitxer des de l'explorador de fitxers.

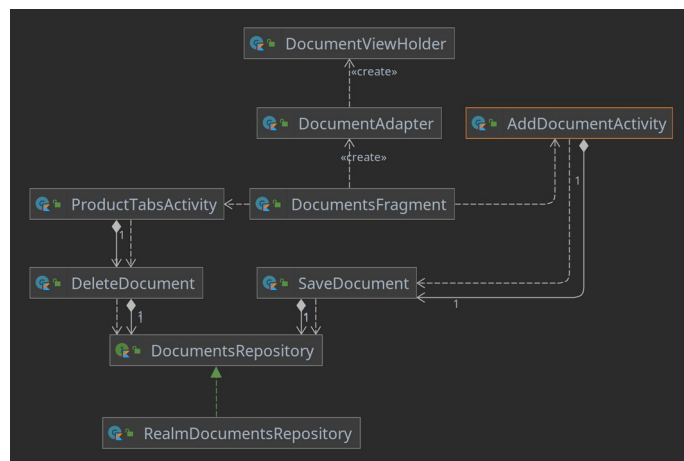


Figura 15: Casos d'ús de documents

Casos d'ús d'incidències

El model d'incidències s'ha evolucionat un xic més que en el plantejament inicial per poder definir diferents actualitzacions per a cada incidència. D'aquesta manera, els fitxers no van adjunts a la incidència sinó a l'actualització d'incidència. Això permetrà poder desar diferents fitxers d'alta, reclamació i finalització... associats a una mateixa incidència. A part, també permet portar un registre més precís dels esdeveniments que van succeint en el transcurs de la incidència. Els documents desats a partir d'una incidència queden associats a aquesta i no es poden eliminar mentre aquesta continuï existint.

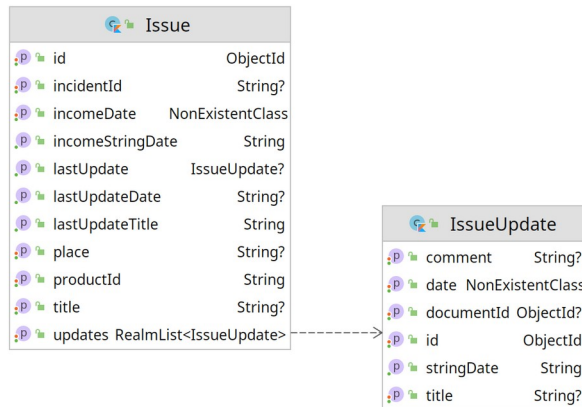


Figura 16: Nou model de dades d'Incidències

Un cop evolucionat el model de dades, les pantalles relacionades amb la gestió d'incidències són tres, el llistat d'incidències, l'edició i creació d'incidències i la creació d'una actualització.

El llistat d'incidències és un recyclerView encarregat d'actualitzar el llistat de manera reactiva, cada registre mostra les dades bàsiques i les dades de l'última actualització. En clicar el registre s'obra l'activitat d'edició de la incidència on es mostra els detalls de la incidència, el llistat d'actualitzacions i el botó d'afegir actualització.

La pantalla d'actualització permet definir un títol, un comentari, la data i adjuntar un document a l'esdeveniment. Per a adjuntar el document s'ha reciclat la pantalla d'afegir document i s'hi desa una etiqueta associada d'incidència.

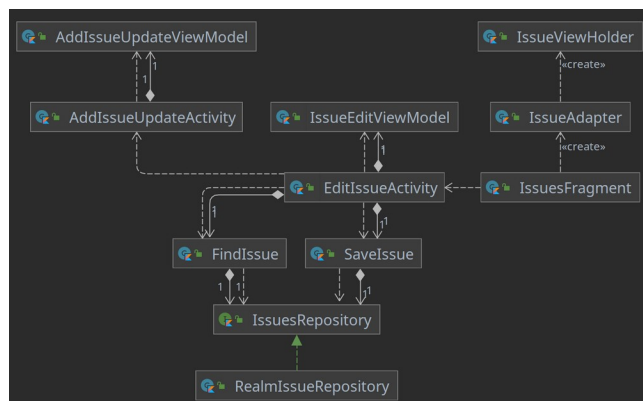


Figura 17: Uml casos d'ús incidència

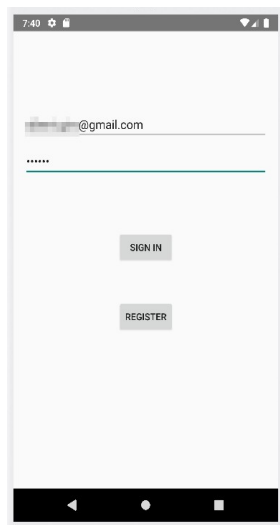
3.2 Tests

A continuació s'han realitzat els tests manuals per tal de validar les funcionalitats de l'aplicació. Les proves s'han fet usant un terminal virtual amb la versió Pie d'Android instal·lada.

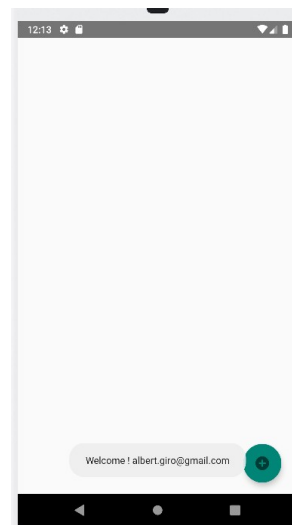
Per fer-ho s'ha seguit els guions plantejats a l'apartat 2.2.1 Cas d'ús. D'aquesta manera, s'ha pogut validar que les funcionalitats establertes per aquesta entrega i validar-ne els fluxos definits.

Cas d'ús CU-001 – Inici de sessió/Registre

Flux bàsic – usuari ja registrat

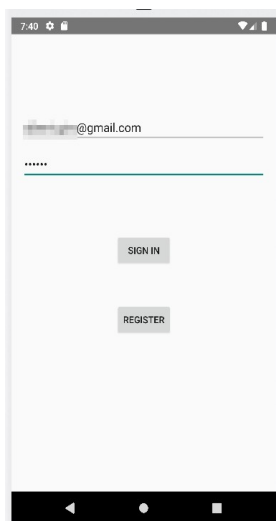


Insereix les dades

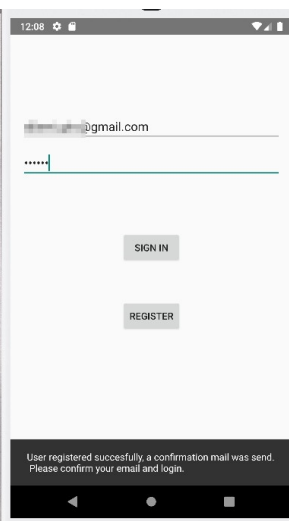


Prem el botó de login

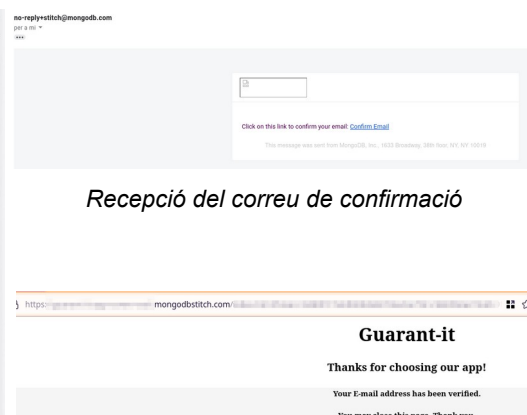
Flux alternatiu – usuari no registrat



Insereix les dades



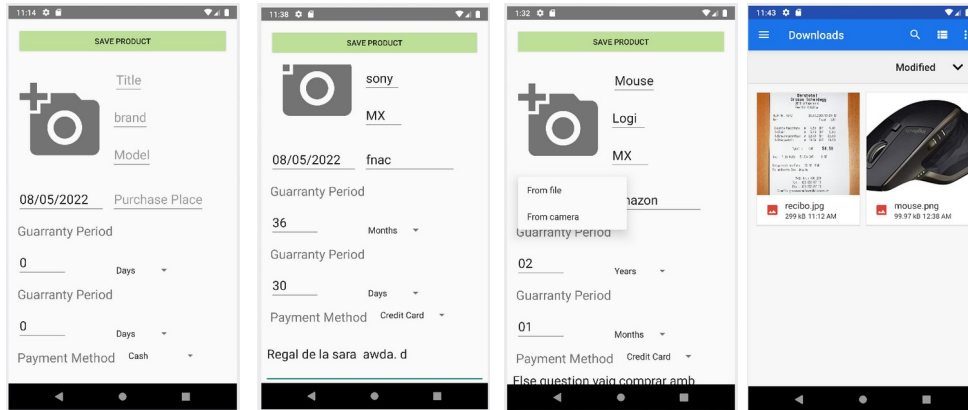
Prem el botó de registre



Confirmació del correu

Cas d'ús CU-002 – Afegir productes

Flux bàsic – una imatge desada el telèfon

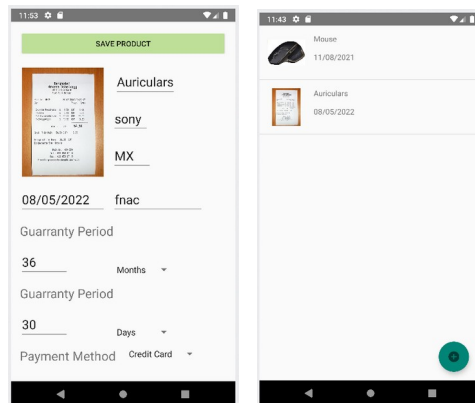


Prem el botó afegir

Omple el formulari

Afegir des de fitxer

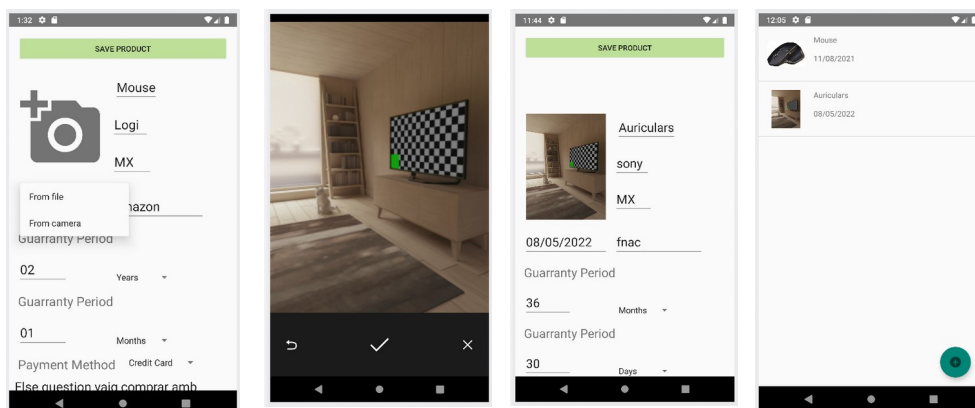
Selecciona imatge



Desa el nou producte

El apareix al llistat

Flux alternatiu – imatge amb la càmera



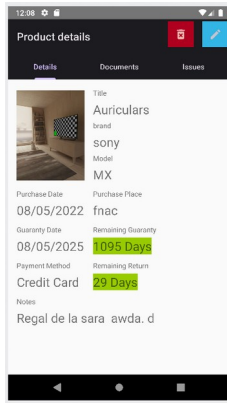
Selecciona des de càmera

Fa la fotografia amb la càmera

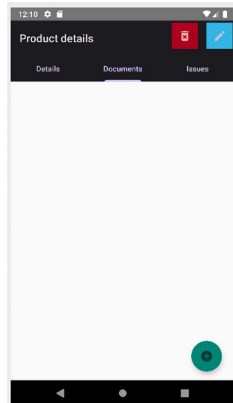
Prem desar el producte

El producte es desa correctament

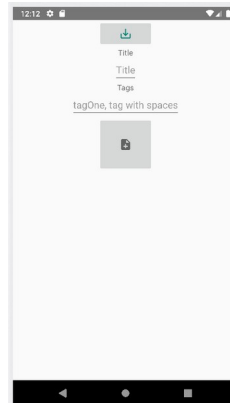
CU-003 – Afegir productes



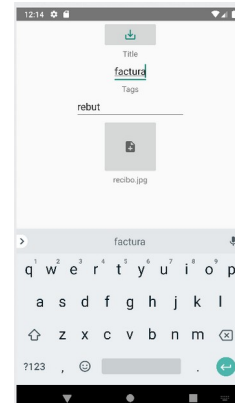
selecciona un producte



Selecciona documents del producte



Prem afegir document

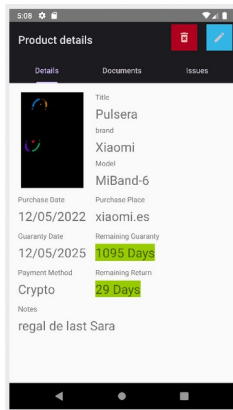


Omple el formulari, seleccione un fitxer

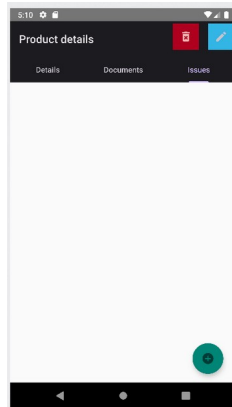


Desa el document i queda associat al producte

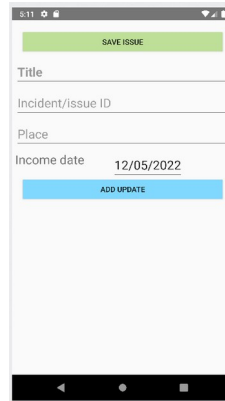
CU-004 – Afegir incidència



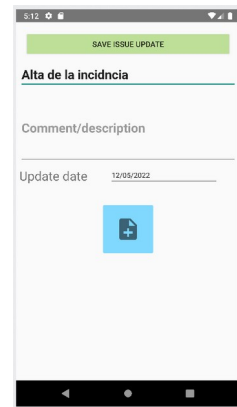
Seleccionar producte



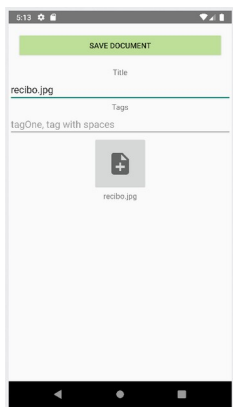
Accedeix a les incidències



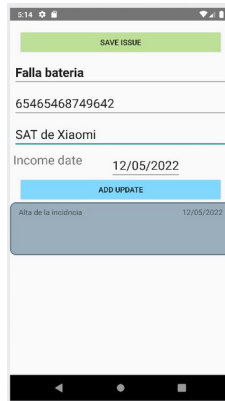
Prem el botó d'afegir



Afegeix un event "alta de la incidència"



Selecciona un fitxer adjunt

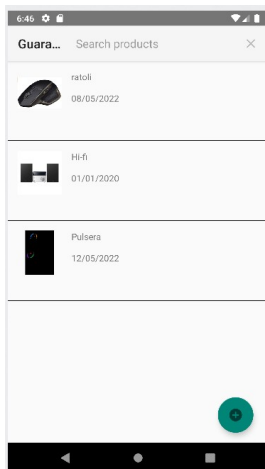


Omple la resta de formulari

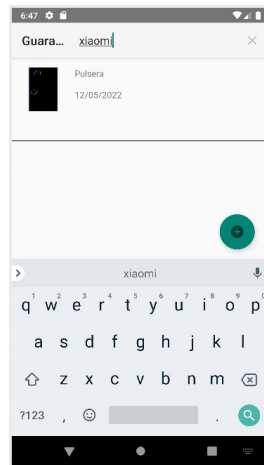


Apareix la nova incidència al llistat

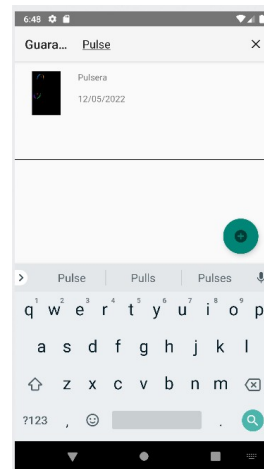
CU-005 – Buscar producte



Prem la lupa del llistat de productes

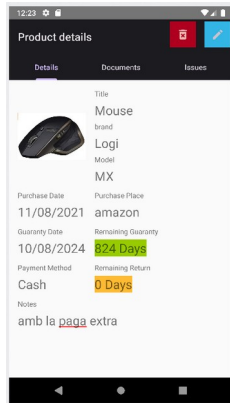


Busca per marca

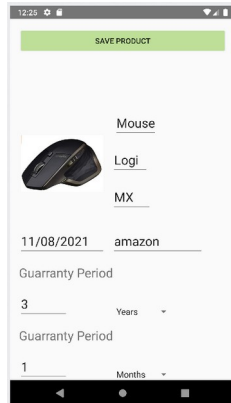


Busca per títol

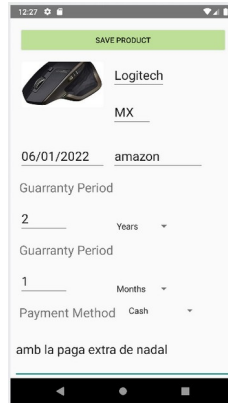
CU-006 – editar productes



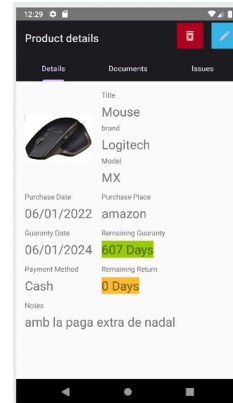
Selecciona un producte



Prem el botó d'editar producte



Modifica alguns camps, la data de compra i el període de garantia



Es desen les dades y es mostren actualitzades

3.3 - Planificació i seguiment

La planificació va començar de manera correcta, però de mica en mica, i d'agut a la inexperiència amb l'ecosistema Android i els seus estàndards varen sorgir un seguit de problemàtiques en l'aplicació dels primers casos d'ús i la implementació que estava seguint.

Això va provocar un retard en els primers casos de Producte que van fer canviar el plantejament que estava donant a l'arquitectura en general. La naturalesa dels objectes Realm i la seva integració per defecte en els Adapters usats pels RecyclerViews varen fer que abandonés el purisme de l'estratègia de mantenir les capes de domini com a objectes "nets" de la influència de l'arquitectura seleccionada en la infraestructura.

Això va suposar un retard en els següents casos d'ús i, al final, a la integració amb el núvol de Dropbox no ha estat possible de dur a terme en el temps que estava estipulat. Aquesta, era una de les funcionalitats que, tot i no afectar el funcionament de l'app, sí que poden suposar una pèrdua de funcionalitat de cara a l'usuari. La pèrdua d'aquesta funcionalitat suposa que els documents només s'emmagatzemen en el dispositiu on s'han enregistrat i queden fora de les dades que se sincronitzen entre dispositius.

4.4. Conclusions

- Lliçons apreses del treball?

La DCU és clau en la definició d'aplicacions, ja que ens aporta la possibilitat de dissenyar l'aplicació amb el punt de vista de l'usuari, i això fa que la usabilitat i les funcionalitats estiguin més adaptades.

He après tecnologies noves molt interessants, com Realm, Mongo o Kotlin que segurament continui investigant. Però aquest aprenentatge també ha suposat un esforç extra en el desenvolupament que ha repercutit negativament en les funcionalitats que s'han acabat entregant. Si la selecció del conjunt tecnològic hagués estat Java i FireBase, hauria facilitat el desenvolupament tot i no suportar les millores quant a disponibilitat que he considerat a l'apartat 3.1 Implementació i casos d'ús.

- Una reflexió crítica sobre l'assoliment dels objectius plantejats al començament.

En general els requisits del treball s'han assolit satisfactòriament. Els cinc requisits funcionals de l'aplicació OF-001/5 queden implementats completament.

Dels quatre objectius no funcionals, només l'ONF-003 "mantenir una còpia de les dades al núvol" ha quedat parcialment acabat. Si bé tota la informació se sincronitza perfectament en temps real, els documents no es desen al núvol i per tant només queden registrats en el dispositiu en el qual s'han definit.

- Una anàlisi crítica del seguiment de la planificació i metodologia al llarg del producte

La metodologia elegida per elaborar el treball ha estat Kanban el qual desfragmenta els treballs en petites tasques per tal de poder ser executades i entregades afegint funcionalitats que aportin valor al conjunt de l'aplicació. Aquesta metodologia no té una planificació prevista en el calendari.

El meu rerefons professional com a desenvolupador de *backend* m'ha fet intentar aportar al desenvolupament d'Android una infraestructura a la qual estic acostumat. La meva inexperiència en el desenvolupament d'aplicacions Android m'ha fet pensar que era la millor solució. Però, tot i no ser una mala opció, la manca

d'experiència ha jugat a la contra i ha endarrerit més del pensat l'elaboració dels primers lliurables del treball.

La metodologia Kanban no ha ajudat a detectar el desviament de recursos que estava patint al principi i s'hi han endarrerit tasques de desenvolupament. Fet que ha ocasionat el no poder entregar com a finalitzada una part important com és la de la sincronització de les dades al núvol de Dropbox.

Potser un canvi de metodologia a una de tipus Scrum on sí que es presenta un seguit de funcionalitats cada dues o tres setmanes hagués fet que es detectés més aviat el desviament de recursos i haver pogut redirigir els esforços abans.

- Les línies de treball futur que no s'han pogut explorar en aquest treball i han quedat pendents.

El següent pas ha de ser el de sincronitzar els documents al núvol per tal de complir els Objectius generals del projecte.

Tot seguit, es podrien anar implementant un seguit de millores pel que fa a:

- La gestió d'incidències:
 - Eliminar incidències i els documents relacionats
 - Editar i eliminar actualitzacions d'incidències
 - Visualitzar els documents des de l'actualització
- Afegir categories i filtratge en el llistat de productes.
- Producte:
 - Afegir algunes dades extres als productes.
 - Estimar la possibilitat de crear camps personalitzats

5. Glossari

Adapters: Component del framework d'Android que s'encarrega de gestionar un llistat de dades del repositori i adaptar-los perquè siguin consumits per la vista.

Android: Sistema operatiu per a dispositius mòbils basat en Linux

Application (capa/layer): Capa de l'arquitectura hexagonal on s'implementen les lògiques dels casos d'ús.

Arquitectura hexagonal: Arquitectura també anomenada ports and adapters, que manté en el centre els objectes de domini i els manté separats de la resta de capes, d'aplicació i infraestructura.

codi font: El codi font són tots aquells documents que es fan servir per poder produir l'aplicació. El codi font està en un format intel·ligible i és el que després de compilar produeix l'aplicació.

Clean architecture: Nom que es dona a les arquitectures que cuiden que l'aplicació no tingui dependències amb altres mòduls o frameworks

DaggerHilt: Llibreria que facilita la injecció de dependències en una aplicació Android.

Domain (capa/layer): Capa on s'ubiquen els objectes de la lògica de negoci. Aquests objectes normalment són els que tenen la lògica de les seves pròpies dades i els esdeveniments que provoquen.

Garantia: Cobertura que dona l'establiment o productor d'un producte pel qual es responsabilitza dels possibles problemes que tingui el producte.

FireBase: Solució que ofereix Google pel desenvolupament d'aplicacions. Ofereix la persistència de dades amb una base de dades Realtime, autorització, emmagatzemant ...

incidències: Qualsevol reclamació o procés establert per solucionar alguna problemàtica relacionada amb un producte.

Infrastructure (capa/layer): Capa de l'arquitectura hexagonal que conté les solucions de persistència i connexió amb altres serveis.

justificant de compra: tiquet, factura o rebut que justifica l'adquisició del producte.

Kanban: És una metodologia de treball de les anomenades "agile". Consisteix a desglossar el treball en tasques més petites que s'agrupen en una pila de tasques pendents anomenada *backlog*. La gestió de cada tasca passa per diferents estats fins que queda en un estat finalitzat. Un recurs no pot assignar-se més d'un nombre definit de tasques per estar treballant simultàniament. Aquesta metodologia és adequada per detectar colls d'ampolla en el procés productiu.

MVC: Arquitectura de desenvolupament Model-Vista-Controlador

MVP: Arquitectura de desenvolupament Model-Vista-Presentació

MVVM: Arquitectura de desenvolupament Model-View-ViewModel

Núvol: Servei d'emmagatzematge i serveis que estan sempre disponibles a través d'una connexió Online.

Offline first: Model de sincronització que prioritza l'emmagatzematge local.

ohMyForm: aplicació de codi obert de formularis en línia que permet ser allotjada de manera personal.

Online First: Model de sincronització que prioritza la connexió a internet.

paper tèrmic: paper usat per imprimir rebuts que no usa tinta sinó que reacciona a la temperatura.

PartitionKey: Clau que utilitza Realm per tal de filtrar quins elements cal retornar d'una col·lecció. Aquesta clau és independent de la consulta que es realitza, es podria considerar que la clau defineix el propietari d'un element.

període de garantia: temps durant el qual el producte està cobert per la garantia.

Període de devolució: temps durant el qual un producte es pot bescanviar o retornar.

PlayStore: La botiga d'aplicacions per a Android més coneguda gestionada per Google.

Realm: Solució d'ajuda al desenvolupament que ofereix MongoDB, alternativa a Firebase.

Reclamació: En el context d'aquesta memòria es fa servir.

RecyclerViews: Element de la interfície d'usuari d'Android que ajuda en la gestió de llistats, permetent només renderitzar els elements necessaris i ajudant a reduir els recursos i optimitzant l'eficiència.

Sincronització: Acció que es realitza per mantenir les dades de dues fonts amb els mateixos elements.

6. Bibliografia

- «Clean Coder Blog». Consulta 26 març 2022. <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>.
- Cockburn, Alistair. «Hexagonal Architecture». *Alistair Cockburn* (blog). Consulta 26 març 2022. <https://alistair.cockburn.us/hexagonal-architecture/>.
- MongoDB Developer Community Forums. «Email Confirmation Script for User Authentication via Email Address - MongoDB Realm», 6 gener 2021. <https://www.mongodb.com/community/forums/community/forums/t/email-confirmation-script-for-user-authentication-via-email-address/13905>.
- MongoDB Developer Community Forums. «Firebase vs Realm - MongoDB Realm», 3 octubre 2020. <https://www.mongodb.com/community/forums/community/forums/t/firebase-vs-realm/9929>.
- «Hilt». Consulta 27 març 2022. <https://dagger.dev/hilt/>.
- Izertis, por. «Componentes de arquitectura de Android, de MVC a MVVM». Solid GEAR, 20 juny 2019. <https://ahorasomos.izertis.com/solidgear/componentes-de-arquitectura-de-android-de-mvc-a-mvvm/>.
- «OhMyForm | OhMyForm is the best open source form solution for the web». Consulta 12 març 2022. <https://ohmyform.com/>.
- «PhotoTicket.App». Consulta 26 febrer 2022. <https://www.phototicket.app/>.
- «Warranty Manager - Apps on Google Play». Consulta 26 febrer 2022. https://play.google.com/store/apps/details?id=io.epix.warranty_manager&hl=en_US&gl=US.
- «Warranty Sured - Your Warranty Manager - Apps on Google Play». Consulta 26 febrer 2022. https://play.google.com/store/apps/details?id=com.jncwebs.warrantysured&hl=en_US&gl=US.

7. Annexos

7.1 Preparació entorn de desenvolupament

El programari seleccionat per desenvolupar el projecte és AndroidStudio, ja que estic familiaritzat amb l'IDE IntelliJ que faig servir habitualment, i crec que és l'opció que, avui dia, ofereix més facilitats pel desenvolupament d'aplicacions android.

Despres de la instal·lació, s'ha creat una aplicació de mostra amb una activitat simple per defecte per a android 8,0 o superior i Kotlin com a llenguatge de programació. Les captures Figura 18 i Figura 19 mostren l'aplicació a l'entorn de desenvolupament i la seva execució.

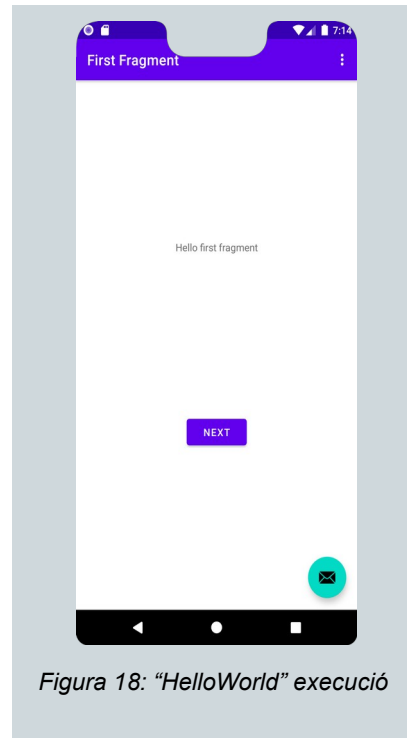


Figura 18: "HelloWorld" execució

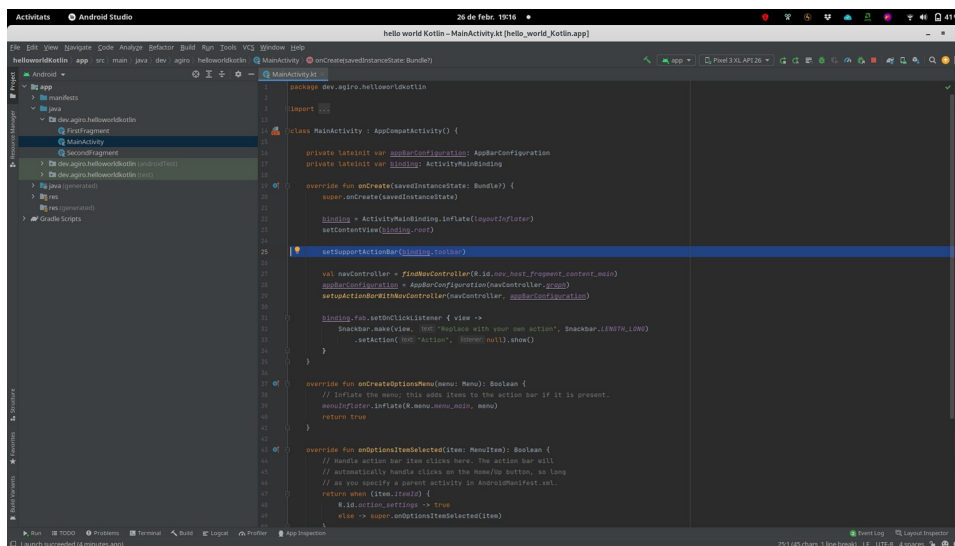


Figura 19: "HelloWorld" Entorn de desenvolupament

7.2 Preparació de l'aplicació al núvol de Realm.io/MongoDB

Configurar l'aplicatiu

Mitjançant gradle creem una variable amb l'id de la nostra app

```
buildTypes {
    def appId = "guarantit" // Replace with proper Application ID
    debug {
        buildConfigField "String", "MONGODB_REALM_APP_ID", "\${appId}"
    }
    release {
        buildConfigField "String", "MONGODB_REALM_APP_ID", "\${appId}"
        minifyEnabled false
        signingConfig signingConfigs.debug
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}
```

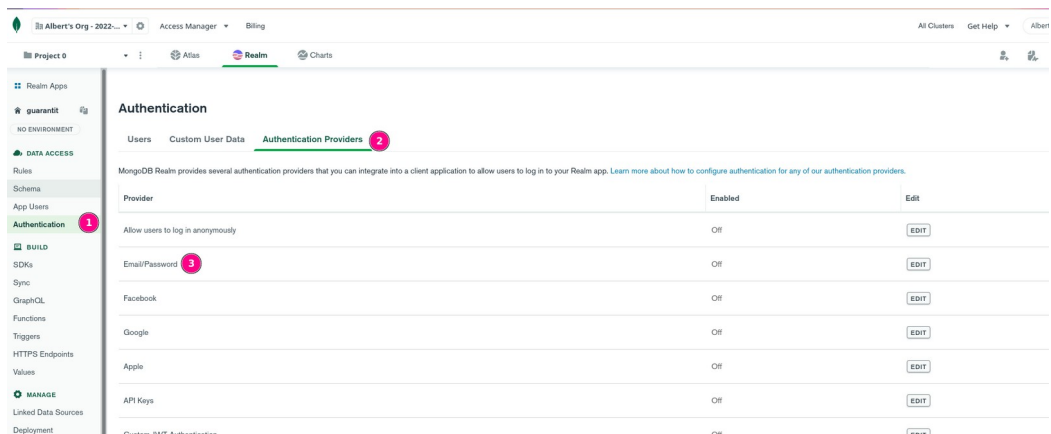
En el moment d'instanciar l'app de Realm es passa com a paràmetre de configuració.

```
@HiltAndroidApp
class GuarantitApplication:Application() {

    override fun onCreate() {
        super.onCreate()
        Realm.init(applicationContext);
        guarantitApp = App(
            AppConfiguration.Builder(BuildConfig.MONGODB_REALM_APP_ID)
                .defaultSyncErrorHandler { session, error →
                    Log.e(TAG(), "onCreate: Could not create Realm App", error)
                }
                .build()
        )
    }
}
```

Definir el tipus d'autenticació

En el primer estadi de l'aplicació, només es contempla la necessitat de registrar-se mitjançant correu electrònic. Per fer-ho anirem a l'apartat de "Data Acces" → "Authentication" → "Authentication providers" i seleccionem la opció "email/password"



Per tal de ser més segurs envers l'usuari es demanarà un correu d'autenticació i la possibilitat de redefinir el mot de pas.

Per poder fer-ho necessitarem un parell d'artefactes una pàgina per a poder rebre les confirmacions i una funció de "reset" del mot de pas. S'ha seguit el procediment trobat el fòrum de mongoDB.¹⁰

Pàgina de confirmació la desarem en el mateix realme.io a la secció de *Hosting*. La següent imatge es mostra la pàgina que s'ha usat on simplement es recull el *token* i es confirma l'usuari.

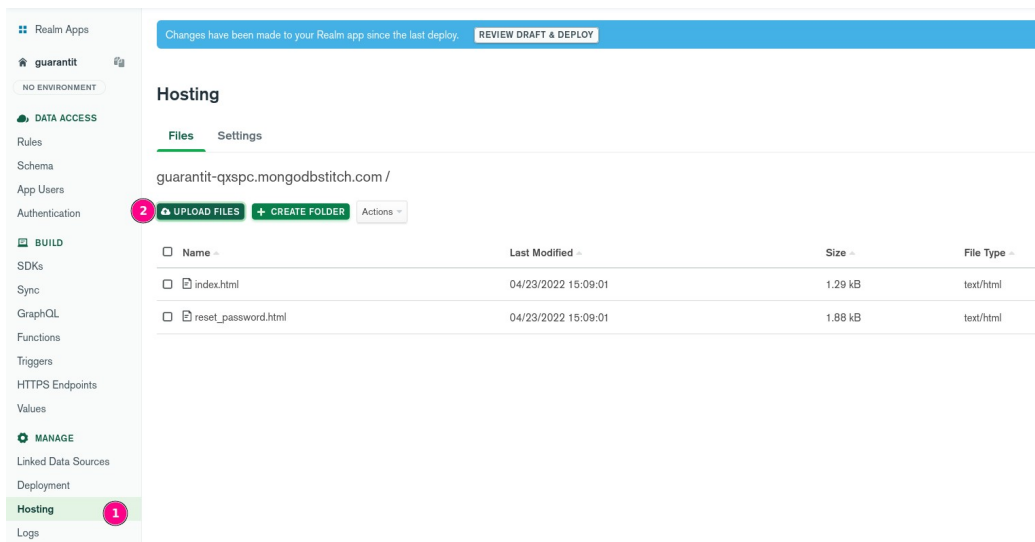
```

1 <html>
2 <head>
3 <script src="https://unpkg.com/realm-web@1.2.0/dist/bundle.iife.js"></script>
4 <script>
5 const APP_ID = "guarant-it-";
6 const app = new Realm.App({ id: APP_ID});
7 //Grab Tokens
8 const params = new URLSearchParams(window.location.search);
9 const token = params.get('token');
10 const tokenId = params.get('tokenId');
11 //Confirm client
12 app.emailPasswordAuth
13   .confirmUser(token, tokenId)
14   .then(() => displayResult('success'))
15   .catch(err => displayResult('error', err))
16 //Display Message depending on result
17 function displayResult(result, err) {
18   const message = document.getElementById("message");
19   if (result === "success") {
20     message.innerText = "Your E-mail address has been verified.\n\n You may close this page. Thank you.";
21   }
22   else if (result === "error") {
23     message.innerText = "Unable to register this user. Please try again to register." + err;
24   }
25 }
26 </script>
27 <title>E-mail Confirmation</title>
28 <style>
29   h1 {text-align: center;}
30   h3 {text-align: center;}
31   h5 {text-align: center; background-color: whitesmoke;}
32   p {text-align: center;}
33   div {text-align: center;}
34 </style>
35 </head>
36 <body>
37   <h1>Guarant-it</h1>
38   <h3>Thanks for choosing our app!</h3>
39   <h5 id = "message"></h5>
40 </body>
41 </html>
42

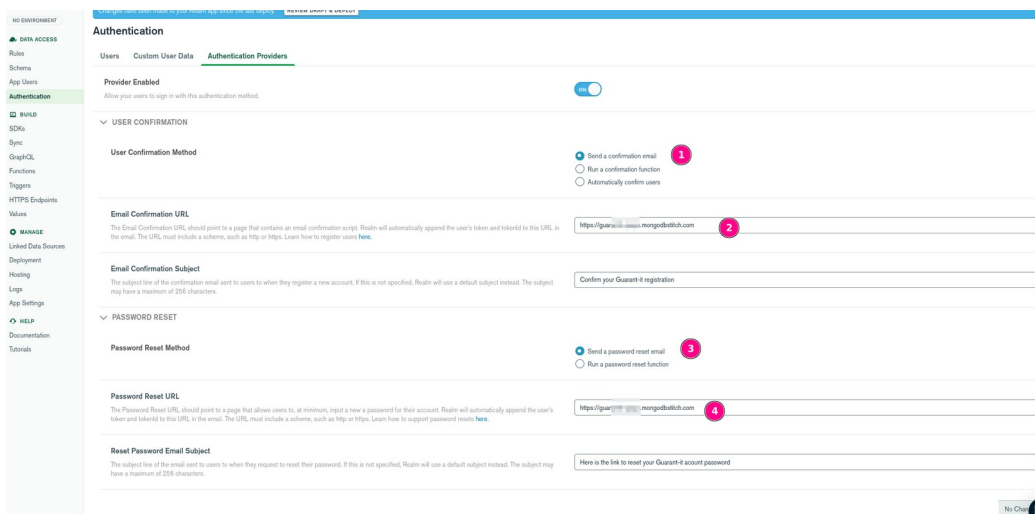
```

I la carreguem al servidor

¹⁰«Email Confirmation Script for User Authentication via Email Address - MongoDB Realm», MongoDB Developer Community Forums, 6 gener 2021, <https://www.mongodb.com/community/forums/community/forums/t/email-confirmation-script-for-user-authentication-via-email-address/13905>.



I el formulari el definim amb correus de confirmació i restabliment de mot de pas amb les pàgines que s'han pujat.



Un cop definits els components del registre, el següent és definir la sincronització del servidor Realm amb la base de dades Atlas. Per fer-ho, s'ha optat pel model "Development Mode" que permet la redefinició del model de dades dinàmicament segons les peticions del model.

Ara que està muntat el sistema de persistència de Realm i associat a la nostra app mitjançant l'identificador en el gradle, Quan l'aplicació iniciï, se sincronitzarà i també ho farà l'esquema que tinguem definit mitjançant els objectes RealmObject, ja que tenim el mode "Developer" seleccionat.

Quan l'aplicatiu estigui llest per anar a producció serà necessari mantenir els esquemes versionats i actualitzar-los manualment segons es vagin modificant.

- Project 0
- Atlas
- Realm
- Charts
- Realms Apps
- Guarant-it_app
- NO ENVIRONMENT
- DATA ACCESS
- Rules
- Schema
- App Users
- Authentication
- BUILD
- SDKs
- Sync
- GraphQL
- Functions
- Triggers
- HTTPS Endpoints
- Values
- MANAGE
- Linked Data Sources
- Deployment
- Hosting
- Logs
- App Settings
- HELP
- Documentation
- Tutorials

Development Mode is ON. Define data models in your client app using Realm SDKs. When ready, turn Dev Mode OFF and view your JSON Schema. **TURN DEV MODE OFF**

Sync has stopped between Atlas and Realm. Fix this issue by attempting to restart sync for Atlas. **RESTART ATLAS SYNC**

Sync ENABLED Latest Sync Event: 05/04/2022 23:04:02 Last Cluster Event Processed: -- Lag: 0 sec Pause Sync Terminate Sync

To adjust most sync details while Sync is enabled, you must terminate Sync first, which will lead to a client reset. With Sync running, you can only edit: (1) Developer Mode and (2) Permissions

Select Development Details

Sync Type
Learn more about the differences between [Partition-based Sync](#) and [Flexible Sync](#).

Partition Based Flexible [PREVIEW](#)

Partition-based Sync allows you to choose a single field across all collections to divide Atlas data into partitions based on the field's value.

Development Mode
Development Mode (Dev Mode) allows you to define and test data models in your client application before you are ready to go into production.

Dev Mode disables drafts (i.e. changes made in your Realm app deploy immediately).

Save Changes

Note: When Sync is enabled, Realm will create a copy of all data from collections with a defined JSON schema in your Atlas cluster.

Define a Database Name
When adding new tables in your mobile app, a new collection will be created in MongoDB. Select the database where you want new collections to be created in.

Choose a Partition Key
Partition keys allow you to split data from your MongoDB collections into Realm for simpler access within your application and definition of permissions. The field you choose as your partition key must exist in the JSON schema of any collection you wish to sync. Learn more about choosing your partition key.

Define Permissions
Select a template or manually define which application users have Read and/or Write access.

This will auto-fill your permissions. You can freely edit on top of a template's JSON.

Select a template

Read

```
1 {
2   - "%partition": "%user.id"
3 }
```

Write

```
1 {
2   - "%partition": "%user.id"
3 }
```

Advanced Configuration (optional)