

Incidències per a la ciutadania

Juan Cifo Servera

Màster Universitari en Enginyeria Informàtica
Desenvolupament d'Aplicacions sobre Dispositius Mòbils

**Jordi Ceballos Villach, Jordi Almirall López
Robert Clarisó Viladrosa**

Gener 2019



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

FITXA DEL TREBALL FINAL

Títol del treball:	Incidències per a la ciutadania
Nom de l'autor:	Juan Cifo Servera
Nom del consultor/a:	Jordi Ceballos Villach Jordi Almirall López
Nom del PRA:	Robert Clarisó Viladrosa
Data de lliurament:	01/2019
Titulació o programa:	Màster Universitari en Enginyeria Informàtica
Àrea del Treball Final:	Desenvolupament d'Aplicacions sobre Dispositius Mòbils
Idioma del treball:	<i>Català</i>
Paraules clau	Incidències ciutadania, <i>Smartphones, Smartcities</i>
Resum del Treball (màxim 250 paraules):	
<p>Avui en dia els ajuntaments fan servir la tecnologia per a realitzar multitud de funcions. Per una banda la gestió administrativa habitual d'un ajuntament i, per una altra banda, com a una nova eina a algun departament.</p> <p>En aquest treball s'ha fet una aplicació mòbil utilitzant eines de programació híbrides a fi d'estudiar les possibilitats dels dispositius mòbils com a eina de comunicació d'incidències dels ciutadans cap als ajuntaments i preparar un prototip que serveixi de base per a futurs projectes i noves funcionalitats.</p>	
Abstract (in English, 250 words or less):	
<p>Nowadays, town councils use technology to perform many tasks. On the one hand, the usual administrative management of the town council, and on the other hand, technology is used as a tool in some department.</p> <p>In this project a mobile application has been made using hybrid programming tools to study the possibilities of mobile devices as an issue reporter tool from citizens to town councils and also to prepare a prototype that will serve as a basis for future projects and new features.</p>	

Índex

1. Introducció	6
1.1. Context i justificació del treball	6
1.2. Objectius del treball	8
1.3. Enfocament i mètode seguit	10
1.4. Planificació del treball	10
1.5. Planificació actualitzada	13
1.6. Breu sumari de productes obtinguts	14
1.7. Breu descripció dels altres capítols de la memòria	14
2. Anàlisi, disseny i arquitectura	15
2.1. Disseny centrat en l'usuari	15
2.1.1. Usuaris i contextos d'ús	16
2.1.1.1. Enquesta	17
2.1.1.2. Entrevistes informals	21
2.1.1.3. Perfils d'usuari	21
2.1.2. Disseny conceptual	22
2.1.2.1. Escenaris d'ús	22
2.1.2.2. Flux d'interacció	26
2.1.3. Prototipatge	27
2.1.3.1. Sketches	28
2.1.3.2. Prototip horitzontal d'alta fidelitat	29
2.1.4. Avaluació	30
2.1.4.1. Qüestionari pre-test	30
2.1.4.2. Escenari i tasques a realitzar	31
2.1.4.3. Qüestionari post-test	31
2.2. Disseny tècnic	31
2.2.1. Casos d'ús	31
2.2.1.1. Funcionament intern	32
2.2.1.2. Decisions de disseny	33
2.2.2. Arquitectura	33
2.2.2.1. API	33
2.2.2.2. Aplicació mòbil	34
3. Implementació	36
3.1. Aplicació mòbil híbrida	36
3.1.1. Frameworks per a desenvolupar aplicacions híbrides	36
3.1.2. Quasar Framework	37
3.1.2.1. Vue	37
3.1.2.2. Extensible	38
3.1.2.3. Totes les plataformes d'un sol cop	38

3.1.2.4. Molts de components responsive i eficients	38
3.1.2.5. Les millors pràctiques de programació per defecte	38
3.1.2.6. Suport d'escriptura de dreta a esquerra	38
3.1.2.7. Dos temes	38
3.1.2.8. Migració progressiva del codi existent	38
3.1.2.9. Quasar CLI	38
3.1.2.10. Desenvolupament ràpid	39
3.1.2.11. Gran comunitat de desenvolupadors	39
3.1.2.12. Compatibilitat amb moltes plataformes	39
3.1.2.13. Componentes traduïbles	39
3.1.2.14. Bona documentació	39
3.1.3. Cordova	39
3.1.4. Firebase	39
3.1.5. Google Maps	40
3.1.6. Llibreries extres	40
3.2. API i autenticació	40
3.2.1. Laravel	41
3.2.2. OAuth2 Bearer Token	41
3.3. Detall del desenvolupament de l'aplicació mòbil	42
3.3.1. Projecte	42
3.3.2. Versions compatibles	42
3.3.3. Llenguatges i utilitats	42
3.3.4. iOS	42
3.3.5. Disseny / Plantilles	43
3.3.6. Emmagatzament	43
3.3.7. Pantalles i rutes	44
3.3.7.1. Pantalla índex	45
3.3.7.2. Pantalla botons	46
3.3.7.3. Pantalla missatge	46
3.3.7.4. Pantalla registre	46
3.3.7.5. Pantalla categories	46
3.3.7.6. Pantalla mapa	46
3.3.7.7. Pantalla imatge	47
3.3.7.8. Pantalla comentari	47
3.3.7.9. Pantalla resum	47
3.3.7.10. Pantalla històric	47
3.3.7.11. Pantalla detall d'incidència	48
3.4. Pantalles de l'aplicació	49
3.4.1. Pantalles bàsiques	49
3.4.2. Pantalles nova incidència	50
3.4.3. Pantalles de consulta d'incidències	51
3.5. Proves	52

3.5.1. Proves manuals	52
3.5.2. Proves de l'API	53
3.5.2.1. Obtenir les dades inicials de l'API	53
3.5.2.2. Crear una incidència	53
4. Conclusions	54
5. Treball futur	55
6. Bibliografia	56
7. Annexos	57
7.1. Annex 1 - Hello world	57
7.2. Annex 2 - API - Dades inicials de funcionament de l'APP	59
7.3. Annex 3 - Error de desplegament	60
7.4. Annex 4 - Configuració de l'aplicació mòbil	61
7.5. Annex 5 - Manual d'usuari	62
7.6. Annex 6 - API - Compilar i executar	63
7.6.1. Requeriments	63
7.6.2. Posta a punt	63
7.7. Annex 7 - APP - Compilar i executar	64
7.7.1. Requeriments	64
7.7.2. Posta a punt	64

1. Introducció

1.1. Context i justificació del treball

Els ajuntaments d'avui en dia fan servir la tecnologia per a realitzar multitud de funcions, tant de caire clàssic, com la gestió administrativa habitual d'un ajuntament, com també a departaments de noves tecnologies.

No és estrany que un ajuntament, o una empresa amb un servei concessionat, faci servir un telèfon intel·ligent per a la realització de tasques comunes, com per exemple, missatgeria entre el personal d'una brigada, seguiment GPS per a la realització de feines de neteja viària¹ i un llarg etcètera.

No obstant això, la comunicació directa entre un ciutadà i l'ajuntament no sempre és tan fàcil. Quan un ciutadà vol comunicar un desperfecte, una incidència o un problema a un ajuntament, sobretot a pobles petits, sovint ho fan de forma presencial, telefònica o via una xarxa social².

Aleshores, la gestió de les incidències que afecten a la ciutadania pot esdevenir caòtica, de difícil seguiment, complicada de comunicar cap a una entitat que solucioni el problema, quasi impossible de fer-ne un ús estadístic i molt difícil d'aconseguir una retroalimentació per part de la ciutadania una vegada la incidència ha estat gestionada.

A més a més, l'ús de telèfons intel·ligents per part de la ciutadania cada vegada és major. Segons un estudi de Deloitte del 2017³, un 92% de la quota de mercat té un smartphone i també cada vegada és més habitual que els usuaris facin servir algun tipus d'aplicació per comunicar-se enlloc de la telefonia convencional.

Els telèfons mòbils intel·ligents tenen les característiques adients per a ser utilitzada com a eina per a la transmissió de les incidències, ja que:

1. Té GPS per tal d'ubicar la incidència en un punt específic i evitar incongruències a l'hora de posicionament
2. Té una càmera per tal d'enregistrar una fotografia, o inclús un vídeo, sobre la incidència
3. Té connexió a Internet en tot moment (si es té una targeta SIM) per la qual cosa la comunicació de la incidència pot ser immediata

És per tot això que utilitzant les noves tecnologies i creant una aplicació mòbil senzilla d'utilitzar s'aconseguiria apropar la ciutadania a l'ajuntament i s'aconseguiria una comunicació fluida, directa i retroalimentada.

Realitzant una cerca a la Play Store és possible trobar alguna aplicació que realitza una funció similar, però que no acaba d'encaixar bé, segons mostren les valoracions. A continuació s'indiquen les aplicacions més destacables de totes les que s'han analitzat a dia 10/11/2018:

- Bústia ciutadana - Barcelona - 3.7 estrelles (115 usuaris)
<https://play.google.com/store/apps/details?id=es.tempos21.bustia>
- Línea verde - Multiajuntament - 3.6 estrelles (854 usuaris)
<https://play.google.com/store/apps/details?id=lineaVerde.App>
- Vinaròs Incidencias - Vinaròs - 4.2 estrelles (26 usuaris)
<https://play.google.com/store/apps/details?id=com.sigogestion.vinaros&roid=com.sigogestion.vinaros>
- Incidencias urbanas Menorca - Menorca - 3.9 estrelles (18 usuaris)
<https://play.google.com/store/apps/details?id=es.silme.fotodenuncia>
- Ajuntament d'Artà - Artà - 5 estrelles (4 usuaris)
<https://play.google.com/store/apps/details?id=com.incidencias.bitgrup>

En primer lloc, veiem la poca participació de la ciutadania a l'hora d'avaluar l'aplicació, la qual cosa pot indicar la poca participació ciutadana degut al desconeixement de l'aplicació. Si és així, des de l'ajuntament caldria fer una promoció de l'aplicació.

En segon lloc veiem que les puntuacions (en estrelles) no són massa elevades. A mode resum, després d'analitzar totes les aplicacions es pot concloure que la insatisfacció dels usuaris roman en els següents punts, per número de queixes:

1. Tipus d'incidències insuficients. Els usuaris es queixen que falten tipus d'incidències o que les que hi ha no s'ajusten a les seves necessitats. També se sol·licita, com a mínim, un tipus d'incidència que sigui "Altres" o "Varis".
2. Aplicació incompleta, falta de funcionalitats. Els usuaris demanen tenir un històric, saber si la seva incidència s'ha llegida, rebre notificacions dels canvis d'estat, etcètera.
3. Poca participació de l'ajuntament. El cas més habitual de frustració ha estat que des de l'ajuntament mai no s'ha contestat a la incidència, ni s'ha gestionat.
4. Errors d'aplicació. L'aplicació es penja o es tanca inesperadament. No es poden recuperar les incidències de l'usuari

En definitiva, per una banda, malgrat existeixin solucions en el mercat que podrien cobrir les necessitats d'aquest projecte, no tenen prou qualitat per poder considerar-se útils. Per altra banda, el cas de Línea Verde es tracta d'una aplicació multiajuntament, la qual cosa perd la personalització que un ajuntament li podria donar, amb els seus colors habituals, logo i distribució.

És per tot això que una vegada aquest projecte hagi finalitzat, immediatament se n'hauria d'obrir un altre per assegurar-ne la qualitat amb la participació constant de l'ajuntament. No té sentit fer un projecte per a apropar la ciutadania a l'ajuntament sense l'opinió i experiència de cada ajuntament.

1.2. Objectius del treball

L'objectiu principal del projecte és oferir un fil directe de comunicació d'incidències entre la ciutadania i l'ajuntament mitjançant una aplicació mòbil de fàcil ús. L'aplicació ha de ser capaç d'anar guiant l'usuari a través d'una sèrie d'etapes per tal de realitzar una incidència. Així mateix, un ciutadà ha de ser capaç de comprovar l'estat de les incidències que ha enviat i poder mantenir una comunicació amb l'ajuntament fins que la incidència se solucioni.

L'abast del projecte conté el desenvolupament inicial de l'aplicació mòbil o prototip i una API senzilla capaç d'enregistrar i servir les incidències. Possibles projectes futurs per a ampliar les funcionalitats d'aquest podrien ser:

1. Desenvolupament d'un *backoffice* senzill, en què l'ajuntament pugui tractar les incidències, donar d'alta nous tipus d'incidències, etcètera
2. Inclusió de funcionalitats extres a l'aplicació d'incidències, com un apartat de notícies. Esdevenir una aplicació per a l'ajuntament i no sols una aplicació d'incidències
3. Desenvolupament d'un *backoffice* complet, en què l'ajuntament pogués redirigir les incidències cap a concessions encarregades de solucionar-les
4. Redissenyar l'aplicació amb la participació de l'ajuntament i ajustar-se a les necessitats de cada un d'ells

Essent més específics, els objectius del projecte són:

1. Estudiar les necessitats de la ciutadania i de l'ajuntament per establir una comunicació directa i fluida
2. Establir les bases per a la creació de productes derivats a partir de la generació d'incidències, com un *backoffice*
3. Creació d'una aplicació que sigui capaç d'enregistrar una incidència en què es guardi una tipus d'incidència, una ubicació, una o més fotografies i un comentari. Les fotografies i el comentari són opcionals
4. L'aplicació ha de notificar qualsevol canvi que s'hagi produït a les incidències
5. L'usuari ha de poder consultar l'estat de les seves incidències
6. Tant l'usuari com l'ajuntament han de poder intercanviar en relació a l'incidència
7. L'ús de l'aplicació ha de poder ser anònim. No s'ha d'enregistrar cap informació que pugui identificar l'usuari si no ho introdueix voluntàriament

Per a complir amb els objectius del projecte es decideix desenvolupar una aplicació híbrida, ja que no es requereix la potència directa de la programació nativa i així el temps de desenvolupament es redueix considerablement. A més a més, cal desenvolupar una API capaç d'enregistrar les incidències i poder-les servir més tard. Finalment, també caldrà utilitzar algun servei de notificaciones PUSH com Firebase de Google.

Els requeriments funcionals de l'aplicació són:

1. L'aplicació descarrega la tipologia d'incidències de l'API
2. L'aplicació permet generar una incidència
 - 2.1. L'aplicació mostra un llistat de tipus d'incidències
 - 2.2. L'aplicació mostra un mapa en què se situa la incidència. També es permet marcar una posició diferent a la actual
 - 2.3. L'aplicació permet fer vàries fotografies, de forma opcional
 - 2.4. L'aplicació permet introduir un comentari, de forma opcional
 - 2.5. Es mostra un resum de la incidència abans d'enviar-la
3. L'aplicació permet consultar les incidències generades
 - 3.1. Es mostra un llistat de les incidències que ha generat l'usuari, mostrant la data de creació i l'estat en què es troba
 - 3.2. Pitjant sobre una incidència se n'obri el detall
 - 3.3. En el detall de la incidència es mostra la mateixa informació com s'ha creat: tipologia, ubicació sobre un mapa, fotografies i comentari
 - 3.4. En el detall de la incidència es permet intercanviar missatges amb l'ajuntament mentre la incidència estigui activa
4. Missatgeria
 - 4.1. L'usuari pot enviar un nou missatge a partir d'una incidència, adjuntant una fotografia i un comentari
 - 4.2. Els missatges estan ordenats per data ascendent i s'identifica qui ha enviat el missatge (si l'usuari o l'ajuntament)
 - 4.3. L'aplicació notifica automàticament a l'usuari quan hi ha un nou missatge
5. API
 - 5.1. L'API permet descarregar amb una sola crida la tipologia d'incidències
 - 5.2. L'API permet descarregar el llistat d'incidències amb el seu detall
 - 5.3. L'API notificarà a l'aplicació qualsevol canvi que es faci o si s'introdueix un nou missatge

Per altra banda, els requeriments no funcionals de l'aplicació són els següents:

1. Característiques i usabilitat
 - 1.1. L'aplicació identificarà l'ajuntament, mostrant el logo i el nom
 - 1.2. L'aplicació guarda en memòria la darrera descàrrega de dades i s'actualitza quan s'obri
 - 1.3. S'integraran, com a mínim, dos idiomes d'ús: català, castellà
 - 1.4. L'aplicació ha d'avisar a l'usuari si s'està processant la informació
2. Disponibilitat i fiabilitat
 - 2.1. L'aplicació ha d'estar disponible tant per Android com per iPhone
 - 2.2. L'aplicació ha de mostrar missatges d'error entenedibles
 - 2.3. L'API ha d'estar disponible, com a mínim, 99.9% del temps
 - 2.4. L'API ha d'enregistrar els errors de les peticions de l'aplicació
3. Seguretat
 - 3.1. L'usuari és anònim exceptuant si ell mateix dona dades de caire personal a les fotografies o en el comentari

- 3.2. Només l'usuari que ha generat una incidència i les persones autoritzades a tractar la incidència han de poder descarregar la incidència

1.3. Enfocament i mètode seguit

Tal com s'ha comentat en els apartats anteriors, cap de les solucions existents al mercat no té una qualitat prou elevada per complir amb els objectius que s'han plantejat. A més a més, un dels objectius principals és poder establir una base per al desenvolupament de productes derivats que puguin accelerar el procés de resolució d'incidències. Per la qual cosa, es considera que la única manera d'aconseguir els objectius és desenvolupar una aplicació completament nova.

En aquest primer projecte, es considera oportú seguir un mètode de cascada enlloc de fer servir metodologies àgils. Primer de tot, i amb més pes, per tal com està plantejat el seguiment del projecte, que convida a seguir aquesta metodologia, i en segon lloc per a poder crear un prototip que pugui ser mostrat a un ajuntament com una solució viable i funcional.

Per tal que el projecte sigui un èxit, i sobretot per establir unes bases de cara a futures versions en què sí que participarà l'ajuntament, és necessari fer un prototip funcional, de durada curta, en què l'ajuntament no cal que participi des del principi. En aquests futurs projectes sí que seria adient fer servir metodologies àgils i que la participació de l'ajuntament sigui constant.

Així doncs, la planificació en cascada consisteix en una sèrie d'etapes ben definides, estructurades i comunes a projectes que fan servir aquestes metodologies: establiment de requeriments, disseny, implementació, verificació i manteniment.

Evidentment, i tal com s'ha comentat varies vegades, aquest projecte depèn completament de la participació de la ciutadania, que són els usuaris finals. És per això que és necessari fer servir les tècniques del disseny centrat en l'usuari, permetent, de forma iterativa, recapitular, revisar i millorar cada una de les etapes anteriors a fi d'obtenir un producte que sigui fàcil de fer servir i satisfactori.

1.4. Planificació del treball

Per a la realització del treball és necessari disposar d'una sèrie de recursos, tant de *software* com de *hardware*.

A nivell de *hardware*, serà necessari disposar, com a mínim, d'un telèfon intel·ligent per a poder realitzar les proves del producte i d'un ordinador per al desenvolupament de la solució i de la memòria. També seria necessari un Mac per tal de poder compilar i publicar la versió

per iPhone, però la publicació a botigues d'aplicacions està fora de l'abast d'aquest projecte i no s'utilitzarà. Així doncs, el *hardware* que s'utilitzarà per aquest projecte és:

Tipus	Model / Característiques	Dedicació al projecte
Ordinador	Intel i5 amb 8GB de RAM	7 mesos
Telèfon intel·ligent	Xiaomi Mi A1 (Android 8.1)	7 mesos

A nivell de *software* és necessari fer servei programes i eines per al desenvolupament de la solució. Per a compilar i publicar la versió per iPhone cal XCode i, per tant, MacOS; però no es considera un requisit en aquest projecte, sinó que es proposa per a un futur projecte en què l'aplicació es publiqui a les botigues d'aplicacions. No obstant això, es realitzaran proves amb un Mac a moments puntuals del projecte per assegurar-ne la compatibilitat.

Així doncs, el *software* utilitzat i amb l'alternativa gratuïta proposada en cas de que sigui de pagament és:

Software utilitzat	Alternativa gratuïta/lliure	Ús
Android Studio	-	SDK d'Android
Microsoft Windows 10	Ubuntu 18.04	Desenvolupament de la solució
yarn (i nodejs)	-	Instal·lació de dependències
PHPStorm	notepad++	IDE de desenvolupament
Granttproject	-	Elaboració del Grantt
Google Docs	LibreOffice	Elaboració de la memòria
Camtasia	?	Elaboració del vídeo de la memòria

La planificació inicial sense detallar és la següent:

Proposta de projecte	juny
Formació en Quasar i Vue	juliol - setembre
Anàlisi de requisits	finals de setembre - principis de octubre
Disseny i implementació	octubre - novembre
Primer prototip	principis de desembre
Proves i modificacions	desembre

Elaboració de la memòria	setembre-desembre
Elaboració de la presentació	mitjans desembre - principis de gener
Presentació	gener

Durant l'estiu, abans de l'inici del projecte, es decideix començar amb la formació amb el framework Quasar⁴ per tal d'agilitzar el procés de desenvolupament.

Des de finals de setembre fins a principis d'octubre s'analitzen els requisits de l'aplicació. A continuació s'entra a la fase de disseny i implementació de la solució que durarà un mes, fins a finals de novembre. A principis de desembre s'espera obtenir un primer prototip funcional de l'aplicació i durant el desembre fer les modificacions i proves necessàries per assegurar la qualitat del producte.

El procés d'elaboració de la memòria és constant durant tot el projecte, fent les modificacions necessàries mentre el projecte es desenvolupa.

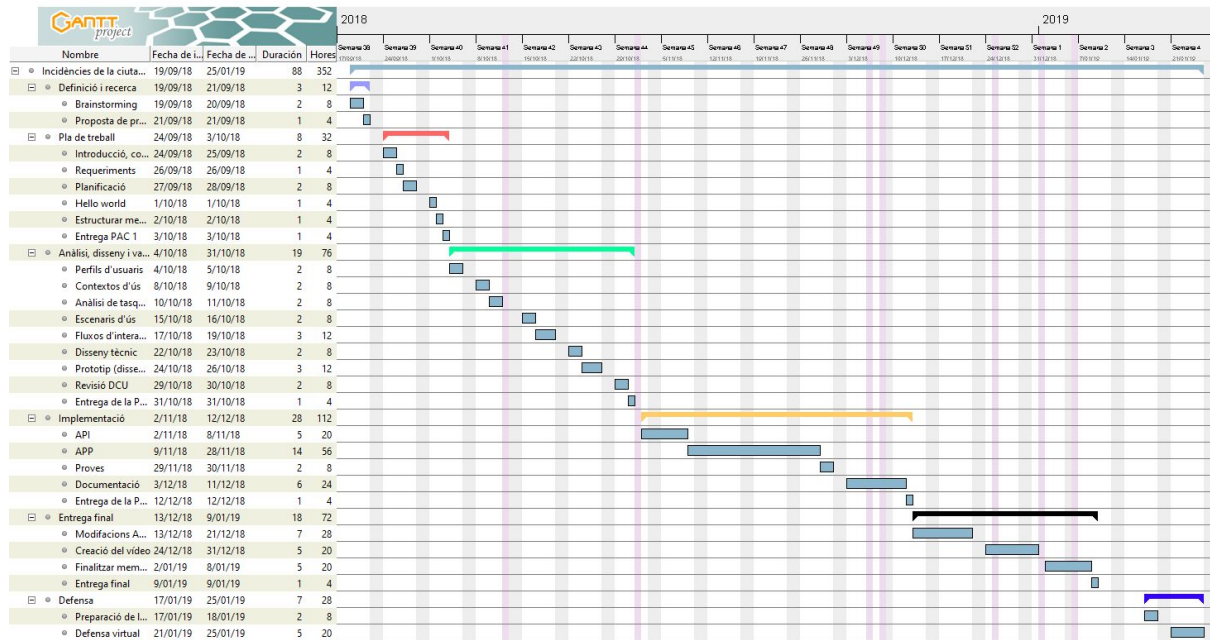
Finalment, s'espera dedicar el darrer mes (des de mitjans desembre fins a principis de gener) l'elaboració de la presentació a fi de poder presentar el projecte durant el gener.

No obstant això, el projecte requereix l'entrega parcial de continguts, per a la qual cosa és necessari establir certes fites arbitràries:

Fita	Data màxima
Capítol 1 (Introducció) de la memòria	03/10/2018
Document DCU i Disseny tècnic (pot servir per fer el capítol 2 de la memòria)	31/10/2018
Capítol 3 (Implementació) de la memòria i prototip de l'aplicació	12/12/2018
Memòria, aplicació final i vídeo del projecte	09/01/2019
Defensa del projecte	21/01/2019

Durant el projecte, es preveu dedicar al projecte 4 hores diàries els dies laborables i cap hora els dies festius o cap de setmana. Si un dia de la setmana no s'ha pogut treballar en el projecte, els dies festius o de cap de setmana es recuperaran les hores. No obstant això, és possible que quan s'apropin les entregues es faci una feina extra per a enllestir-les.

A partir d'aquesta informació i amb el detall de cada entrega parcial es pot generar un Gantt detallat amb les hores per a cada tasca:



Imatge 1: Diagrama de Gantt

Tal com s'ha comentat anteriorment, cal tenir en compte que a més a més de la planificació del projecte també s'ha fet una formació prèvia en la utilització de les eines que es faran servir, reduint considerablement el temps de desenvolupament.

1.5. Planificació actualitzada

A dia 11/12/2018, segons la planificació inicial s'hauria d'haver assolit:

1. APP quasi al complet (mancant canvis menors)
2. API quasi al complet (mancant canvis menors)
3. Memòria quasi completa

Ara bé, degut a factors no planificats (instal·lació de l'API a un servidor) i a endarreriments de desenvolupament (s'havia afegit *Bearer Authentication* i no s'havia previst des d'un principi) el projecte es trobava en el següent estat:

1. APP completada. Pendent de revisar i algun canvi menor (fitxers d'idioma)
2. API al 80%:
 - a. Quedava pendent assignar les incidències als usuaris que la generen a partir de la *Bearer Authentication*
 - b. Quedava pendent filtrar les incidències de l'usuari
 - c. Revisió completa i canvis menors
3. Memòria al 60%
 - a. Durant la fase d'implementació s'havia pensat re-estructurar, redefinir i revisar tots els capítols ja realitzats de la memòria
 - b. Mancances greus al capítol de *Proves*
 - c. Calia completar la informació del capítol de *Disseny tècnic - arquitectura*

- d. El capítol d'implementació havia de contenir més detalls d'implementació confusos, com la *Bearer Authentication*. A més, calia afegir annexos útils relacionats amb la implementació
- e. Revisió general de tota la memòria per millorar-ne la qualitat

En definitiva, el projecte estava endarrerit i es varen prendre les següents mesures:

1. La primera setmana del període *Entrega final* es duplicarien les hores de feina previstes per acabar d'implementar les aplicacions
2. La segona setmana del mateix període es duplicarien les hores de feina previstes per repassar, revisar i millorar els capítols de la memòria existent
3. La resta de planificació es mantenia igual

Finalment, gràcies a les mesures aplicades, s'ha pogut acabar el projecte tal com s'havia previst inicialment, a més d'afegir les funcionalitats addicionals detectades.

1.6. Breu sumari de productes obtinguts

Realitzant aquest projecte s'esperen obtenir els següents productes:

1. Aplicació mòbil híbrida executable tant a dispositius Android com a dispositius iPhone
2. La memòria del projecte, en què es detalla el contingut, l'abast i el recorregut del projecte
3. Manuals d'usuari, manuals d'instal·lació i altres documents accessoris
4. Presentació del projecte que inclou els trets més destacats del projecte
5. Vídeo de presentació en què es resumirà i explicarà el procés que ha seguit el projecte i els resultats obtinguts

1.7. Breu descripció dels altres capítols de la memòria

El contingut de la memòria està dividit en els següents capítols:

1. Introducció. Capítol que detalla els objectius, el context i la planificació de treball a seguir.
2. Anàlisi, disseny i avaluació. Capítol que detalla l'anàlisi de l'aplicació, les decisions de disseny i tècniques d'avaluació del producte.
3. Implementació. Capítol que detalla les eines utilitzades, els mètodes seguits, les proves plantejades, i tot allò que estigui relacionat amb la fase d'implementació de la solució.
4. Conclusions. Capítol que detalla l'experiència obtinguda, els problemes detectats amb les propostes de solucions i propostes de futur.
5. Bibliografia.
6. Treball futur. Capítol que es descriuen possibles projectes derivats d'aquest.
7. Annexos. Capítol que engloba tota aquella informació complementària del projecte, com manuals i exemples.

2. Anàlisi, disseny i arquitectura

Tal com s'ha exposat anteriorment, aquest projecte intenta descobrir les necessitats que té la ciutadania per comunicar les incidències del seu municipi amb la creació d'un prototip d'una aplicació que caldrà ser revisada i re-estudiada amb l'ajuda de l'ajuntament.

No obstant això, tal com es demana, s'aplicarà el disseny centrat en l'usuari, malgrat sigui massa prest per fer-ne un ús real i útil. De tota manera, aquest primer anàlisi enfocat a les necessitats dels potencials usuaris pot establir les bases del que s'ha de fer o no s'ha de fer a futurs projectes.

2.1. Disseny centrat en l'usuari

L'aplicació final que es vol construir en aquest projecte podrà ser utilitzada per una gran diversitat de persones. És per això que en aquest projecte es faran servir les tècniques del *disseny centrat en l'usuari* (DCU).

El DCU és una filosofia de disseny que té per objectiu la creació de productes o solucions que resolguin les necessitats dels usuaris finals, a més d'aconseguir la màxima satisfacció i la millor experiència d'ús amb el menor esforç per part de l'usuari.

Ara bé, en definitiva, en aquest projecte es farà servir com una sèrie de fases iteratives per aconseguir el millor disseny per a l'usuari final.

- Anàlisi

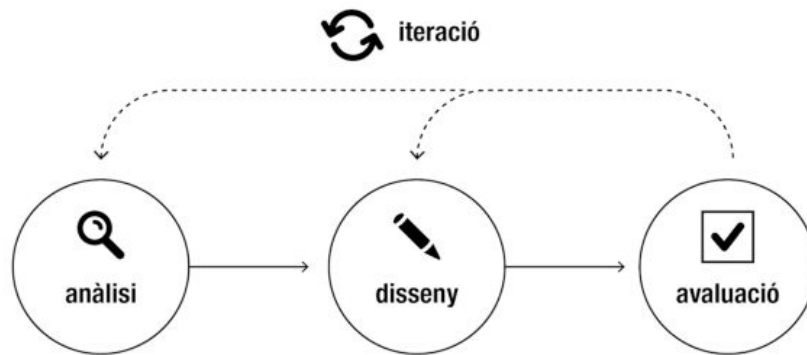
Procés en què es coneix l'usuari, s'estableixen els requisits per tal de fer el disseny.

- Disseny

Procés en què es desenvolupa el disseny conceptual: els perfils dels usuaris, les persones i els escenaris i es traspassa al disseny de maquetes i l'arquitectura de la informació: classificació de targetes i prototipatge.

- Avaluació

Procés en què s'avalua el disseny per millorar-lo o validar-lo amb tècniques com l'avaluació heurística o el recorregut cognitiu.



2.1.1. Usuaris i contextos d'ús

L'objectiu de la primera fase és conèixer les característiques dels usuaris, les seves necessitats i objectius. A més a més, de l'elaboració de contextos d'ús.

Existeixen varis mètodes d'investigació i anàlisi en el DCU en dispositius mòbils per tal d'obtenir informació referent a les necessitats, objectius i contextos dels usuaris, com la tècnica del *shadowing* o el mètode de diari; que són compatibles amb els mètodes habituals del DCU, com la investigació contextual, les entrevistes en profunditat, les dinàmiques de grup o les enquestes.

En primer lloc, el *shadowing*, o mètode de seguiment, és un mètode d'anàlisi en què l'investigador vigila i observa discretament els participants voluntaris mentre duen a terme les seves tasques habituals. Ara bé, aquest mètode només s'aplica a l'entorn en què tingui sentit analitzar les activitats dels participants, no durant tot el dia. En relació en aquest projecte, no es considera una tècnica útil per a la creació de perfils i es decideix no fer-la servir. Pot ser aquesta tècnica té sentit en projectes en què els participants formin part d'un col·lectiu reduït, o que la informació que aportin sigui molt valuosa, però no en una aplicació per a la generació d'incidències de la ciutadania.

En segon lloc, el mètode del diari consisteix en què els participants recopilin informació sobre el que es vol investigar. En relació en aquest projecte, es decideix no fer servir aquesta tècnica degut a que carregar de feina extra als participants per a un projecte que de desenvolupament d'una aplicació àgil i ràpida no té sentit. Aquest mètode només aportaria informació esbiaixada o molt adornada.

Així doncs, es considera més oportú fer servir tècniques més personals en què es podran conèixer les necessitats dels participants així com els motius que els impulsarien a notificar incidències.

2.1.1.1. Enquesta

El mètode d'enquestes és el menys invasiu, voluntari, imparcial i fàcil per a aconseguir informació sobre els usuaris finals. La part més important d'aquest mètode és el desenvolupament d'una enquesta breu però suficient per a obtenir la informació desitjada: amb massa preguntes esdevé avorrida, amb massa poques no s'aconsegueix la informació suficient i preguntes sense relació amb el que es vol descobrir no té sentit.

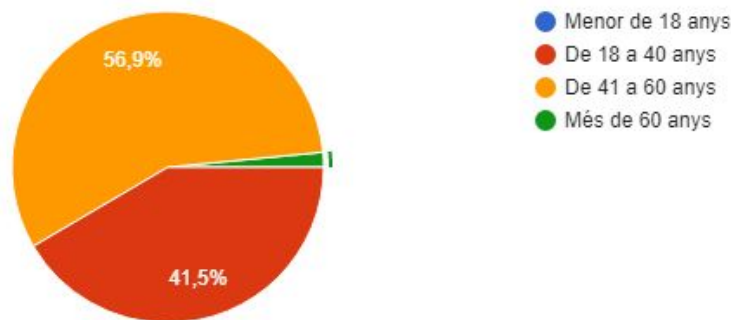
Així doncs, després de vàries modificacions, temptatives i reduccions, l'enquesta consta de només 6 preguntes breus i poc invasives per a què més gent la pugui finalitzar. El detall de l'enquest pot ser consultat a l'Annex 2 - Enquesta.

En total han participat 65 persones, la qual cosa, malgrat no es pugui considerar un estudi representatiu degut a la poca participació i essent necessari aplicar alguna normalització de les dades, serveix per detectar quines funcionalitats consideren útil, poc útil o imprescindible una petita part de la població.

En primer lloc, per contextualitzar les dades de l'enquesta, cal saber que la majoria de gent que ha participat es troba entre els 41 i els 60 anys, mentre que sols hi ha un participant que superi els 60 anys.

Quina franja d'edat tens?

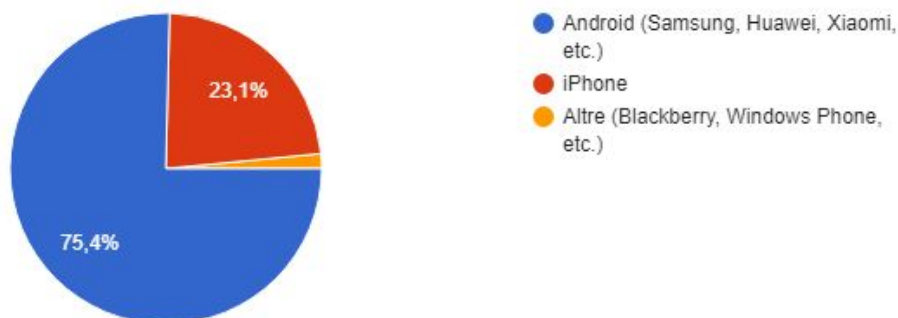
65 respuestas



En segon lloc, una quarta part de la mostra té un dispositiu amb Android, mentre que la resta, exceptuant la persona major de 60 anys, té un iPhone.

Quin tipus de mòbil tens?

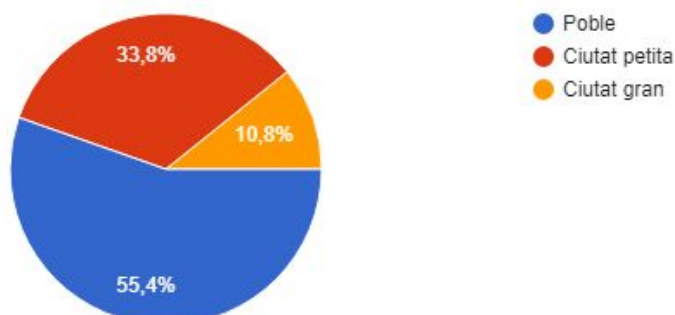
65 respuestas



Finalment, la majoria de la mostra viu a un poble, una tercera part a una ciutat petita i un deu per cent viu a una ciutat gran.

On vius?

65 respuestas



Així doncs, a partir de les característiques anteriors, la majoria de participants de l'enquesta són persones entre 41 i 60 anys, que viuen a pobles i fan servir telèfons amb Android.

A continuació, es demana de quina manera comuniquen les incidències actualment. Es pot veure que quasi la meitat de gent comunica les incidències només a familiars i amics, un 16.9% no ho comunica a ningú i un 15% de passada a algú de l'ajuntament o a un policia. A més a més, 3 persones ja fan servir una aplicació per fer comunicar les incidències però no han especificat quina. Per altra banda, el 84.6% faria servir un servei de notificació d'incidències anònim si l'ajuntament l'oferís.

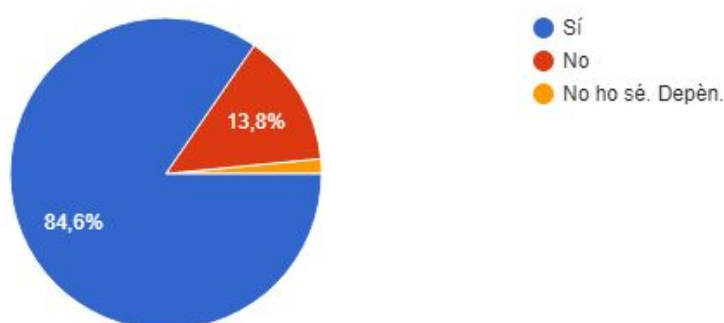
Si hi ha una incidència al teu municipi, ho comuniques?

65 respuestas



Si l'ajuntament oferís un servei anònim de notificació d'incidències, ho faries servir?

65 respuestas



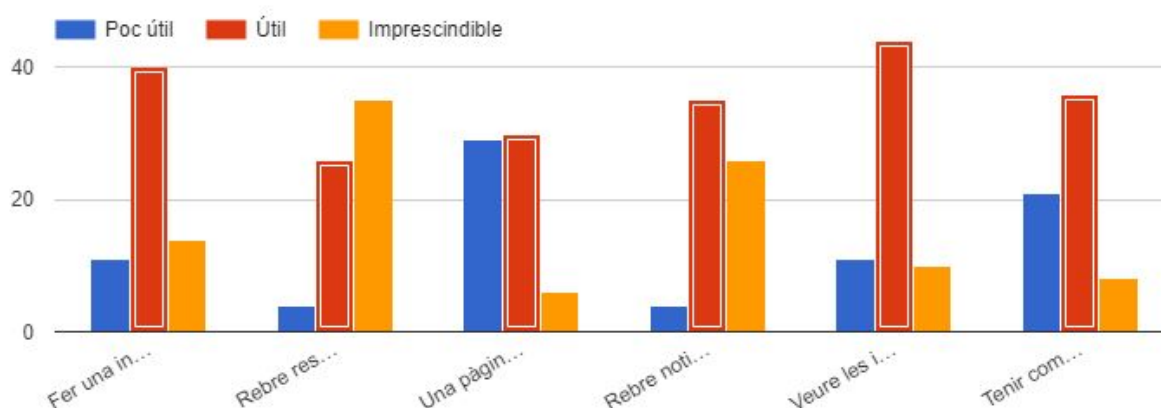
A partir dels resultats de les dues preguntes anteriors s'ha descobert, tot i la poca quantitat de gent que ha participat, que:

1. La gent vol un sistema per comunicar incidències a l'ajuntament i al mateix temps no les està comunicant o bé només les comunica als familiars
2. Malgrat ja existeixen eines o maneres que es fan servir per comunicar les incidències -telèfon, aplicacions, directament a l'ajuntament-, la gent vol un sistema per comunicar les incidències a l'ajuntament
3. Dues persones han confós el concepte d'incidència amb el concepte d'emergència, ja que han especificat que avisarien a l'112

Així doncs, amb els tres punts anteriors es confirma la importància de tenir en compte l'usuari en el disseny de l'aplicació i adaptar-la a les seves necessitats.

Finalment, s'ha demanat que especifiquessin si 6 funcionalitats eren útils, poc útils o imprescindibles a una aplicació per a la comunicació d'incidències cap a l'ajuntament.

Quines funcionalitats consideraries útils per a una aplicació de notificació d'incidències?



Funcionalitat	Poc útil	Útil	Imprescindible	Total
Fer una incidència de forma anònima	11	40	14	65
Rebre resposta per part de l'ajuntament	4	26	35	65
Una pàgina web amb "Les meves incidències"	29	30	6	65
Rebre notificacions vàries de l'ajuntament	4	35	26	65
Veure les incidències d'altra gent	11	44	10	65
Tenir compte d'usuari	21	36	8	65

En primer lloc, es confirma, una vegada més, que la opció més desitjada pels possibles usuaris és tenir resposta de l'ajuntament, motiu pel qual altres aplicacions tenen tan mala puntuació a les botigues d'aplicacions.

En segon lloc, una funcionalitat que considera molt útil o quasi imprescindible aquest conjunt de persones és fer servir l'aplicació per alguna cosa més que no només fer incidències, sinó també rebre notificacions vàries de l'ajuntament. En tercera instància, es considera molt útil fer incidències anònimes i veure les incidències que ha fet l'altra gent.

És doncs, a futurs projectes, que l'ajuntament hauria d'estudiar si és interessant que la resta de ciutadans vegin totes les incidències creades pels usuaris, com acte de transparència, o, en canvi, pot ser perjudicial de cara a la protecció de dades o per altres motius.

Finalment, el que els participants no veuen útil és afegir una pàgina web amb les incidències ni tampoc tenir un compte d'usuari. Sembla doncs, que l'ús seria esporàdic i instantani, sense interès en guardar un històric una vegada s'hagi solucionat la incidència.

2.1.1.2. Entrevistes informals

Abans de realitzar l'enquesta, es varen realitzar entrevistes informals a algunes persones que hi participarien.

L'objectiu de les entrevistes no era entrar en detall de cada funcionalitat, sinó conèixer de forma global quines necessitats, problemes i idees tenien els diferents grups en relació a la comunicació d'incidències del municipi.

Per una banda, amb sector més jove que ha participat (persones d'entre 25 i 35 anys) es va fer una entrevista grupal en què es va demanar com seria el sistema -no específicament una aplicació- ideal per a notificar les incidències que detecten al seu municipi.

Per altra banda, a una persona del sector d'entre 41 i 60 també se li va fer una entrevista informal consultant quin seria el sistema ideal per comunicar les incidències.

En definitiva, totes les persones varen coincidir en què la millor manera per notificar una incidència seria avisant directament a l'ajuntament a través d'una xarxa social o bé via missatgeria instantània. No obstant això, també varen preocupar-se que aquest fet no és anònim i que a l'hora de notificar una incidència només la comunicarien per aquestes vies si fos molt important.

2.1.1.3. Perfils d'usuari

A partir de la informació obtinguda es poden definir dos perfils d'usuari:

Usuari actiu o generador	
Característiques del perfil	<p>Perfil format per persones que generen incidències de forma més o menys freqüent.</p> <p>Les persones d'aquest perfil poden ser de qualsevol edat i amb un dispositiu Android o iPhone. Viuen a pobles o ciutats. Actualment ja comuniquen incidències per altres vies: comunicació directa amb l'ajuntament o via xarxes socials.</p> <p>Aquest conjunt d'usuaris tenen com objectiu notificar les incidències del seu municipi i voler ser notificats quan es tractin les incidències. Alguns faran incidències de forma anònima i d'altres voldrien donar el seu no.</p>
Context d'ús	<p>Aquest conjunt d'usuaris farà servir l'aplicació en el mateix moment que detecti una incidència i la vulgui notificar a l'ajuntament.</p>
Tasques	<ul style="list-style-type: none">• Ubicar una incidència a un mapa• Seleccionar el tipus d'incidència

	<ul style="list-style-type: none"> • Escriure un text descrivint la incidència • Adjuntar alguna fotografia • Aportar més informació a una incidència ja existent • Tancar una incidència seva
Llistat de característiques	<ul style="list-style-type: none"> • Veure les seves incidències generades • Veure els comentaris que ha fet l'ajuntament • Saber l'estat en què es troben les seves incidències

Usuari passiu o visualitzador	
Característiques del perfil	<p>Perfil format per persones que només visualitzen les incidències d'altra gent.</p> <p>Les persones d'aquest perfil poden ser de qualsevol edat i amb un dispositiu Android o iPhone. Viuen a pobles o ciutats. Actualment no comuniquen incidències a l'ajuntament, només entre familiars i amics.</p> <p>Aquest conjunt d'usuaris tenen com objectiu veure les incidències del seu municipi</p>
Context d'ús	Aquest conjunt d'usuaris farà servir l'aplicació en qualsevol moment que vulguin saber les incidències obertes del seu municipi
Tasques	<ul style="list-style-type: none"> • Poder veure les incidències generades d'altra gent
Llistat de característiques	<ul style="list-style-type: none"> • Veure totes les incidències obertes del municipi

2.1.2. Disseny conceptual

A partir de la informació obtinguda anteriorment, es poden elaborar els escenaris d'ús.

2.1.2.1. Escenaris d'ús

E1 - Crear una incidència sense imatge en el moment en què es produeix	
Perfil	Usuari generador
Context	L'usuari, caminant de nit pel carrer, troba una farola fusa i no es veu res.
Objectiu	Notificar a l'ajuntament que una farola està fusa
Tasques	<ul style="list-style-type: none"> • Crear una incidència a la seva ubicació només fent servir text
Necessitats d'informació	Categoria d'incidència d'enllumenat públic

Funcionalitats	<ul style="list-style-type: none"> ● Incidència a la seva ubicació ● Poder escriure text
Tasques	<ul style="list-style-type: none"> ● Seleccionar el tipus d'incidència d'enllumenat públic ● Ubicar la incidència ● Escriure un text ● Enviar

E2 - Crear una incidència només amb imatge en el moment que ho detecta	
Perfil	Usuari generador
Context	L'usuari, caminant pel carrer, veu un contenidor al terra i tot el rebuig enmig del carrer
Objectiu	Notificar a l'ajuntament que un contenidor està tirat al carrer
Tasques	<ul style="list-style-type: none"> ● Crear una incidència a la seva ubicació només fent servir imatge
Necessitats d'informació	Categoria d'incidència de neteja viària
Funcionalitats	<ul style="list-style-type: none"> ● Incidència a la seva ubicació ● Poder adjuntar imatges
Tasques	<ul style="list-style-type: none"> ● Seleccionar el tipus d'incidència de neteja viària ● Ubicar la incidència ● Fer una foto ● Enviar

E3 - Crear una incidència detallada en el mateix lloc on es produeix	
Perfil	Usuari generador
Context	Un usuari passejant detecta que en tota una mançana no hi ha llum, i sap que és des de fa una setmana
Objectiu	Notificar a l'ajuntament que no hi ha llum en tota una mançana des de fa una setmana, fent servir imatges i text explicatiu
Tasques	<ul style="list-style-type: none"> ● Crear una incidència a la seva ubicació
Necessitats d'informació	Categoria d'incidència d'enllumenat públic
Funcionalitats	<ul style="list-style-type: none"> ● Incidència a la seva ubicació ● Poder escriure text ● Poder adjuntar imatges

Tasques	<ul style="list-style-type: none"> ● Seleccionar el tipus d'incidència de neteja viària ● Ubicar la incidència ● Escriure un text ● Fer una foto ● Enviar
---------	--

E4 - Registrar-se a l'aplicació	
Perfil	Usuari generador
Context	Un usuari vol registrar-se a l'aplicació perquè s'ha de canviar de mòbil i vol conservar les seves incidències
Objectiu	Crear un compte d'usuari
Tasques	<ul style="list-style-type: none"> ● Registrar-se
Necessitats d'informació	<ul style="list-style-type: none"> ● El seu correu electrònic ● La contrasenya que vulgui fer servir
Funcionalitats	<ul style="list-style-type: none"> ● Registre
Tasques	<ul style="list-style-type: none"> ● Obrir la finestra de registre ● Introduir el seu correu electrònic i una contrasenya

E5 - Iniciar sessió a l'aplicació	
Perfil	Usuari generador
Context	Un usuari vol iniciar sessió amb el seu compte d'usuari
Objectiu	Iniciar sessió
Tasques	<ul style="list-style-type: none"> ● Iniciar sessió
Necessitats d'informació	<ul style="list-style-type: none"> ● El seu correu electrònic ● La contrasenya que va fer servir
Funcionalitats	<ul style="list-style-type: none"> ● Iniciar sessió
Tasques	<ul style="list-style-type: none"> ● Obrir la finestra d'inici de sessió ● Introduir el seu correu electrònic i la seva contrasenya

E6 - Crear una incidència a una altra ubicació	
Perfil	Usuari generador

Context	Un usuari, en arribar a casa, vol notificar que hi ha un arbre caigut al parc on sol anar. Un amiga seva li ha enviat una foto amb l'arbre caigut
Objectiu	Notificar una incidència a una altra ubicació i fent servir imatges de la seva galeria
Tasques	<ul style="list-style-type: none"> • Crear una incidència des d'una altra ubicació • Adjuntar imatges de la seva galeria • Escriure un text
Necessitats d'informació	<ul style="list-style-type: none"> • On s'ha produït la incidència • Fotografies de la incidència
Funcionalitats	<ul style="list-style-type: none"> • Seleccionar una altra ubicació per fer una incidència • Adjuntar imatges de la galeria • Escriure un text
Tasques	<ul style="list-style-type: none"> • Seleccionar el tipus d'incidència de neteja viària • Ubicar la incidència a un altre lloc • Escriure un text • Adjuntar una foto de la galeria • Enviar

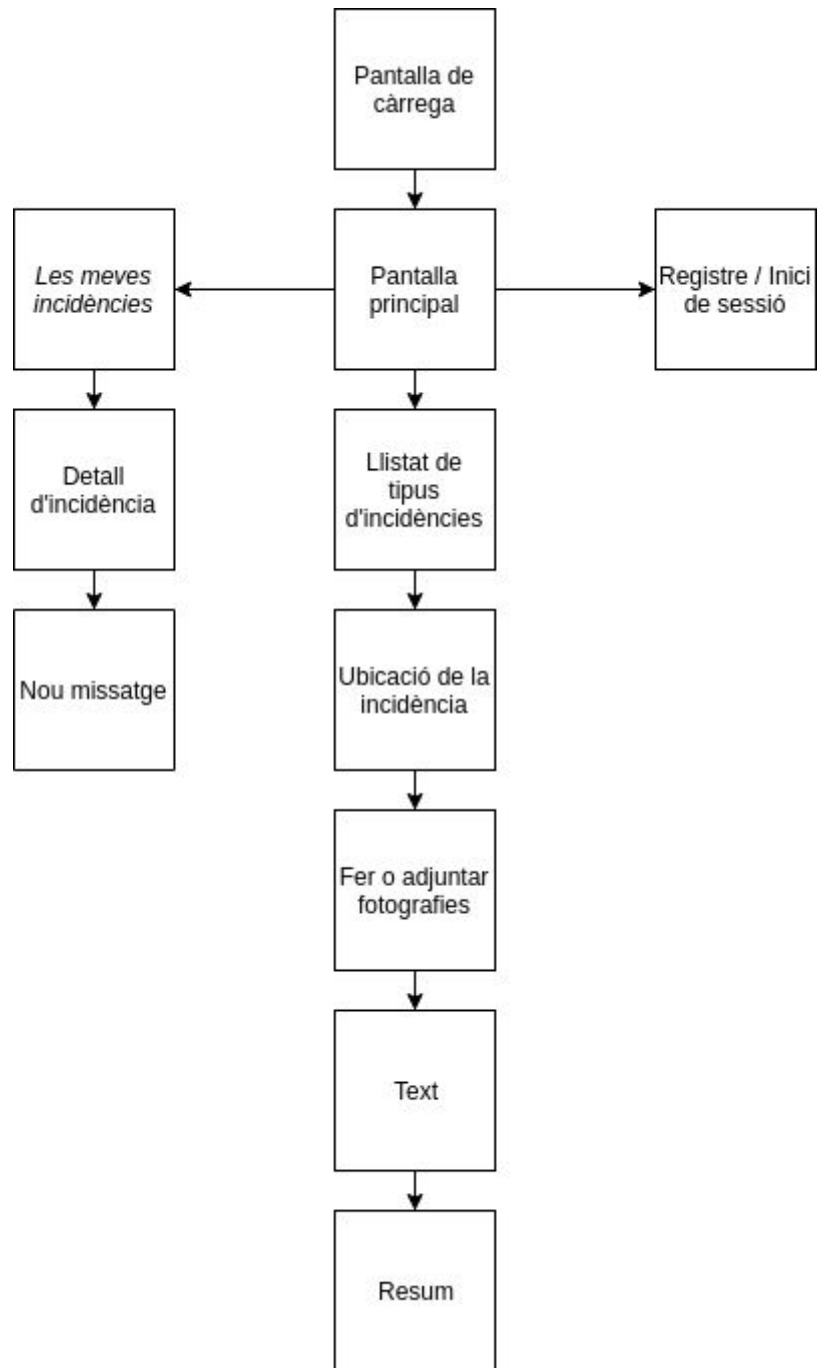
E7 - Rebre una notificació quan la incidència es tracti	
Perfil	Usuari generador
Context	Un usuari vol adonar-se'n quan l'ajuntament tracti la seva incidència
Objectiu	Conèixer quan la incidència s'ha resolt
Tasques	-
Necessitats d'informació	<ul style="list-style-type: none"> • Una incidència generada anteriorment
Funcionalitats	<ul style="list-style-type: none"> • Rebre notificacions PUSH
Tasques	<ul style="list-style-type: none"> • Esperar fins que arribi la notificació

E8 - Iniciar sessió a l'aplicació	
Perfil	Usuari generador
Context	Un usuari vol aportar més informació a una incidència que ha generat anteriorment
Objectiu	Escriure un text amb més informació sobre un problema notificat anteriorment

Tasques	<ul style="list-style-type: none"> • Veure les seves incidències • Escriure un text
Necessitats d'informació	<ul style="list-style-type: none"> • Incidència generada anteriorment
Funcionalitats	<ul style="list-style-type: none"> • Veure incidències generades per l'usuari • Escriure un text nou
Tasques	<ul style="list-style-type: none"> • Veure les incidències anteriors • Seleccionar la incidència a la qual es vol afegir més informació • Escriure un text addicional • Enviar

2.1.2.2. Flux d'interacció

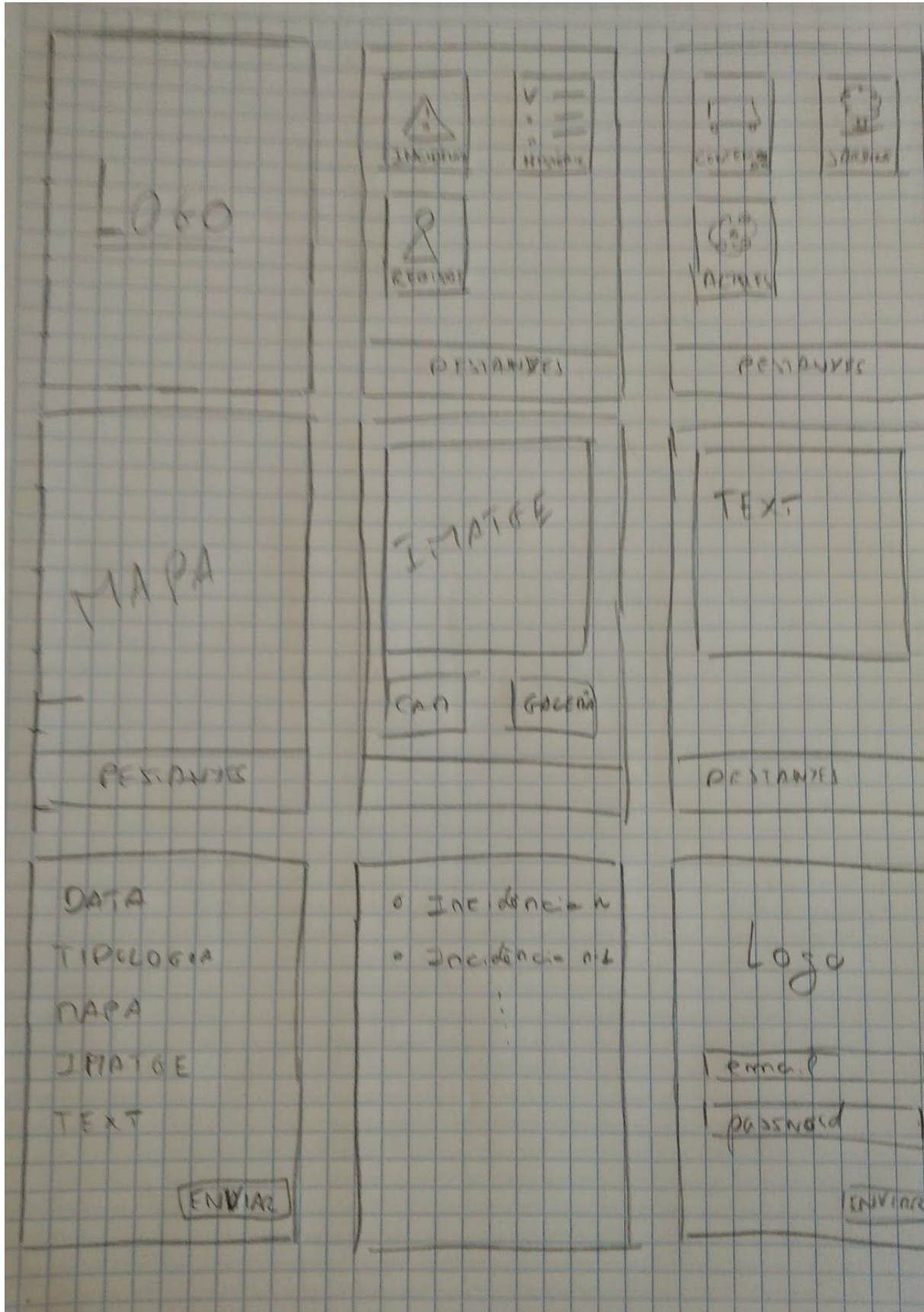
L'ús de l'aplicació ha de ser senzill i accessible per a qualsevol tipus d'usuaris, per això el flux ha de ser molt guiat. Una vegada l'usuari hagi generat una incidència l'aplicació torna a mostrar la pantalla principal.



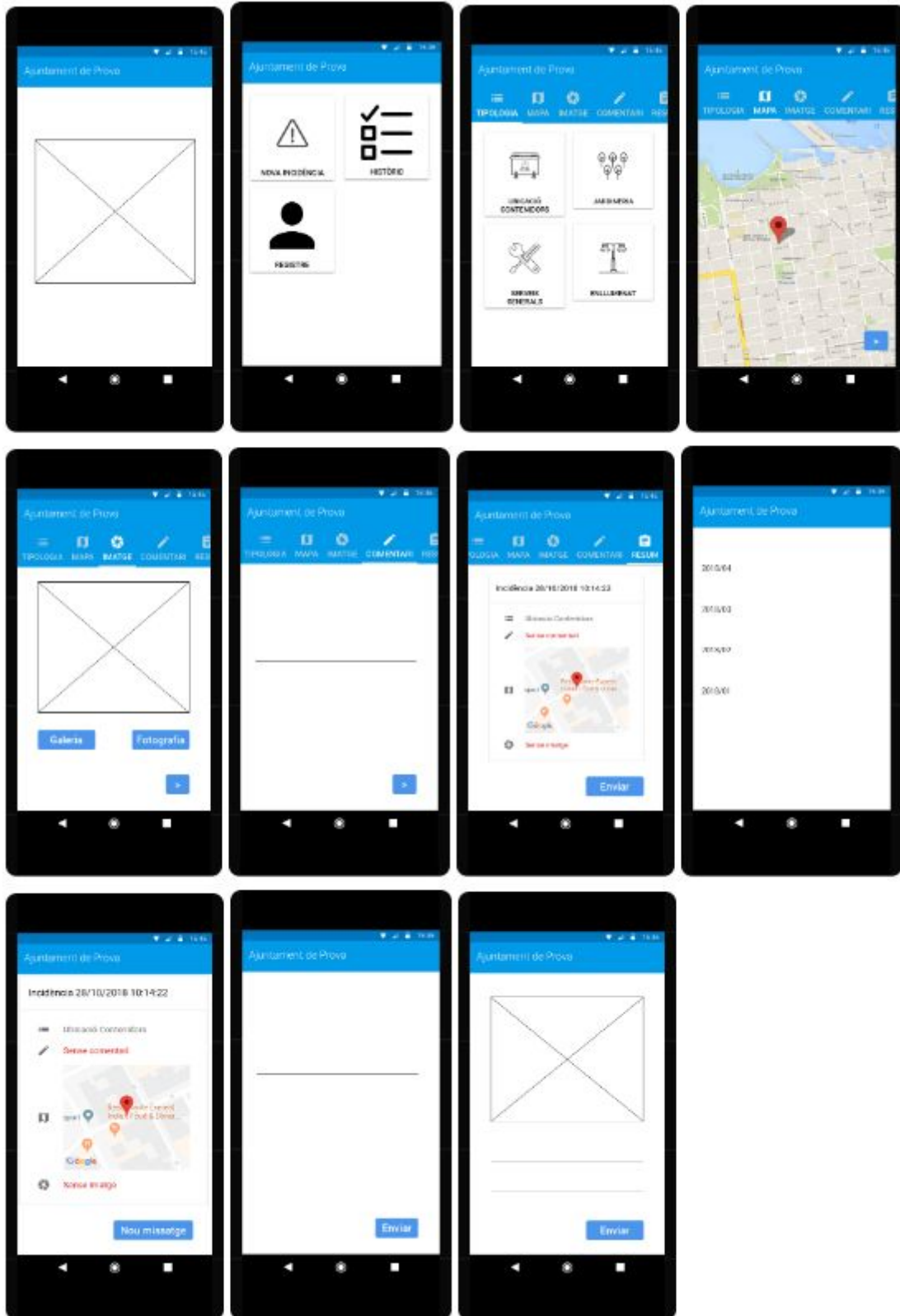
2.1.3. Prototipatge

A continuació es mostra el prototipatge d'aquesta primera versió de l'aplicació

2.1.3.1. Sketches



2.1.3.2. Prototip horitzontal d'alta fidelitat



A continuació s'explica breument de la funcionalitat de cada pantalla:

1. Pantalla de càrrega en què es connectarà a l'API per descarregar el llistat de tipologies i altra informació útil
2. Pantalla principal amb els botons de nova incidència, veure l'històric i registre. A futures versions l'aplicació podria obtenir el llistat de botons de l'API. Pitjant a un botó s'obri una altra pantalla
3. Pantalla amb llistat de tipus d'incidències. Pitjant sobre un tipus d'incidència es passa a la pròxima pantalla
4. Pantalla de mapa on es veu on se situa la incidència. Pitjant sobre un punt del mapa la incidència se situa a un altre lloc
5. Pantalla d'imatges amb dos botons per adjuntar la imatge des de la galeria o treure una fotografia. La part superior mostra totes les imatges dins d'una galeria.
6. Pantalla per introduir el text
7. Pantalla resum en què es mostra tota la informació que s'ha introduït a les etapes anteriors
8. Pantalla amb llistat de les incidències generades per l'usuari en ordre descendent
9. Pantalla de detall d'una incidència generada per l'usuari
10. Pantalla per afegir més informació a la incidència generada
11. Pantalla d'inici de sessió o de registre

2.1.4. Avaluació

Per a avaluar el prototip es realitzarien diverses tasques:

1. Filtre inicial de participants
2. Qüestionari pre-test
3. Tasques a realitzar
4. Qüestionari post-test

El primer filtre es faria segons el públic objectiu que es vol aconseguir, llavors en aquesta primera iteració quedarien excloses les persones que no facin servir un Andorid o un iPhone.

2.1.4.1. Qüestionari pre-test

Per aquest prototip, el qüestionari pre-test seria el següent. Evidentment, caldria replantejar-lo a la versió definitiva i per a cada iteració del DCU.

1. Quina edat tens?
2. Vius a un poble, a una ciutat petita (menys de 100.000 habitants) o a una ciutat gran?
3. Quina professió tens?
4. Sols portar el mòbil fora de casa?
5. Tens un Android o un iPhone?
6. Saps instal·lar una aplicació al mòbil?
7. Tens instal·lades moltes aplicacions al mòbil?
8. Si veus un desperfecte al teu municipi, a qui ho notifiqués? I com?

2.1.4.2. Escenari i tasques a realitzar

A continuació es podria presentar un context en què l'usuari hagi de realitzar una incidència a partir d'una sèrie de tasques:

Avisa a l'ajuntament que al carrer fa 2 setmanes que no es neteja al carrer de qui al davant. Pots anar-hi i realitzar la incidència in situ o bé fer-la des d'aquí.

L'escenari comporta les següents tasques:

1. Triar la tipologia d'incidències corresponent a neteja viària
2. Ubicar la incidència on correspongui (ja sigui des del mateix carrer o a distància)
3. Fer fotografies i/o un text explicatiu del que passa
4. Enviar la incidència a l'ajuntament

2.1.4.3. Qüestionari post-test

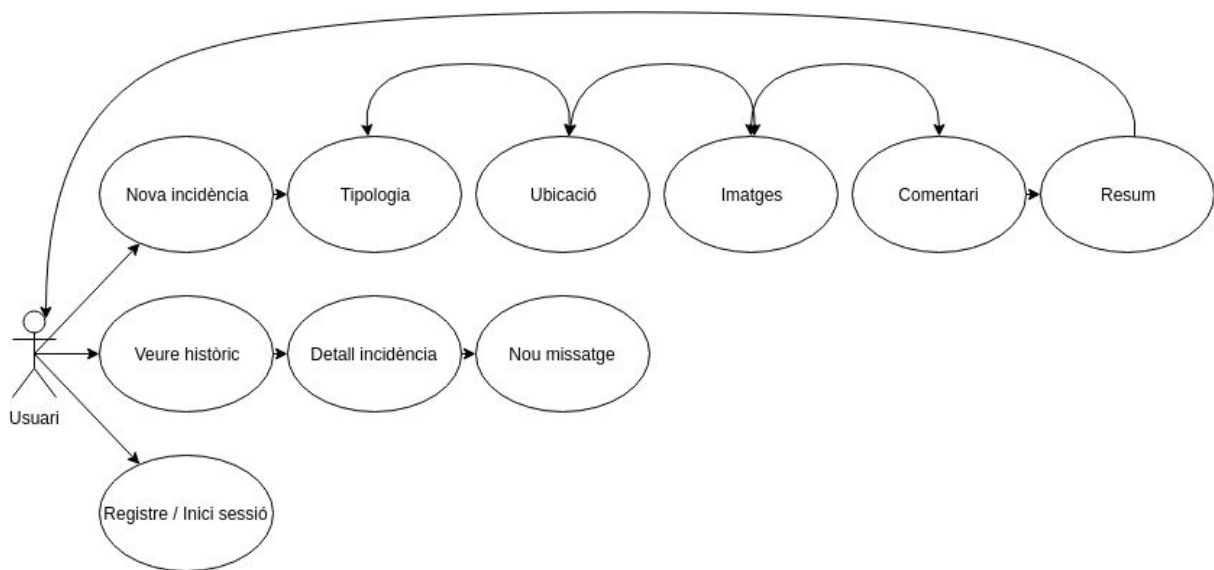
Finalment, es podria fer un qüestionari post-test a l'usuari per a què avaluï com ha anat la prova. La primera versió del qüestionari podria ser el següent. Cada pregunta podria ser avaluada d'1 a 5 en què 1 significaria "Totalment en desacord" i 5 "Totalment d'acord". A més a més, a cada pregunta es podria afegir una resposta explicativa.

1. Creus ha estat fàcil d'utilitzar?
2. Has sabut quines passes havies de fer per a realitzar la incidència?
3. Has pogut comunicar la incidència?
4. Has trobat en falta alguna funcionalitat?
5. Canviaries l'ordre en què es mostren les opcions (tipus, mapa, imatge i comentari)?
6. Faries servir aquesta aplicació per a realitzar incidències?

2.2. Disseny tècnic

2.2.1. Casos d'ús

Tal com s'ha comentat anteriorment, l'aplicació requereix d'una senzillesa extrema per a què pugui ser utilitzada amb èxit per a qualsevol tipus d'usuaris. És per això que totes les passes han de ser molt guiades i sense pèrdua. A continuació es mostra un diagrama UML amb el processus que pot tenir l'aplicació a partir de la pantalla principal.



Imatge 2: UML d'accions de l'aplicació

Aleshores, un usuari que vulgui notificar una incidència hauria de realitzar les següents passes:

1. Obrir l'aplicació
2. Pitjar sobre el botó de Nova incidència, identificat amb una imatge i un text
3. Triar una tipologia d'incidència
4. Mostrar l'ubicació en què es produeix la incidència o bé oferir canviar-la per una altra
5. Opcionalment, permetre fer o adjuntar una o varies fotografies
6. Opcionalment, permetre escriure un text
7. Veure un resum abans d'enviar-se la incidència per a què l'usuari confirmi que el que vol notificar és tal com ho exposa
8. Enviar

Així mateix, si un usuari vol veure una incidència que ja ha publicat haurà de:

1. Obrir l'aplicació
2. Pitjar sobre el botó d'Històric, identificat amb una imatge i un text
3. Triar la incidència que vulgui

A més a més, si vol afegir nova informació d'una incidència que ja ha publicat, a part de les accions anteriors haurà de:

4. Pitjar el botó de nou missatge
5. Escriure un text
6. Enviar

2.2.1.1. Funcionament intern

L'aplicació requereix Internet en cinc moments:

1. Obtenció de les dades inicials (tipologia d'incidències)
2. Enviament d'una incidència
3. Consultar l'històric i veure el detall d'una incidència
4. Enviar un text amb més informació per a una incidència
5. Registre o inici de sessió

Quan l'aplicació s'obri per primera vegada es connecta a l'API per obtenir les dades inicials. Si no aconseguís descarregar-les l'aplicació mostraria un missatge d'error i es tancaria.

En canvi, als altres casos (incloent les altres vegades que es vulgui obtenir les dades inicials) en què es necessita Internet, si en el moment de realitzar l'acció no es pot connectar a l'API l'aplicació mostraria un missatge d'error però no es tancaria: seria decisió de l'usuari si tancar l'aplicació o esperar a tenir Internet.

2.2.1.2. Decisions de disseny

Numeració de les incidències

Es decideix numerar les incidències en format ANY/Número de quatre dígit (per exemple, 2018/0001). D'aquesta manera és fàcil identificar de quin any són les incidències.

Botons de la pantalla inicial

A la versió d'aquest projecte, es proposa fer servir quatre botons a la pantalla principal: Nova incidència, Històric, Ajuda i Registre. Però a futures versions, un afegit interessant seria que des del *backoffice* es poguessin ordenar i establir els botons.

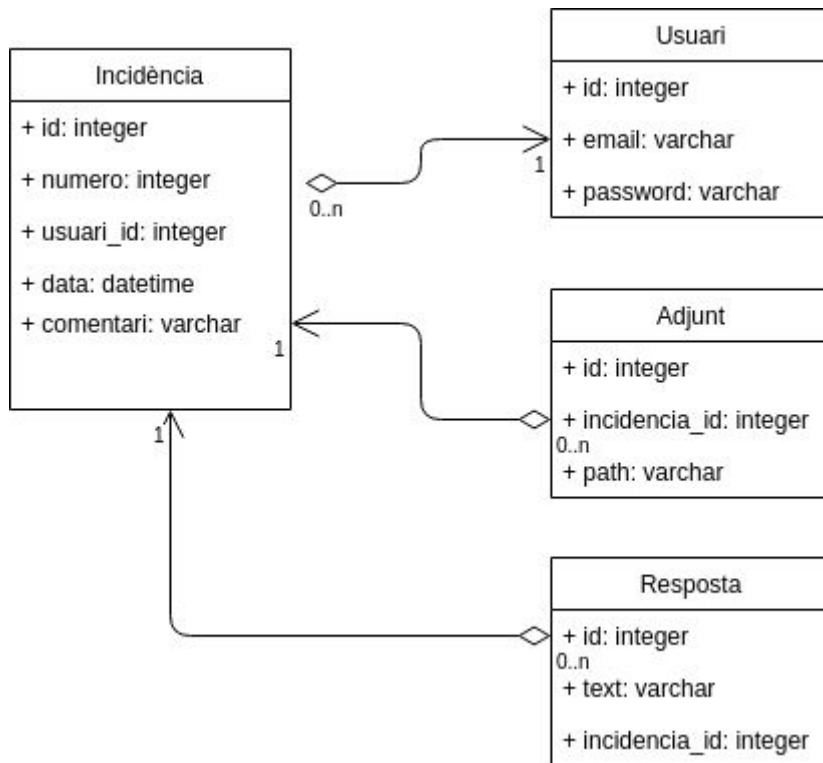
Categories d'incidències

Des de l'API s'obtenen totes les categories d'incidències que es mostren a la plataforma. És al *backoffice* on s'estableixen quines imatges, quins textos i quin ordre tenen els tipus d'incidències.

2.2.2. Arquitectura

2.2.2.1. API

L'API ha de guardar les incidències a la part del servidor. Malgrat aquesta part escapa de l'abast del projecte, es pot veure a grans trets com hauria de guardar-se la informació a la base de dades:



2.2.2.2. Aplicació mòbil

A l'aplicació mòbil es farà servir el LocalStorage per emmagatzemar les dades en format JSON. No existiran models de dades pròpiament dits ni una estructura de base de dades relacional.

En aquest cas, tan sol existirà un objecte Incidència que guardarà tota la informació que es vagi demanant:

- id (integer, generat a la part de l'API)
- codi (string, generat a la part de l'API amb el format any/número)
- categoria_id (enter)
- lat (double)
- lng (double)
- data (string)
- comentari (string)
- adjunts (array):
 - id (integer, generat a la part de l'API)
 - path (string, generat a la part de l'API)
- respostes (array)
 - id (integer, generat a la part de l'API)
 - text (string)

A més a més, els botons de la pantalla inicial, les categories d'incidència i els missatges no es guarden com entitats relacionals, sinó com un JSON al LocalStorage.

3. Implementació

La fase d'implementació ha consistit en desenvolupar el prototip utilitzant les tecnologies que s'han comentat anteriorment a més a més de moltes llibreries existents al mercat.

3.1. Aplicació mòbil híbrida

De manera molt resumida, una aplicació mòbil híbrida és una aplicació que combina elements natius i web.

Mentre que les aplicacions natives permeten una personalització absoluta, accés a tots els perifèrics i utilitzar totes les eines dels SDK de cada plataforma, també incrementen molt el temps de desenvolupament, ja que s'ha de desenvolupar l'aplicació per a cada plataforma (Android, iOS). Per altra banda, les aplicacions webs s'executen directament al navegador i no cal realitzar adaptacions per a cada dispositiu (si fan servir navegador que compleixen els estàndards), ara bé, l'accés a perifèrics és més limitat que a aplicacions natives i no es pot publicar a botigues d'aplicacions.

Per tot això, les aplicacions híbrides intenten combinar els avantatges d'ambdues vessants: Per una banda, permeten accedir a perifèrics, realitzar un sol desenvolupament i que sigui aprofitable a múltiples plataformes i publicable a botigues d'aplicacions. A canvi, però, el rendiment de les aplicacions podria ser inferior que a les aplicacions natives.

En aquest projecte sens dubte la millor opció és una aplicació híbrida, ja que els perifèrics que es necessiten són limitats (càmera i GPS), s'ha de poder publicar a Apple Store i a Play Store, i els el temps de desenvolupament és limitat.

3.1.1. Frameworks per a desenvolupar aplicacions híbrides

Avui en dia existeixen alguns de frameworks per a desenvolupar aplicacions híbrides, essent els més coneguts Ionic (Angular|Google), Xamarin (C#|Microsoft), React Native (React|Facebook) i Quasar (Vue|Evan You). Amb qualsevol d'aquests frameworks s'hagués pogut desenvolupar satisfactòriament l'aplicació d'aquest projecte, però cal tenir en compte una sèrie de factors a l'hora de decidir quin framework encaixa millor.

En primer lloc, Ionic fou un dels primers frameworks que feren capaç el desenvolupament d'aplicacions híbrides, per tant té una experiència sòlida però està basat en Angular (Google), per la qual cosa la corba d'aprenentatge és molt elevada⁵.

En segon lloc, Xamarin està basat en C# i malgrat sigui d'una gran companyia com Microsoft, també implicaria un sobre esforç d'aprenentatge.

En tercer lloc, React Native no té una corba d'aprenentatge tan pronunciada com Ionic-Angular, i malgrat Facebook hagi aportat una gran documentació, la sintaxi del codi no és amable per un desenvolupador habitual de JavaScript.

Finalment, Quasar i Vue malgrat no tinguin el suport d'una gran multinacional al darrera i no siguin els frameworks més utilitzats⁶ sí que tenen una excel·lent documentació, fan servir els estàndards JavaScripts habituals, la corba d'aprenentatge és molt baixa i de seguida s'obtenen resultats visibles. A més a més, la possibilitat de poder utilitzar tots els components existents de cordova i de npm fa que l'aplicació pugui fer servir les darreres tecnologies de programació.

A nivell de rendiment, no es trobarien diferències notables per a una aplicació de les característiques d'aquest projecte.

En definitiva, l'elecció del framework per aquest projecte ha estat encertada i un èxit. A l'annex 4, s'especifiquen els fitxers bàsics de configuració de l'aplicació, com per exemple la URL de l'API REST o com canviar els colors de l'aplicació; la resta de configuracions és per defecte com vénen amb Quasar.

3.1.2. Quasar Framework

L'eina principal per a desenvolupar la solució híbrida ha estat Quasar Framework, un framework de codi obert amb la base del framework de JavaScript Vue⁷, que ajuda a desenvolupadors web a crear webs i aplicacions (SPA, PWAs, aplicacions de mòbil, etc.). Quasar permet escriure només un codi i poder-lo executar a diferents plataformes fent servir les eines integrades.

A més, Quasar ofereix una gran multitud de components visuals llestos per fer servir i molt fàcils de configurar, en aquest projecte s'han fet servir botons, pestanyes, barra d'eines, icones, classes CSS, panells, notificacions, llistes, etcètera. De totes les eines disponibles de Quasar se n'han fet servir sols algunes, ja que també té selectors, entrades de formulari, valoracions, etcètera.

L'ús de funcionalitats o eines ja existent és molt fàcil d'integrar a Quasar, ja que sempre fa servir Vue i els estàndards JavaScript, així doncs, afegir un client de crides AJAX o llibreries de tractament de dates és molt fàcil.

Finalment, a nivell tècnic compta amb els següents avantatges⁸:

3.1.2.1. Vue

El framework JavaScript Vue ha suposat un canvi radical a l'hora de programar JavaScript, és molt fàcil desenvolupar mòduls aprofitables a solucions existents i veure els resultats de seguida. A més a més, la curva d'aprenentatge és molt bona.

3.1.2.2. Extensible

Quasar permet afegir qualsevol llibreria JavaScript i compilar-la amb el codi (gràcies a eines habituals web com webpack).

3.1.2.3. Totes les plataformes d'un sol cop

Amb un sol codi es poden crear automàticament pàgines web responsive, PWA (Progressive Web Apps), aplicacions de mòbils que semblen natives mitjançant Cordova i aplicacions d'escriptori.

3.1.2.4. Molts de components responsive i eficients

Quasar inclou molts de components que són responsives i estan pensats en ser eficients, tant per mòbil com per web (gràcies a Vue). La prioritat del framework és el rendiment i ser responsive.

3.1.2.5. Les millors pràctiques de programació per defecte

Les millors eines habituals web i les bones pràctiques de programació web també s'apliquen a Quasar per defecte: Minificació HTML/CSS/JS, ECMAScript 6, funcions d'accessibilitat, etcètera.

3.1.2.6. Suport d'escriptura de dreta a esquerra

Si l'idioma ha de ser escrit de dreta a esquerra, Quasar ho converteix automàticament.

3.1.2.7. Dos temes

Quasar ve amb el tema Material (Android) i iOS per defecte. La qual cosa és ideal per crear aplicacions que semblin natives per a dispositius mòbils d'ambdues plataformes.

3.1.2.8. Migració progressiva del codi existent

Degut a que fa servir Vue, i per tant UMD (Unified Module Definition), permet integrar el codi existent molt fàcilment i anar migrant la solució de manera progressiva.

3.1.2.9. Quasar CLI

Quasar CLI és una eina que facilita el desenvolupament gràcies al conjunt d'utilitats (algunes importades de la programació web) que inclou, entre altres:

- Hot-reload mantenint l'estat (inclús a un mòbil connectat o un emulador). Fent un canvi al codi, es compila i executa al dispositiu en qüestió de segons sense perdre la informació de pantalla
- Mantenir l'estat si hi ha hagut errors de compilació
- ESLint (opcional)
- Transformació del codi ES6 a codi suportat a qualsevol plataforma

3.1.2.10. Desenvolupament ràpid

Gràcies al conjunt de components i al Quasar CLI, molt ràpidament es veuen resultats i també serveix, en certa manera, d'eina de prototipatge.

3.1.2.11. Gran comunitat de desenvolupadors

Molts de desenvolupadors fan servir Quasar a dia d'avui i és fàcil trobar solucions de problemes habituals per Internet. A més a més, el framework s'actualitza sovint.

3.1.2.12. Compatibilitat amb moltes plataformes

Google Chrome, Firefox, IE11/Edge, Safari, Opera, iOS, Android, Windows Phone, Blackberry.

3.1.2.13. Componentes traduïbles

Inclou l18n per defecte i si l'idioma desitjat no està disponible és pot afegir molt fàcilment.

3.1.2.14. Bona documentació

La documentació és concisa, concreta i completa, la qual cosa fa que sigui molt fàcil trobar la documentació d'un component o de l'API del framework.

En definitiva, tots els punts anteriors fa que Quasar Framework sigui una de les millors opcions avui en dia per a desenvolupar aplicacions híbrides.

3.1.3. Cordova

Quasar depèn de Cordova per crear i executar aplicacions sobre Android i iOS. Malgrat això, és molt senzill afegir el mòdul de Cordova, ja que Quasar ve preparat per fer-ho. A l'annex 7.1 es mostra un exemple bàsic de com integrar-lo.

En la programació de la solució híbrida és quasi transparent l'ús de Cordova ja que Quasar ho gestiona tot.

3.1.4. Firebase

Firebase és la plataforma de Google per a desenvolupar aplicacions web i mòbils.

Després d'avaluar varies de com comunicar l'aplicació mòbil amb una base de dades en el núvol es va decidir fer una API en PHP, però una altra opció hagués pogut ser utilitzar tota la tecnologia de Firebase. Ara bé, degut a que a futurs projectes seria interessant realitzar un *backoffice* per a fer el tractament de les incidències, es va considerar una millor opció la realització d'una base de dades relacional a un servidor d'Internet.

No obstant això, sí que es fa servir Firebase per a rebre missatges Push amb notificacions de canvis a les incidències o altres missatges.

Al projecte s'adjunta la configuració d'exemple de Firebase per tal que no hagi de ser necessari configurar-lo.

3.1.5. Google Maps

Es fa servir Google Maps per mostrar la ubicació de la persona que vol fer una incidència. Ara bé, no es fa servir l'API de Google Maps directament, sinó que s'utilitza el complement `googlemaps` de Cordova que té una API per a realitzar les accions més comunes, com ubicar a la geoposició actual i crear un marcador.

3.1.6. Llibreries extres

A més a més de les llibreries per defecte que inclou Quasar, també se n'han fet servir d'altres per aconseguir la solució. Algunes d'elles es veuran en més detall al subcapítol de detalls de desenvolupament.

Aplicació base (Quasar):

- `moment`: Llibreria de tractament de dates
- `axios`: Client HTTP
- `vuex-persistedstate`: Llibreria per guardar l'estat a la memòria local, de tal manera l'aplicació es pot tancar i tornar a obrir recuperant l'estat

Cordova:

- `camera`: Complement per accedir a la càmera
- `geolocation`: Complement per accedir a la ubicació
- `file`: Component per accedir al sistema de fitxers
- `inappbrowser`: Component per obrir pàgines web dins de l'aplicació
- `firebase`: Component per interactuar amb Firebase (en aquest projecte només es fa servir per els missatges Push)
- `googlemaps`: Component per fer servir Google Maps

3.2. API i autenticació

Per a la recepció i enviament de dades a l'aplicació mòbil s'ha creat una API en PHP fent servir el framework Laravel.

Existeixen moltes possibilitats a l'hora de desenvolupar una API REST de forma eficient, o inclús es podria fer servir només eines del núvol, com Firebase, per emmagatzemar la informació. Ara bé, s'ha elegit Laravel per la facilitat de desenvolupament, la gran documentació, i les possibilitats d'ampliació que té. Tal com s'ha comentat, un possible projecte futur seria la implementació d'un *backoffice* complet en què es pugui gestionar tota la informació de la base de dades; aleshores, tota la informació ja estaria disponible a la base de dades del servidor obtinguda via l'API REST.

Malgrat el detall de l'API i la conjunció amb el *backoffice* queden fora de l'abast d'aquest projecte, a continuació se'n descriuen les eines utilitzades:

3.2.1. Laravel

Laravel és un framework PHP amb la filosofia de crear un codi elegant i simple que permet ser reutilitzable fàcilment. És per això que és un framework idoni per a desenvolupar una API REST.

En aquest projecte s'ha fet servir la darrera versió del framework existent, la 5.7.

A l'annex 7.2 es mostra un exemple d'un fitxer de rutes i d'un controlador en què es torna la informació necessària per a què l'aplicació mòbil funcioni.

3.2.2. OAuth2 Bearer Token

Per a assegurar la autenticació dels usuaris es decideix fer servir OAuth2 amb Bearer Token. La combinació d'aquesta tecnologia amb Laravel és molt fàcil d'implementar i és prou segura per autenticar els usuaris.

Quan un usuari inicia l'aplicació per primera vegada es connecta al servidor i demana un *token*. L'aplicació guarda aquest *token* i el farà servir per autenticar-se, generar incidències i obtenir l'historial.

3.3. Detall del desenvolupament de l'aplicació mòbil

3.3.1. Projecte

L'aplicació s'ha desenvolupat fent servir l'IDE PHPStorm (que també és compatible amb frameworks JavaScript, com Vue).

L'estructura del projecte ha continuat amb els estàndards de Quasar⁹. Dins de la carpeta src hi ha tot el codi font de l'aplicació i dins de la carpeta src-cordova la solució per executar-se amb Cordova.

De les carpetes generades per defecte, s'han fet modificacions a les següents:

- i18n: S'han creat tots els fitxers d'idioma necessaris per l'aplicació
- layouts: Hi ha dos dissenys: el per defecte utilitzat a tota l'aplicació i el de la generació d'una nova incidència
- pages: S'han creat totes les pàgines de l'aplicació
- plugins: S'ha afegit el complement i18n
- router: S'han creat les rutes de l'aplicació
- store: S'han creat els magatzems necessaris per guardar la informació obtinguda per l'API i la generació d'una incidència

3.3.2. Versions compatibles

Android 4.4 - Android 9

A l'hora del desenvolupament de la solució es fa servir cordova-android 7 que soporta Android amb versió SDK¹⁰ des de la 19 (Android 4.4 - Kit Kat) fins a la 27 (Android 8.1 - Oreo), encara que també és compatible amb Android 9.

iOS 8 i superiors

A l'hora del desenvolupament de la solució es fa servir cordova-ios 4.5.4 que soporta dispositius amb iOS 8 o superior.

3.3.3. Llenguatges i utilitats

Durant tot el desenvolupament del projecte s'han utilitzat les eines més modernes del mercat per tal d'obtenir un resultat òptim, per exemple s'ha fet servir EcmaScript 6, webpack i yarn.

3.3.4. iOS

Malgrat l'avantatge de fer aplicacions híbrides és no haver de programar més d'una vegada el codi font, la veritat és que per publicar sobre Apple Store és necessari compilar amb un Mac i disposar d'un iPhone o iOS.

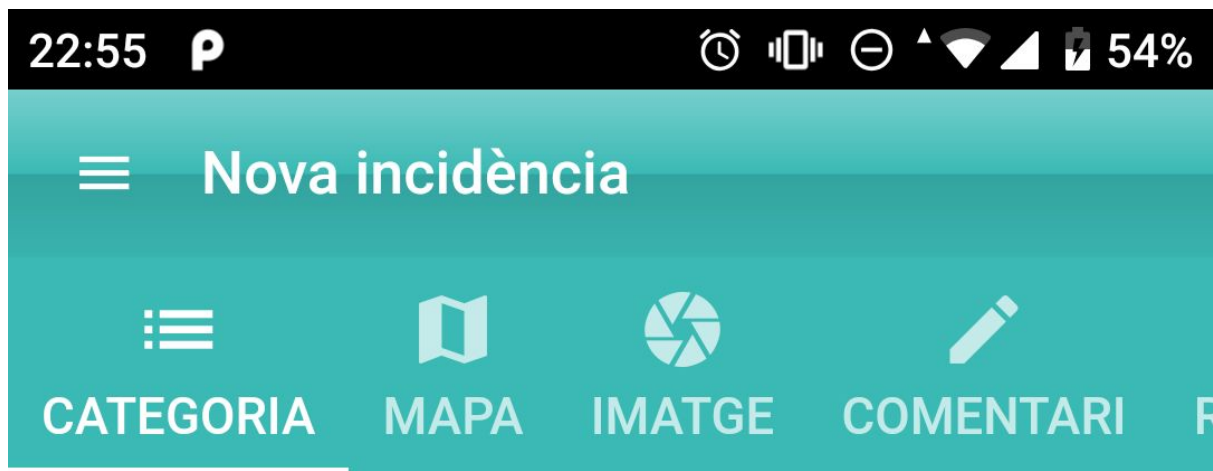
Per això, per reduir el cost del projecte, tot el desenvolupament es farà sobre un Windows i només a la fase final es compilarà sobre un Mac. Abans d'això, però, s'ha avaluat el resultat d'una compilació de Quasar Framework sobre un Mac i el resultat ha estat un èxit.

3.3.5. Disseny / Plantilles

L'aplicació té dues plantilles, la comuna que es fa servir a totes les pantalles i una específica per a la pantalla de generació d'una nova incidència.

Per una banda, la plantilla de generació d'una nova incidència es fa servir a totes les pantalles que intervenen en la creació d'una nova incidència: Categoria, Mapa, Imatge, Comentari i Resum.

La diferència amb la plantilla comuna és el conjunt de pestanyes que defineixen els components d'una incidència, de tal manera que de forma fàcil es pot tornar a un punt anterior per canviar la informació.



Imatge 3: Plantilla de generació d'una nova incidència

Per altra banda, a la plantilla comuna sols apareix el nom de l'aplicació i es fa servir per a totes les altres pantalles.



Imatge 4: Plantilla comuna

3.3.6. Emmagatzament

Tal com s'ha comentat anteriorment, l'aplicació careix d'una base de dades relacional, sinó que fa servir un simple objecte JSON per guardar la informació.

L'aplicació fa servir vuex per guardar l'estat d'una nova incidència i tota la informació obtinguda de l'API. Ara bé, degut a que LocalStorage només permet l'emmagatzematge de valors senzills (cadena de text i números) és necessari transformar tota la informació a a una cadena de text JSON, però això la llibreria vuex-persistedstate ho fa automàticament.

Així doncs, l'aplicació consta de tres emmagatzaments diferents:

API

On es guarda tota la informació necessària per fer servir l'aplicació (botons, categories, estats i missatges):

```
export default {  
  botons: [],  
  estats: [],  
  categories: [],  
  missatges: []  
}
```

Incidència

On es guarda tota la informació d'una nova incidència

```
export default {  
  categoria: {},  
  codi: null,  
  lat: null,  
  lng: null,  
  comentari: null,  
  adjunts: []  
}
```

Incidències

On es guarda l'històric de les incidències (obtingut via l'API)

```
export default {  
  items: []  
}
```

3.3.7. Pantalles i rutes

Quasar fa servir *vue-router*¹¹ per canviar de pantalla. Les rutes es configuren al fitxer `src/router/routes.js` i s'especifica de quina plantilla hereda el disseny.

En aquest projecte hi ha dos grups de pantalles: les que hereden de la plantilla comuna i les que hereden de la plantilla de generació d'incidència.

```

export default [
  {
    path: '/',
    name: 'index',
    component: () => import('layouts/default'),
    children: [
      {path: '', component: () => import('pages/index')},
      {path: 'text/:codi', name: 'text', component: () => import('pages/text')},
      {path: 'botons', name: 'botons', component: () => import('pages/botons')},
      {name: 'registre'...},
      {name: 'historial'...},
      {name: 'detall-incidencia'...},
      {name: 'nou-missatge'...},
    ],
  },
  {
    path: '/incidencia',
    name: 'incidencia',
    component: () => import('layouts/incidencia'),
    children: [
      {name: 'categoria'...},
      {path: 'mapa', name: 'mapa', component: () => import('pages/mapa')},
      {path: 'imatge', name: 'imatge', component: () => import('pages/imatge')},
      {name: 'comentari'...},
      {path: 'resum', name: 'resum', component: () => import('pages/resum')},
    ],
  },
  { // Always leave this as last one
    path: '*',
    component: () => import('pages/404'),
  },
];

```

Imatge 5: Captura del fitxer `src/router/routes.js`

Per una banda, les pantalles del primer grup són les necessàries per accedir a les funcionalitats extres de l'aplicació: consultar l'historial d'incidències, registre d'usuari, botons de la pantalla inicial i pantalles amb un text definit.

Per altra banda, les pantalles del segon grup són totes aquelles que componen una incidència: Categoria, Mapa, Imatge, Comentari i Resum.

3.3.7.1. Pantalla índex

La pantalla índex és la primera que es carrega quan s'obri l'aplicació i serveix com a pantalla de càrrega mentre s'obtenen les dades inicials a partir l'API.

En concret, les funcions que fa són: Obtenir les dades de l'API i guardar-ho a l'emmagatzematge API, generar un token de Firebase i enviar-lo al servidor via l'API. Després, es canvia automàticament a la pantalla de botons.

3.3.7.2. Pantalla botons

La pantalla de botons és la principal i que espera l'acció de l'usuari. Mostra el llistat de botons que s'han definit al back-office. Un botó està compost per propietats: títol, imatge i enllaç. El camp enllaç pot tenir diferents valors que realitzen diferents accions:

- INCIDENCIES: S'obri la pantalla de generació d'una nova incidència
- HISTORIAL: S'obri la pantalla d'historial d'incidències
- AJUDA: S'obri una pantalla de missatge amb el codi AJUDA
- REGISTRE: S'obri la pantalla de registre
- Altres: Qualsevol altra valor s'obrirà dins de l'aplicació com una pàgina web

3.3.7.3. Pantalla missatge

La pantalla de missatge mostra el contingut d'un missatge HTML a la pantalla. Mitjançant l'API s'obtenen tots els missatges identificats per un codi.

```
<template>
|   <q-page padding>
|       <div v-html="text"></div>
|   </q-page>
|</template>
```

Imatge 6: Template de la pantalla de missatge

3.3.7.4. Pantalla registre

La pantalla registre permet que l'usuari doni un correu electrònic i contrasenya per registrar-se

L'aplicació fa una crida al servidor via l'API amb les dades i torna a la pantalla de botons si ha tengut èxit.

3.3.7.5. Pantalla categories

La pantalla de categories mostra totes les categories d'incidència obtingudes de l'API. Seleccionar la categoria canvia automàticament a la pantalla de mapa. Quan es tria una categoria es guarda l'identificador a l'objecte Incidència.

3.3.7.6. Pantalla mapa

La pantalla de mapa carrega el mapa de Google Maps i posiciona a l'usuari a la seva ubicació.

En aquesta pantalla també es demana permís a l'usuari per obtenir la seva ubicació. La petició d'obtenció d'ubicació és automàtic gràcies al component cordova-plugin-googlemaps. La única configuració que cal fer al complement és especificar la clau per fer servir l'API de Google al fitxer `src-cordova/config.xml`. A més a més, el text que s'especifica a `LOCATION_WHEN_IN_USE_DESCRIPTION` és el missatge que apareixerà als dispositius iOS quan es demani el permís. A dia d'avui el complement no permet que el missatge sigui traduïble.

```

<plugin name="cordova-plugin-googlemaps" spec="2.3.6">
  <variable name="API_KEY_FOR_ANDROID" value="" />
  <variable name="PLAY_SERVICES_VERSION" value="15.0.1" />
  <variable name="ANDROID_SUPPORT_V4_VERSION" value="26.1.0" />
  <variable name="API_KEY_FOR_IOS" value="" />
  <variable name="LOCATION_WHEN_IN_USE_DESCRIPTION" value="This app wants to get your location" />
  <variable name="LOCATION_ALWAYS_USAGE_DESCRIPTION" value="This app wants to get your location" />
</plugin>

```

Imatge 7: Configuració del component de mapa

Quan l'usuari confirma la ubicació es guarda la latitud i longitud a l'objecte Incidència.

3.3.7.7. Pantalla imatge

La pantalla d'imatge permet adjuntar una o més imatge a la incidència. La imatge pot ser obtinguda a partir de la càmera o del sistema de fitxers.

El component de Cordova cordova-plugin-camera permet agafar imatges del sistema de fitxer i també fer fotografies. La programació d'una opció o una altra és exactament igual, l'únic camp diferent és a les opcions de la crida de la funció

```
navigator.camera.getPicture: CAMERA O PHOTOLIBRARY
```

Per altra banda, per guardar la imatge es llegeix amb FileReader i com a DataURL. La imatge es guarda a l'*array* adjunts de l'objecte Incidència.

3.3.7.8. Pantalla comentari

La pantalla comentari mostra un textarea per poder afegir un comentari a la incidència. Es guarda a l'atribut comentari de l'objecte Incidència.

3.3.7.9. Pantalla resum

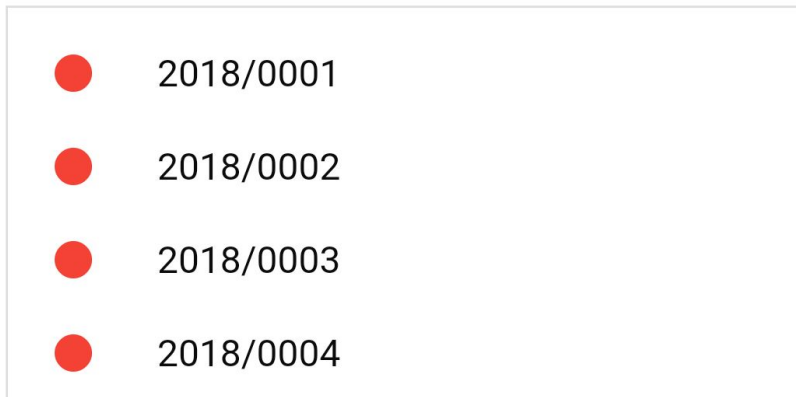
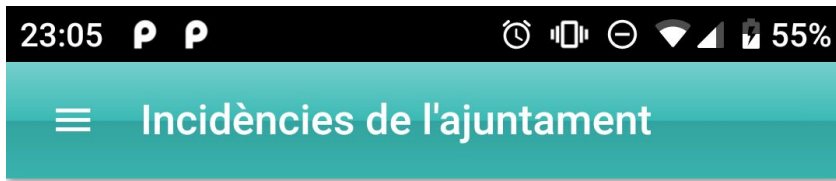
La pantalla de resum mostra tot el contingut de la incidència recuperant la informació del LocalStorage.

3.3.7.10. Pantalla històric

A la pantalla d'històric es fa una crida al servidor per obtenir el darrer estat de les incidències i mostrar-ho en llistat.

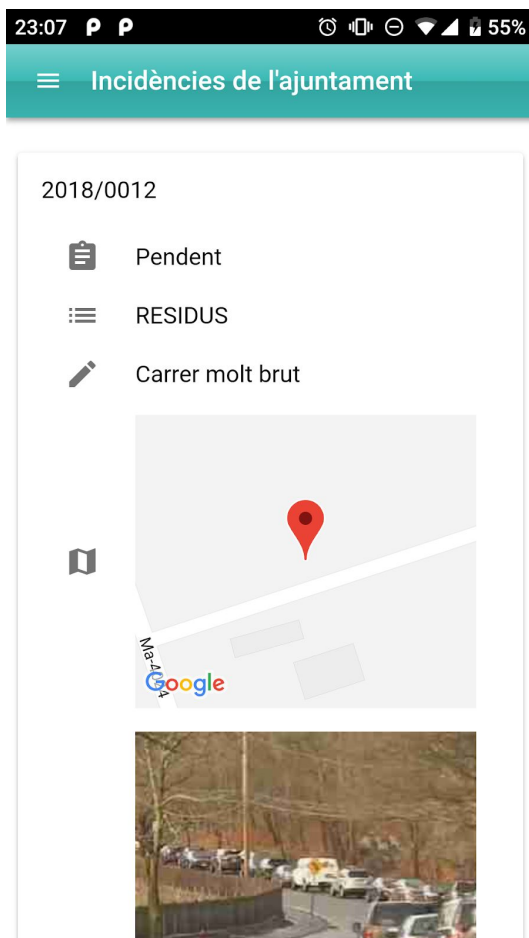
S'ha d'avaluar l'ordre de les incidències, si ordenar-les per ordre ascendent de creació o bé descendent. En aquesta versió s'ha decidit order-ho per ordre ascendent de creació.

A més a més, una petita icona mostra l'estat de la incidència: vermell si està pendent, taronja si està en curs i verd si està finalitzada.



Imatge 8: Captura d'històric d'incidències

3.3.7.11. Pantalla detall d'incidència



Imatge 9: Pantalla de detall d'incidència

La pantalla del detall de la incidència mostra totes les propietats que componen la incidència.

Amb el component de Cordova googlemaps es mostra la ubicació de la incidència.

Per no carregar en excés la memòria del telèfon, només es guarden les URL de les imatges del servidor enlloc de les imatges codificades en base64 o similar. La desavantatge d'això és que es requereix Internet per mostrar-les.

A la part inferior hi ha un botó que canvia a la pantalla per a la generació d'un nou missatge.

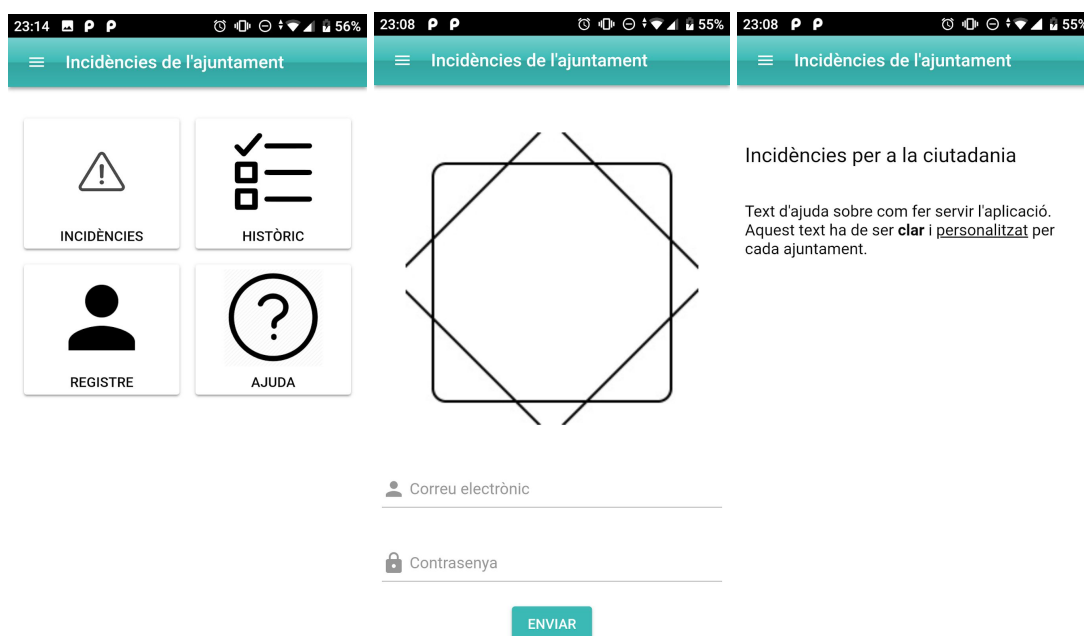
3.4. Pantalles de l'aplicació

A continuació es mostren els detalls de les pantalles de l'aplicació.

Cal tenir en compte que tots els botons i totes les categories són multi-idioma, així com també ho és l'aplicació. L'aplicació detecta l'idioma que té el dispositiu i mostra els textos i les imatges en consonància, si des de l'API no han especificat una traducció, es mostrarà l'idioma per defecte que hagi triat l'ajuntament.

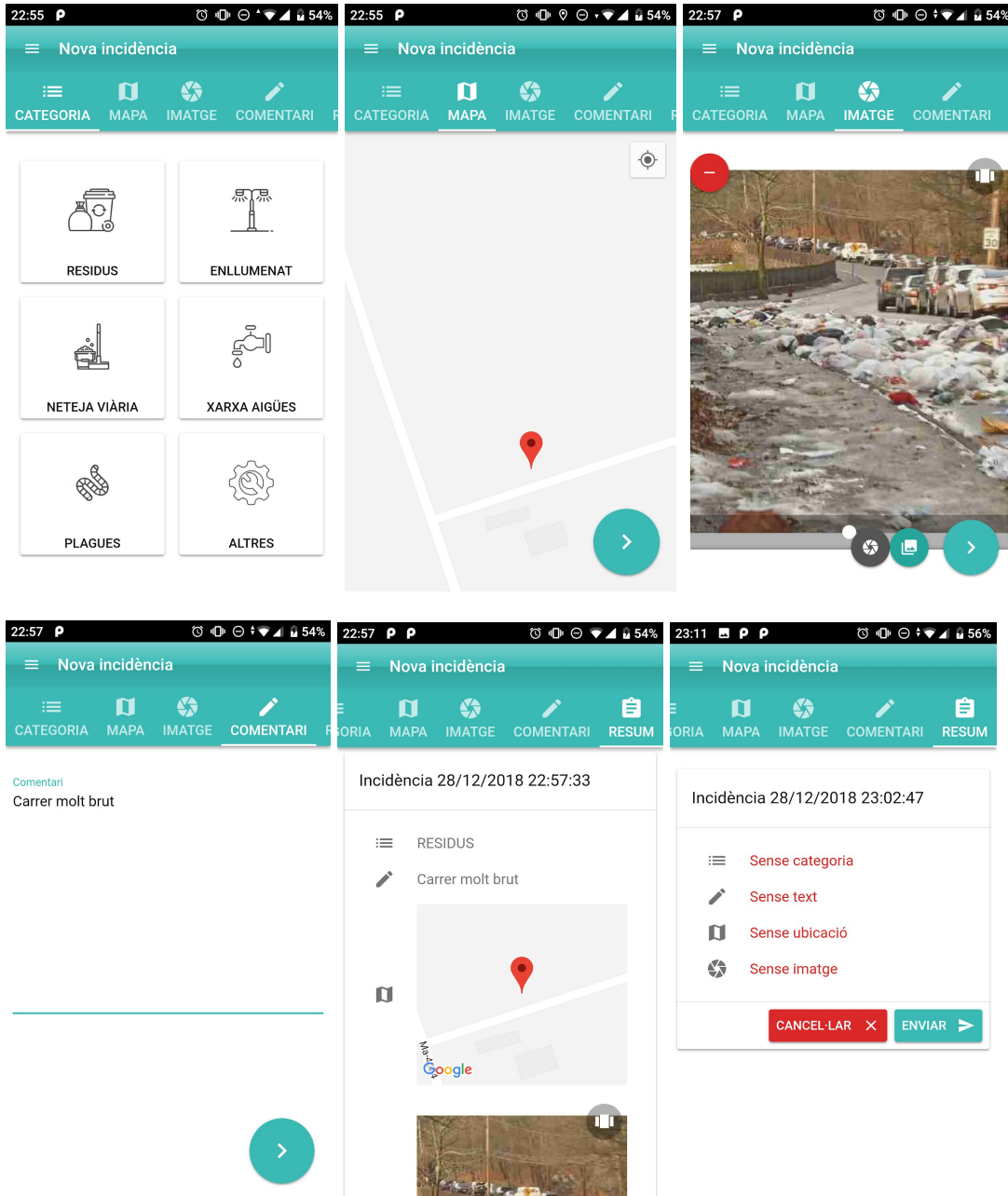
3.4.1. Pantalles bàsiques

Les pantalles bàsiques són la de càrrega (“SplashScreen”), la principal en què es mostren els botons, la pantalla de registre (que s'hi accedeix via un botó configurable o bé via el menú lateral) i la pantalla d'ajuda.



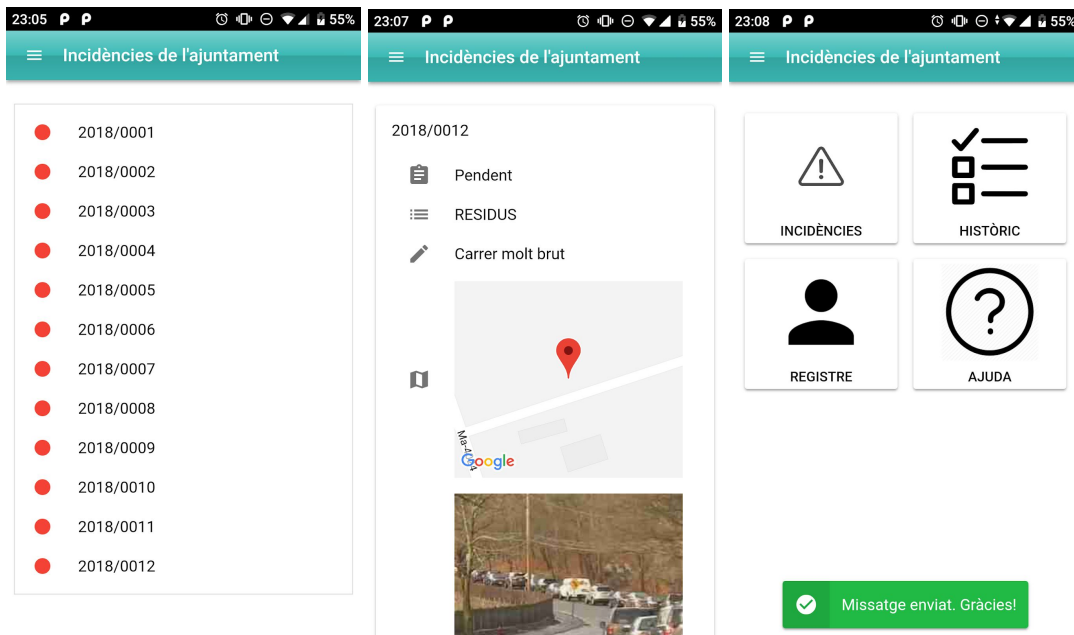
3.4.2. Pantalles nova incidència

Hi ha 5 pantalles de nova incidència: Categoria, Mapa, Imatge, Comentari i Resum



3.4.3. Pantalles de consulta d'incidències

Hi ha 3 pantalles de consulta d'incidències passades: Històric, Detall i Nou missatge



3.5. Proves

El framework Quasar no té una eina de proves per defecte o *out of de box*, però està en procés de crear-se¹². Així doncs, es decideix fer les proves de forma manual.

La plataforma es prova sempre sobre un Xiaomi Mi A1 amb Android 8.1 (Oreo) i sobre un Moto G4 amb Android 7 (Nougat). Finalment, abans de l'entrega final es demanaran a voluntaris provar l'aplicació sobre altres dispositius Android i també es provarà sobre un iPhone.

3.5.1. Proves manuals

A continuació es descriuen un conjunt de proves manuals que es fan cada vegada que es presenti una nova versió. Aquestes proves podran canviar durant el transcurs del projecte per adaptar-se a les noves necessitats.

Pantalla	Prova
Inici	Es mostra el logo Carrega la informació de l'API
Botons	Es mostra el llistat de botons amb text i imatge Pitjar botó INCIDÈNCIES obri la pantalla de categoria d'incidència
Incidència - Categoria	Es mostra el llistat de categories d'incidència Pitjar un botó obri la pantalla de mapa
Incidència - Mapa	Es mostra el mapa Es demana permís per fer servir la ubicació (la primera vegada) El mapa se situa a la ubicació de l'usuari i es mostra un marcador El botó continuar obri la pantalla d'imatge
Incidència - Imatge	Es mostra que no hi ha cap imatge seleccionada Pitjar el botó de Càmera permet fer una foto i mostrar-la al llistat d'imatges Pitjar el botó de Galeria permet agafar una foto de la galeria i mostrar-la al llistat d'imatges En qualsevol dels dos casos anteriors, es demana permís per fer servir la càmera o accedir al sistema de fitxers (la primera vegada) El botó continuar obri la pantalla de comentari
Incidència - Comentari	Es mostra un camp de text amb l'etiqueta Comentari Es pot introduir un comentari de qualsevol mida El botó continuar obri la pantalla de resum

Incidència - Resum	<p>Es mostra la informació introduïda anteriorment, i només la introduïda</p> <p>S'avisava en vermell si algun camp està incomplet</p> <p>La posició GPS i la categoria són obligatòries per enviar la incidència</p> <p>El botó enviar envia la incidència al servidor i obri la pantalla de botons inicials</p> <p>El botó cancel·lar esborra la incidència i obri la pantalla de botons inicials</p>
Històric	<p>Es mostra una finestra de càrrega</p> <p>S'obtenen les incidències de l'API</p> <p>Es mostren les incidències obtingudes</p> <p>Pitjar sobre una incidència mostra el detall de la incidència</p>
Detall d'incidència	<p>Es mostra tota la informació de la incidència</p> <p>El botó de nou missatge obri la finestra de nou missatge</p>
Nou missatge	<p>Es mostra un camp de text amb l'etiqueta Missatges</p> <p>Es pot introduir un missatge de qualsevol mida</p> <p>El botó confirmar envia el missatge al servidor</p>

3.5.2. Proves de l'API

Malgrat l'API és una petita part del projecte i no se n'especifiquen els detalls, també cal fer-ne proves. A continuació es descriuen breument les proves que s'han fet, per a més detall, s'inclou el codi font de les proves.

Laravel permet fer proves unitàries i de funcionalitat molt fàcilment fent servir phpunit. En aquest cas, s'han fet dues proves de funcionalitat:

3.5.2.1. Obtenir les dades inicials de l'API

La prova de funcionalitat intenta obtenir les dades inicials necessaris per a l'APP via l'API: Fa una connexió al servidor, i espera veure una categoria d'incidència, un botó i un missatge.

3.5.2.2. Crear una incidència

La prova de funcionalitat comprova que la categoria i la ubicació són indispensables per a la creació d'una incidència, així com que l'usuari hagi generat correctament un *token* d'autenticació.

4. Conclusions

L'objectiu principal d'aquest projecte era estudiar la viabilitat de la utilització d'un telèfon mòbil per apropar la ciutadania a l'ajuntament mitjançant l'enviament i tractament de les incidències del seu municipi, i a més, ser un punt de partida per a la creació d'una aplicació ciutadana personalitzada per cada ajuntament.

Per això, havent estudiat l'oferta actual d'aplicacions destinades a la transmissió d'incidències d'un municipi cap a un ajuntament, és clar que hi ha un interès en apropar la ciutadania a l'ajuntament, sobretot a l'àrea de les incidències de cada dia.

A més a més, havent fet un petit estudi de disseny centrat en l'usuari, s'ha pogut detectar *a priori* quines funcionalitats són les més desitjades i útils pels usuaris finals de l'aplicació, que res tenen a veure a amb la primera concepció del projecte.

També, havent fet un breu anàlisi de les tecnologies actuals per a la creació d'aplicacions mòbils, es pot concloure que és factible crear una aplicació mòbil per a múltiples plataformes amb poc temps i pocs recursos.

No obstant això, i malgrat el resultats hagin estat satisfactoris, les decisions preses en aquest projecte obliguen que l'aplicació depengui de servidors externs per a funcionar.

Així mateix, tot i que el projecte ha estat enfocat des d'un principi a realitzar una aplicació prototip, es troba a faltar l'estudi de l'impacte social i econòmic que tendria l'ús de les noves tecnologies dins d'aquest àmbit. No obstant això, a partir d'aquest projecte es proposa fer-ne un estudi i seguiment.

Amb tot això, es pot concloure que sí és possible crear una aplicació destinada a l'enviament d'incidències adaptada a cada ajuntament. Ara bé, també s'ha detectat que el punt que provoca més frustració als usuaris és la manca de resposta dels ajuntaments després d'haver tramitat una incidència. Aleshores, no té sentit que un ajuntament dediqui esforços i recursos en la creació d'una aplicació personalitzada, si després no dedica temps al seguiment diari.

En definitiva, es pot considerar el projecte un èxit perquè ha assolit els objectius proposats, però cal tenir en compte que aquest projecte obri les portes a altres molt interessants que completarien la solució i la farien útil.

5. Treball futur

Aquest projecte vol ser un punt de partida de cara a realitzar aplicacions que apropin la ciutadania amb el seu ajuntament, i la part de tramitació d'incidències és només una petita part de totes les funcionalitats que ofereixen les noves tecnologies.

A partir d'aquest projecte poden sortir moltes idees que completarien la solució:

Per una banda, es podria ampliar l'aplicació mòbil per a què tingués més funcionalitats, com un apartat de notícies, adjuntar vídeos o notes de veu a les incidències. També es podrien afegir funcions socials, com iniciar sessió amb una xarxa social o compartir una incidència.

Per altra banda, la part de servidor podria créixer cap a una aplicació de gestió. Es podria fer tota una plataforma web de gestió d'incidències, en què es podria canviar tota la configuració de l'aplicació (categories d'incidències, botons i missatges) i poder fer el tractament de totes les incidències. Inclús, es podria obrir una secció per a què les empreses concessionàries poguessin gestionar les incidències que anessin destinades a elles (neteja viària, recollida, enllumenat, aigües).

Finalment, un projecte de caire més humà, però possiblement el més útil i interessant, seria fer el seguiment i millorar la resposta des de l'ajuntament de les incidències generades; estudiar la manera per a què la solució sigui útil, fàcil i dinàmica per a poder resoldre el problema per a què ha estat creada: apropar el ciutadà a l'ajuntament.

6. Bibliografía

1. "Nuevas tecnologías aplicadas a la gestión de residuos sólidos" 30 jul.. 2015, <https://www.esmartcity.es/comunicaciones/nuevas-tecnologias-aplicadas-gestion-residuos-solidos-urbanos-limpieza-viaria>. Se consultó el 30 sept.. 2018.
2. "Comunicación con la ciudadanía en línea | Ayuntamiento de Barcelona." <https://ajuntament.barcelona.cat/es/content/canales-de-comunicaci%C3%B3n-ciudadana-en-l%C3%ADnea>. Se consultó el 30 sept.. 2018.
3. "Tendencias de consumo móvil en España - Deloitte." <https://www2.deloitte.com/es/es/pages/technology-media-and-telecommunications/articles/consumo-movil-espana.html>. Se consultó el 30 sept.. 2018.
4. "Quasar Framework." <https://quasar-framework.org/>. Se consultó el 30 sept.. 2018.
5. "Why Angular JS So Difficult – Allen Kim – Medium." 7 feb.. 2017, <https://medium.com/allenhwkim/why-angular-js-so-difficult-57032dd327a4>. Se consultó el 28 dic.. 2018.
6. "ionic vs quasar-framework vs react-native | npm trends." <https://www.npmtrends.com/ionic-vs-quasar-framework-vs-react-native>. Se consultó el 28 dic.. 2018.
7. "Vue.js." <https://vuejs.org/>. Se consultó el 1 dic.. 2018.
8. "Introduction to the Quasar Framework - Quasar Framework." 19 sept.. 2018, <https://quasar-framework.org/guide/introduction-to-quasar.html>. Se consultó el 1 dic.. 2018.
9. "Directory Structure - Quasar Framework." 25 jul.. 2018, <https://quasar-framework.org/guide/app-directory-structure.html>. Se consultó el 1 dic.. 2018.
10. "Android Platform Guide - Apache Cordova" <https://cordova.apache.org/docs/en/latest/guide/platforms/android/>. Se consultó el 1 dic.. 2018.
11. "Vue Router - Vue.js." <https://router.vuejs.org/>. Se consultó el 1 dic.. 2018.
12. "GitHub - quasarframework/quasar-testing: [WORK IN PROGRESS]" <https://github.com/quasarframework/quasar-testing>. Se consultó el 9 dic.. 2018.
13. "Configure developer options - Android Developers." 25 abr.. 2018, <https://developer.android.com/studio/debug/dev-options?hl=es-419>. Se consultó el 29 dic.. 2018.

7. Annexos

7.1. Annex 1 - Hello world

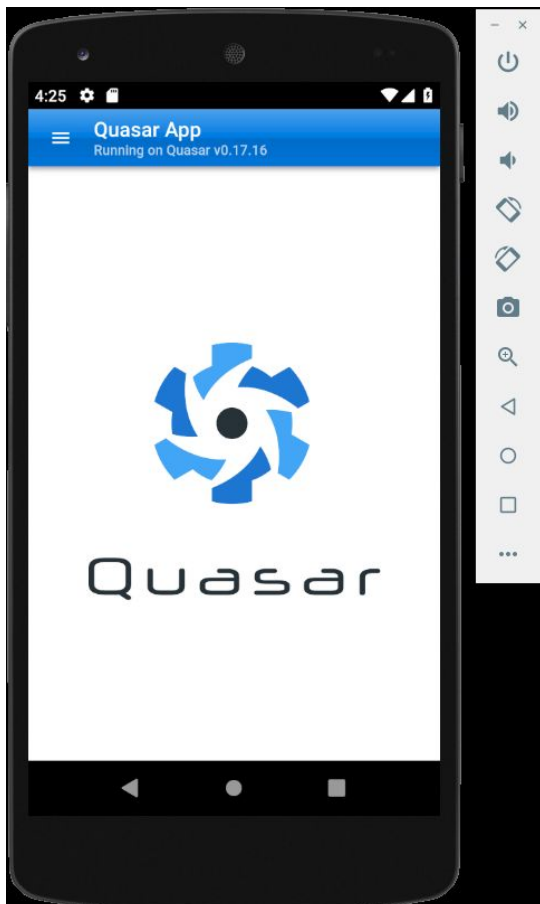
La tecnologia triada per a la realització de l'aplicació del projecte és Quasar. Quasar és un framework de codi obert basat en Vue que permet crear pàgines web i aplicacions mòbils *responsives*. La generació de les aplicacions per a Android i iOS és mitjançant Apache Cordova, però també es podria fer servir Quasar per crear SPA (Single Page App), aplicacions d'escriptori i PWAs (Progressive Web App).

Per a la generació d'una aplicació "Hello world" cal instal·lar el framework seguint les passes descrites a la pàgina web de Quasar:

1. Instal·lar les eines de desenvolupament clàssiques per a desenvolupar (Android Studio o XCode)
 - a. Descarregar i instal·lar l'aplicació i els SDK necessaris
 - b. Descarregar i instal·lar yarn (i nodejs)
2. Instal·lar vue-cli
 - a. yarn global vue-cli
3. Instal·lar les eines de comandes de Quasar
 - a. yarn global add quasar-cli
4. Generar un projecte nou
 - a. quasar init helloworld
 - b. Aquí es pot especificar mòduls addicionals per afegir al projecte, com multidioma, enregistrament de l'estat, etcètera. Tanmateix, aquests mòduls es poden afegir més endavant
5. Afegir el mode cordova
 - a. quasar mode -a cordova
6. Afegir la plataforma mòbil desitjada (mitjançant cordova)
 - a. (dins del directori de cordova: src-cordova) cordova platform add android
7. Comprovar requeriments
 - a. cordova requirements
 - b. En aquest punt es comprovarà si es té Android Studio instal·lat, l'SDK actualitzat i un emulador
8. Executar el helloworld
 - a. quasar dev -m cordova -T android

Per a editar el codi font es pot fer servir qualsevol IDE per HTML/Javascript. En aquest cas es decideix utilitzar PHPStorm, ja que el projecte també tindrà una API feta en PHP.

Imatge 10: Captura de pantalla del codi font



Imatge 11: Captura de pantalla de l'emulador

7.2. Annex 2 - API - Dades inicials de funcionament de l'APP

La tecnologia triada per al desenvolupament de l'API és el framework Laravel, que permet el desenvolupament d'API de forma molt senzilla.

A continuació es mostra la programació necessària per tornar les dades essencials per a què l'aplicació mòbil funcioni (botons, categories d'incidències i missatges):

- /routes/api.php

```
Route::name('index')->get('/', 'ApiController@index');
```

- /app/Http/Controllers/APIController.php

```
/**
 * Totes les dades
 *
 * @return array
 */
public function index()
{
    return [
        'categories' => \App\Categoria::all(),
        'botons' => \App\Boto::all(),
        'missatges' => \App\Missatge::all(),
    ];
}
```

Categoria, Boto i Missatge són models que estenen Eloquent

Amb la programació anterior, es retorna un array amb el format JSON amb totes les categories, botons i missatges de la base de dades. Degut a que es fa servir la ruta dins del fitxer api.php ja s'inclou la configuració necessària per a fer servir l'API.

7.3. Annex 3 - Error de desplegament

Quan hi ha una nova versió de cordova-android o de les llibreries de maven de vegades Cordova no aconsegueix compilar correctament l'aplicació.

La solució més senzilla és esborrar el directori `src-cordova/platforms` i tornar a compilar.

```
37 actionable tasks: 8 executed, 29 up-to-date
FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':app:transformClassesWithDesugarForDebug'.
> com.android.build.api.transform.TransformException: java.lang.RuntimeExc

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --

* Get more help at https://help.gradle.org
```

Imatge 12: Exemple d'error de compilació

7.4. Annex 4 - Configuració de l'aplicació mòbil

Tota la configuració de l'aplicació se centralitza al fitxer `quasar.conf.js`, i el camp més important per canviar és la `API_URL`, que defineix la URL que té el servidor de l'API REST.

Per altra banda, l'aplicació mòbil té dos mètodes de funcionament: producció i desenvolupament. A nivell de programació, la principal diferència és que per la configuració de desenvolupament és que és necessari establir un proxy per atacar l'API.

Si l'aplicació s'executa en desenvolupament, les rutes `/api` i `/storage` s'han de transformar a peticions vàlides contra l'API per tal de no violar els control d'accés HTTP (CORS). En canvi, a producció les crides s'han de fer automàticament al servidor de l'API.

Quasar estableix automàticament el flag de desenvolupament o de producció segons la comanda que es faci servir. A més a més, s'han configurat dos accesos ràpids al fitxer `package.json` per fer-ho més fàcil:

android-dev:

```
quasar dev -m cordova -T android
```

android-prod:

```
quasar build -m cordova -T android && cd src-cordova && cordova run android --device
```

No cal canviar cap paràmetre per a què l'aplicació faci servir la `API_URL` en producció i el proxy a desenvolupament, ja que al fitxer de configuració ja es detecta quin mètode es fa servir:

```
env: {  
  URL: ctx.dev ? '' : '' + API_URL + '',  
},
```

7.5. Annex 5 - Manual d'usuari

L'aplicació Incidències intenta que sigui molt fàcil de fer servir i autoexplicativa. Totes les passes que es fan per crear una incidència són seqüencials i sempre es pot rectificar una acció.

Per a generar una incidència cal fer:

1. Prémer sobre el botó *Incidències*
2. Seguir les passes necessàries per a la generació d'una incidència:
 - a. Triar una categoria
 - b. Ubicar l'incidència sobre un mapa
 - c. (Opcional) Adjuntar una o més fotografies
 - d. (Opcional) Descriure la incidència
 - e. Revisar que la informació introduïda és correcta
 - f. Enviar la incidència

Per a consultar l'històric d'incidències cal fer:

1. Prémer sobre el botó *Històric* (o bé a *Les meves incidències* del menú)
2. Elegir una incidència de la llista
 - a. El color dins d'un cercle indica l'estat de la incidència
 - i. Vermell: Pendent de revisar
 - ii. Taronja: En curs
 - iii. Verda: Resolta
3. Veure el detall de la incidència generada

Per a enviar un nou missatge cal fer:

1. Anar a l'històric i seleccionar una una incidència
2. Pitjar el botó de la part inferior de la pantalla amb un globus i línies horitzontals
3. Escriure el missatge
4. Enviar

Per a registrar-se cal fer:

1. Prémer sobre el botó *Registre*
2. Introduir un correu electrònic i una contrasenya
3. Enviar
4. A partir d'ara, les incidències estaran lligades al correu electrònic proporcionat, i es rebran notificacions via correu dels canvis fets

7.6. Annex 6 - API - Compilar i executar

L'API està feta amb PHP i fa servir el framework Laravel.

7.6.1. Requeriments

1. Servidor web (apache2, PHP, nginx, ...)
2. Base de dades (mysql, sqlite, oracle, ...)
3. PHP 7.2 amb el connector per a la base de dades que es faci servir
4. composer

7.6.2. Posta a punt

El projecte adjuntat a la memòria ja ve preparat per a funcionar sense necessitat de crear una base de dades o preparar un servidor web més enllà del propi del PHP, però el més habitual és que s'hagi de crear apart. Per això, cal assegurar-se de:

1. Existeix el fitxer `.env` a l'arrel del projecte
 - a. Si no existeix, cal copiar-lo del `.env.example` i configurar la connexió a la base de dades i establir una clau de xifrat
2. (Només SQLite) Existeix el fitxer `database/database.sqlite`

Instal·lació de dependències

Per a descarregar i instal·lar les dependències cal executar composer a l'arrel del projecte:

```
composer install
```

Preparar la informació bàsica de base de dades per a funcionar

El projecte inclou un *seeder* per a fer servir l'aplicació mòbil directament sense haver de configurar res. Per carregar-lo cal fer a l'arrel del projecte (esborra i recrea tota la base de dades, incloent les dades de prova):

```
php artisan migrate:fresh --seed
```

(Opcional) Preparar servidor web

El projecte ja ve preparat per servir l'API sense necessitat d'un servidor web més que el propi de PHP, útil per a desenvolupament, però no aplicable a producció:

```
php artisan serve --host 0.0.0.0  
(o si es té npm/yarn instal·lat) yarn serve
```

Prova de funcionament

Per a comprovar que l'API està escoltant correctament, es pot provar amb un navegador qualsevol d'accedir a <http://localhost:8000/api> (o un altre port assignat) i veure informació de categories d'incidències, missatges, etcètera.

7.7. Annex 7 - APP - Compilar i executar

L'aplicació està feta amb Quasar i necessita node, npm i l'SDK dels dispositius per a funcionar (consultar l'Annex 1 - Hello World per a més informació).

7.7.1. Requeriments

1. SDK d'Android i/o iOS
2. yarn (o npm) i nodejs
3. git

7.7.2. Posta a punt

Cal tenir un emulador en marxa o bé un dispositiu connectat via USB amb la depuració habilitada¹³.

Instal·lació de dependències

Per a descarregar i instal·lar les dependències cal executar yarn a l'arrel del projecte:

```
yarn
```

Configurar la URL de l'API

Al fitxer quasar.conf.js cal ajustar la variable API_URL a la del servidor de l'API

```
const API_URL = 'http://IP:PORT'
```

Executar

S'han preparat dos scripts npm per a fer més fàcil l'execució:

- `yarn android-dev`: compila i executa l'aplicació en mode desenvolupament
- `yarn android-prod`: compila i executa l'aplicació en mode producció (però amb alguns flags de desenvolupament per control d'errors, per preparar l'aplicació a mode per publicar cal afegir el flag `--release` a la crida de cordova)

(Opcional) Firmar i publicar

Malgrat la publicació a botigues d'aplicacions quedi fora de l'abast del projecte, per a signar i alinear l'aplicació APK es pot fer servir, per Android, apk-signer. Caldrà crear una keystore i un certificat per aplicació.