

One click assistance

Maria Guarch Alarcón

Grau d'Enginyeria Informàtica
TFG - Desenvolupament web

Gregorio Robles Martínez
Santi Caballe Llobet

12/06/2019



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

FITXA DEL TREBALL FINAL

Títol del treball:	One click assistance
Nom de l'autor:	Maria Guarch Alarcón
Nom del consultor/a:	Gregorio Robles Martínez
Nom del PRA:	
Data de lliurament:	12/06/2019
Titulació o programa:	Pla d'estudis de l'estudiant
Àrea del Treball Final:	Desenvolupament web
Idioma del treball:	Català
Paraules clau	Angular, Firebase, Google maps
Resum del Treball:	
<p>El present treball final de grau consisteix a una aplicació web dirigida a empreses d'assistència en carretera. Actualment, quan hi ha un accident o incidència en carretera, els clients es comuniquen mitjançant la línia telefònica amb l'operador per indicar les dades d'ubicació i sol·licitar els serveis d'assistència que necessita. Malauradament, en ocasions, aquestes dades arriben incorrectament a l'operador i/o l'empresa d'assistència. A més, en cas de disputa entre companyies asseguradores, poder obtenir imatges en el moment de l'accident pot ser molt útil per esbrinar qui n'és responsable.</p> <p>Aquest treball final de grau pretén donar solució a aquests inconvenients mitjançant una aplicació que permet ubicar alertes en un mapa i establir un xat amb l'operador que permet enviar imatges.</p> <p>Per dur a terme aquest projecte s'ha desenvolupat una aplicació amb Angular, utilitzant Firebase com a Backend as a Service, que proporciona eines com l'autenticació d'usuaris i base de dades en temps real.</p> <p>S'han assolit els objectius tècnics de desenvolupar una aplicació que resolgui els requeriments esmentats i també els objectius personals, d'ampliar els coneixements de programació amb un llenguatge frontend com és Angular.</p> <p>Com a conclusió personal, després de molts anys desenvolupant backend, l'enfoc frontend és molt diferent i aporta nous paradigmes als que he hagut d'adaptar-me. La corva d'aprenentatge ha estat dura al principi però ha estat satisfactori poder treballar amb altres esquemes que sortien del meu marc mental.</p>	

Abstract in English:

This final degree project consists of a web application aimed at road assistance companies. Nowadays, when there is an accident or incident on the road, the clients communicate over the telephone line with the operator to indicate the location and to request the assistance services they need. Occasionally, this information arrives incorrectly at the operator and/or the assistance company. Furthermore, in the event of a dispute between insurance companies, being able to obtain images at the time of the accident may be very useful for finding out who has been responsible.

This project is intended to provide a solution to these problems with an application that allows sending alerts to be placed on a map and a chat that enables attached images to be sent to the operator.

To accomplish this project, I have developed an application with Angular framework, using Firebase as a Backend as a Service, which provides users authentication and real time database.

Technical goals have been achieved since this application accomplishes the specific requirements commented above. And for the personal goals, I have widening my programming knowledge with a frontend language such as Angular.

As a personal conclusion, after many years of backend development, the frontend approach is very different and brings us paradigms that I have needed to adapt. The learning curve has been hard at first but it has been satisfying to be able to work with other schemes that come out of my mental frame.

Index

1. Introducció	6
1.1 Context i justificació del treball	6
1.2 Objectius del treball	6
1.3 Enfocament i mètode seguit	8
1.4 Planificació del treball	9
Tasques a realitzar	10
1.5 Resum del producte obtingut	11
1.6 Resta de capítols	11
2. Tecnologies	12
2.1 Tecnologies utilitzades	12
Angular 2+	12
Firebase	13
2.2 Tecnologies relacionades	14
Typescript	14
Bootstrap	14
2.3 Altres eines	15
3. Disseny i prototip	16
4. Arquitectura	22
4.1 Disseny de l'arquitectura del projecte	22
4.2 Estructura projecte Angular	23
4.3 Estructura Firebase Database	24
5. Implementació	27
5.1 Configuració de l'entorn de treball	27
5.2 Autenticació d'usuaris	30
5.3 Alta d'alerta	34
5.4 Llistat d'alertes	35
5.5 Xat	36
5.6 Editar perfil d'usuari	38
6. Pressupost	39
7. Conclusions	40
8. Bibliografia	41
9. Annexos	42
9.1 Accés aplicació	42
9.2 Repositori GIT	42

1. Introducció

1.1 Context i justificació del treball

Actualment les companyies d'assistència en carretera continuen utilitzant la línia telefònica com a mitjà de comunicació amb els seus clients per donar d'alta incidències. En cas d'accident, l'usuari, degut a l'estrés que pot provocar aquesta situació, pot donar indicacions inexactes a la companyia, o bé es poden interpretar malament aquestes indicacions i enviar l'assistència a una ubicació errònia. És per això que és imprescindible disposar de la ubicació exacta per reduir el temps que passa des que succeeix l'accident fins que arriba l'assistència.

Donat que gairebé tothom disposa de telèfon mòbil, i que disposem de la tecnologia, en aquest cas Google Maps, és lògic pensar que podem aprofitar-nos d'aquest fet per crear una aplicació que permeti ubicar el punt de l'accident en un mapa en temps real. A més, en el moment de l'accident es poden captar imatges que poden ser crucials a l'hora d'un possible peritatge en cas de disputa entre les companyies d'assegurança.

El projecte pretén donar solució a aquests requeriments. Per una part, l'usuari pot enviar una alerta geoposicionada, i d'altra banda, pot enviar imatges en temps real mitjançant un xat amb l'operador de la companyia.

1.2 Objectius del treball

L'objectiu principal és fer una aplicació web responsive que permeti als usuaris enviar alertes que s'ubiquin en un mapa i comunicar-se amb l'operador mitjançant un xat que permeti enviar imatges.

Com a objectiu personal, amb aquest treball vull aprendre Angular, ja que m'interessa aprendre un llenguatge de desenvolupament de frontend i també l'ús de Firebase com a backend.

Perfils d'usuaris

Aquesta aplicació és un webapp amb una interfície diferent per a cada perfil d'usuari. Els perfils són l'operador de l'empresa d'assistència i el client.

L'operador de l'empresa disposa de coneixements avançats sobre l'ús de l'aplicació. La seva tasca consisteix a donar resposta a les alertes que es reben i gestionar l'assistència que l'usuari necessita. L'aplicació permetrà a l'usuari sol·licitar ambulància, grúa o policia i també afegir comentaris o enviar fotografies que poden ser útils en cas d'accident.

Els clients poden tenir diferents graus de coneixements sobre l'ús de noves tecnologies, per tant l'eina ha de ser molt intuïtiva i clara. A més, en cas d'accident l'usuari ha de poder sol·licitar assistència de manera ràpida i fàcil. Així doncs, la interfície per als clients disposa d'un únic botó per enviar alertes. Un cop enviada l'alerta, l'aplicació demana algunes dades per completar informació i finalment s'obre un xat amb un operador.

Funcionalitats

L'aplicació ha de permetre als clients dur a terme les següents accions:

1. Donar-se d'alta

El procés d'alta ha de ser fàcil però ha d'incloure la validació de l'e-mail per evitar altes errònies. Es demanaran les dades bàsiques i l'usuari podrà completar el seu perfil un cop hagi validat el seu correu electrònic.

S'utilitzarà Firebase Authentication, l'autenticació es farà mitjançant el correu electrònic de l'usuari i contrasenya.

2. Editar perfil

Un cop validat el correu electrònic, els usuaris podran editar les seves dades bàsiques i completar-les. L'usuari podrà canviar el seu nom, cognom, fotografia de perfil i la matrícula del seu cotxe.

3. Enviar alerta

El webapp permetrà a l'usuari enviar una alerta en un sol click. L'aplicació detectarà la geolocalització de l'usuari i posicionarà l'alerta en el mapa.

S'obrirà una pantalla on l'usuari podrà sol·licitar serveis com l'ambulància, grúa o policia i afegir alguns comentaris. Després s'obrirà un xat per comunicar-se amb l'operador.

4. Llistat d'alertes

L'usuari podrà veure el llistat d'alertes que ha enviat, amb les dades bàsiques de l'alerta i el xat.

5. Xat

L'aplicació permetrà establir una conversa amb l'operador. El xat s'implementarà amb Firestore. El xat permetrà adjuntar imatges, amb previsualització, que poden ser útils en cas d'accident.

Des de la interfície dels operadors, l'aplicació ha de permetre:

1. Veure les alertes

L'aplicació permetrà veure les alertes ubicades en un mapa, també hi haurà un llistat d'alertes. El mapa s'adaptarà a les alertes de manera que es puguin veure totes a la pantalla. Aquest llistat mostrarà qui va enviar l'alerta, on estava ubicada i la data d'inici i fi de l'alerta (en cas que estigui tancada).

2. Xat

Quan l'operador faci click a una alerta se li obrirà el xat amb l'usuari. Al lateral esquerra del xat hi haurà la informació bàsica de l'usuari i els serveis que ha sol·licitat, al lateral dret hi ha un mapa amb la ubicació de l'alerta.

3. Tancar alertes

L'operador podrà tancar les alertes des de llistat d'alertes o bé, des del xat amb l'usuari. Tancar una alerta consisteix a posar-hi data de finalització.

1.3 Enfocament i mètode seguit

Aquest treball final de grau s'ha desenvolupat completament desde zero utilitzant Angular i Firebase. L'estratègia és aprofitar totes les funcionalitats que proporciona Firebase, sobretot per l'autenticació d'usuaris i la base de dades en temps real per al xat. Hi ha llibreries d'Angular i Firebase que faciliten molt la feina.

També s'han utilitzat templates de Bootstrap per a facilitar la tasca de maquetació a fi de fer les pantalles responsive.

S'han assolit tots els objectius desitjables mitjançant aquesta estratègia, i personalment he après Angular, i també maquetació en scss, que era un dels objectius personals que pretenia.

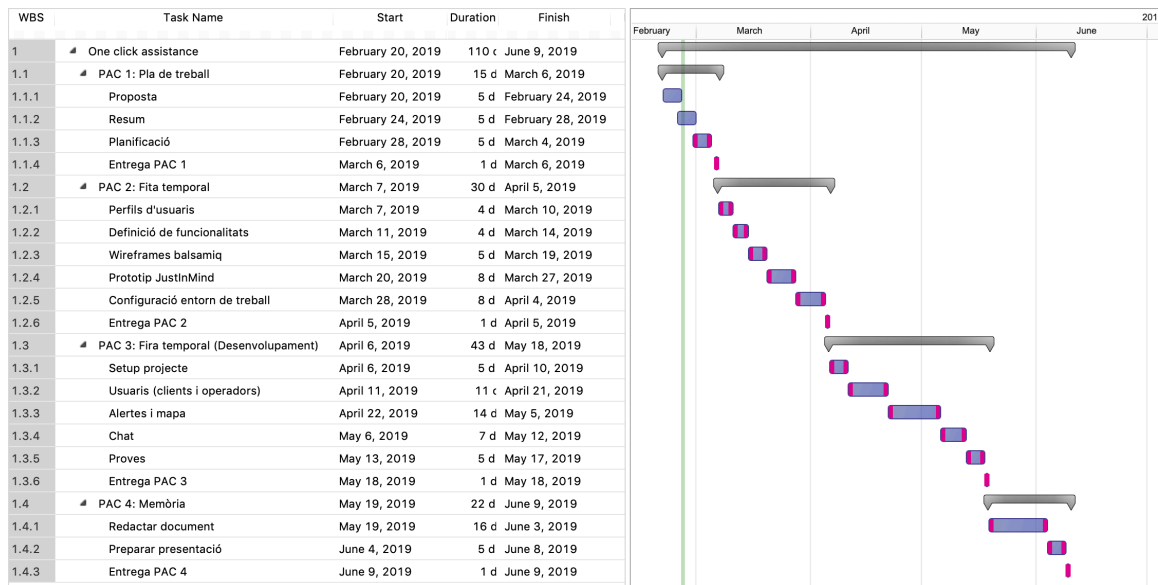
1.4 Planificació del treball

One click assistance	February 20, 2019	110 h	June 9, 2019
PAC 1: Pla de treball	February 20, 2019	15 d	March 6, 2019
Proposta	February 20, 2019	5 d	February 24, 2019
Resum	February 24, 2019	5 d	February 28, 2019
Planificació	February 28, 2019	5 d	March 4, 2019
Entrega PAC 1	March 6, 2019	1 d	March 6, 2019
PAC 2: Fita temporal	March 7, 2019	30 d	April 5, 2019
Perfils d'usuaris	March 7, 2019	4 d	March 10, 2019
Definició de funcionalitats	March 11, 2019	4 d	March 14, 2019
Wireframes bàsic	March 15, 2019	5 d	March 19, 2019
Prototip JustInMind	March 20, 2019	8 d	March 27, 2019
Configuració entorn de treball	March 28, 2019	8 d	April 4, 2019
Entrega PAC 2	April 5, 2019	1 d	April 5, 2019
PAC 3: Fira temporal (Desenvolupament)	April 6, 2019	43 d	May 18, 2019
Setup projecte	April 6, 2019	5 d	April 10, 2019
Usuaris (clients i operacions)	April 11, 2019	11 h	April 21, 2019
Alertes i mapa	April 22, 2019	14 d	May 5, 2019
Chat	May 6, 2019	7 d	May 12, 2019
Proves	May 13, 2019	5 d	May 17, 2019
Entrega PAC 3	May 18, 2019	1 d	May 18, 2019
PAC 4: Memòria	May 19, 2019	22 d	June 9, 2019
Redactar document	May 19, 2019	16 d	June 3, 2019

La planificació inicial s'ha seguit en quant a fites temporals però el càlcul d'hores no ha coincidit amb les hores dedicades finalment. Es va fer un càlcul optimista, intentant avançar les entregues a les dates previstes al calendari de l'assignatura però finalment s'han hagut de dedicar més hores de les esperades i no s'han pogut complir les dates d'entrega.

Possiblement el costum d'adaptar-se a la planificació de l'aula ha fet que aquest intent per assolir les tasques abans de la data d'entrega no s'hagi pogut dur a terme.

Planificació



Tasques a realitzar

- Implementar el registre d'usuaris. Els usuaris hauràn de validar el seu correu electrònic, i tindràn la opció de resetejar la seva contrasenya.
- Establir els diferents tipus d'usuari: client i operador. Els clients visualitzaran una interfície diferent dels operadors.
- Editar perfil d'usuari. Permetrà introduir les dades relatives al client i així com el vehicle assegurat.
- Enviar alertes. Permetrà triar entre els diferents serveis que necessita l'usuari: ambulància, policia i/o grúa. Totes les alertes enviaran les dades bàsiques de l'usuari i obriran un chat amb l'operador.
- Posicionar alertes en el mapa.
- Implementar xat.
- Formulari de dades per les alertes que demanarà si l'usuari necessita ambulància, grúa i/o policia i també una descripció del sinistre. Adicionalment es podrien demanar imatges del sinistre i/o matrícula del cotxe implicat.

1.5 Resum del producte obtingut

Finalment s'ha desenvolupat una aplicació web responsive que compleix els requeriments inicials. L'aplicació permet donar-se d'alta com a usuari, amb el rol *user*, mitjançant correu de verificació. A més, l'aplicació permet als usuaris editar les seves dades i resetejar la contrasenya.

Un cop donat d'alta, l'usuari pot enviar alertes que s'ubiquen en el mapa i un cop enviada l'alerta, s'obre un xat que permet enviar imatges amb previsualització. També hi ha un llistat amb l'històric de les alertes on es pot veure també el xat de cada alerta.

L'usuari *admin* pot veure les alertes que s'han rebut ubicades en un mapa. Quan l'usuari *admin* fa click sobre l'alerta, s'obre el xat amb l'usuari, i es mostren les dades de l'alerta, hora, dades d'usuari, serveis sol·licitats, observacions i ubicació. Tant aquesta pantalla com les opcions del menú varien en funció del rol de l'usuari. Per l'usuari *admin* apareixen més opcions ja que es considera que aquest usuari utilitza l'aplicació des d'un ordinador d'escriptori o portàtil, i en canvi l'usuari amb rol *user* pot utilitzar l'aplicació des del mòbil, amb la qual cosa es mostren menys dades des d'aquest perfil d'usuari.

Del plantejament inicial, l'únic que no s'ha resolt és el fet de poder donar d'alta l'usuari amb el rol *admin*. Oferir una interfície per poder fer-ho seria un problema de seguretat, i en aquest moment s'ha d'establir l'usuari *admin* des de Firebase, editant el rol de l'usuari.

D'altra banda, tampoc s'ha permès que els usuaris amb rol *user* puguin tancar l'alerta com s'havia plantejat inicialment. Només es permet als usuaris *admin* poder tancar alertes.

1.6 Resta de capítols

En aquest treball final de grau s'expliquen les tecnologies que s'han utilitzat per desenvolupar el projecte, el disseny i prototip, l'arquitectura, la implementació i les conclusions.

Les tecnologies en les que s'aprofundeixen són principalment Angular i Firebase, i totes les eines que proporciona Firebase que s'han utilitzat. També s'anomenen algunes eines bàsiques com, per exemple, per la maquetació Bootstrap i Git per control de versions, etc.

Es mostra també el disseny i prototip inicial, fet amb JustInMind. Bàsicament són captures de pantalla indicant quines són les funcionalitats que satisfà cadascuna d'elles.

En l'apartat d'arquitectura es poden veure els diagrames amb els diferents components de l'aplicació, i també el disseny de la base de dades de Cloud Firestore.

A continuació s'analitzen les parts de codi que considero més rellevants i destacables, sobretot les que fan referència als objectes d'Angular *Observable* i *Promise*, i també l'asincronicitat dels mètodes.

Finalment s'aporten les conclusions personals després d'haver treballat en aquest projecte, i també s'exposen algunes idees sobre possibles altres usos de l'aplicació o ampliacions que podrien aportar-li valor al projecte.

2. Tecnologies

Darrerament el desenvolupament web ha anat evolucionant cap a aplicacions client, deixant enrera les aplicacions servidor que es desenvolupaven fins fa pocs anys. L'enfoc de desenvolupar interfícies d'usuari responsive que consumeixin una API s'ha estés de manera que llenguatges com Angular s'han posat en primera línia a l'hora de triar llenguatge pel frontend.

D'altra banda, la implementació d'un xat estableix un requeriment bàsic per al seu desenvolupament que consisteix a emmagatzemar els missatges en una base de dades en temps real. Aquest servei l'ofereix Firebase de Google, amb l'eina Cloud Firestore. A més, Firebase també ofereix el servei d'autenticació. Per tant, Firebase com a Backend as a Service (Baas) i Angular són una bona combinació per al desenvolupament d'una aplicació d'aquestes característiques.

2.1 Tecnologies utilitzades

Angular 2+

Angular és un framework per aplicacions web desenvolupat en *TypeScript*, de codi obert i mantingut per Google que s'utilitza per crear i mantenir aplicacions web. Angular combina els templates d'HTML amb els Components, és a dir, enllaça els atributs dels components per presentar les dades a la vista.

Angular disposa d'una eina, Angular CLI, que permet crear components mitjançant la línia de comandes. Un dels avantatges d'Angular és precisament l'ús dels components ja que aquests poden rebre i retornar variables i això facilita la reutilització del codi.

Les llibreries més destacades que s'han utilitzat en aquest projecte són les d'Angular Firebase i també Google Maps component (Agm).

Firebase

Firebase és una plataforma per al desenvolupament d'aplicacions web i mòbil que està integrada amb Google Cloud Platform. Firebase es troba dins dels BaaS (Backend as a service), que proporciona eines per als desenvolupadors com l'autenticació, base de dades en temps real, i hosting, així com analytics i notifikacions push.

Firebase té una versió gratuïta i també té versió de pagament. En aquest projecte s'ha configurat la versió de pagament i s'han utilitzat les eines de base de dades en temps real, autenticació, storage i hosting.

Angular té llibreries per Firebase dins del package Angular/fire. A continuació es fa un resum de l'ús que se n'ha fet de cadascuna de les llibreries.

Firebase Auth

Firebase Auth s'ha utilitzat per a donar d'alta usuaris, autenticar-los, verificar el correu electrònic, restablir la contrasenya i per sortir de l'aplicació. Per a poder afegir dades sobre els usuaris, s'ha hagut de crear una taula users a la base de dades de Firebase i a l'hora de donar-se l'alta cal implementar un codi per a poder actualitzar les dades que es guarden de l'usuari a Firebase.

Dels usuaris es guarden el nom, l'e-mail, la foto, la matrícula i el seu rol. (Només es donen d'alta usuaris amb role 'user' ja que altrament qualsevol usuari es podria donar d'alta com "admin").

Firebase Cloud Firestore

Les llibreries d'Angular que treballen amb Firebase permeten l'actualització en temps real del contingut de la pàgina sense necessitat de recarregar-la. Els objectes que retorna les crides als mètodes de la llibreria d'AngularFirestore són de tipus *Observable*. Aquests

objectes permeten “subscriure’s” als canvis que es produeixen a la col·lecció de dades de Firebase en temps real, i mostrar-los en temps real a l’HTML.

Firestore hosting

Firestore te un servei de hosting de contingut web orientat a programadors. Permet fer deploy en una sola comanda i l’aplicació s’allotja sota el subdomini [firebaseapp.com](https://one-click-1554500008958.firebaseio.com).

L’aplicació web desenvolupada per aquest projecte final de grau es pot trobar al següent enllaç:

<https://one-click-1554500008958.firebaseio.com>

2.2 Tecnologies relacionades

Typescript

Typescript és un llenguatge de programació de codi obert, desenvolupat per Microsoft. Typescript és una evolució de Javascript que incorpora el llenguatge tipat i els objectes basats en classes. És una extensió de Javascript, per tant és compatible amb qualsevol codi Javascript preexistent. Quan el codi Typescript es compila es converteix en codi Javascript. Aquest llenguatge es pot utilitzar en aplicacions web i servidor.

Els tipus de variables que admet Typescript són, String, Number, Boolean, Array, Tuple, Enum, Any, Void i Never.

Bootstrap

En aquest projecte s’ha utilitzat Bootstrap per a la maquetació de les pàgines. Bootstrap és un framework CSS que inclou la combinació d’HTML i CSS per a la creació de pàgines web de manera ràpida i fàcil. De fet, Bootstrap 4 utilitza Sass i els seus fitxers tenen l’extensió .scss. A més, bootstrap permet implementar pàgines responsive amb el sistema grid, basat en columnes, que fa les pàgines s’adaptin a qualsevol pantalla, fins i tot mòbil.

Bootstrap proporciona plantilles o templates per a crear formularis, botons, menús, barres de navegació i altres elements com llistes, etc..

2.3 Altres eines

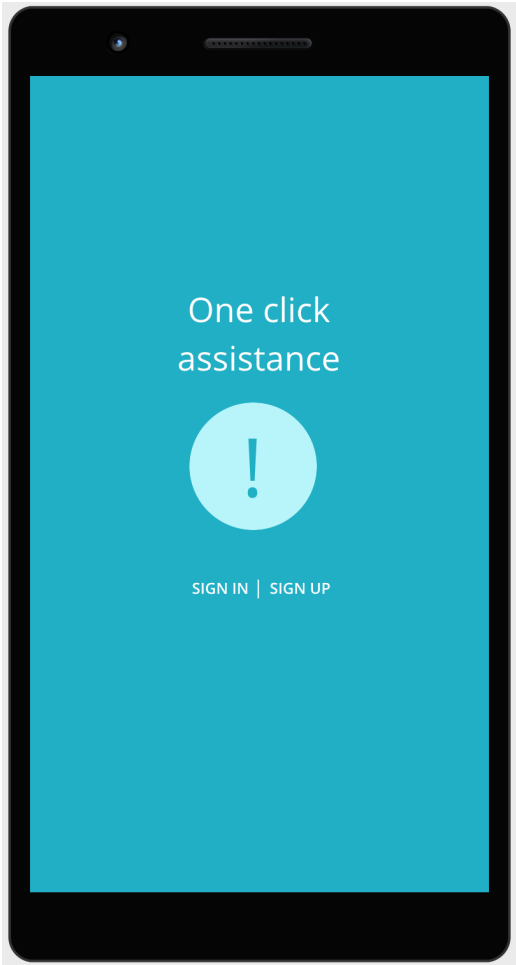
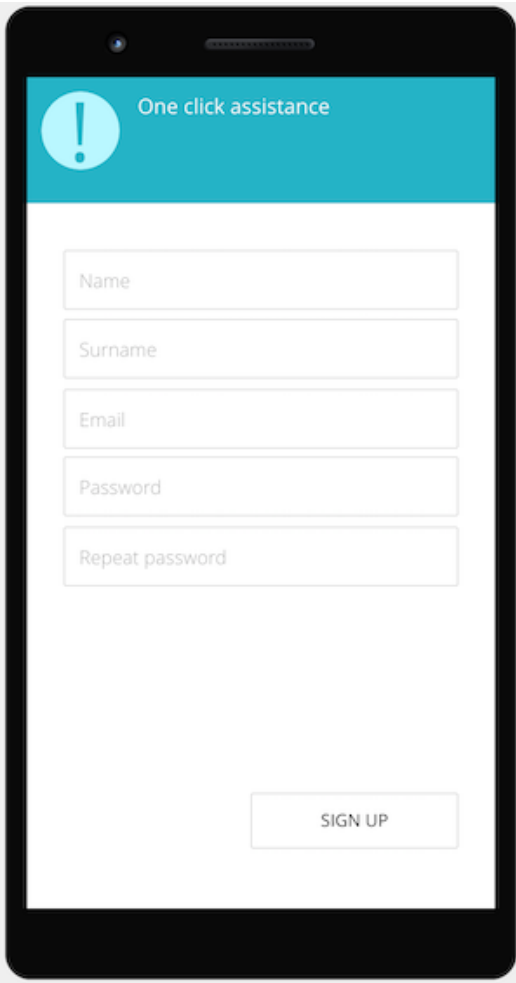
npm

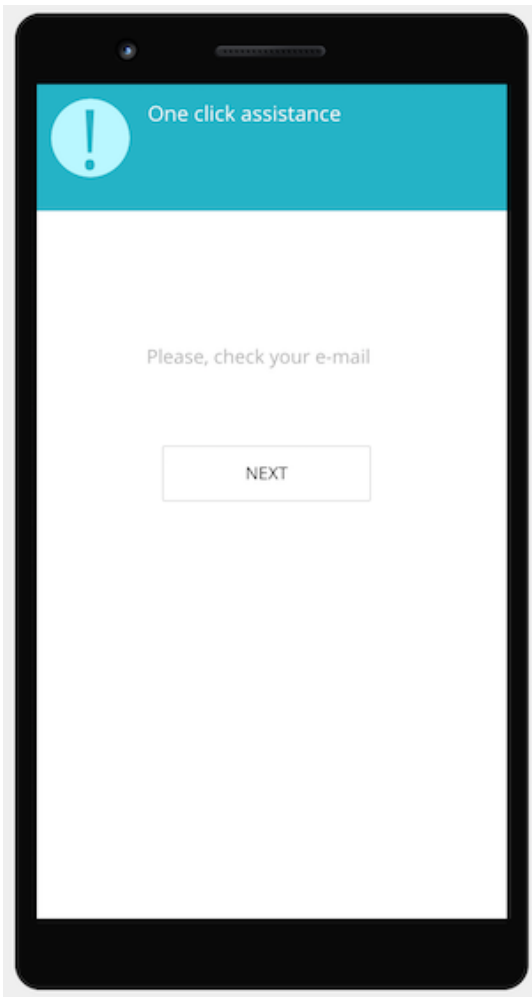
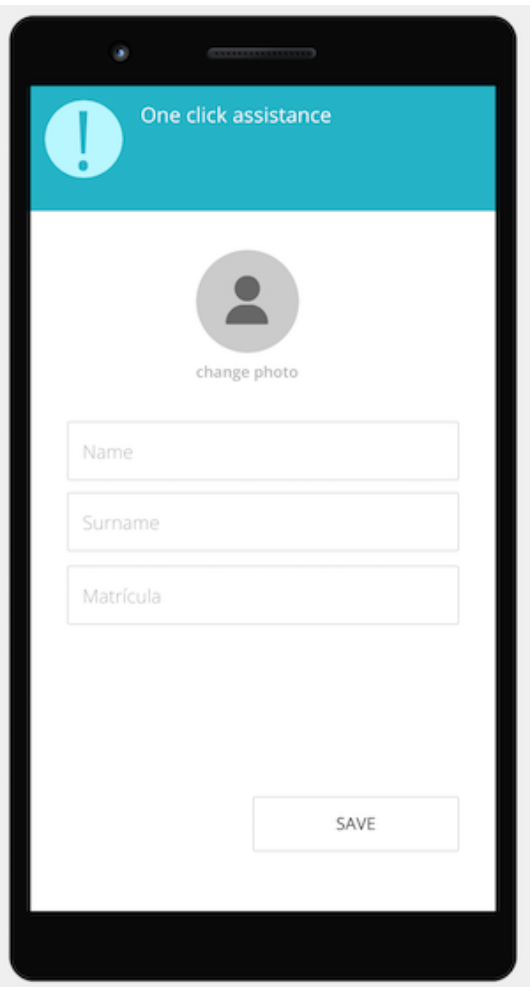
npm és el gestor de paquets de node.js. Cal instal·lar primer node.js i npm per poder instal·lar Angular cli.

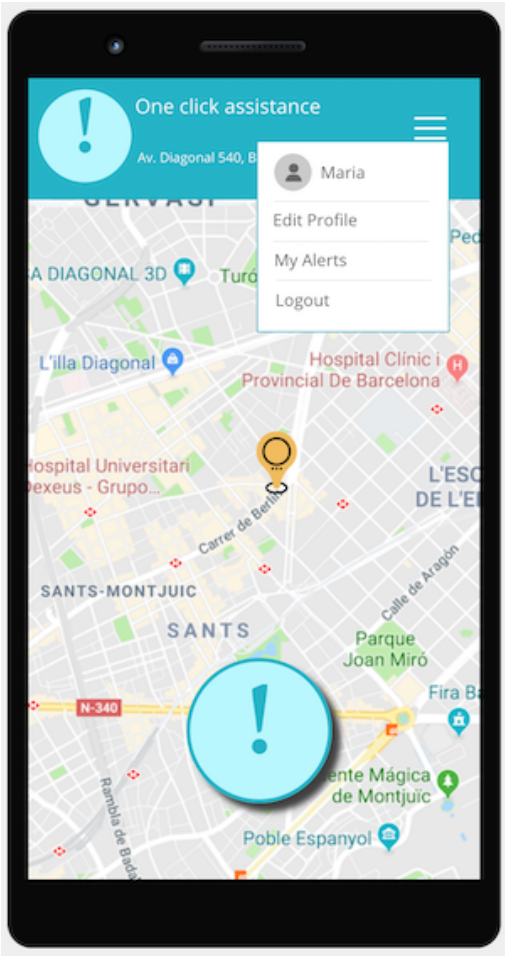
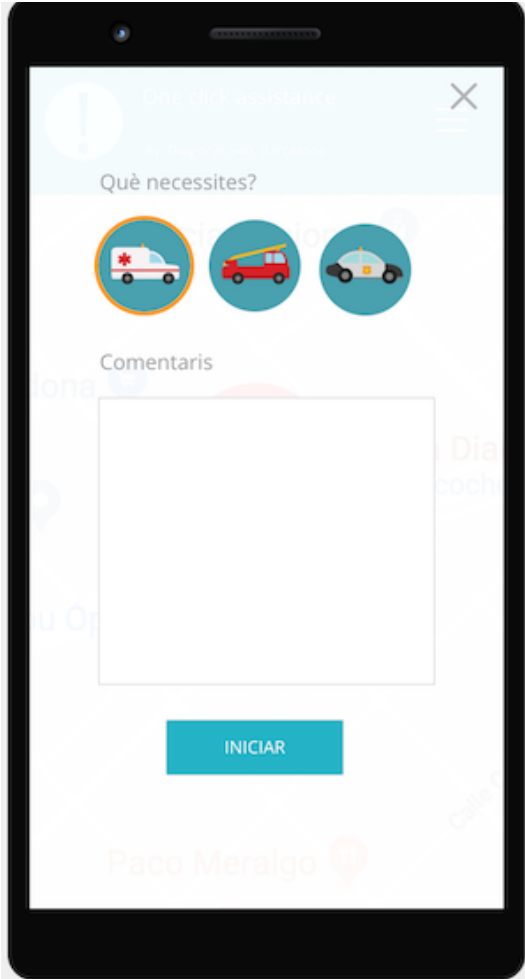
Git

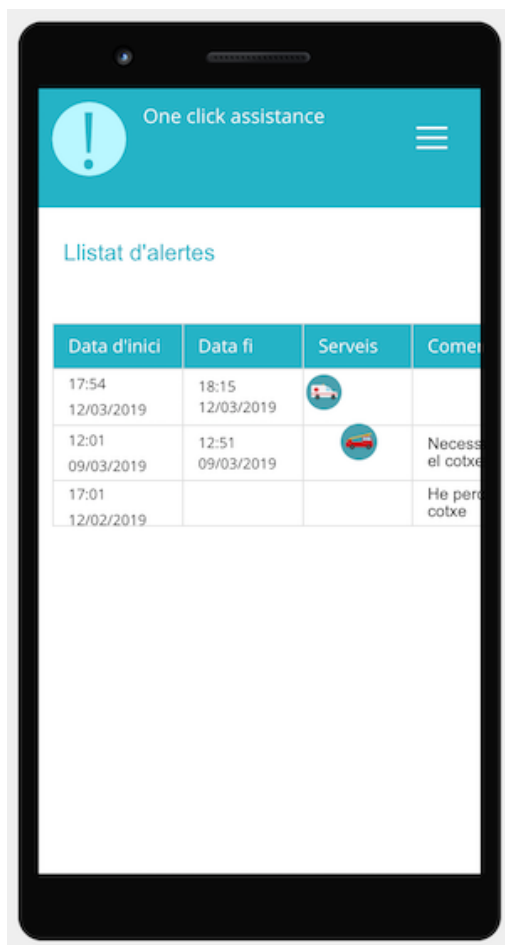
Git és un software de control de versions que permet portar un registre dels canvis que es van fent en el codi. L'ús habitual de Git és per compartir canvis entre programadors i fer merge de les aportacions de cadascú. En aquest projecte s'ha utilitzat només per tenir històric de canvis i poder tornar a una versió anterior en cas que hi hagués algun error en el codi.

3. Disseny i prototip

	
Pantalla inicial	Sign up
<p>Des d'aquesta pantalla l'usuari pot fer login o donar-se d'alta.</p>	<p>Des d'aquesta pantalla l'usuari pot donar-se d'alta a l'aplicació. Les dades que es demanen en el formulari d'alta són:</p> <ul style="list-style-type: none"> - Nom - Cognom - Email - Contrasenya

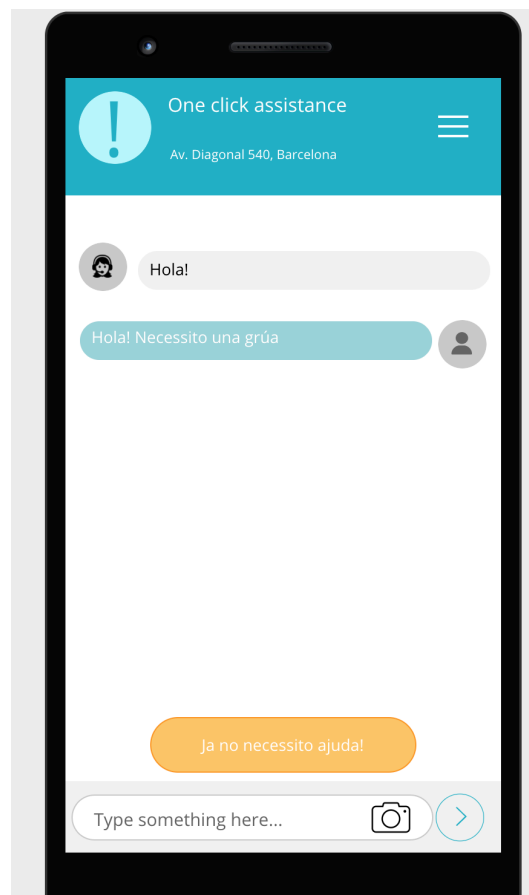
	
<p>Confirmation</p> <p>Quan l'usuari prem el botó "Sign up", l'aplicació envia un correu electrònic a l'usuari amb un link per confirmar l'alta. Si l'usuari prem el botó "Next" sense haver validat el compte, l'aplicació li mostra un missatge d'error.</p>	<p>Edit profile</p> <p>L'usuari pot editar la seva foto de perfil, nom, cognoms i matrícula del vehicle que té assegurat.</p>

 <p>The screenshot shows the 'One click assistance' app interface. At the top, there's a header with a blue background and a white exclamation mark icon. Below the header, the text 'One click assistance' is displayed. A menu overlay is visible on the right side, showing the user's name 'Maria' and options: 'Edit Profile', 'My Alerts', and 'Logout'. The background is a map of Barcelona with a location pin and a large blue exclamation mark icon.</p>	 <p>The screenshot shows the 'One click assistance' app interface. At the top, there's a header with a blue background and a white exclamation mark icon. Below the header, the text 'One click assistance' is displayed. The main screen is titled 'Què necessites?' (What do you need?) and features three circular icons representing different services: a medical service (ambulance), a fire service (fire truck), and a police service (police car). Below the icons, there's a section titled 'Comentaris' (Comments) with a large text input area. At the bottom, there's a blue button labeled 'INICIAR' (Start).</p>
<p>Map</p> <p>Des d'aquesta pantalla l'usuari pot enviar una alerta. També es pot accedir al menú d'opcions.</p> <p>El menú permet accedir a l'edició del perfil, al llistat d'alertes i sortir de l'aplicació.</p>	<p>New alert</p> <p>Quan l'usuari envia una alerta, l'aplicació li ofereix la possibilitat de seleccionar els serveis que necessita per ser atès. També permet afegir comentaris.</p>



Llistat d'alertes

L'usuari podrà consultar el seu llistat d'alertes amb la data d'inici i fi, els serveis sol·licitats i els comentaris, així com també el xat de cada alerta.

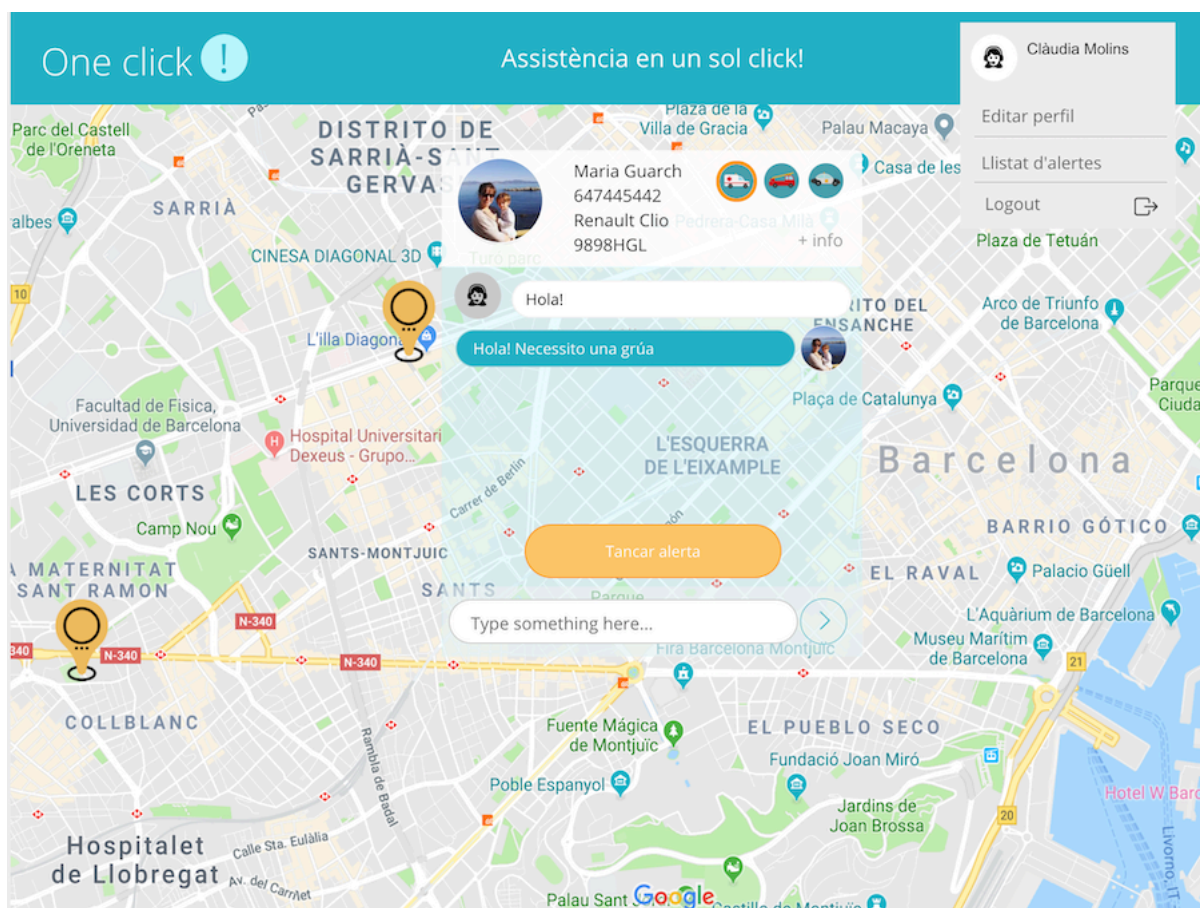


Xat

El xat permetrà a l'usuari enviar imatges de l'accident. A més, des d'aquesta pantalla l'usuari podrà tancar l'alerta en cas d'haver rebut l'assistència que havia sol·licitat.

Mapa operador

Les alertes es mostraran en un mapa i quan l'usuari fa click sobre l'alerta, s'obre un xat des d'on pot veure la informació sobre l'alerta, pot enviar missatges a l'usuari i també tancar l'alerta.



Llistat d'alertes

Des del llistat d'alertes, l'operador podrà visualitzar alertes que ja no apareixen en el mapa perquè han estat tancades. També podrà tancar les alertes que es trobin en estat obert.

One click !

Assistència en un sol click!



Clàudia Molins

Editar perfil

Llistat d'alertes

Logout

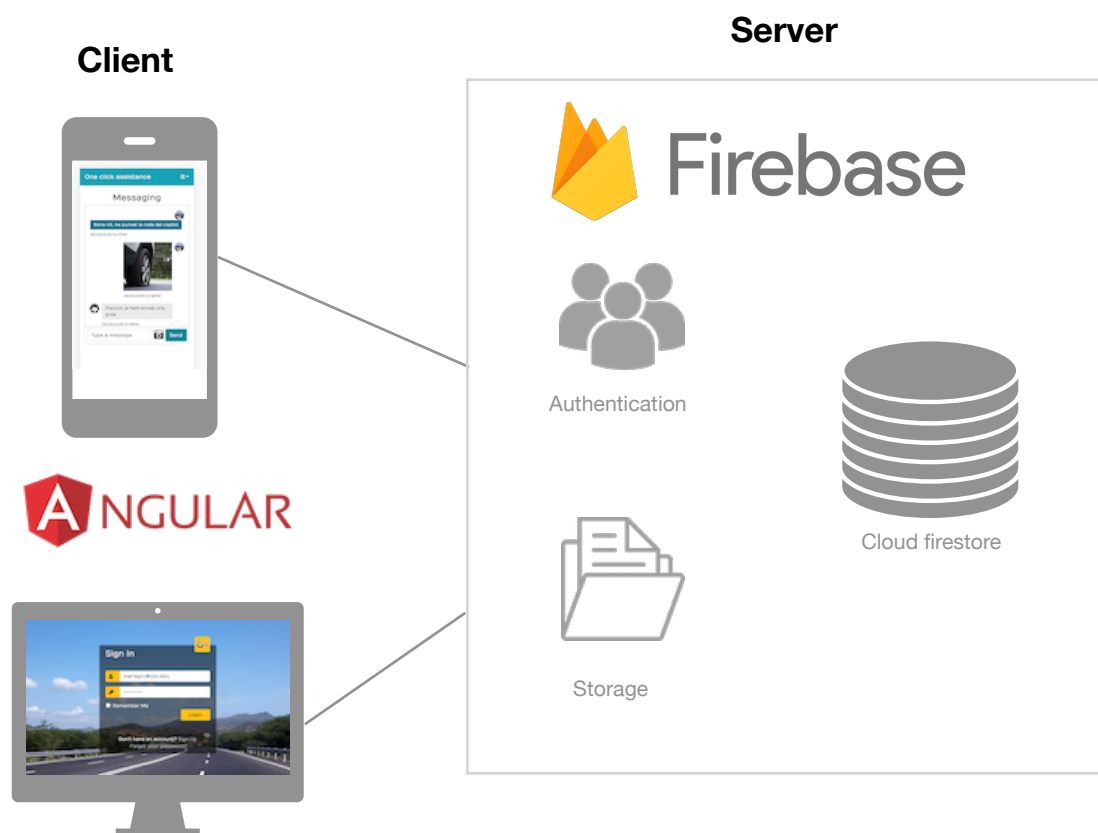
Vista mapa

ID Alerta	Usuari	Serveis	Comentaris	Data d'inici	Data fi	Estat	Accions
1	Maria Guarch			18:32 12/03/2019		Oberta	TANCAR
2	Xavier Garcia		Necessito una grúa perquè el cotxe no te bateria	10:11 09/03/2019	11:32 09/03/2019	Tancada	
3	Clàudia Martí		He perdut les claus del cotxe	12:35 07/03/2019	12:59 07/03/2019	Tancada	

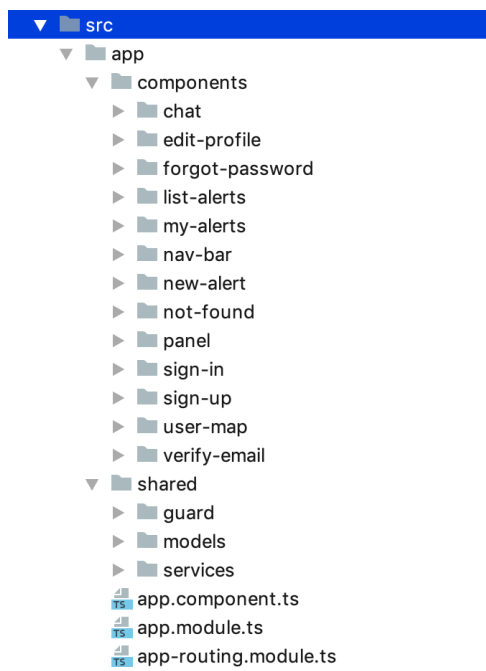
4. Arquitectura

El model d'arquitectura d'aquest projecte és bàsicament una aplicació client desenvolupada amb Angular que fa ús d'un Backend as a Service (BaaS), en aquest cas, Firebase. Es tracta doncs d'una arquitectura de dues capes, on l'aplicació client i Firebase manipulen les dades de forma directa.

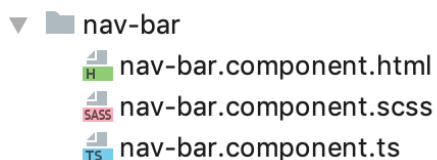
4.1 Disseny de l'arquitectura del projecte



4.2 Estructura projecte Angular



L'aplicació Angular es divideix en components, i dins de cada component trobem tres arxius.



En aquest exemple, trobem el fitxer `nav-bar.component.ts` que conté el codi TypeScript, en el fitxer `nav-bar.component.scss` hi trobem la fulla d'estil amb el codi scss i al fitxer `nav-bar.component.html` hi trobem el codi HTML amb les etiquetes html i les variables vinculades al component.

A més, s'ha creat una carpeta *shared* on es troben els serveis, *services*, que comparteixen molts dels components com són l'autenticació, el xat, o l'upload d'imatges.

Dins també de la carpeta *shared*, hi trobem els guards que permeten protegir l'accés a les pàgines segons si l'usuari està autenticat o no, o segons el seu rol. Dins les rutes de l'aplicació, en el fitxer `app-routing.module.ts`, s'especifica el guard que es vol fer servir per una determinada ruta. Per exemple:

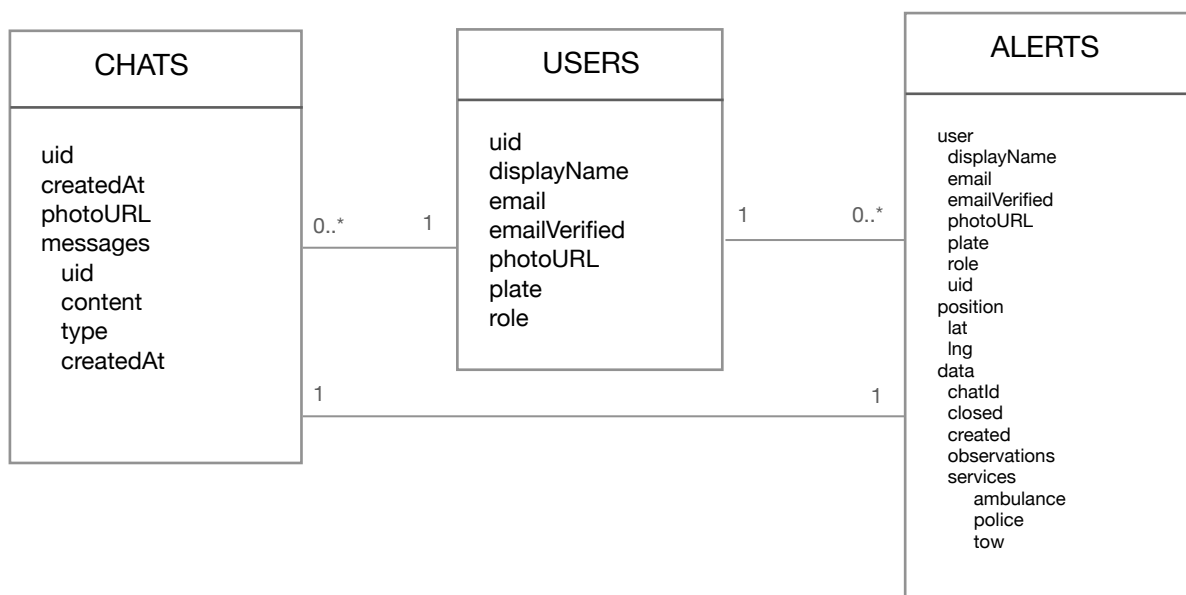
```
{ path: 'list-alerts', component: ListAlertsComponent, canActivate:
[AuthGuard]}
```

Aquesta instrucció significa que només pot accedir al component list-alerts l'usuari Admin.

Finalment, el fitxer app.module.ts és on s'especifiquen totes les dependències de les llibreries, i on s'han d'afegir tots els import que calguin. Les variables d'entorn es troben al fitxer environment.ts, hi ha un fitxer environment per cada entorn de desenvolupament (producció o testing).

4.3 Estructura Firebase Database

S'han creat les taules: users, alerts i chats. L'arquitectura de la base de dades Cloud Firestore no segueix l'estructura d'una base de dades relacional ja que es tracta d'una base de dades NoSQL i no s'estableixen relacions. Si bé, conceptualment, l'esquema entitat relació correspon amb el model que es mostra a continuació. Per exemple, en un model de base de dades relacional, no es guarden les dades d'usuari dins d'una altra taula ja que només amb l'identificador, fent una consulta SQL es pot fer join de totes les taules i agafar les dades de l'usuari. En canvi, en aquest model es guarden les dades d'usuari en el document d'alerta, ja que el concepte és diferent i no es pot fer una consulta per crear les dades.



Users

L'estructura de dades que es guarda per als usuaris és la següent:

```
user:
  uid
  displayName
  email
  emailVerified
  photoURL
  plate
  role
```

L'aplicació permet editar les dades dels usuaris. El fet que el component d'Angular tingui vinculat un objecte Observable a la vista, fa que quan es canvien les dades a la base de dades de Firebase, automàticament s'actualitzin les dades, per exemple, de la barra de navegació, on hi ha el displayName de l'usuari.

Alertes

L'estructura de les alertes segueix el següent esquema:

```
data:
  chatId
  created
  closed
  observations
  services { ambulance, police, tow}
position:
  lat
  lng
user:
  uid
  displayName
  email
  emailVerified
  photoURL
  plate
  role
```

El llistat d'alertes s'ordena per data de creació descendent i id d'usuari, per tant, s'ha hagut de crear un índex a tal efecte.

ID de colección	Campos indexados	Alcance de la consulta	Estado
alerts	user.uid Ascendente data.created Descendente	Colección	Habilitado

Xats

L'estructura que es guarda a la base de dades pels xats és la següent:

Chat

```
uid
photoUrl
createdAt
messages:
  content
  createdAt
  type
  uid
```

El xat permet enviar imatges, per això s'ha creat el camp type per definir si un missatge és de tipus text o imatge. Així, a la vista HTML es diferencia entre un i altre. L'enviament d'imatges s'ha implementat en un preview que permet previsualitzar la imatge i esborrar-la abans d'enviar-la.

5. Implementació

A continuació es detallen algunes de les particularitats més destacables d'aquest treball, i de la seva implementació. S'explica breument la configuració de l'entorn de treball i s'entra en detall d'alguns dels components que han requerit un estudi més en profunditat del llenguatge de desenvolupament i també de les llibreries d'angular i firebase.

5.1 Configuració de l'entorn de treball

Aquests són els passos que cal seguir per a configurar l'entorn de treball segons la documentació d'Angular Documentation[1].

En primer lloc, cal instal·lar node i npm

```
brew install npm  
brew install node
```

Podem comprovar la versió instal·lada amb la comanda següent:

```
node -v
```

A continuació, cal instal·lar angular cli.

```
npm install -g @angular/cli
```

Crear el projecte

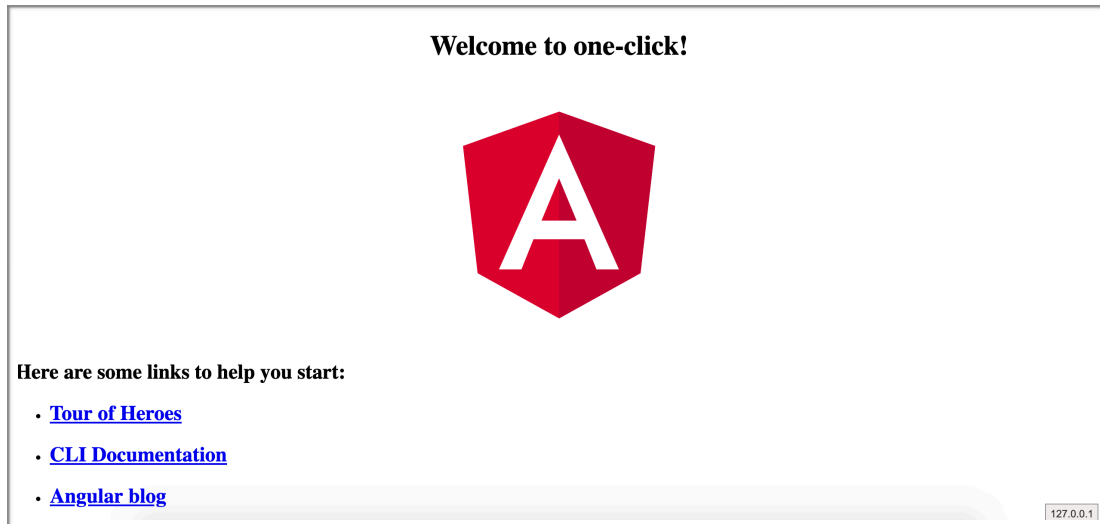
```
ng new one-click
```

```
[maria@mguarch:~/Sites$ ng new one-click  
[? Would you like to add Angular routing? Yes  
? Which stylesheet format would you like to use? SCSS [ http://sass-lang.com/d  
ocumentation/file.SASS_REFERENCE.html#syntax ]  
CREATE one-click/README.md (1025 bytes)  
CREATE one-click/.editorconfig (246 bytes)  
CREATE one-click/.gitignore (629 bytes)  
CREATE one-click/angular.json (3922 bytes)  
CREATE one-click/package.json (1308 bytes)  
CREATE one-click/tsconfig.json (435 bytes)  
CREATE one-click/tslint.json (1621 bytes)  
CREATE one-click/src/favicon.ico (5430 bytes)
```

Un cop creat el projecte, cal iniciar el servidor, que escoltarà al port 42000 per defecte

```
ng serve
```

Podem comprovar que el projecte s'ha creat correctament a la URL <http://localhost:4200>



A continuació es detallen alguns paquets necessaris per al desenvolupament del projecte:

```
npm install firebase @angular/fire --save
```

Caldrà instal·lar Angular Google Maps[2], per a fer-ho cal executar aquesta comanda:

```
npm install @agm/core --save
```

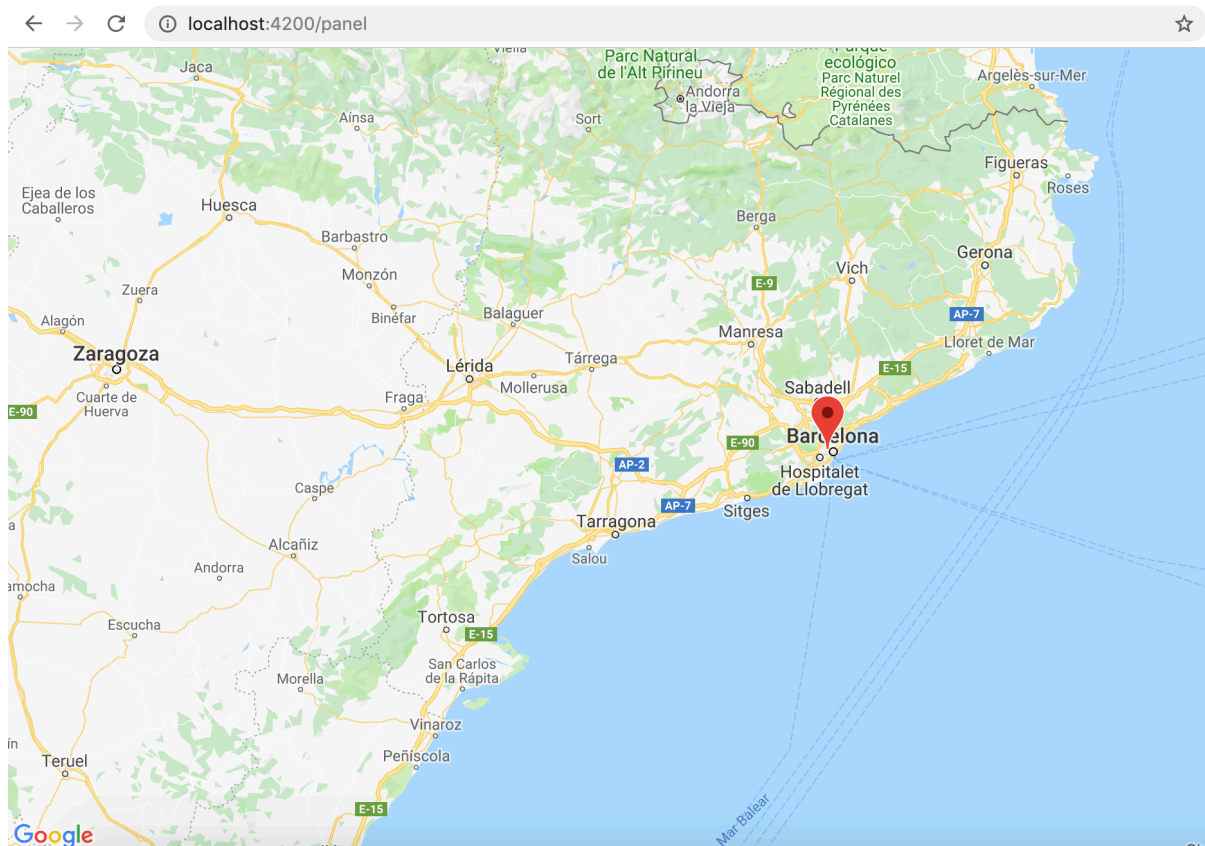
Per a fer funcionar Angular Google Maps, necessitarem una apiKey vàlida. Per obtenir-la cal seguir els passos del següent enllaç:

<https://developers.google.com/maps/documentation/javascript/get-api-key?hl=en#key>

/app/app.module.ts

```
@NgModule({
  imports: [AgmCoreModule.forRoot({
    apiKey: 'AIzaSyBFQHL50RXDI_R5m3i6sbxAdZbBpFQTMc4'
  })
])
```


Amb aquesta configuració, ja podem veure el mapa.



Afegir Firebase al projecte

Per afegir Firebase al projecte cal insertar aquest codi en el document `environment.ts` que conté les variables de configuració[5].

```
firebase: {
  apiKey: "AIzaSyBqXm1GoV24qT9CyixJmuv2cRpT2XE92jU",
  authDomain: "one-click-1554500008958.firebaseio.com",
  databaseURL: "https://one-click-1554500008958.firebaseio.com",
  projectId: "one-click-1554500008958",
  storageBucket: "one-click-1554500008958.appspot.com",
  messagingSenderId: "456653242268",
  appId: "1:456653242268:web:88350ac9a249b575"
}
```

5.2 Autenticació d'usuaris


S'ha utilitzat Firebase Authentication per a l'autenticació d'usuaris. S'ha creat un servei dins de *shared/services* anomenat *auth.service.ts*. Aquest servei gestiona l'alta, el login i logout de l'aplicació.

A més, s'han implementat els guards, dins de *shared/guard*, per al control d'accés a l'aplicació, en funció de si l'usuari està loguejat o no, i segons el seu rol pot accedir a unes pantalles o unes altres.

Per implementar l'autenticació d'usuaris s'ha seguit una de les lliçons de Firebase OAuth Login With Custom Firestore User Data[3] i Full Angular 7 Firebase Authentication System[4].

Sign up

S'ha implementat l'alta d'usuaris amb verificació de correu electrònic. Aquest servei el proporciona Firebase. Per dur a terme l'alta mitjançant correu electrònic i contrasenya amb correu de verificació, cal activar a Firebase aquest mètode d'accés.

 Correo electrónico/contraseña

Habilitar ☒

Permite a los usuarios registrarse con su dirección de correo electrónico y contraseña. Nuestros SDK también proporcionan la verificación de la dirección de correo, la recuperación de contraseñas y las primitivas de cambio de dirección de correo. [Más información](#)

Vínculo del correo electrónico (acceso sin contraseña)

Habilitar ☐

Cancelar

Guardar

La llibreria Firebase Auth proporciona el mètode *createUserWithEmailAndPassword()*, un cop creat, en el retorn de la crida, s'envia l'e-mail de verificació amb la crida al mètode *this.afAuth.auth.currentUser.sendEmailVerification()*.

Quan es crea un usuari nou, cal crear també el document dins la taula de *users* del Cloud Firestore, on guardem altres dades com per exemple, la matrícula. Per poder guardar i vincular aquestes dades amb l'usuari de Firebase, cal fer merge de les dades de l'usuari a la taula *users* i el de la pròpia autenticació de Firebase.

Login

En el codi següent es mostra un exemple d'un component, en aquest cas, Sign in, que vincula l'atribut userData, que conté email i password, amb el formulari de la vista de login.

```
import {Component, OnInit} from '@angular/core';
import {AuthService} from "../../shared/services/auth.service";

@Component({
  selector: 'app-sign-in',
  templateUrl: './sign-in.component.html',
  styleUrls: ['./sign-in.component.scss']
})
export class SignInComponent implements OnInit {

  public userData = {
    email: null,
    password: null
  };

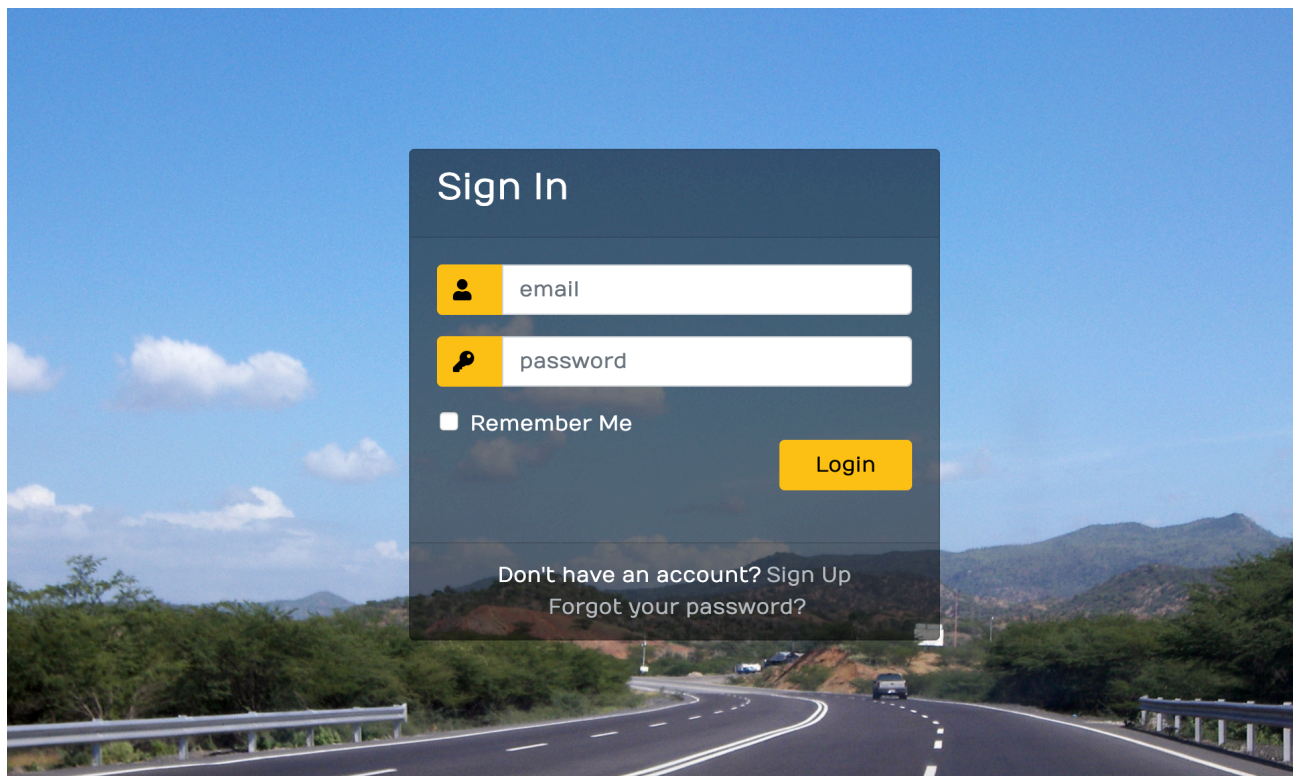
  constructor(public authService: AuthService) {
  }

  ngOnInit() {
  }
}
```

En les etiquetes del codi HTML es vinculen les variables del component, utilitzant l'etiqueta NgModel.

```
<input [(ngModel)]="userData.email" type="text" name="email"
class="form-control" placeholder="email">

</div>
<div class="input-group form-group">
  <div class="input-group-prepend">
    <span class="input-group-text"><i class="fas fa-key"></i></span>
  </div>
  <input [(ngModel)]="userData.password" type="password" name="password"
class="form-control" placeholder="password">
</div>
```



Email confirmation

Un cop donat d'alta, l'usuari rep un e-mail de verificació amb un enllaç, i un cop verificat, s'actualitza el camp *email/Verified* del usuari i l'usuari ja pot loguejar-se.

Firebase permet editar la plantilla de l'e-mail de verificació des de la consola, dins de *Authentication -> Plantillas -> Verificación de correo electrónico*.

Forgot password

En cas que l'usuari no recordi la seva contrasenya pot recuperar-la fent click al link *Forgot your password?*

S'obre un formulari on l'usuari ha d'introduir el seu correu electrònic, on s'enviarà un link per a restablir la seva contrasenya. Aquest link porta a un altre formulari on introduir la nova contrasenya.

Reset your password

for **mariagrc@uoc.edu**

New password



SAVE

Aquest procés forma part del servei d'autenticació de Firebase. Per invocar aquest mètode només cal cridar el mètode *sendPasswordResetEmail()* de l'objecte *FirebaseAuth*.

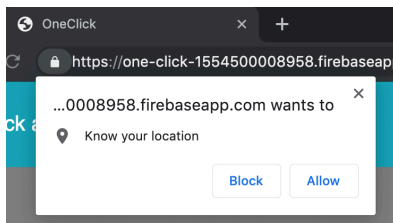
Sign out

El mètode *signOut()* també està implementat en la llibreria de *Firebase Auth*, l'única particularitat que cal destacar és que degut a l'asincronicitat, cal executar qualsevol altre instrucció posterior dins del block *then* per a que aquesta tingui efecte. Es mostra l'exemple a continuació:

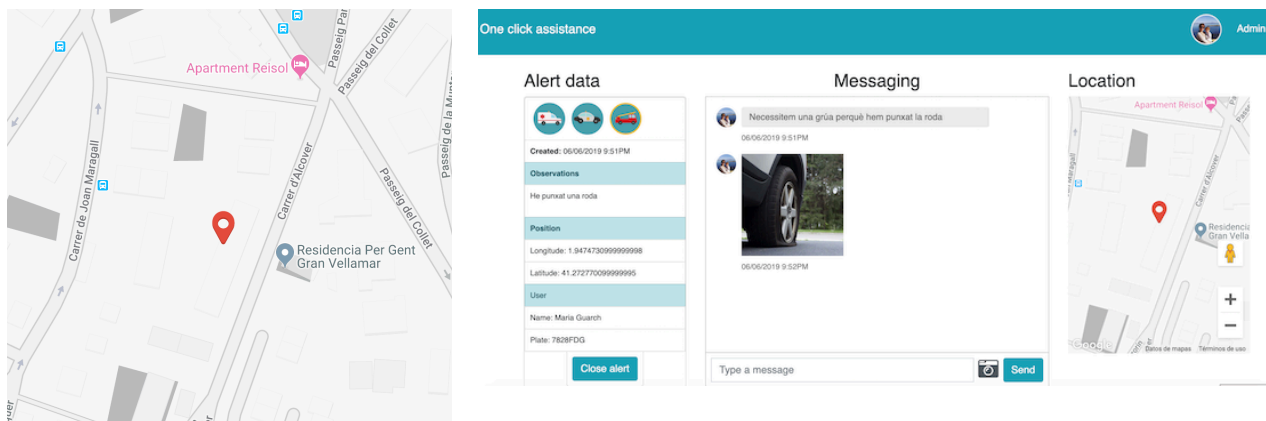
```
this.afAuth.auth.signOut().then(() => {  
  this.router.navigate(['sign-in']);  
});
```

5.3 Alta d'alerta

La primera vegada que l'usuari accedeix a l'aplicació, el navegador li demana permisos per obtenir la seva ubicació.















En donar permisos, el mapa es centra en la ubicació de l'usuari. Per fer una nova alerta només cal fer click al botó centrat en el mapa i s'obre un formulari que permet introduir comentaris en una caixa de text, i també triar els serveis que l'usuari necessita. A continuació, s'obre un xat. En aquest moment, a l'usuari administrador li apareix una alerta en el mapa. En fer click al marcador, s'obre la pantalla amb el xat i les dades de l'alerta.



El codi necessari per mostrar les alertes en el mapa és el següent:

```
<agm-map
  (mapReady)="setUserMarker( $event )"
  [maxZoom]="18"
  [fitBounds]="true"
  [latitude]="this.lat"
  [longitude]="this.lng"
  [styles]="customStyle" #agmMap>
  <agm-marker *ngFor="let alert of alerts"
    (markerClick)="openChat( alert )"
    [iconUrl]="icon"
    [latitude]="alert.position.lat"
    [longitude]="alert.position.lng" [agmFitBounds]="true">
  </agm-marker>
</agm-map>
```

5.4 Llistat d'alertes

One click assistance Admin ▾							
ID	User	Services	Observations	Start date	End date	Status	Actions
xPxUUh...	Maria Guarch	  	He punxat una roda	29/05/2019 10:16PM		Open	view close
KYd2RB...	Maria Guarch	  	Hem tingut un accident	29/05/2019 10:16PM		Open	view close
yWOkzg...	Maria Guarch	  	Test	29/05/2019 1:35AM	29/05/2019 10:23PM	Closed	view
BfMK7Q...	Maria Guarch	  	Test	29/05/2019 1:35AM	29/05/2019 10:23PM	Closed	view

Per obtenir un llistat d'una col·lecció de Cloud Firestore, cal un objecte del tipus *AngularFirestoreCollection*.

Declarem l'objecte en el component:

```
alertsCollection: AngularFirestoreCollection<any>;
```

I en el mètode *getAlerts()* obtenim la col·lecció "alerts".

```
this.alertsCollection = this.db.collection<any>(
  'alerts', ref => ref.orderBy('data.created', 'desc')
);
```

Després cal assignar l'objecte *Observable* que tenim mapejat amb la vista el llistat que retorna el collection.

```
this.alerts = this.alertsCollection.snapshotChanges().pipe(
  map(alerts => {
    return alerts.map(a => {
      const data = a.payload.doc.data();
      const id = a.payload.doc.id;
      return { id, ...data };
    });
  }));
```

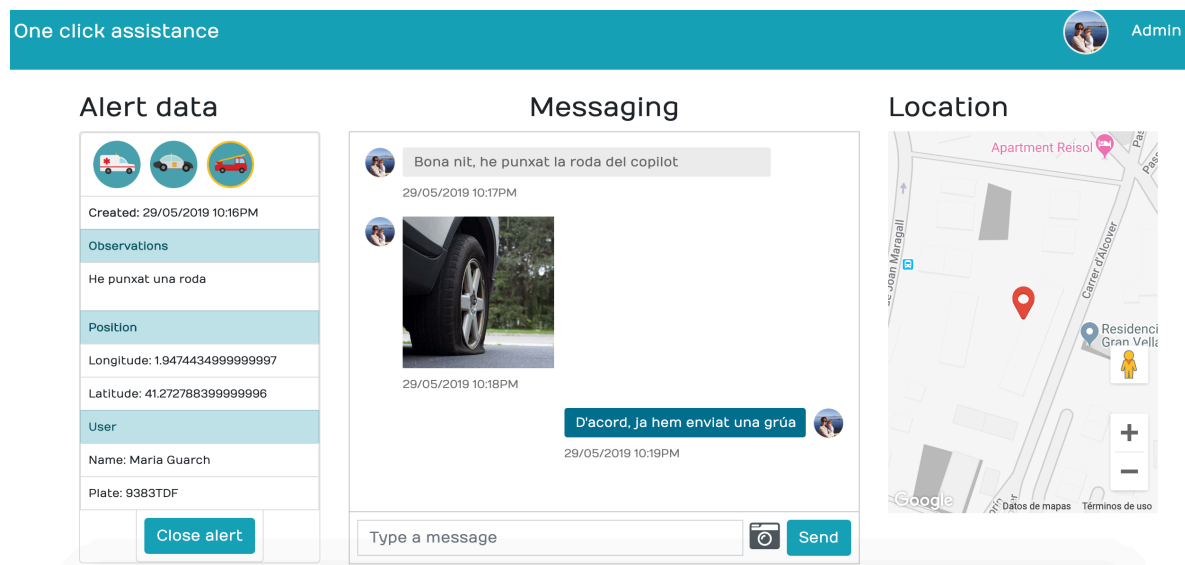
Així, qualsevol canvi que es produeixi en les dades del llistat es veuen reflectides a la vista sense haver de recarregar la pàgina.

5.5 Xat

La vista del xat per l'usuari amb el rol *admin* ofereix molta més informació ja que es considera que l'usuari amb aquest rol utilitza un ordinador de sobretaula o portàtil.

A l'esquerra hi ha tota la informació referent a l'alerta: els serveis que ha demanat l'usuari, la data, observacions o comentaris, posició i el nom de l'usuari i la matrícula del vehicle.

Al centre de la pantalla hi trobem el xat, que permet enviar imatges amb previsualització. I finalment, a la dreta trobem el mapa amb l'alerta ubicada en el punt on ha succeït l'accident o incidència.



En canvi, l'usuari amb el rol *user* només pot veure el xat ja que l'aplicació està pensada per ser utilitzada des de dispositius mòbils.

Una de les singularitats dels objectes *Observables* és que el valor de les variables es perd fora de l'scope del mètode *subscribe*. Així, per poder obtenir el ID del xat que s'acaba de crear i poder vincular-ho a l'alerta, calia fer un *Promise* amb les variables *resolve* i *reject* on emmagatzemar el resultat d'afegir un nou xat a la col·lecció de Firestore, que retorna l'ID que necessitem.

A continuació es mostra el codi que retorna l'objecte *Promise* amb les dades que retorna el mètode *add* del collection. Referències Angular Firebase lessons[6], Build a Group Chat with Firestore [7] i Build a real time chat with Angular 4+ and Firebase [8].


```
public create() {
  return new Promise((resolve, reject) => {
    return this.auth.user.subscribe(
      user => {
        const data = {
          uid: user.uid,
          photoURL: user.photoURL,
          createdAt: Date.now(),
          count: 0,
          messages: []
        };
        const chat = this.afs.collection('chats').add(data);
        chat.then(value => {
          resolve(value.id);
        })
      }
    );
  });
};
```

L'altra particularitat destacable de la programació amb Angular és l'asincronicitat. Cal tenir en compte que els processos no tenen perquè executar-se en l'ordre que es troben les instruccions en el codi. Així doncs, en el cas del xat, per poder desar les dades de la imatge que acabem de pujar, primer cal esperar que la imatge hagi acabat de pujar al servidor. Així doncs, en el mètode d'upload de la imatge cal indicar a la capçalera del mètode que la funció és asíncrona, i també, quan es fa la crida, cal indicar que esperem a que acabi per obtenir-ne el resultat.

Per indicar que el mètode s'executa de manera asíncrona es fa com es mostra a continuació:

```
async uploadImage(upload: Upload) { ... }
```

I per fer la crida al mètode sendMessage, cal indicar-ho de la següent manera:

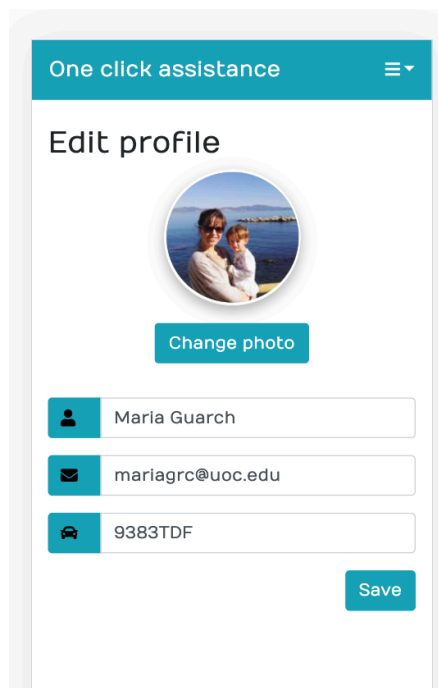
```
await this.uploadService.uploadImage(this.currentUpload)
```

Si un mètode conté alguna crida a un altre mètode asíncron, aquest també ha d'incloure a la capçalera l'async.

5.6 Editar perfil d'usuari

L'edició de perfil permet pujar la imatge de l'usuari. A aquest efecte s'ha habilitat Firestore per emmagatzemar les imatges que es pugen.

S'ha creat un servei anomenat *upload.service.ts* dins de *shared/services*.



A fi de poder fer el preview de la imatge, calia implementar a la vista, una imatge pel preview, que anomenem *imageShow*:

```
<img *ngIf="imageShow" [src]="imageShow"
    class="rounded-circle z-depth-1-half avatar-pic userImage">
```

I en el component, cal emprar un *FileReader* per obtenir la imatge i assignar-la a *src* de *imageShow*. Referència: [Stackoverflow\[9\]](#).

```
this.selectedFiles = event.target.files;

if (event.target.files && event.target.files[0]) {
  var reader = new FileReader();

  reader.onload = (event: ProgressEvent) => {
    this.imageShow = (<FileReader>event.target).result;
  }

  reader.readAsDataURL(event.target.files[0]);
}
```

6. Pressupost

El pressupost que s'adjunta a continuació considera les hores de gestió i desenvolupament del projecte amb un preu hora de 60 euros.

	Valoració en hores	Preu
Disseny de l'aplicació, UX i prototip	16	960€
Setup del projecte	8	480€
Implementació		
• Gestió d'usuaris	40	2400€
▸ Rols i seguretat		
▸ Login amb verificació		
▸ Recuperar contrasenya		
▸ Editar perfil amb imatge		
• Gestió d'alertes	24	1440€
▸ Crear alertes		
▸ Llistat d'alertes		
▸ Filtres segons rol d'usuari		
• Mapa	16	960€
▸ Posicionar alertes		
▸ Filtres segons rol d'usuari		
• Xat	32	1920€
▸ Implementació		
▸ Enviar imatges		
Total	136	8160€

7. Conclusions

El desenvolupament d'aquesta aplicació m'ha permès aprendre tecnologies que m'interessen especialment ja que el desenvolupament frontend, i en concret el llenguatge Angular, obre moltes possibilitats en el món laboral. Cal a dir que el paradigma canvia completament a tot el que havia après fins ara, ja que sempre he treballat amb PHP i base de dades relacionals. El canvi és considerable i la corva d'aprenentatge ha estat una mica dura al principi.

D'altra banda, quant a la planificació inicial i sobre com s'ha anat desenvolupant el treball, a nivell de fites s'ha anat complint fins al final però no ha estat fàcil complir les dates previstes inicialment. En un principi la planificació era molt optimista, retallant els temps d'entrega respecte l'establert en l'aula però malauradament no s'han pogut fer entregues abans de la data límit.

Personalment considero que he assolit pràcticament tots els objectius proposats. Tècnicament l'aplicació respon als requeriments inicials, si bé, fora bo trobar una solució per poder donar d'alta usuaris amb rol d'administrador de manera segura, però en qualsevol cas caldria la intervenció humana per validar aquestes altes.

Els objectius personals que pretenia els he assolit amb escreix. He après les bases d'Angular partint de zero i he assolit els coneixements mínims per poder desenvolupar aplicacions seguint la documentació i els tutorials.

El desenvolupament d'aquest projecte m'ha donat altres idees de possibles usos d'aquest tipus d'aplicació. Un altre ús que es podria fer d'aquesta aplicació seria per exemple, com a xarxa social. Es podria incloure en el desenvolupament la gestió de contactes i grups, i es podria enviar al grup d'amics l'alerta per quedar amb els amics en un punt de la ciutat. Els amics del grup podrien rebre l'alerta en el mapa i posar les indicacions de com arribar mitjançant Google Maps.

8. Bibliografia

1. Angular documentation

<https://angular.io/docs>

2. Angular Maps

<https://angular-maps.com/guides/getting-started/>

3. Firebase OAuth Login With Custom Firestore User Data

<https://angularfirebase.com/lessons/google-user-auth-with-firestore-custom-data/>

4. Full Angular 7 Firebase Authentication System

<https://www.positronx.io/full-angular-7-firebase-authentication-system/>

5. Firebase documentation

<https://firebase.google.com/docs/>

6. Angular Firebase lessons

<https://angularfirebase.com/lessons/>

7. Build a Group Chat with Firestore

<https://angularfirebase.com/lessons/build-group-chat-with-firestore/>

8. Eliran Elnasi, (05/2017) Build a real time chat with Angular 4+ and Firebase [consulta: 17/05/2019] <https://www.linkedin.com/pulse/build-real-time-chat-angular-4-firebase-eliran-elnasi/>

9. Stackoverflow

<https://stackoverflow.com>

9. Annexos

9.1 Accés aplicació

L'aplicació es troba en el següent enllaç:

<https://one-click-1554500008958.firebaseio.com>

Usuari amb rol "admin"

usuari: mariagrc@gmail.com

contrasenya: testing

Usuari amb rol "user"

usuari: mariagrc@uoc.edu

contrasenya: testing

9.2 Repositori GIT

<https://github.com/mariagrc/one-click>