

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Creació d'una app per a troubleshooting de grues amb variadors de freqüència

MEMÒRIA

Autor: Álvaro Mijancos Martínez
Director: Jordi Juliench Sáenz-López
Ponent: Samuel Galceran Arellano
Convocatòria: Juny 2020



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquesta memòria recull la informació associada al procés d'estudi i de realització de com crear i desenvolupar una aplicació per a dispositius mòbils fent ús de l'entorn de treball informàtic Ionic Framework.

Té l'objectiu d'exposar el procediment seguit per a conèixer la naturalesa d'aquest framework, així com les seves característiques, la seva estructura, el seu funcionament i el possible producte que es pot obtenir.

També pretén presentar el contingut de l'aplicació disponible per a l'usuari i com accedir a ell, donant detall dels recursos que conté l'app i de les seves funcionalitats. Es podria entendre com una guia per a l'usuari que s'acaba de descarregar i instal·lar l'aplicació en un dispositiu propi.

Una altra informació que s'exposa, d'una forma més tècnica però molt breu, és la descripció i el funcionament de les principals accions de l'aplicació que s'han programat en el seu desenvolupament.

En aquest projecte també es contempla el procés d'optimització realitzat a l'aplicació un cop aquesta ja estava desenvolupada, així com possibles millores aplicables en un futur per a acabar de perfeccionar-la.

A més a més, s'ha estudiat l'impacte ambiental que ha suposat la realització del projecte i el seu pressupost.

Sumari

RESUM	3
SUMARI	5
1. GLOSSARI	9
2. PREFACI	12
2.1. Origen del projecte	12
2.2. Motivació	13
2.3. Requeriments previs	13
3. INTRODUCCIÓ	15
3.1. Objectius del projecte	15
3.2. Abast del projecte	15
4. TIPUS D'APLICACIONS	16
4.1. Aplicacions natives	16
4.2. Aplicacions web	16
4.2.1. Aplicacions web progressives o PWA (Progressive Web Apps)	17
4.3. Aplicacions híbrides	17
4.4. Decisió	17
5. LENGUATGES DE DESENVOLUPAMENT WEB	19
5.1. HTML	19
5.2. CSS	19
5.3. Sass	20
5.4. JavaScript	20
5.5. TypeScript	21
6. PROGRAMES DE DESENVOLUPAMENT D'APLICACIONS	22
6.1. Angular	22
6.2. Apache Cordova	23
6.3. Ionic Framework	24
7. ENTORN DE TREBALL DE IONIC FRAMEWORK	26
7.1. Node.js & npm	26
7.2. Ionic CLI	26
7.3. Visual Studio Code	27
7.4. Capacitor	27

7.5. Firebase	28
7.6. Ionicons.....	29
7.7. Ionic Creator	29
8. ESTRUCTURA DE PROJECTE	30
8.1. App.....	30
8.1.1. App.module.ts.....	31
8.1.2. App.html.....	31
8.1.3. App.component.ts	31
8.1.4. App.scss	31
8.1.5. Main.ts	31
8.2. Assets	31
8.3. Pages.....	31
8.4. Services	32
8.5. Theme.....	32
9. APLICACIÓ	33
9.1. Pàgina d'inici i estructura	33
9.1.1. Capçalera	33
9.1.2. Contingut	33
9.1.3. Peu de pàgina.....	34
9.2. Secció Errors	36
9.2.1. Pàgina d'errors.....	36
9.2.2. Pàgina de presentació d'error	37
9.2.3. Pàgina del solucionador d'errors.....	38
9.2.4. Pàgina final d'un error	39
9.2.5. Pàgina Report.....	40
9.2.6. Pàgina Summary	41
9.3. Secció FAQs	42
9.4. Secció Vídeos Tutorials	44
9.5. Secció Documents.....	45
10. PRINCIPALS ACCIONS I CARACTERÍSTIQUES	47
10.1. Esdeveniment click	47
10.2. Navegació	48
10.3. Transmissió i visualització de paràmetres	48
10.4. Plugin Call Number	49
10.5. Idiomes	50
10.6. Base de dades i enviament del formulari.....	51

11. ANDROID	53
11.1. Configuració i exportació del projecte	54
11.2. Android Studio	54
11.3. Icon & Splash Screen	54
11.4. Android Emulator	55
11.5. Arxiu apk	56
12. OPTIMITZACIÓ	57
13. FUTURES MILLORES	58
14. IMPACTE AMBIENTAL	60
15. PRESSUPOST	62
CONCLUSIONS	63
AGRAÏMENTS	65
BIBLIOGRAFIA	67

1. Glossari

API: són les sigles en anglès d'Interfície de Programació d'Aplicacions (*Application Programming Interface*). És un conjunt de funcions prèviament implementades que ofereixen al programador una interfície a través de la qual poden comunicar-se amb un sistema determinat, afegint-li noves funcionalitats. En el cas d'una aplicació mòbil seria el mitjà a través del qual l'aplicació pot comunicar-se amb els recursos del sistema del dispositiu mòbil com poden ser la càmera, l'agenda de contactes, arxius, etc.

App: abreviatura d'aplicació.

Arxiu apk: un arxiu amb extensió .apk (*Android Application Package*, en català Paquet d'Aplicació d'Android) és l'arxiu que s'utilitza per distribuir i instal·lar aplicacions per a dispositius que utilitzen el sistema operatiu Android.

Arxiu json: és un tipus d'arxiu que permet emmagatzemar dades i informació, i també transferir-la.

Atribut: és una variable o propietat interna d'una classe. Poden ser de tipus numèric, text, booleà, o fins i tot un altre objecte.

Back-end: és la part del desenvolupament web o d'aplicacions que s'encarrega de què tot el funcionament i la lògica d'una pàgina web o una aplicació funcioni. És el conjunt d'accions que succeeixen però no veiem com.

Classe: element constructiu dels llenguatges de programació. Està definit per un conjunt d'atributs i mètodes, que confeccionen l'estat i el comportament de la classe, respectivament. És també un model o plantilla per crear objectes que estaran definits amb els mateixos atributs i mètodes de la classe a partir de la qual s'han creat.

Compilador: és un programa informàtic que tradueix un programa escrit en un llenguatge de programació en un altre llenguatge de programació.

ECMA: Associació Europea de Fabricants d'Informàtica, de les sigles en anglès *European Computer Manufacturers Association*, és l'organisme internacional de normalització del sector de la comunicació i la informàtica a Europa.

Emulador: En informàtica, un emulador és un software que permet executar programes o videojocs en una plataforma (sigui una arquitectura de hardware o un sistema operatiu) diferent d'aquella per a la qual van ser escrits originalment. A diferència d'un simulador, que només tracta de reproduir el comportament del programa, un emulador tracta de modelar de

forma precisa el dispositiu de manera que el programa desitjat funcioni com si estigués sent utilitzat en l'aparell original.

Entorn de desenvolupament integrat (IDE): de les sigles en anglès *Integrated Development Environment*, un entorn de desenvolupament integrat és una aplicació informàtica que proporciona serveis per facilitar al programador el desenvolupament d'un software.

Entorn de treball o *framework*: estructura tecnològica de suport que serveix com a base per l'organització i desenvolupament d'un software.

Entorn en temps d'execució o *runtime environment*: software que proveeix serveis a un altre programa que té la característica de poder executar-se i modificar-se al mateix temps.

Esdeveniment: acció detectada per un programa informàtic.

Front-end: és la part visual i interactiva d'una aplicació mòbil o una pàgina web, la part que l'usuari pot veure i amb la que pot experimentar.

Interfície d'usuari: és el conjunt de controls i mitjans sensorials a través del qual l'usuari pot comunicar-se amb una màquina, un ordinador o un dispositiu. Comprèn tots els punts de contacte entre l'usuari i l'equip.

Llibreria: en informàtica, és un programa integrat dins d'un altre programa per proporcionar-li diverses funcionalitats i serveis.

Metallenguatge o *superset*: tipus de llenguatge que té la característica de poder incorporar parts i funcionalitats d'un altre llenguatge, degut a que tenen una sintaxi compatible.

Mètode: és una acció que una classe pot realitzar.

Objecte: és un element constructiu dels llenguatges de programació definit per uns atributs i mètodes. Es pot crear a partir d'una classe, fet que el faria compartir atributs i mètodes amb la classe a partir de la qual s'ha creat.

Open-source: referit a un software que no inclou cap tipus d'encryptació, de forma que és accessible al públic. Tothom pot veure, modificar i distribuir el codi de la forma que considerin convenient.

Plugin: és una petita aplicació que s'implementa en una altra aplicació per agregar-li una funció nova i generalment molt específica. En el cas del desenvolupament d'aplicacions mòbils, un plugin seria una aplicació addicional, un paquet de codi, que és executada per l'aplicació principal i que permet a aquesta fer ús d'un recurs específic del sistema d'un

dispositiu mòbil a través de l'API. Per tant, existiria un plugin per a la càmera, un altre plugin per l'agenda de contactes, etc.

Troubleshooting: és el terme utilitzat per a la resolució de problemes, sovint aplicada, per a reparar productes o processos fallits en una màquina o sistema. Consisteix en una recerca sistemàtica i lògica de la font del problema per resoldre'l i fer que el producte o procés torni a estar operatiu.

URL: Localitzador de Recursos Uniforme, de les sigles en anglès *Uniform Resource Locator*, és la direcció web dels diferents recursos disponibles a Internet, única per a cadascun d'ells.

Wizard: en informàtica, també conegut com a assistent, és un servei per a l'usuari format per un conjunt de pàgines per guiar-lo i orientar-lo pas a pas per a la realització d'una tasca determinada.

Youtube: lloc web dedicat a compartir vídeos.

2. Prefaci

2.1. Origen del projecte

A finals de febrer de 2020, vaig començar a treballar amb un conveni de pràctiques a l'empresa Tecnotrans Bonfiglioli S.A., situada a la Zona Franca de Barcelona.

Aquesta empresa és una filial de la multinacional italiana Bonfiglioli, i es centra en l'assemblatge i la distribució de variadors de freqüència i motoreductors industrials.

En el moment de la meva incorporació, em van incloure en un projecte destinat a resoldre possibles problemes que poguessin experimentar els clients amb els variadors de freqüència de l'empresa en aplicacions per a grues. Aquest procés de resolució d'errors, també anomenat *troubleshooting*, inicialment, consistia a crear un wizard solucionador d'errors perquè els clients el poguessin utilitzar com a assistent i poder resoldre els seus problemes sense haver de tenir un coneixement expert en el funcionament dels variadors o haver de contactar directament amb el servei tècnic de Tecnotrans Bonfiglioli.

Després de valorar de quina forma seria més fàcil, més còmode i més útil per a l'usuari tenir accés i fer ús d'aquest troubleshooting, es va decidir que es distribuïria en format aplicació de mòbil. Vaig passar a formar part de l'equip de desenvolupament del projecte, on la meva tasca seria, precisament, desenvolupar l'aplicació que contindria els wizards solucionadors d'errors per a poder dur a terme el troubleshooting.

A mesura que el projecte anava avançant, es va veure un gran potencial en l'aplicació i es va voler ampliar el seu contingut amb diferents seccions, i no restringir-la únicament als wizards solucionadors d'errors.

Degut a que la meva participació en aquest projecte requeriria una gran implicació i que era un projecte de gran interès personal, vaig decidir, d'acord amb l'empresa, aprofitar aquest projecte per a desenvolupar el meu Treball de Fi de Grau.

Vull destacar que l'empresa no té cap equip especialitzat ni cap precedent previ en desenvolupament d'aplicacions mòbils, i que jo he sigut l'única persona que s'ha encarregat de generar l'aplicació en si. La informació que aquesta conté, com ara els passos dels wizards, ha estat desenvolupada i proporcionada per les altres persones que componien l'equip de desenvolupament del projecte de troubleshooting.

2.2. Motivació

La meua principal motivació per a aquest projecte era poder arribar a crear un producte tan sol·licitat i utilitzat en l'actualitat com és una aplicació mòbil. Un element d'ús diari en la societat i que pràcticament tot el món coneix i utilitza però que, sense dubte, poca gent es para a pensar de quina forma es generen i existeixen. La possibilitat de desenvolupar una aplicació des de zero i a més a més, poder veure el seu resultat en primera persona, instal·lant-la en el meu propi dispositiu mòbil, era un gran repte al qual no tenia gens de por d'enfrontar-m'hi.

També tenia un gran interès per explorar i descobrir un món totalment desconegut per a mi com és el món de la programació. Fer créixer el meu coneixement aprenent un conjunt de llenguatges nous, essencials per a qualsevol programador informàtic, era una de les principals motivacions.

I com no, el de poder dur a terme un bon treball a l'empresa i que estiguessin satisfets d'haver apostat per mi a l'hora de contractar-me. Formar part d'un projecte innovador a la feina, sense cap tipus de precedent, i poder crear un producte el qual ajudés a l'empresa a créixer era un dels grans al·licients d'aquest projecte. A més, no es tractaria d'un producte qualsevol, sinó d'un producte final que seria utilitzat directament pels clients.

2.3. Requeriments previs

Per a poder realitzar aquest projecte es requereixen uns coneixements bàsics en programació. Si, a més, es disposa d'un domini en els llenguatges de desenvolupament web que utilitza Ionic Framework, com són HTML, CSS i JavaScript, i en el funcionament del framework Angular per a programar, es té molt guanyat.

3. Introducció

3.1. Objectius del projecte

Els objectius d'aquest projecte són analitzar i comprendre com es desenvolupen les aplicacions mòbils, estudiant quins llenguatges són necessaris i de quins programes informàtics es requereixen. Per això, primer s'haurà de conèixer el tipus d'aplicacions existents i escollir el tipus adequat i que més s'ajusti al projecte.

També es té com a objectiu crear una aplicació híbrida amb l'entorn de treball Ionic Framework, aprenent el funcionament d'aquest framework i la seva estructura de projecte, fins a obtenir com a resultat una aplicació instal·lable i funcional en un dispositiu mòbil. Aquesta aplicació ha de poder ajudar a resoldre els problemes que tinguin els usuaris amb els variadors de freqüència de l'empresa Tecnotrans Bonfiglioli, i satisfer les necessitats i expectatives d'aquests perquè puguin veure-la com una gran eina i complement per al seu dia a dia en el treball.

3.2. Abast del projecte

Aquest projecte també pretén exposar les característiques, la funcionalitat i perquè es requereixen els llenguatges utilitzats en el desenvolupament de l'aplicació, així com els diferents programes implicats per a aconseguir crear aquesta aplicació. No només els programes imprescindibles per a poder generar una aplicació amb Ionic Framework, sinó també aquells que han estat utilitzats com a complements per a aquest entorn de treball i poder dotar de certs serveis i característiques a l'aplicació de manera que fos molt més completa i de major valor i qualitat.

També s'expliquen les característiques i el contingut de les diferents pàgines i seccions que confeccionen l'aplicació, amb la intenció de donar a entendre perquè el client pot veure una utilitat i atracció en aquestes parts.

Aquest projecte no té la intenció d'explicar el codi escrit per a desenvolupar l'aplicació, ja que seria massa extens, però sí donar a entendre d'una manera superficial com funcionen i perquè serveixen les principals accions de l'aplicació.

4. Tipus d'aplicacions

Actualment pràcticament tothom coneix el que és una aplicació mòbil. En l'àmbit de la informàtica una aplicació es defineix com un programa informàtic dissenyat com a eina per permetre i/o ajudar a l'usuari a realitzar una o diversos tipus de tasques. Una aplicació mòbil o *app*, per tant, és una aplicació informàtica dissenyada i construïda per ser executada en un telèfon intel·ligent, en una tauleta tàctil i en altres dispositius mòbils.

Per a poder desenvolupar-les existeixen diverses alternatives diferents: aplicacions natives, aplicacions web i aplicacions híbrides [1] [2] [3].

4.1. Aplicacions natives

Les aplicacions natives són aquelles que es generen per a funcionar en un sistema operatiu concret.

El seu principal inconvenient és que cada sistema operatiu té el seu propi llenguatge per al desenvolupament d'aplicacions natives. Les aplicacions per a dispositius iOS es desenvolupen emplenant llenguatge Objective-C, mentre que les aplicacions per a dispositius Android s'han de desenvolupar utilitzant Java. Això implica que si es vol crear una aplicació que pugui ser utilitzada tant en dispositius Android com iOS aquesta s'haurà de generar dues vegades, amb el seu llenguatge corresponent en cada cas.

Com a avantatges principals no necessiten connexió a internet per a funcionar i tenen un gran rendiment, ja que han estat creades per a un sistema operatiu en concret. Tenen la possibilitat d'accedir a tots els recursos del tant sistema com del hardware, com poden ser la càmera, el GPS, l'agenda, etc.

4.2. Aplicacions web

Són aplicacions a les quals s'accedeix des del navegador del dispositiu a través de la seva direcció web o *URL*.

Els seus avantatges és que no necessiten ser instal·lades en el dispositiu, es poden executar des de qualsevol sistema operatiu sense haver de crear diverses aplicacions i són senzilles de crear, ja que utilitzen els llenguatges de desenvolupament web típics i coneguts com són HTML, CSS i JavaScript.

Pel que fa als seus inconvenients necessiten connexió a internet per a poder funcionar i

tenen un accés molt limitat als recursos del dispositiu i del hardware, de manera que l'experiència d'ús de l'usuari sol ser pitjor que en els altres casos. A més, no es poden publicar en plataformes de distribució com Google Play Store i AppStore per a la seva comercialització.

4.2.1. Aplicacions web progressives o PWA (Progressive Web Apps)

Les PWA [4] [5] [6] vindrien a ser una versió millorada de les aplicacions web. Aquest tipus d'aplicacions sí que permeten ser utilitzades sense connexió a internet. El seu URL s'ha d'ancorar a la pantalla principal del dispositiu de manera que quan s'obri l'aplicació en comptes de fer-ho en el navegador ho farà com si fos una aplicació nativa.

4.3. Aplicacions híbrides

Per últim, les aplicacions híbrides són una combinació de les anteriors. Aquest tipus d'aplicacions recullen els millors aspectes dels altres tipus. Per un costat, per al seu desenvolupament utilitzen els llenguatges típics del desenvolupament de les pàgines web HTML, CSS i JavaScript. Aquest fet permet mantenir el mateix codi per a diferents sistemes operatius i fa que no s'hagi de generar la mateixa aplicació dues vegades com en les aplicacions natives. I per un altre costat, tenen un gran accés als recursos del sistema i del seu hardware. S'han d'instal·lar en el dispositiu i es poden publicar en les plataformes digitals de distribució.

Com a inconvenient cal dir que el seu rendiment és inferior les aplicacions natives.

4.4. Decisió

Un cop analitzades les opcions disponibles i les característiques de cada una d'elles es va decidir optar per l'aplicació híbrida. Els motius eren clars, poder desenvolupar una aplicació que fos disponible en el màxim mercat possible, generant un únic codi per als sistemes operatius referència Android i iOS i publicant l'aplicació en les plataformes de distribució, i que permetés a l'usuari tenir una experiència excel·lent, amb una gran varietat de funcionalitats. A més, amb una aplicació híbrida es podrien utilitzar la gran majoria de recursos del dispositiu.

Després de fer una recerca sobre els diferents frameworks disponibles per desenvolupar aplicacions mòbils híbrides, es va optar per escollir Ionic Framework [7] per a realitzar aquest projecte.

Aquest framework és un dels més famosos i punters en el desenvolupament híbrid. Es

considera que amb ell s'obtenen unes de les aplicacions mòbils híbrides amb major rendiment i qualitat. Es va fundar l'any 2013, sent un dels primers en el sector.

Com a principals factors per a aquesta decisió, a més dels ja mencionats, s'estima que la seva corba d'aprenentatge de Ionic té un elevat pendent, el que significa que permet guanyar i assolir uns grans coneixements en poc temps. També compta amb una gran comunitat d'usuaris, fet que facilita l'accés a un gran contingut de documentació i suport. I per últim, el seu ús és gratuït i no cal contractar un pla de pagament per obtenir grans resultats.

Alguns exemples de grans empreses que han desenvolupat aplicacions o posseeixen aplicacions desenvolupades amb Ionic Framework són McDonald's Turquia, amb la seva aplicació oficial per demanar menjar online, la marca Diesel, que també compta amb la seva pròpia aplicació creada amb Ionic Framework o fins i tot Airbus, concretament Airbus Helicopters, que està fent ús de Ionic per proporcionar aplicacions als seus empleats.

5. Llenguatges de desenvolupament web

5.1. HTML

HTML [8] [9] són les sigles en anglès de *HyperText Markup Language*, llenguatge de marques d'hipertext. És un llenguatge per a l'elaboració de contingut visual de les pàgines web. Aquest contingut poden ser textos, imatges, vídeos, gràfics, enllaços, etc. HTML5 és l'última versió d'aquest llenguatge.

S'anomena llenguatge de marques perquè utilitza unes marques o etiquetes per definir les parts de l'estructura de la pàgina web i els elements que la compondran. Cada element necessita una marca d'obertura i una marca de tancament per a definir-lo.

Per exemple, per definir la capçalera d'una pàgina, en anglès *header*, s'utilitzarien les marques següents: `<header></header>`. La primera marca seria la marca d'obertura i la segona la de tancament, i entremig de les dues marques aniria el contingut que es volgués afegir a la capçalera.



Figura 1: Logo HTML. Font: [10].

5.2. CSS

És un llenguatge de disseny gràfic que s'utilitza per definir i crear la presentació d'un document estructurat amb un llenguatge de marques. Les seves sigles corresponen a *Cascading Style Sheets* [11], en català fulls d'estil en cascada.

És un llenguatge molt utilitzat, sobretot per establir l'estil el disseny visual de pàgines web creades amb HTML. Va ser creat per separar la part de l'estructura i contingut d'una pàgina web de la part de la presentació i el format d'aquesta.

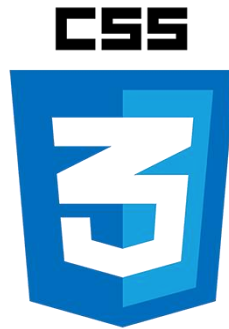


Figura 2: Logo CSS. Font: [12].

5.3. Sass

Sass [13] és un metallenguatge de CSS. Les seves sigles signifiquen *Syntactically Awesome Stylesheets*, en català fulls d'estil sintàcticament impressionants. És un llenguatge que, igual que CSS, serveix per definir i crear el disseny de la presentació de pàgines web, però que conté un major nombre de característiques. Sass permet incorporar funcionalitats de CSS, ja que tenen una sintaxi compatible. A l'hora d'utilitzar Sass per al desenvolupament de pàgines web, quan aquestes s'executen, el seu codi es tradueix a CSS.



Figura 3: Logo Sass. Font: [14].

5.4. JavaScript

JavaScript [15] S'utilitza principalment per crear contingut dinàmic en les pàgines web com poden ser accions que s'activen al pulsar botons, finestres amb avisos a l'usuari o animacions. No només s'encarrega del contingut dinàmic del front-end d'una pàgina web o aplicació, sinó que també incorpora funcionalitats per treballar des del back-end i enviar informació i ordres al servidor.

És un llenguatge de programació interpretat, això vol dir que el seu codi s'executa en temps real sense la necessitat de què un compilador tradueixi el seu codi en un altre llenguatge.

JavaScript és alhora la implementació més popular de l'especificació ECMAScript [16], publicada per ECMA International. Aquesta especificació s'encarrega de regir com han de ser interpretats i funcionar certs llenguatges de programació.

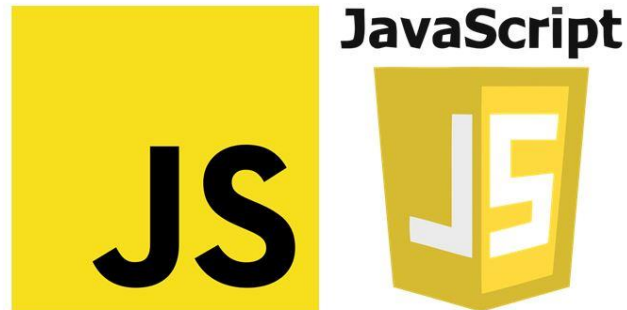


Figura 4: Logos JavaScript. Fonts: [17] [18].

5.5. TypeScript

TypeScript [19] és un llenguatge de programació de codi obert desenvolupat i mantingut per Microsoft. És també un superset de JavaScript, per tant, entén i estén la seva sintaxi i qualsevol codi escrit en JavaScript hauria de funcionar sense cap problema en TypeScript. A diferència de JavaScript, TypeScript és un llenguatge compilat, per tant per a la seva lectura i execució aquest llenguatge serà traduït a JavaScript a través d'un compilador. Inclou diverses característiques addicionals que faciliten la seva escriptura de codi i fan que sigui més versàtil, comparat amb JavaScript.



Figura 5: Logo TypeScript. Font: [20].

6. Programes de desenvolupament d'aplicacions

6.1. Angular

Angular [21] és un framework de JavaScript, open-source, mantingut per Google i dissenyat per construir aplicacions web d'una sola pàgina SPA (Single Page Application). Que sigui un framework de JavaScript significa que és un framework amb gran part del seu codi pre escrit en JavaScript que permet un desenvolupament més fàcil en aplicacions basades en JavaScript. El seu llenguatge per al desenvolupament d'aplicacions és HTML, Sass i TypeScript.

Una aplicació web SPA creada amb Angular és una aplicació d'una sola pàgina, el que significa que la navegació entre seccions i pàgines de l'aplicació, així com la càrrega de dades, es realitza de forma instantània cridant al servidor sense haver de refrescar el navegador. Això és degut al fet que una vegada oberta l'aplicació aquesta es carrega sencera, i Angular el que fa es anar canviant la vista de la pàgina a partir de la ruta continguda en el servidor, amb què l'usuari va movent-se a través de l'aplicació.



Figura 6: Logo Angular. Font: [22].

Angular separa el que és el front-end i el back-end. Per un costat el que es fa és generar la vista de les pàgines amb plantilles HTML, i per un altre genera una lògica creada en els components de TypeScript que controlen aquestes vistes. Això és possible gràcies al seu sistema *two-way data binding*, una forma bidireccional d'enllaçar la lògica que conté la informació de l'aplicació amb les vistes. Hi ha dos tipus de data-binding:

- Event binding: responen a la interacció de l'usuari en modificar algun input de l'aplicació, actualitzant la informació de l'aplicació.
- Property binding: permeten modificar o afegir valors des de la lògica de la informació de la nostra aplicació a la vista HTML.

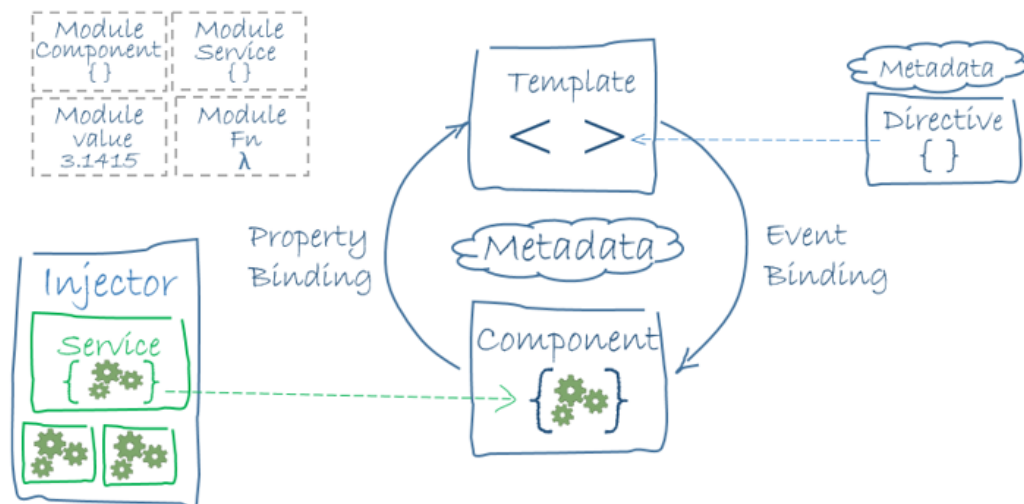


Figura 7: Esquema de l'arquitectura d'Angular. Font: [23].

6.2. Apache Cordova

Apache Cordova [24] es defineix com un framework open-source per al desenvolupament d'aplicacions mòbil. Permet utilitzar els llenguatges web estàndards HTML, CSS i JavaScript per crear aplicacions híbrides o multiplataforma.

Encara que, pel que és especialment coneguda, és per ser un framework que proporciona unes API de JavaScript i un gran conjunt de plugins que permeten a l'aplicació accedir a les funcionalitats i recursos natius del dispositiu mòbil com poden ser la càmera, els contactes, els arxius... i que transforma aquestes aplicacions desenvolupades amb els llenguatges de programació web en aplicacions natives completes. Cordova conté les eines de compilació necessàries per empaquetar aplicacions web i transformar-les en aplicacions per a iOS i Android.

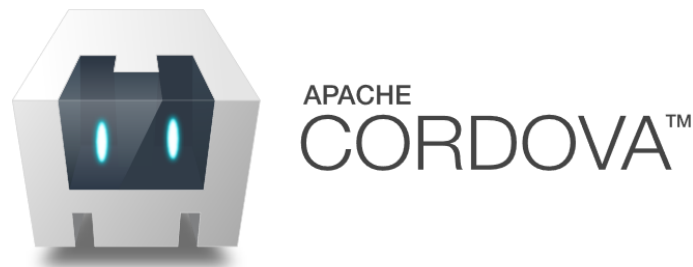


Figura 8: Logo Apache Cordova. Font: [25].

6.3. Ionic Framework

Ionic Framework és un kit d'eines d'interfície d'usuari de codi obert per crear aplicacions web progressives o aplicacions mòbils híbrides útils i de gran qualitat a partir dels llenguatges de programació web HTML, CSS i JavaScript.

Ionic es centra en el front-end de les aplicacions, és a dir, en les anomenades experiències d'usuari i interfície d'usuari, UX i UI respectivament (*User Experience i User Interface*). Això estaria constituït pels controls de l'usuari, les interaccions, les animacions i els gestos. Ionic és en si una gran llibreria de components d'interfície d'usuari que s'utilitzen com a elements constructius d'una aplicació.

Per tant, per a poder crear una aplicació completa Ionic incorpora i utilitza altres frameworks que el complementin per a complir amb totes les necessitats que pot requerir una aplicació.



Figura 9: Logo Ionic. Font: [26].

En el moment del seu llançament Ionic estava basat en Angular i en Apache Cordova. Utilitzava Angular per construir l'aplicació en si i Apache Cordova per fer ús de les seves API i plugins per comunicar les funcions de l'aplicació amb els recursos del dispositiu, així com per traduir el seu codi en una forma llegible per a poder exportar-ho i crear una aplicació nativa del sistema operatiu desitjat.

Actualment Ionic permet als seus usuaris utilitzar diferents frameworks de JavaScript pel desenvolupament d'aplicacions com són Angular, React i Vue, aquest últim en fase de construcció. A partir de maig de 2019 va llençar el seu propi framework proveïdor d'APIs i constructor del codi web en aplicació nativa Capacitor com a alternativa a Apache Cordova.

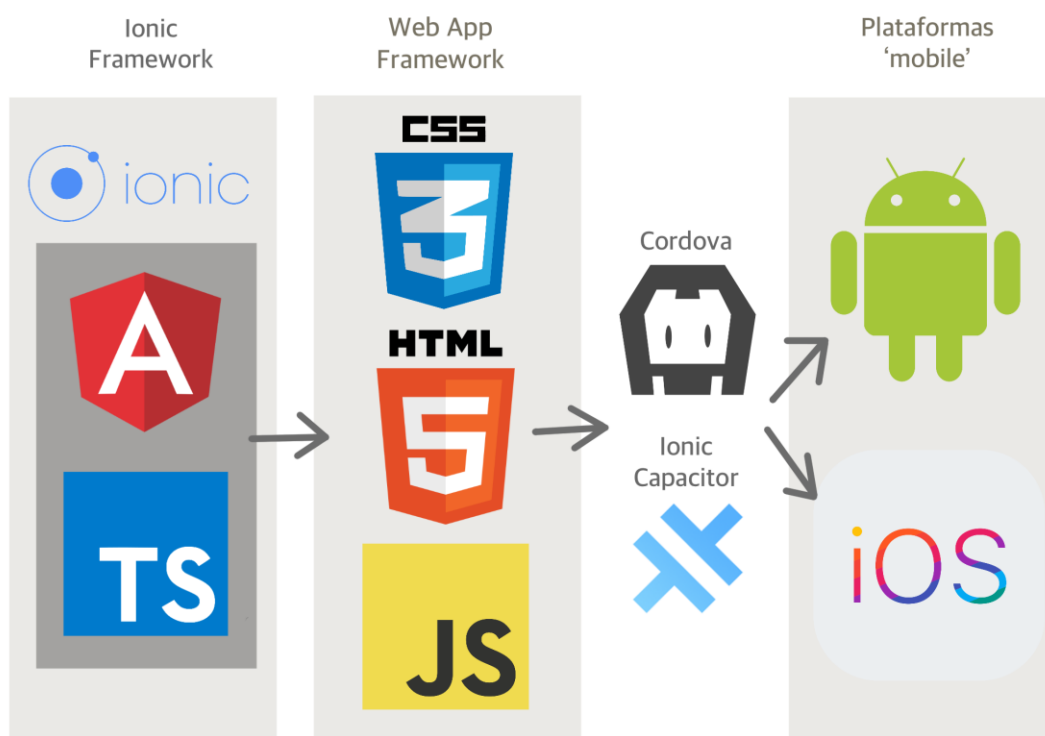


Figura 10: Programes i llenguatges utilitzats, i etapes del procés per obtenir una aplicació amb Ionic Framework. Font: [27].

7. Entorn de treball de Ionic Framework

7.1. Node.js & npm

Node.js [28] i npm [29] són els programes necessaris i els únics requerits per a la instal·lació i el correcte funcionament de Ionic framework.

Node.js és un entorn en temps d'execució que permet escriure el llenguatge JavaScript en el back-end. S'utilitza principalment per a oferir serveis pel desenvolupament web. És imprescindible per a Ionic per a construir diverses de les seves eines de desenvolupament d'aplicacions, entre altres l'Ionic CLI del que es parla en el següent apartat.

Npm és l'administrador dels paquets en els quals es troben emmagatzemats els serveis de Node.js. A través de npm s'instal·la Ionic, juntament amb diverses de les seves dependències, és a dir, aquells programes i arxius els quals Ionic necessita per a poder funcionar.



Figura 11: Logos de Node.js i npm. Fonts: [30] [31].

7.2. Ionic CLI

La paraula CLI correspon a les sigles en anglès *Command-line Interface*. És un mitjà a través del qual, utilitzant línies de text simples, s'interactua amb un programa.

L'Ionic CLI [32] serà l'eina utilitzada per a crear les aplicacions desenvolupades amb Ionic. Té un gran nombre de funcionalitats com generar les pàgines de l'aplicació, donar ordres per executar l'aplicació en el servidor, descarregar llibreries o donar suport d'ajuda.

7.3. Visual Studio Code

És un altre imprescindible per al desenvolupament d'aplicacions. És un editor de text de codi a través del qual podrem escriure el codi de la nostra aplicació. Qualsevol editor de text compatible amb Ionic podria ser utilitzar, encara que Visual Studio Code [33] és el que es recomana des de la pàgina de Ionic Framework.

És un editor de text desenvolupat per Microsoft i mundialment conegut gràcies a algunes de les seves característiques com ser gratuït, open-source i disponible per Windows, macOS i Linux. A més a més, accepta un gran nombre de llenguatges de programació.



Figura 12: Logo Visual Studio Code. Font: [34].

7.4. Capacitor

Capacitor [35] és una plataforma open-source que permet transformar aplicacions web en aplicacions natives executables en dispositius iOS i Android. Aquesta serà la seva principal funció dins del projecte. És el successor d'Apache Cordova i el framework recomanat i promocionat per dur a terme aquesta tasca amb les aplicacions construïdes amb Ionic Framework, ja que està desenvolupat i mantingut per aquesta mateixa empresa.

Capacitor també proporciona un conjunt d'APIs que permeten accedir a les funcionalitats natives dels dispositius. Encara més, Capacitor és compatible amb la majoria de plugins d'Apache Cordova, de forma que també es podrà fer ús d'ells.



Figura 13: Logo Capacitor. Font: [36].

7.5. Firebase

Firebase [37] és una plataforma que ofereix un conjunt de recursos i eines pel desenvolupament d'aplicacions web i aplicacions mòbil, desenvolupada per Google l'any 2014.

Firebase permet integrar els seus serveis dins de projectes propis amb l'objectiu de facilitar la gestió de l'aplicació, augmentar la base dels seus usuaris i la seva monetització. Un gran avantatge és que la major part d'aquests serveis són gratuïts.

Permet sincronitzar fàcilment dades del projecte sense la necessitat d'administrar connexions o escriure una lògica de sincronització complexa. També utilitza un conjunt d'eines multiplataforma que fan que aquestes es puguin integrar d'una forma senzilla tant en aplicacions web com en aplicacions mòbils. En el cas d'aquestes últimes és compatible amb els grans sistemes operatius com són iOS i Android.

Dos dels seus serveis més rellevants són Firebase Database i Firebase Storage. El primer proporciona una base de dades en temps real que permet que la informació de l'aplicació sigui emmagatzemada en el núvol i que els usuaris puguin accedir a ella i modificar-la en temps real. Porta incorporada una funcionalitat que, en cas que l'usuari perdi la connexió a l'hora de realitzar els canvis, guarda els canvis de forma local en el dispositiu i un cop l'usuari torna a tenir connexió els sincronitza amb la base de dades de forma automàtica.

Per una altra banda, el servei de Firebase Storage permet realitzar càrregues i descàrregues segures d'arxius des de l'aplicació. Pot ser utilitzat per emmagatzemar imatges, vídeos, àudios, arxius PDF o qualsevol contingut generat per l'usuari.



Figura 14: Logo Firebase. Font: [38].

7.6. Ionicons

Ionicons [39] és una col·lecció d'icones d'accés públic creada i utilitzada per Ionic. Conté més de 100 icones confeccionades per a poder usar-se en aplicacions web, aplicacions mòbils i aplicacions d'escriptori.

Per tant, aquesta col·lecció és de gran utilitat i el seu ús és molt senzill i intuïtiu. És la font que ha proveït totes les icones que s'han incorporat en les diferents pàgines de l'aplicació.

Ionicons està integrat per defecte en Ionic Framework, per tant no cal instal·lar-ho si l'aplicació es construeix amb Ionic. Tot i això, també és possible utilitzar aquesta col·lecció si s'utilitzen altres frameworks de desenvolupament.

7.7. Ionic Creator

És una utilitat de Ionic Framework que permet crear una aplicació mòbil sense la necessitat d'escriure cap línia de codi. Consisteix en un sistema *Drag and Drop*, és a dir, arrossegar els elements que es volen per a l'aplicació i deixar-los anar en la posició desitjada.

Ionic Creator [40] proporciona una visualització d'un telèfon mòbil amb l'aplicació que estem construint oberta, i una col·lecció d'elements que poden ser incorporats a les aplicacions com per exemple llistes, botons, imatges, paràgrafs, etc. Aquests elements només s'han de seleccionar amb el ratolí, arrossegar fins a la part de l'aplicació on es volen situar i deixar anar. També permet configurar algunes funcions com la navegació entre diferents pàgines de l'aplicació.

El problema és que de forma gratuïta és una eina bastant limitada, per aconseguir una aplicació completa s'hauria de contractar un pla de pagament. Tot i això permet construir una primera idea del que es vol com a aplicació, crear el codi d'aquesta primera idea i descarregar-lo per a poder exportar-lo a l'editor i poder continuar l'aplicació sense haver de partir de zero.

8. Estructura de projecte

Per al desenvolupament de l'aplicació s'ha fet servir la versió 3 de Ionic Framework. Al tractar-se d'una versió antiga, aquesta utilitza ionic-angular per al desenvolupament de l'aplicació, en comptes d'un dels frameworks de JavaScript. Consisteix una mescla d'Angular pur i alguns elements propis de Ionic. Aquesta versió permet escollir entre Cordova i Capacitor per passar l'aplicació a nativa i utilitzar els recursos propis dels dispositius. També cal afegir que, a l'utilitzar Angular, s'utilitza el llenguatge Sass en comptes de CSS per a decorar l'aplicació.

Aquest capítol no pretén explicar l'organització de totes les carpetes del projecte, sinó que es centra en el contingut més rellevant de la carpeta `src`, que és la que conté emmagatzemat tot el codi utilitzat per al desenvolupament de l'aplicació i que és pràcticament l'única que construeix i modifica el programador.

A continuació s'expliquen les seves principals subcarpetes.



Figura 15: Carpeta `src` i les seves subcarpetes en Visual Studio Code. Font: pròpia.

8.1. App

Aquesta carpeta conté tots els arxius necessaris per engegar l'aplicació i l'estructura principal sobre la qual es construeix aquesta.

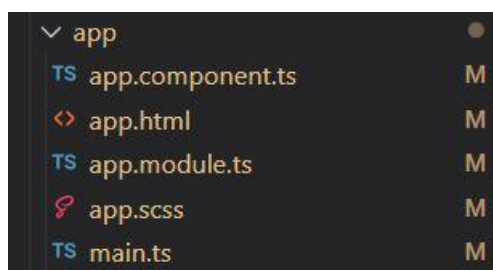


Figura 16: Carpeta `app` i els seus arxius, en Visual Studio Code. Font: pròpia.

8.1.1. App.module.ts

És un arxiu TypeScript en el qual s'han de declarar totes les dependències i els components de l'aplicació. Això correspon bàsicament a les pàgines que constitueixen l'aplicació, als diferents serveis que poden ser utilitzats en aquesta i a les classes dels plugins i les llibreries importades.

8.1.2. App.html

En aquest arxiu HTML s'han de definir els elements del front-end de l'aplicació que es vol que siguin comuns en totes les pàgines de l'aplicació. Per exemple, un menú lateral al qual l'usuari pogués accedir independentment de la pàgina en què es trobés o una barra inferior amb diferents opcions que es mantingués en tota l'aplicació.

8.1.3. App.component.ts

En aquest arxiu TypeScript es defineixen els aspectes inicials de l'aplicació, com la primera pàgina que es visualitza en obrir l'aplicació o l'idioma que apareix per defecte. També es defineixen les funcions que s'han assignat als elements de l'arxiu app.html, en cas que això s'hagi fet.

8.1.4. App.scss

És l'arxiu de Sass on s'han d'incloure els estils de disseny que es vulguin fer servir d'una forma global a l'aplicació. També s'utilitza per importar altres arxius Sass.

8.1.5. Main.ts

És un arxiu TypeScript que es genera de forma automàtica i s'encarrega d'inicialitzar l'aplicació.

8.2. Assets

Aquesta carpeta conté els diferents recursos que pugui necessitar l'aplicació. Aquests recursos poden ser imatges que es vulguin mostrar a l'usuari, arxius de text PDF, arxius json, icones, etc.

8.3. Pages

Cada pàgina de l'aplicació té una carpeta pròpia amb el nom que se li ha assignat. Pages és la carpeta que conté totes les carpetes de totes les pàgines creades.

La carpeta d'una pàgina de l'aplicació conté tres arxius, un arxiu HTML, un altre Sass i un altre de tipus TypeScript, cada un amb una funcionalitat diferent. Això és degut al fet que, com ja s'ha mencionat anteriorment, Angular separa el front-end del back-end.

En l'arxiu HTML es programa la part visual de la pàgina i en l'arxiu TypeScript la seva lògica. Les funcions que s'assignen als elements de l'arxiu HTML perquè l'usuari pugui interactuar amb ells, són definides en l'arxiu TypeScript.

En l'arxiu Sass es defineixen els estils que es vulguin aplicar a la pàgina concreta.

8.4. Services

Un servei és una classe amb un objectiu concret i definit. Tot i que es programen en un arxiu TypeScript i certament s'atribueixen a elements de pàgines, no es defineixen en l'arxiu TypeScript propi d'una pàgina perquè Angular prefereix separar les funcions que tenen a veure amb l'experiència d'usuari de les accions com obtenir dades d'un servidor o validar el compte d'un usuari. Aquestes últimes accions les deleguen als serveis.

Doncs bé, en la carpeta services es trobaran continguts aquests arxius TypeScript de serveis, en cas que es requereixen per a l'aplicació.

8.5. Theme

Aquesta carpeta únicament conté un arxiu anomenat variables.scss. Aquest és un arxiu Sass que s'utilitza per definir totes les variables que es vulguin utilitzar d'una forma global en l'aplicació.

La diferència entre aquest arxiu i l'arxiu app.scss de la carpeta app és que en variables.scss es defineixen variables, com pot ser un color específic que es vulgui definir per a després poder utilitzar-ho en algun element, mentre que en l'arxiu app.scss es defineixen estils, és a dir, un conjunt de característiques d'un element en particular.

9. Aplicació

Aquest apartat pretén explicar en detall el contingut de l'aplicació i les funcionalitats de les diferents seccions i pàgines que la componen, així com exposar els motius pels quals pot ser d'interès per a l'usuari i per a l'empresa.

9.1. Pàgina d'inici i estructura

La pàgina d'inici consisteix en la base de l'estructura de tota l'aplicació. A partir d'ella es pot arribar a totes les parts que la constitueixen i és el punt comú que les uneix.

9.1.1. Capçalera

La part superior de la pàgina té una capçalera del color blau fosc corporatiu de l'empresa amb el títol "Inici" en color blanc en ella. Aquest és un component visual que es manté en totes les pàgines de l'aplicació, amb el títol corresponent de la pàgina on es troba l'usuari.

9.1.2. Contingut

En el cos de la pàgina es veuen quatre apartats, cada un d'ells amb la seva icona i títol corresponent i amb una fletxa al costat dret que apunta cap a la dreta. Els títols són: "Errores", "FAQs", "Vídeos Tutoriales" i "Documentos". Aquestes són les quatre seccions de l'aplicació, a les quals s'accedeix tocant en qualsevol part del seu apartat. Sempre que es navegui a una secció, o simplement es passi d'una pàgina de l'aplicació a una altra, es podrà retornar a la pàgina anterior fent clic sobre la fletxa blanca orientada cap a l'esquerra que apareix en el costat esquerre de la capçalera.

Secció Errors

La secció "Errores" és la que inclou el llistat dels diferents errors que poden obtenir els operaris amb els variadors de freqüència de l'empresa. Accedint a cadascun d'ells es pot realitzar un wizard assistent, basat en preguntes sobre les característiques i la configuració de l'equip, per intentar solucionar l'error obtingut i fer que el variador torni a funcionar correctament.

Secció FAQs

La secció "FAQs" és una secció bastant popular i utilitzada per totes les empreses. Les seves sigles, en anglès *Frequently Asked Questions*, corresponen en català a preguntes fetes freqüentment. I bàsicament consisteix en això, en un seguit de preguntes relacionades

amb els variadors, presentades en forma de llista, que es considera que poden ser de gran utilitat per a l'usuari. És una de les seccions que més s'ha potenciat, ja que s'ha considerat que el client pot valorar molt aquesta ajuda i fer un gran ús d'ella. Les preguntes abasten un ampli conjunt de situacions en què l'usuari es pot trobar, de forma que la secció pugui servir a l'operari com a font d'informació i de consultes sense haver de contactar personalment amb el servei tècnic.

Secció Vídeos Tutorialis

La secció "Vídeos Tutorialis" consisteix en un seguit de petits vídeos que mostren com realitzar certes accions sobre els equips. Aquesta secció va sorgir amb la idea de complementar les FAQs, afegint valor visual a les explicacions, ja que diuen que una imatge val més que mil paraules. També part dels vídeos d'aquesta secció es troben en algunes de les passes dels wizards solucionadors d'errors, quan es tracta d'alguna configuració específica que s'ha de realitzar i es disposa del vídeo explicatiu.

Secció Documents

Per últim, aquesta secció pretén posar a disposició dels clients certs manuals o arxius perquè se'ls pugui descarregar en format digital en el seu dispositiu mòbil.

9.1.3. Peu de pàgina

El peu de pàgina està format per una fila amb quatre icones: una casa, un telèfon, una bola del món i un engranatge. De la mateixa forma que la capçalera, aquest peu de pàgina serà el mateix per a totes les pàgines de l'aplicació. Cada icona és un botó que porta associat una funció diferent.

La casa permet retornar a la pàgina d'inici des de qualsevol pàgina de l'aplicació fent clic sobre ella. A diferència de la navegació entre diferents pàgines, aquesta acció no permetrà tornar a la pàgina en la qual s'estava prèviament.

Clicant sobre el telèfon la funció que s'activa és la de trucar a servei tècnic de Tecnotrans Bonfiglioli. Es considera una acció molt útil la de poder trucar directament al servei tècnic mitjançant un botó, pel fet que simplifica la feina a l'usuari. Aquest no ha de marcar cap número, ni anotar-ho o aprendre-se'l abans de teclejar-ho en el dispositiu per a poder fer la trucada.

La icona del globus terraquí és un enllaç a la pàgina web de l'empresa. Clicant sobre ella s'obrirà el navegador del dispositiu mòbil en la pàgina web oficial de Tecnotrans. També és una eina que simplifica la feina al client per si vol consultar algun tipus d'informació.

Per últim, la icona de l'engrenatge és la que permetrà a l'usuari canviar l'idioma de l'aplicació. S'ha considerat que la distribució i venda de l'aplicació pot arribar a traspasar les fronteres nacionals i, probablement, els operaris d'empreses d'altres països no sàpiguen espanyol, per tant s'ha d'adequar l'aplicació perquè tothom pugui gaudir del seu ús. En clicar sobre l'engrenatge apareixerà una petita finestra enmig de la pantalla indicant els idiomes disponibles. L'usuari haurà de seleccionar la llengua que desitja i confirmar-ho prement OK. En aquest moment la visualització de l'aplicació serà la mateixa però amb l'idioma que ha escollit l'usuari.

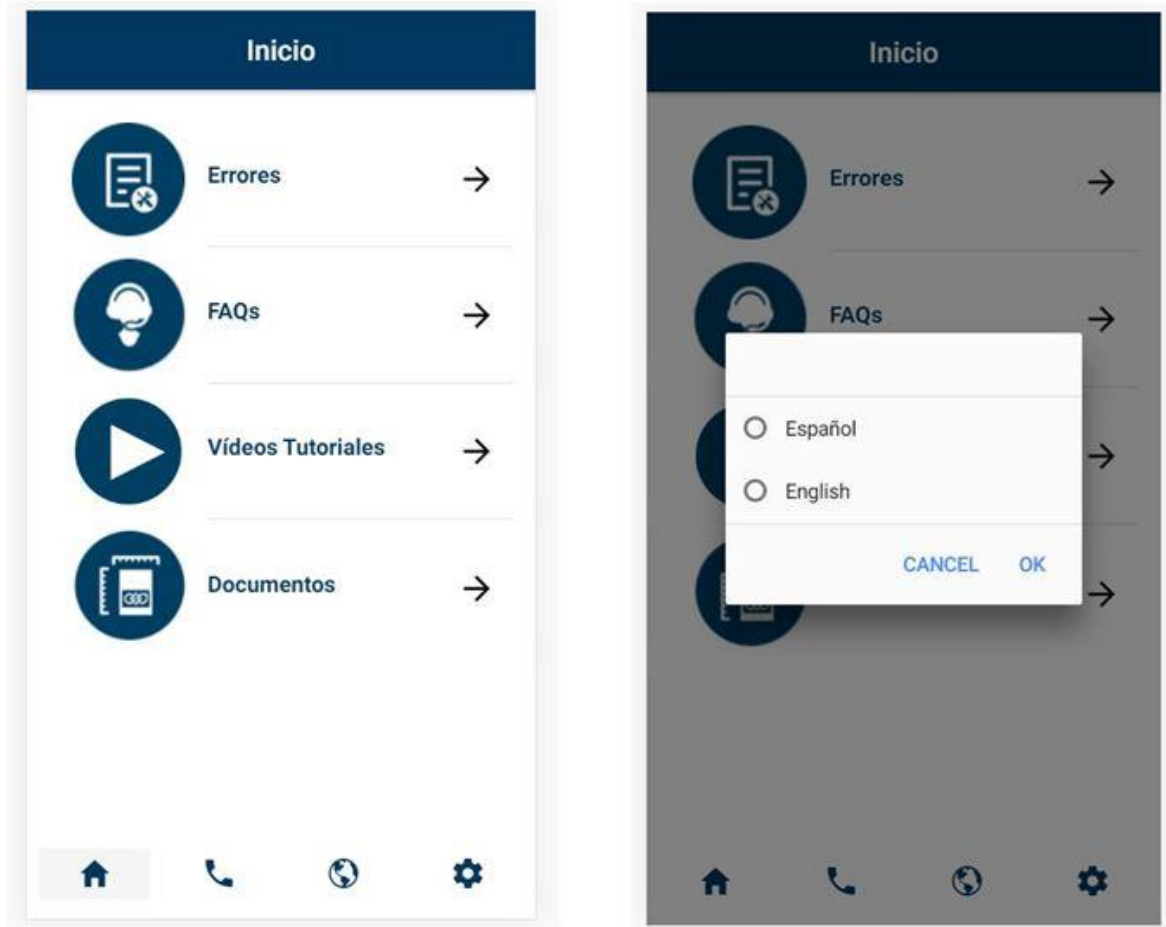


Figura 17: Pàgina d'inici de l'aplicació i finestra emergent per canviar l'idioma. Font: pròpia.

9.2. Secció Errors

9.2.1. Pàgina d'errors

Aquesta pàgina està formada per la capçalera amb el títol de la secció, un buscador i un llistat de codis d'error.

El buscador duu a terme la seva funció òbvia de, en escriure un text, mostrar en pantalla únicament aquells elements que es corresponen al que s'ha inserit en el buscador.

Cada codi d'error és un element que té la funcionalitat implementada de, al ser polsat, traslladar a l'usuari a la pàgina de presentació de l'error seleccionat.

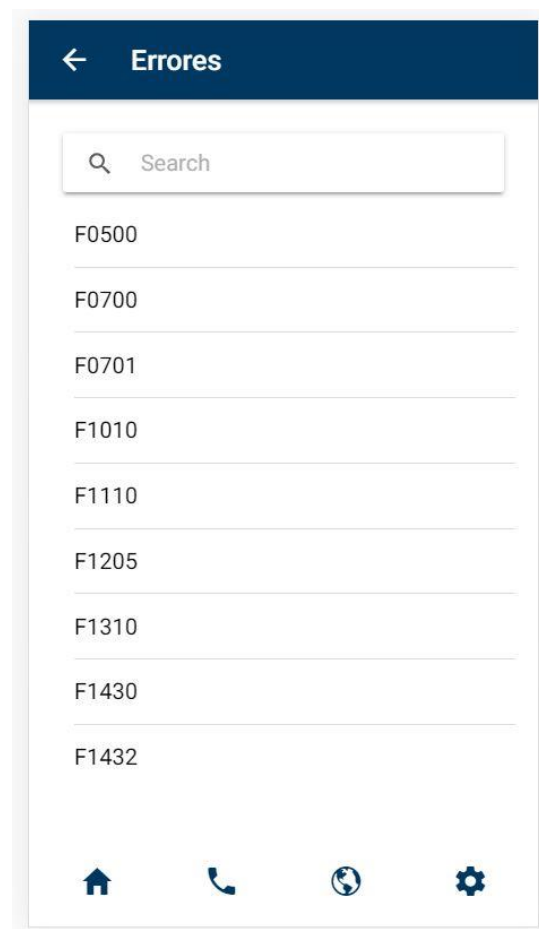


Figura 18: Pàgina d'errors. Font: pròpia.

9.2.2. Pàgina de presentació d'error

En entrar a un error la primera pàgina que apareix és la seva pàgina de presentació, on es mostra informació general sobre l'error escollit.

El títol de la capçalera és el codi de l'error en qüestió.

En el contingut de la pàgina es veu el títol de l'error, quin component ha provocat l'error i la seva causa. També apareix informació addicional, com per exemple possibles situacions que hagin sigut les responsables de produir la causa que ha desencadenat en l'error.

A sota de tota aquesta informació de text es troba un botó, amb la frase "Go to wizard", el qual si es prem condueix a l'usuari a començar un solucionador basat en preguntes sobre l'equip, amb l'objectiu d'ajudar a l'usuari a resoldre el seu error.

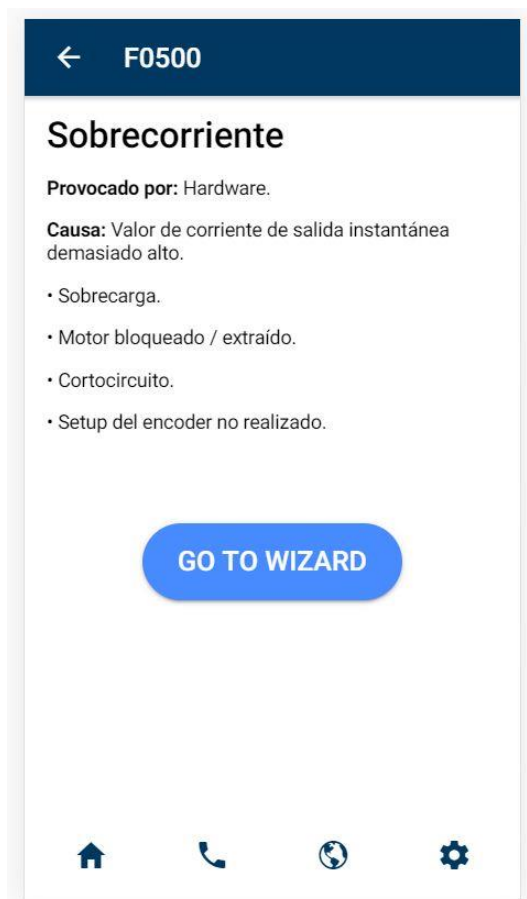


Figura 19: Pàgina de presentació de l'error F0500. Font: pròpia.

9.2.3. Pàgina del solucionador d'errors

En prémer el botó de la pàgina de presentació d'un error l'usuari comença el wizard solucionador de l'error.

En la pantalla es mostra una pregunta relacionada amb les condicions en les quals es troba l'equip, la seva configuració o les accions dutes a terme abans que succeís la fallada entre d'altres. Sempre hi haurà, com a mínim, dos botons amb dues respostes possibles, normalment corresponents a Sí i No.

En polsar el botó de la resposta escollida per l'operari, es passa a una altra pàgina exactament igual però amb una pregunta diferent. Aquest procés continua fins que l'usuari aconsegueix solucionar el seu error o es determina que l'equip es troba avariats.

Les preguntes de les pàgines del solucionador d'errors poden venir acompanyades d'imatges, enllaços per descarregar manuals, vídeos tutorials o explicacions d'un procediment a realitzar.

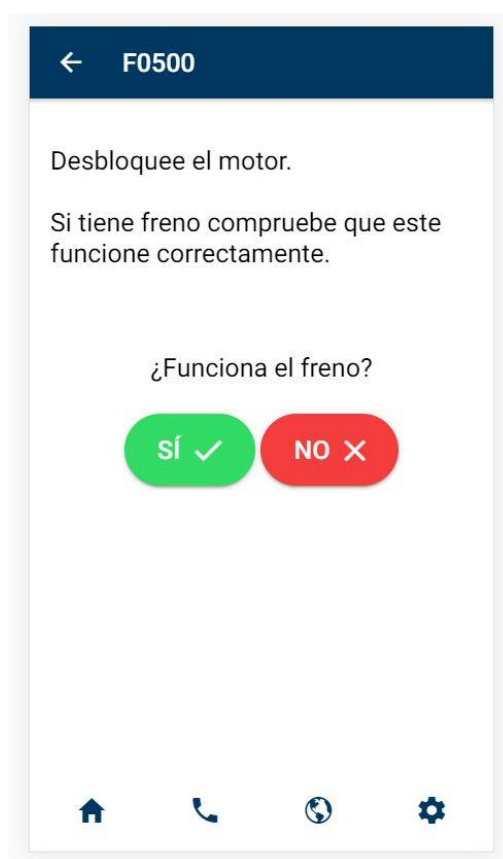


Figura 20: Pàgina del wizard de l'error F0500. Font: pròpia.

9.2.4. Pàgina final d'un error

Un cop s'ha completat el wizard solucionador d'errors es pot arribar a tres situacions diferents: que s'hagi solucionat l'error i l'equip torni a funcionar, que es determini el motiu pel qual l'equip no funcionava i es doni una indicació de com solucionar-ho, o que no es pugui saber amb exactitud que és concretament el que està provocant l'error i s'hagi de contactar amb el servei tècnic de Tecnotrans Bonfiglioli.

En el cas de la primera situació s'arriba a una pàgina que diu "Problem Solved" amb una emoticona somrient i amb el polze aixecat indicant satisfacció.

En la segona situació s'arriba a una pàgina on s'indica el motiu de l'error i, si és possible, també s'indica com solucionar-ho.

En el cas que no s'hagi pogut determinar com solucionar l'error amb el wizard, s'arriba a una pàgina on s'expressa que es truqui a servei tècnic i apareix un botó que, clicant-lo, directament connecta amb servei del dispositiu mòbil per fer trucades i truca al número de servei tècnic de l'empresa.

En les dues primeres situacions, a sota de les parts ja explicades que apareixen a cadascuna de les pàgines, es troba un element comú. Es tracta d'una secció que diu "Por favor, a continuación rellene el siguiente formulario". Clicant sobre aquesta secció es dirigeix a l'usuari a la pàgina Report, on haurà d'emplenar un petit formulari que s'enviarà a Tecnotrans per tenir constància de com es resolen els errors. Aquesta pàgina està explicada en més detall en el següent punt.

Després hi ha un element que és comú en els tres casos. Consisteix en un botó amb la frase "Go To Summary", el qual si es clica condueix a una pàgina amb un resum del què s'ha anat contestant al llarg del solucionador d'errors. Aquesta pàgina, referida com Pàgina Summary, també es troba explicada en més detall en els següents punts.

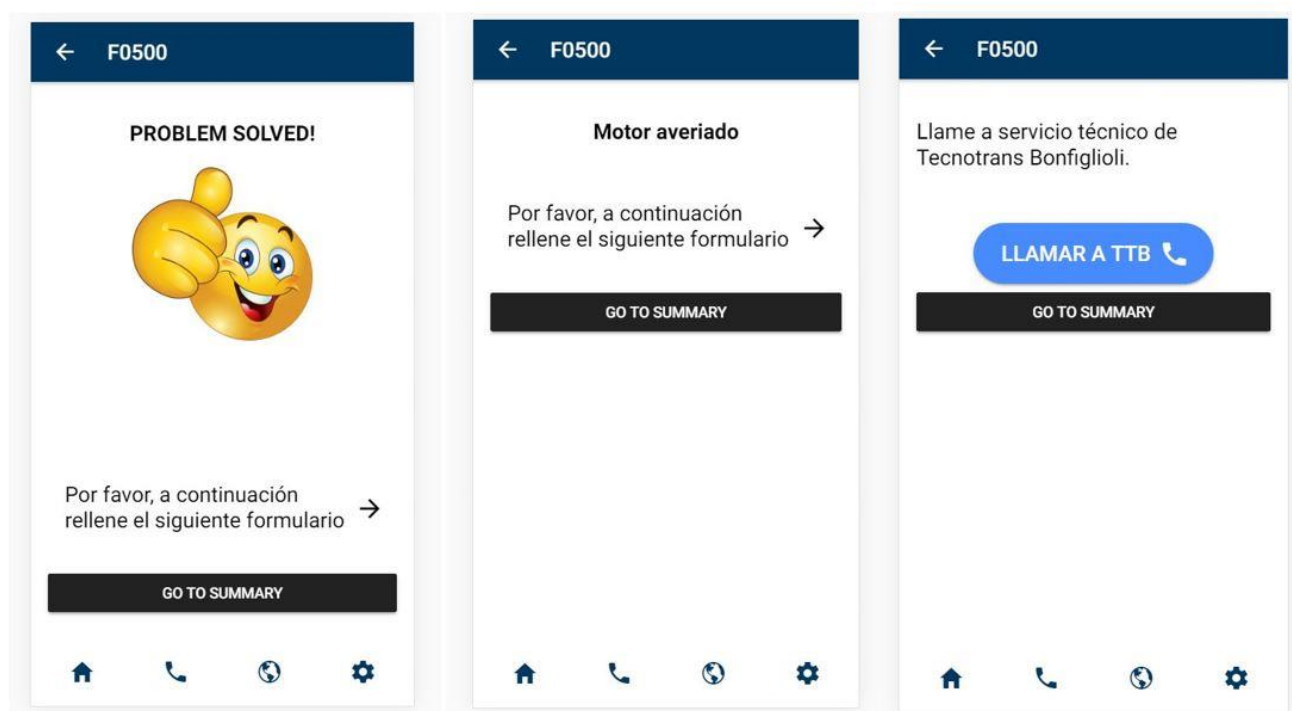


Figura 21: Pàgines finals possibles dels errors. Font: pròpia.

9.2.5. Pàgina Report

Aquesta és una pàgina a la qual s'arriba sempre després d'haver acabat el wizard d'algun error. El seu objectiu és recollir informació i enviar-la a una base de dades per a poder tenir una certa retroacció de l'ús de l'aplicació.

Principalment es desitja saber quant ús s'està fent de l'aplicació i quines empreses són les que l'estan utilitzant. També interessa saber quins són els errors que s'estan consultant amb més freqüència i si l'usuari ha pogut arribar a solucionar-los o a trobar el seu desencadenant.

A través d'aquesta informació es pot estimar l'impacte de l'aplicació sobre els clients i també permet fer un estudi de quins són els errors més consultats i poder millorar els equips, en el cas que hi hagi una tendència a produir-se algun error en concret.

La pàgina conté un formulari que està format per cinc apartats obligatoris i un opcional a complementar per l'usuari a l'hora d'enviar el formulari. Els sis apartats corresponen a entrades de text. Els cinc primers són: el nom de l'usuari, l'empresa en la qual treballa, la ubicació d'aquesta, el codi de l'error que acaba de consultar i un correu electrònic de contacte. L'apartat opcional correspon a comentaris que l'usuari vulgui fer.

A sota d'aquestes entrades hi ha un botó d'enviar que només s'activa quan les entrades de caràcter obligatori s'han complimentat. En polsar-lo s'envia el formulari amb la seva informació a la base de dades i apareix un missatge emergent que dura 3 segons en la pantalla informant que el formulari s'ha enviat correctament.

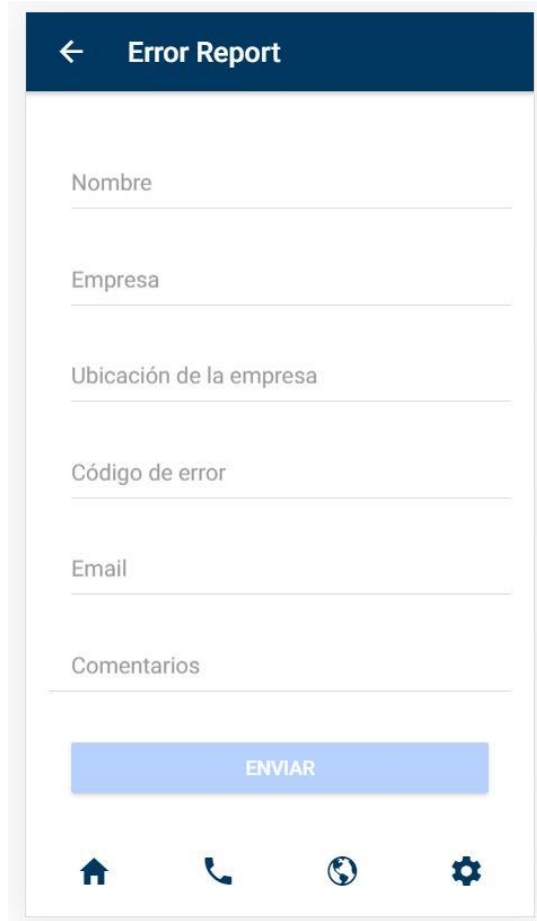


Figura 22: Pàgina Report amb el botó d'enviar desactivat. Font: pròpia.

9.2.6. Pàgina Summary

De la mateixa manera que la pàgina Report, a aquesta pàgina s'accedeix una vegada l'usuari ha completat un wizard solucionador d'algun error.

L'objectiu d'aquesta pàgina és mostrar a l'usuari les respostes que ha anat donant a les preguntes que li anaven apareixent al llarg del wizard solucionador d'errors. Aquesta informació pot ser de gran valor si finalment l'usuari no ha pogut solucionar l'error i es posa en contacte amb Tecnotrans Bonfiglioli, i poder posar en situació als tècnics de les passes que ha seguit i informar d'alguna possible modificació que hagi pogut fer sobre l'equip.

La pàgina Summary, com ja s'ha mencionat anteriorment, simplement presenta una llista

amb les respostes a les preguntes que s'han respost al llarg del wizard. Aquesta llista s'obté gràcies al fet que cada vegada que es passa per una pàgina del solucionador d'errors i s'accedeix a una altra, s'envien uns paràmetres amb la resposta de l'usuari. Fins que s'arriba a la pàgina final d'un error, on es recullen en una llista tots els paràmetres que s'han anat enviant. En accedir a la pàgina de Summary, aquesta llista també s'envia en forma de paràmetre des de la pàgina final d'error fins a la pròpia pàgina de Summary, i es mostra a l'usuari.

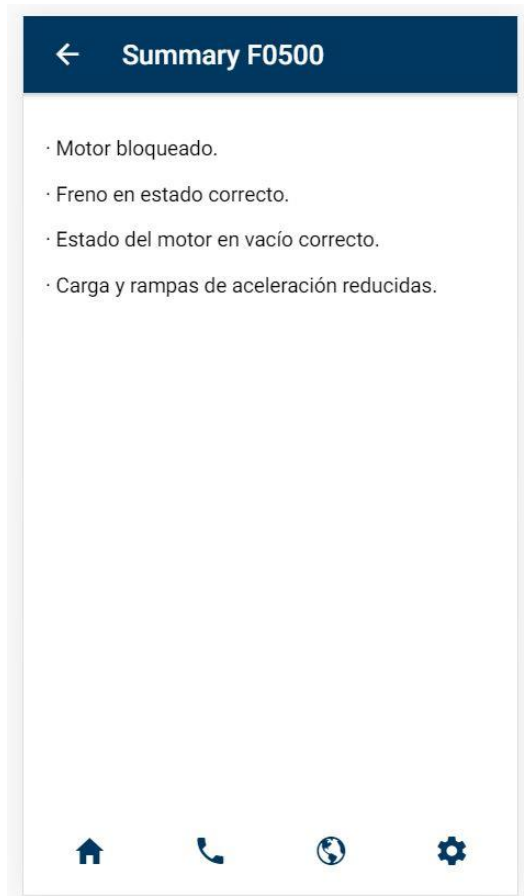


Figura 23: Pàgina Summary de l'error F0500. Font: pròpia.

9.3. Secció FAQs

Aquest és un apartat que pràcticament qualsevol portal web de qualsevol empresa incorpora, ja que és una gran ajuda per als usuaris.

És un dels grans potencials de l'aplicació, degut a que permet resoldre molts dubtes interessants als clients sobre els equips sense la necessitat d'haver de contactar personalment amb l'empresa.

La pàgina d'aquesta secció està constituïda per un buscador on es pot escriure el dubte que s'està buscant o la paraula clau d'aquesta, de forma que s'agilitza el procés de recerca de l'explicació o la pregunta que s'està buscant.

A sota del buscador es troben en forma de llista tots els enunciats de les preguntes disponibles, separades per un rètol segons la seva temàtica.

Per accedir a les respostes de les preguntes només cal clicar sobre aquestes, i es redirigirà a una pàgina on es veu la pregunta seleccionada en negreta i a continuació la seva resposta.

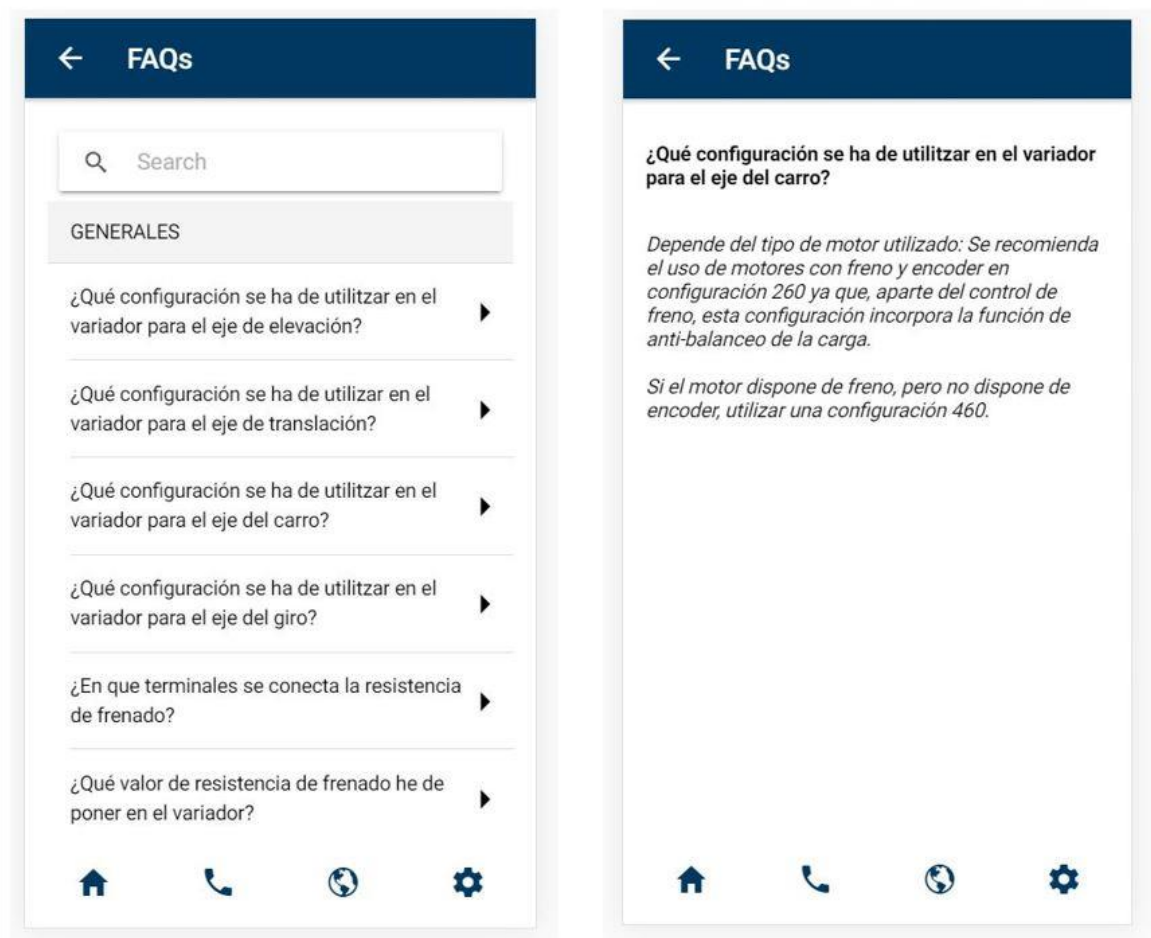


Figura 24: Pàgines de la secció FAQs. Font: pròpia.

9.4. Secció Vídeos Tutorials

La secció de vídeos tutorials és un apartat bastant intuïtiu. Com ja s'ha mencionat anteriorment, té l'objectiu de complementar la secció de FAQs aportant contingut audiovisual a possibles dubtes que tinguin els usuaris. Consisteix en un conjunt de vídeos amb explicacions curtes, però pràctiques, sobre com realitzar certes configuracions o modificacions sobre els equips que distribueix Tecnotrans Bonfiglioli.

Entrant en aquesta secció des de la pàgina d'inici, clicant sobre el seu apartat corresponent, s'arriba a una pàgina amb un format semblant al de la pàgina d'inici. Pel moment només hi ha un apartat, amb la seva imatge, títol i fletxa en el costat esquerre corresponent, però es té la intenció d'ampliar el seu contingut en el futur. Aquests apartats són els temes sobre els quals l'aplicació conté un conjunt vídeos tutorials. Clicant sobre la fila de cada tema es passa a una altra pàgina, ara sí, amb una llista amb els títols de cada vídeo que l'usuari disposa per a la seva visualització.

Per accedir a cada vídeo, de la mateixa manera que anteriorment, l'usuari ha de clicar sobre el títol del vídeo perquè el redirigeixi a una altra pàgina de l'aplicació. En aquesta pàgina, finalment, es troba, contingut en un marc, el vídeo tutorial escollit.

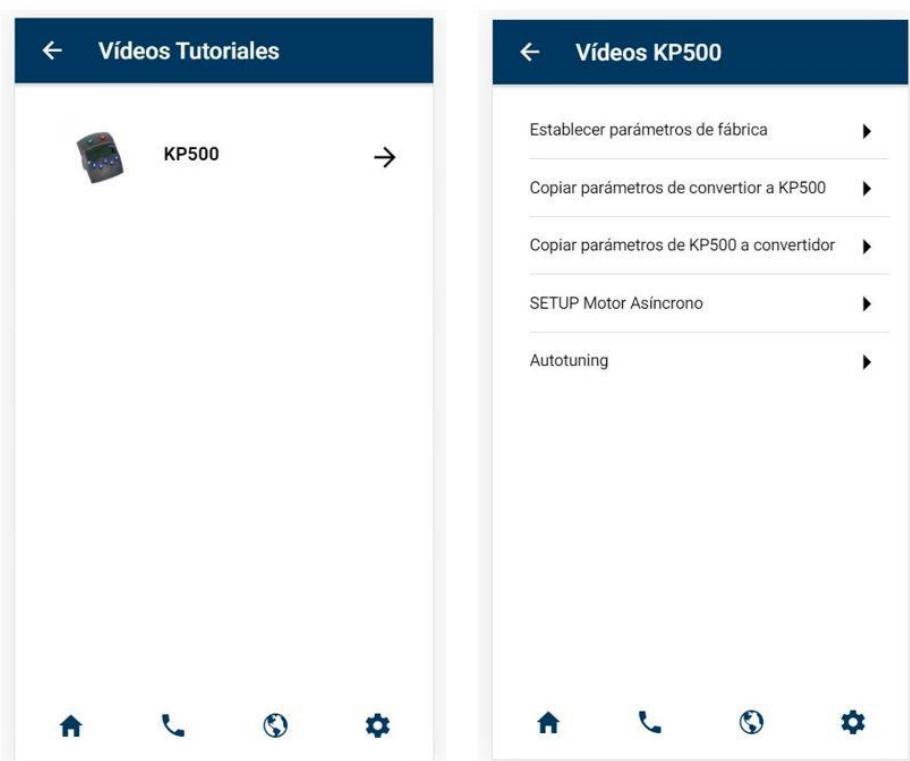


Figura 25: Pàgines de la secció d'errors prèvies a la pàgina que conté el vídeo. Font: pròpia.

La visualització dels vídeos és molt senzilla. Aquests es troben en format miniatura de Youtube, incrustats en un marc. Això és degut a que realment són enllaços a vídeos d'un canal privat de l'empresa a Youtube. Aquesta configuració també permet a l'usuari la possibilitat de poder utilitzar tota la pantalla del seu dispositiu per a reproduir el vídeo, tant en una posició vertical de l'aparell com horitzontal.



Figura 26: Pàgina que conté un vídeo tutorial. Font: pròpia.

9.5. Secció Documents

A aquesta secció s'accedeix de la mateixa manera que a les altres, des de la pàgina d'inici fent clic sobre el seu apartat. És una secció molt senzilla i bàsica però que pot ser de gran utilitat per a l'usuari.

En entrar a la pàgina s'observen quatre apartats. El primer és un apartat genèric i correspon a les solucions mecatròniques i als variadors. En clicar sobre ell apareix un botó a sota seu per descarregar el catàleg de producte dels diferents variadors que distribueix l'empresa i les seves possibles funcionalitats.

Els següents tres apartats corresponen als tres tipus de variadors més utilitzats en

aplicacions per a grues. En clicar sobre ells, de la mateixa manera que amb el primer apartat, apareixen a sota seu diferents botons per descarregar en el dispositiu documents i manuals referents a les característiques, al funcionament i a les aplicacions de cada un dels variadors de freqüència.

En clicar sobre cada apartat no només apareixen els documents associats a aquest, sinó que la fletxa de la dreta passa d'apuntar cap a baix a fer-ho cap a dalt. Si es torna a clicar sobre l'apartat, es restableix la seva visualització inicial. Els botons per descarregar els documents s'amaguen i la fletxa de la dreta torna a apuntar cap a baix.

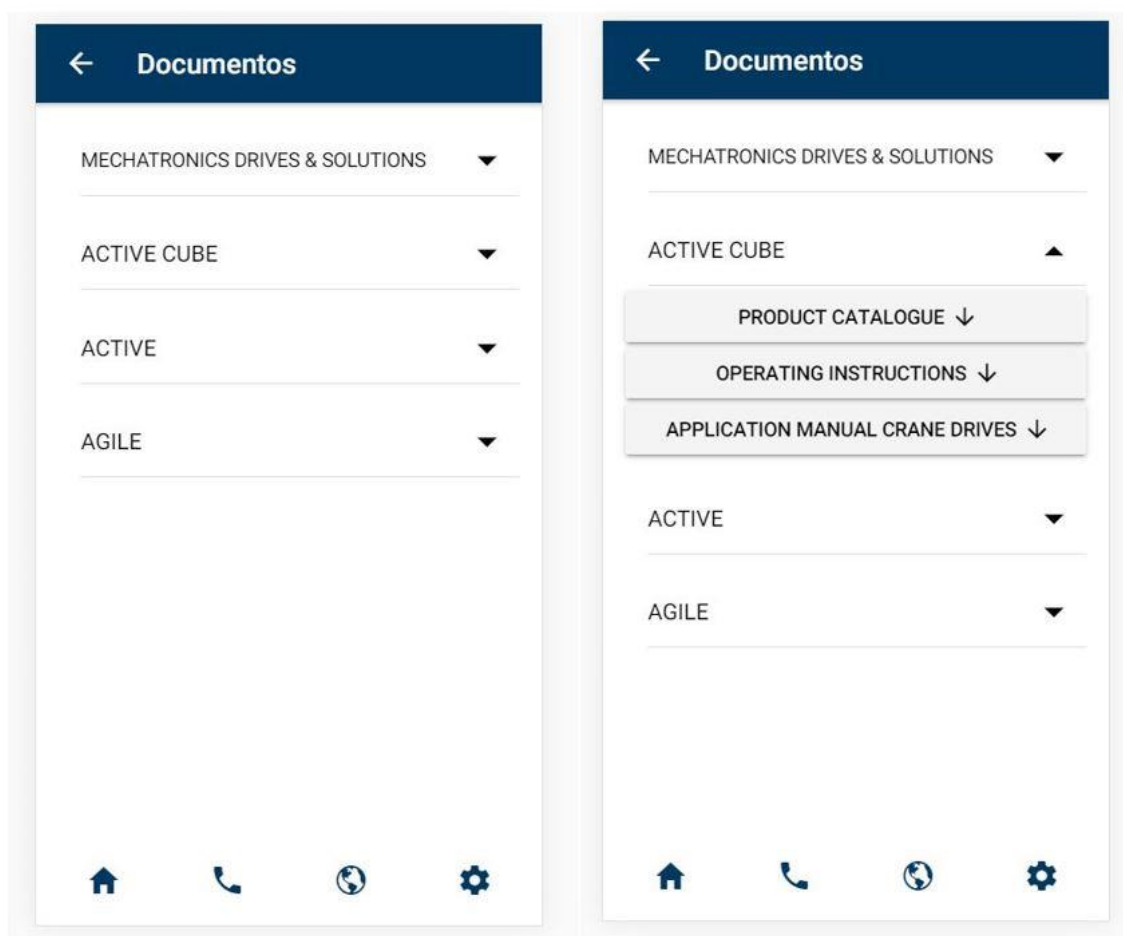


Figura 27: Pàgina de la secció Documents. Font: pròpia.

10. Principals accions i característiques

Aquest apartat pretén explicar, d'una forma molt superficial, com funcionen algunes de les principals accions que formen part del comportament de l'aplicació i quins recursos s'han utilitzat per a programar-les.

Abans caldria fer una breu explicació de com funciona Angular per a posar-se en context.

Angular utilitza classes i objectes per construir la lògica de les seves pàgines i poder fer el two-way data binding, ja mencionat anteriorment.

Una classe és un element constructiu dels llenguatges de programació. Està definida per un conjunt d'atributs i mètodes, que confeccionen l'estat i el comportament de la classe, respectivament. Els atributs són variables o propietats internes de la classe que poden ser de tipus numèric, text, booleà o fins i tot un objecte, mentre que els mètodes són accions que poden realitzar les classes.

Un objecte és també un element constructiu dels llenguatges de programació definit per uns atributs i mètodes, però normalment aquest es crea a partir d'alguna classe, que serveix com a plantilla. L'objecte creat a partir d'una classe estarà definit pels atributs i mètodes de la classe a partir de la qual s'ha creat, a part dels propis que se li vulguin afegir.

10.1. Esdeveniment click

L'esdeveniment *click* [41] és un dels recursos més importants i més utilitzats dins del desenvolupament de l'aplicació.

El propòsit d'aquest esdeveniment és atribuir una funció a un element del front-end quan es clica sobre ell. Com per exemple, atribuir la funció de canviar de pàgina a un botó quan aquest es clica, una acció molt recurrent en el funcionament de l'aplicació.

L'esdeveniment s'ha d'escriure en l'arxiu HTML, dins de la marca d'obertura de l'element al qual volem afegir-li la funció al ser clicat. La forma en què s'ha d'escriure és *(click)= "nom de la funció()*". Dins dels parèntesis es pot escriure el nom d'algun paràmetre en cas que la funció el requereixi.

Per una altra banda, la funció s'ha de definir en l'arxiu TypeScript de la mateixa pàgina que l'arxiu HTML on s'ha presentat aquesta funció associada a l'esdeveniment *click*. En l'arxiu TypeScript s'ha de definir la funció amb exactament el mateix nom amb el qual s'ha escrit en l'arxiu HTML, en cas contrari l'aplicació no reaccionarà quan es faci clic sobre l'element.

Una vegada dins de la funció, per a poder definir quina acció durà a terme aquesta, s'ha de fer ús dels mètodes de les classes. A aquests mètodes, sovint, se li ha de proporcionar algun tipus d'informació perquè puguin dur a terme la seva tasca. Es pot definir qualsevol acció que sigui realitzable per Ionic i Angular.

10.2. Navegació

La navegació consisteix en el mètode que permet moure's entre les diferents pàgines de l'aplicació.

La classe que permet realitzar la navegació és *NavController* [42]. Aquesta classe té un gran nombre de mètodes, encara que per al desenvolupament de l'aplicació principalment només s'han utilitzat dos: *push* i *setRoot*.

El mètode *push* s'utilitza per passar d'una pàgina a una altra de forma que queda un rastre del camí que es va seguint. Això permet que, en cas que es vulgui retornar a les pàgines des de les quals s'ha accedit a la pàgina actual en la qual es troba l'usuari, es pugui fer sense cap problema.

Utilitzant la classe *NavController* i el mètode *push*, juntament amb l'esdeveniment *click*, s'han definit les funcions per a què els elements de l'aplicació, com per exemple els botons dels wizards, defineixen els camins que seguirà l'usuari.

El mètode *setRoot* també permet navegar entre diferents pàgines de l'aplicació, però en aquest cas no deixa cap rastre del camí que s'ha seguit. Directament dirigeix a l'usuari a la pàgina desitjada, sense la possibilitat de tornar enrere.

El seu ús ha estat emprat per retornar a la pàgina d'inici des de qualsevol pàgina de l'aplicació. S'ha definit la funció juntament amb l'esdeveniment *click*, i se li ha assignat al botó de la caseta de la barra inferior de l'aplicació.

10.3. Transmissió i visualització de paràmetres

La transmissió de paràmetres no es realitza de forma independent, sinó que es duu a terme conjuntament amb la navegació. Quan es passa d'una pàgina de l'aplicació a una altra, no només es canvia la visualització que observa l'usuari a la pantalla, sinó que també es té la possibilitat que s'enviïn paràmetres entre les pàgines que s'ha navegat.

Per tant, per a la transmissió de paràmetres entre pàgines de l'aplicació són necessàries dues classes: *NavParams* i *NavController*. La classe *NavController* és l'encarregada d'enviar

els paràmetres des de la pàgina d'origen, i la classe *NavParams* [43] és qui els recull en la pàgina de destí.

El procediment per enviar els paràmetres consisteix en generar un objecte i empaquetar en ell els paràmetres. A aquest objecte se li han d'afegir atributs. Cada atribut està compost per un nom i un valor. El nom és la forma de referir-se a l'atribut, i el valor és la informació que conté. En cas que es vulgui mostrar en la pantalla de l'aplicació el paràmetre, el que es podrà mostrar és el valor de l'atribut, mai el seu nom.

Utilitzant el mètode *push*, mencionat en l'apartat anterior, s'envia l'objecte que conté els paràmetres a la pàgina de destí a la que es navega.

El mètode utilitzat per recollir l'objecte enviat s'anomena *get* i pertany a la classe *NavParams*.

Amb la intenció de seguir enviant els paràmetres a una altra pàgina o fer-ne ús d'ells en la pàgina en què es reben, es crea un altre objecte en la pàgina de destí i se li assignen els atributs de l'objecte rebut.

Per a poder visualitzar aquests paràmetres en la nostra aplicació simplement s'ha de dur a terme una interpolació entre els arxius del front-end i el back-end. És un procés molt senzill que consisteix a escriure en l'arxiu HTML, dins de claus dobles, el nom de l'objecte que s'ha rebut i que conté els paràmetres, seguit del nom de l'atribut que es vol mostrar de la següent manera: `{{nom_objecte.nom_atribut}}`. El que l'usuari podrà veure a través de la pantalla serà el valor de l'atribut.

Aquesta classe ha estat utilitzada bàsicament per a tenir un registre del qual l'usuari ha respost i el camí que ha seguit durant el wizard del solucionador d'errors. Cada vegada que l'usuari passa per una pàgina es defineix un nou atribut, que té un valor diferent en funció del què ha respost en aquesta pàgina. Quan s'arriba a la pàgina Summary en finalitzar un error, s'obté un objecte amb un conjunt d'atributs que són el registre de tots els paràmetres que s'han anat enviant pàgina a pàgina durant el wizard. Finalment, fent una interpolació, els paràmetres es visualitzen en la pàgina Summary perquè l'usuari pugui tenir constància de les seves respostes, durant el wizard.

10.4. Plugin Call Number

El plugin Call Number [44] és qui permet connectar l'aplicació amb el recurs del dispositiu mòbil que permet establir les trucades, amb l'objectiu evident, de fer una trucada telefònica.

Per a poder utilitzar aquest plugin, primer aquest s'ha d'instal·lar en l'aplicació des de

Node.js a través del seu administrador npm mitjançant el CLI de Ionic.

Tots els plugins porten associats una classe que permet donar les ordres per utilitzar el plugin en qüestió. En aquest cas la classe s'anomena *CallNumber*.

El mètode que permet fer trucades s'anomena *callNumber*, i se li ha de proporcionar el número de telèfon al qual es desitja trucar.

La manera perquè l'usuari pugui fer una trucada des de l'aplicació involucra a l'esdeveniment *click* i a la classe *CallNumber*. Mitjançant l'esdeveniment *click*, en l'arxiu HTML s'atribueix a un botó la funció per fer la trucada, i aquesta funció es defineix en l'arxiu TypeScript utilitzant la classe *CallNumber* i el seu mètode *callNumber*.

10.5. Idiomes

Per a poder dotar l'aplicació de diferents idiomes s'ha d'utilitzar la llibreria *ngx-translate* [45] [46]. Aquesta llibreria s'ha d'instal·lar a l'aplicació des del servei Node.js a través del seu administrador npm utilitzant el CLI de Ionic. Un cop instal·lada la llibreria s'han d'importar les dependències necessàries per a poder utilitzar-la, com són les diferents classes que s'utilitzen per a la seva configuració i ús.

Primerament s'han de crear uns arxius de tipus json. Un arxiu json és un tipus d'arxiu que permet emmagatzemar dades i informació i també transferir-la. En el cas de l'aplicació, els arxius json contindran el text que es vulgui presentar a l'usuari en diferents idiomes. Per tant, es necessita un arxiu json per a cada idioma que es vulgui utilitzar.

Les dades contingudes en aquests arxius consten de dos elements principals, les claus i els valors. Les claus són el nom amb el qual es refereix a la dada que volem accedir, i el valor és la informació que conté aquesta dada. Per exemple, si es vol mostrar el títol d'una pàgina en diferents idiomes, la clau podria ser anomenada "títol" i el valor seria el títol en si en l'idioma que es vulgui tenir. Per a tenir una frase en dos idiomes la clau ha de ser la mateixa en els dos arxius json, i la frase, el valor d'aquesta clau, ha de ser en l'idioma que li correspongui en el seu respectiu arxiu.

Pel que fa a les classes necessàries per a poder definir en quin idioma es vol veure el contingut de l'aplicació i també per a canviar a un altre, la classe més important és *TranslateService*.

Mitjançant el seu mètode *setDefaultLang* es defineix l'idioma per defecte en el qual apareixerà l'aplicació quan l'obri l'usuari. Per a poder-ho establir se li ha d'aportar el nom de l'arxiu que conté l'idioma que volem fixar per defecte.

L'altre mètode més utilitzat és *use*. Amb aquest mètode es podrà canviar l'idioma que s'està utilitzant en aquell moment en l'aplicació. De la mateixa manera que amb el mètode anterior, a *use* també se li ha de proporcionar el nom de l'arxiu json que conté l'idioma al qual es vol canviar.

La forma per a canviar d'idioma que està implementada en l'aplicació és a través del botó amb la icona de l'engranatge de la barra inferior, present a totes les pàgines. Utilitzant l'esdeveniment *click* i una funció definida, quan es clica sobre aquesta icona apareix una petita finestra enmig de la pantalla amb els diferents idiomes disponibles. S'ha de seleccionar l'idioma que es vol utilitzar i confirmar-ho amb el botó OK.

Quan es selecciona l'idioma el que realment s'està fent és seleccionar el nom de l'arxiu json que es vol utilitzar, i en clicar sobre OK s'està cridant a la funció que conté el mètode *use* per a canviar l'idioma.

10.6. Base de dades i enviament del formulari

Per a crear una base de dades on rebre el formulari que envien els usuaris de l'aplicació en acabar un wizard solucionador d'algun error s'ha fet servir Firebase, en concret el seu servei de Database.

El primer pas és crear un projecte en Firebase i sincronitzar-lo amb l'aplicació. Per a poder establir una connexió amb Firebase es necessiten dues llibreries que s'han d'instal·lar des de Node.js a través del seu administrador npm utilitzant el CLI de Ionic, *angularfire2* i *firebase*.

Per sincronitzar el projecte de Firebase amb l'aplicació s'ha d'utilitzar la classe *AngularFireModule*, de la llibreria *angularfire2*, i el mètode *initializeApp*, al qual se li han de proporcionar unes credencials que s'obtenen en el servei de Database, dins del projecte de Firebase.

D'aquesta manera ja s'ha aconseguit connectar l'aplicació amb la base de dades. La següent passa és configurar el formulari per preparar-lo per ser enviat. Utilitzant les classes *FormBuilder* i *FormGroup* i el mètode *group* es crea un objecte on es guarden en forma d'atribut les credencials introduïdes per l'usuari en els diferents apartats del formulari. En el procés de creació d'aquest objecte es defineix si un apartat és obligatori d'omplir o no amb la classe *Validators* i el mètode *required*.

Per últim, per enviar el formulari és necessari crear un servei on es defineixi una funció per a poder dur a terme aquest enviament. En el servei s'utilitza la classe *AngularFirestore* juntament amb el seu mètode *add*, que és l'encarregat d'enviar el formulari i afegir-lo a una

col·lecció en la base de dades de Firebase. En clicar el botó que hi ha a sota del formulari en l'aplicació, es crida, mitjançant l'esdeveniment *click*, a la funció del servei que conté el mètode *add*. A aquest mètode se li ha de proporcionar l'objecte que conté les credencials introduïdes per l'usuari. També se li ha d'indicar a quina col·lecció de la base de dades es vol afegir el formulari enviat. Si aquesta col·lecció no existeix, Firebase Database la crea quan se li envia el primer formulari.

11. Android

Un cop l'aplicació ja està totalment desenvolupada, el següent pas és transformar-la en una aplicació nativa per a poder instal·lar-la en un dispositiu mòbil com a tal i que així el seu rendiment i l'experiència d'usuari siguin majors i de més qualitat.

Segons fonts estadístiques de portals web com Statcounter i Net Market Share, a data d'abril 2020, els sistemes operatius Android i iOS, són els sistemes operatius més utilitzats en dispositius mòbils en tot el món, amb un percentatge de més del 99% conjuntament.

Tot i això, entre aquests dos sistemes operatius també hi ha una gran diferència, ja que Statcounter informa que, també a data d'abril 2020, els dispositius mòbils amb sistema operatiu Android suposen un 70,68% dels mòbils en el món, mentre que iOS un 28,79% [47]. Net Market Share també ofereix unes dades similars, els mòbils amb Android com a sistema operatiu suposarien un percentatge del 73,53% dels mòbils en el món i els mòbils amb iOS un 25,99% [48].

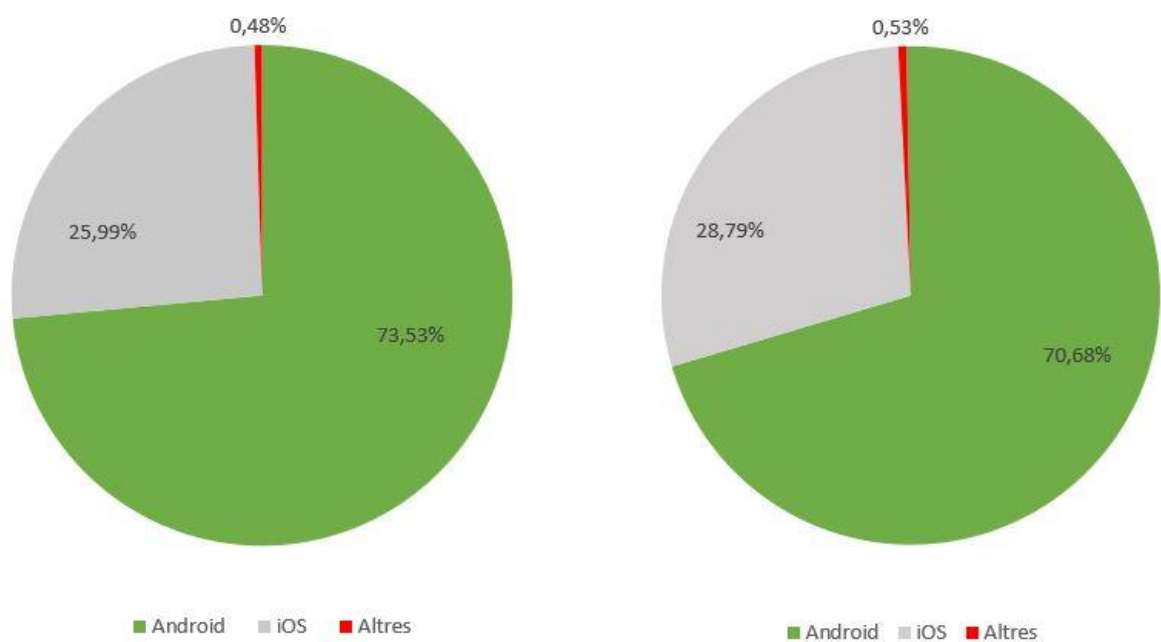


Figura 28: Gràfics de l'ús dels sistemes operatius en mòbils. Font: pròpia.

Per tant, s'ha decidit primer adaptar i crear l'aplicació per al sistema operatiu Android, amb l'objectiu de poder fer arribar l'aplicació al mercat més ampli possible.

11.1. Configuració i exportació del projecte

Per a poder transformar l'aplicació creada amb Ionic Framework en una aplicació nativa per a qualsevol sistema operatiu, el projecte s'ha de configurar i adaptar per a poder ser llegit per l'entorn de desenvolupament integrat propi del sistema operatiu al qual es vol transformar l'aplicació.

L'aplicació ha estat integrada amb Capacitor, per tant aquesta ha sigut la plataforma que ha permès configurar el projecte per a poder ser exportat a Android Studio, l'entorn de desenvolupament integrat d'Android.

La forma en la qual aquesta adaptació s'ha dut a terme ha sigut utilitzant unes poques línies de comandes en el CLI de Ionic [49], que han permès que aquest procés s'hagi finalitzat d'una forma senzilla i ràpida.

11.2. Android Studio

Android Studio [50] és el entorn de desenvolupament integrat d'Android. Qualsevol aplicació mòbil que es vulgui crear per a poder ser instal·lada com una aplicació nativa en els dispositius mòbils els quals el seu sistema operatiu és Android han de ser generades a partir d'aquesta aplicació informàtica.

Degut a que l'aplicació ha estat programada en la seva totalitat amb Ionic Framework, dins d'Android Studio pràcticament no s'ha de modificar ni afegir res, simplement comprovar que aquesta funciona correctament en els dispositius Android i generar l'arxiu apk de l'aplicació nativa que s'ha d'instal·lar en aquests dispositius.

11.3. Icon & Splash Screen

L'icon i l'splash screen són dos components de l'aplicació que resulta més senzill de configurar directament amb Android Studio que amb Ionic Framework.

L'icon és la icona que apareix en la pantalla del dispositiu en què s'ha instal·lat l'aplicació i que és l'accés per obrir aquesta aplicació. Android Studio té un servei al qual proporcionant-li la imatge que es vol com a icona fa els ajustos necessaris per adaptar i convertir aquesta imatge en la icona per a l'aplicació.

Per un altre costat, quan es parla de splash screen es refereix a la primera pàgina que l'usuari veu quan obre l'aplicació. És una pàgina transitòria de benvinguda que apareix mentre s'està carregant l'aplicació, que sol durar uns tres segons, i que acostuma a contenir

el logo o el nom de l'empresa a qui pertany l'aplicació.

La seva configuració és un procés diferent del de la icona. Quan s'obre un projecte d'una aplicació en Android Studio, aquest crea una carpeta que conté les imatges de l'splash screen. S'ha de buscar aquesta carpeta i reemplaçar les imatges que apareixen per defecte per l'splash screen per les desitjades per a l'aplicació.

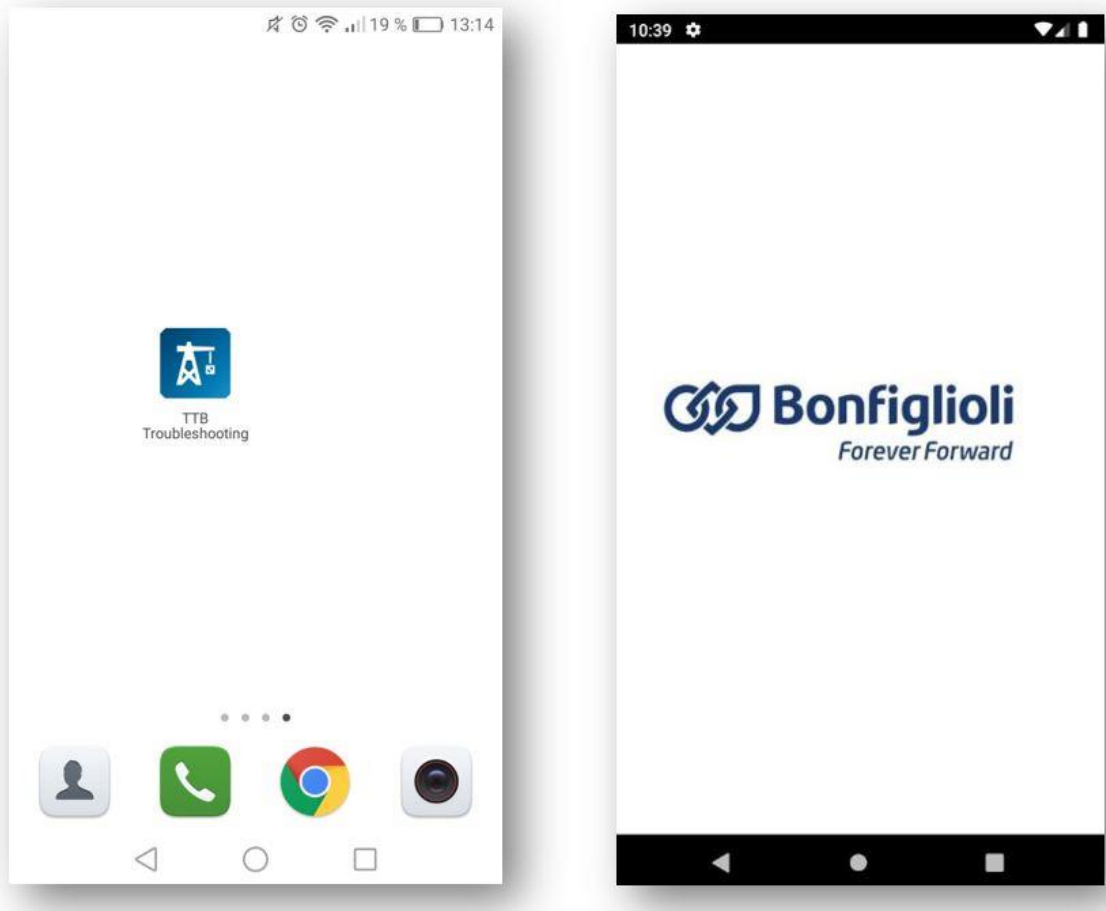


Figura 29: Icona i Splash Screen de l'aplicació. Font: pròpia.

11.4. Android Emulator

Per a poder testejar el correcte funcionament de l'aplicació en un dispositiu mòbil sense la necessitat de dependre de tenir un dispositiu físic que utilitzi el sistema operatiu Android es fa servir Android Emulator [51].

Android Emulator és un emulador d'Android Studio que permet crear dispositius virtuals en un ordinador amb l'objectiu mencionat: provar l'aplicació en diferents dispositius d'Android sense necessitat de comptar amb ells físicament.

L'emulador proporciona gairebé totes les funcions d'un dispositiu Android real. Es poden simular trucades i missatges de text entrants, especificar la ubicació del dispositiu, utilitzar diferents velocitats de xarxa, accedir a Google Play Store i molt més.

Per tant, utilitzant Android Emulator s'ha creat un dispositiu virtual en la qual s'ha comprovat el correcte funcionament de l'aplicació abans de crear l'arxiu apk i provar l'aplicació en un dispositiu físic real.

11.5. Arxiu apk

Un cop s'ha confirmat que l'aplicació creada amb Ionic Framework funciona de la forma desitjada en els dispositius Android, mitjançant el dispositiu virtual creat amb Android Emulator, només queda crear l'arxiu apk.

L'arxiu d'extensió .apk és l'arxiu d'Android que conté totes les carpetes i arxius necessaris perquè l'aplicació pugui ser instal·lada i funcionar en els dispositius Android. Aquest és l'arxiu amb el qual es distribueixen i s'instal·len les aplicacions en els dispositius amb sistema operatiu Android.

Utilitzant Android Studio, simplement s'ha d'indicar que es vol generar l'arxiu apk i el programa el crea i guarda en el nostre ordinador [52]. Després, aquest arxiu s'ha de descarregar des del dispositiu en el qual es vol instal·lar l'aplicació, sempre que sigui un dispositiu Android, i en un minut aproximadament ja es disposa de l'aplicació en el dispositiu llesta per a poder utilitzar-la.

12. Optimització

Un cop l'aplicació estava acabada es va considerar que el pes de l'arxiu apk era massa gran per al tipus d'aplicació que és i que es necessitava fer una optimització.

Aquesta optimització va consistir a reduir el nombre de pàgines creades per als diferents wizards solucionadors d'errors. Degut a que el contingut que apareix en cada una de les pàgines dels wizards és diferent, en un primer moment es va crear una pàgina pròpia per a cada pàgina del wizard. Tenint en compte això, i que es tenien nou errors, el número total de pàgines destinades únicament a la secció Errors era 129.

Però totes aquestes pàgines compartien l'estructura de disseny en què el contingut era mostrat. Depenent de la pàgina hi havia una breu explicació o indicació d'alguna consigna a realitzar, una pregunta, i dos botons com a respostes a la pregunta. Gràcies a aquesta característica es va aconseguir que per a realitzar la secció Errors només calessin 45 pàgines.

El mètode va ser el següent: en la pàgina de presentació de cada error, en el seu arxiu TypeScript, es va crear un objecte que contenia en forma d'atributs totes les explicacions, preguntes i respostes corresponents a les pàgines del seu wizard. D'aquesta forma l'objecte s'anava transmetent de pàgina en pàgina proporcionant el contingut que es volia mostrar en cada pàgina.

Per exemple, la pàgina 1 en començar un wizard és la mateixa per a tots els errors, però depenent des de quin error s'accedeix a aquesta pàgina 1 l'objecte que se li envia és diferent, i per tant l'explicació, la pregunta i les respostes que es volen presentar de forma visual en ella també són diferents. La pàgina 1 és comuna per a tots els errors, però l'objecte que li arriba és únic per a cada error, per tant el contingut que es mostra en la pàgina 1 també és únic per a cada error.

Aquest mètode d'optimització també s'ha implementat en aquelles pàgines que tenien un disseny comú per a diferents errors com pot ser la pàgina final d'un error.

Gràcies a aquesta optimització es va aconseguir passar de 12 MB que pesava l'arxiu apk inicialment a 9 MB, el que suposa una reducció 30% del seu pes.

13. Futures millores

En aquest punt s'expliquen les futures millores aplicables que es poden realitzar en el projecte per obtenir una aplicació de major qualitat i que pugui ser més comercial i arribar a un mercat d'usuaris més ampli.

Respecte a les funcionalitats que podrien ser incorporades a l'aplicació, una podria ser la creació i registre de comptes d'usuari per accedir a l'aplicació. D'aquesta forma l'usuari podria tenir uns certs serveis personalitzats per l'ús de l'app que farien que es sentís més còmode. A més a més, l'empresa podria controlar d'una forma més eficient quins clients estan fent servir la seva aplicació.

Un dels serveis personalitzats que es podrien oferir al client en cas de tenir un compte d'usuari creat en l'aplicació podria ser un apartat de servei de missatgeria remot amb l'empresa. En aquesta secció l'usuari podria obrir un xat amb l'empresa per comunicar alguna incidència, demanar ajuda amb algun dels equips, informar d'avaries o comportaments estranys. Sens dubte un servei que el client valoraria molt positivament i que faria que la comunicació entre l'usuari i l'empresa pogués ser més dinàmica i personalitzada.

Una altra secció que podria ser afegida a l'aplicació podria ser una relacionada amb les novetats de productes o serveis oferits per l'empresa. Seria un espai publicitari amb l'objectiu de cridar l'atenció de l'usuari i poder seduir-lo per firmar nous contractes.

Una altra possible millora seria aplicar a l'aplicació un disseny professional. Normalment, en el desenvolupament d'aplicacions, per un costat hi ha un professional que s'encarrega de realitzar el disseny i per un altre, hi ha un programador que s'encarrega de desenvolupar l'aplicació en si, de forma que cadascú es pot centrar en una tasca determinada i el producte final és molt més complet. En el cas d'aquest Treball de Fi de Grau tant el disseny com el funcionament de l'aplicació ha estat desenvolupat per la mateixa persona.

Una altra possible millora seria actualitzar la versió amb la qual s'ha desenvolupat l'aplicació. L'app d'aquest projecte ha fet servir la versió 3 de Ionic Framework, però en l'actualitat ja està disponible fins a una versió 5. Tot i això, la versió 5 és massa actual, va ser llançada al febrer de 2020, i potser per un programador inexpert pot ser difícil aconseguir un resultat satisfactori degut a la poca documentació no oficial que hi ha en internet. Però des de finals de gener de l'any 2019 la versió 4 ha suposat un gran canvi en la forma de treballar amb Ionic que ha permès als seus usuaris poder desenvolupar aplicacions més modernes i completes.

Amb la finalitat de distribuir l'aplicació d'una forma més professional, una altra possible

millora seria aconseguir publicar l'aplicació en Google Play Store, la plataforma oficial de distribució d'aplicacions per a dispositius que utilitzen el sistema operatiu Android. Qualsevol aplicació existent que es vulgui comercialitzar per a dispositius Android es troba en aquesta plataforma. El fet d'enviar l'arxiu apk als clients directament com a forma de distribució és un mètode molt poc professional i que pot generar certes crítiques i rebuig per part dels usuaris. A més a més, amb l'aplicació publicada en Google Play Store és més fàcil distribuir les actualitzacions futures de la pròpia aplicació. Així també es mostra certa transparència, ja que en Google Play Store els usuaris poden deixar comentaris sobre l'aplicació i el seu funcionament, valoracions i suggeriments.

Per arribar al mercat més ampli possible també s'hauria de considerar exportar l'aplicació desenvolupada amb Ionic Framework al sistema operatiu iOS i generar l'aplicació nativa per a aquest sistema operatiu de la mateixa forma que s'ha fet amb Android. Això permetria arribar a més del 99% dels dispositius mòbils en tot el món i considerar que qualsevol persona en el món es podria descarregar i fer ús de l'aplicació.

Igual que Android, iOS també disposa de la seva pròpia plataforma per a la distribució d'aplicacions per a dispositius que utilitzen el seu sistema operatiu, en aquest cas anomenada App Store. Per tant, un cop generada l'aplicació nativa per iOS també s'hauria de considerar publicar-la en App Store per a la seva distribució.

També convindria testear el bon funcionament dels wizards solucionadors d'errors provocant intencionadament errors en variadors de freqüència amb prototips de grues i comprovar que aquest són eficaços. Encara més, un cop confirmat això, es podria aplicar el mateix mètode però anat a visitar el client i dur-lo a terme en grues reals en comptes de prototips.

14. Impacte ambiental

Aquest apartat pretén recollir i valorar els possibles impactes que ha pogut causar la realització d'aquest projecte en el medi ambient.

Es considera que aquest projecte no ha tingut un gran impacte, ja que ha consistit en el desenvolupament d'una aplicació mòbil i l'única eina requerida ha sigut un ordinador, en aquest cas, un ordinador portàtil.

Per tant, al no tractar-se d'una creació d'un element físic, no s'han hagut d'utilitzar materials extres, que en el seu procés de fabricació podrien haver suposat un impacte negatiu.

Tampoc s'ha realitzat cap desplaçament per fer reunions o treballar a l'empresa, a causa del confinament, i per tant, no s'ha contaminat per l'ús dels mitjans de transport.

No s'ha tingut en compte el procés de fabricació de l'ordinador com a possible impacte ambiental, ja que aquest no es va adquirir amb el propòsit exclusiu del desenvolupament de l'aplicació i ja es posseïa amb anterioritat.

Per últim, es consideraria que l'únic impacte ambiental que ha pogut causar la realització del projecte ha sigut el consum energètic de l'ordinador. Un ordinador consumeix una mitja de 200 W/h, si aquest s'ha utilitzat una mitja de 6 hores diàries durant els 80 dies aproximats que s'han dedicat al projecte, es podria dir que s'han consumit 96 kW d'electricitat.

També cal valorar les millores en l'àmbit ambiental que pot suposar l'ús de l'aplicació.

En cas que un client tingués un variador de freqüència espatllat i no sabés com arreglar-ho, aquest variador s'hauria d'enviar a TecnoTrans Bonfiglioli perquè fos examinat, i si finalment es pogués arreglar, s'hauria d'enviar un altre cop al client. Aquests enviaments suposen una contaminació deguda als gasos contaminants generats pel mitjà de transport que s'ha encarregat de fer els desplaçaments. Gràcies a l'aplicació de troubleshooting, una gran part d'aquests errors els podrà solucionar directament el client, i per tant, els desplaçaments per enviar i retornar els variadors no es duren a terme i la contaminació deguda a aquests no es produirà.

En cas que un tècnic de Bonfiglioli s'hagués de desplaçar fins a la ubicació del client per examinar el variador, l'aplicació també generaria uns certs beneficis a nivell ambiental. Gràcies al formulari enviat pel client en acabar un error, el servei tècnic ja té una informació prèvia de quin és l'estat del variador i quines comprovacions s'han dut a terme per examinar-lo i poder solucionar el problema.

Per tant, aquest fet estalvia cert temps al tècnic per la revisió del variador, i en cas que es necessitessin eines addicionals per a examinar el variador o alguns recanvis, el tècnic ja els portaria amb ell. Potser si el tècnic no hagués tingut la informació prèvia obtinguda a través del formulari, en una primera inspecció no hauria portat aquestes eines addicionals o recanvis. Això implicaria que el tècnic hauria de fer més desplaçaments per recollir el material i tornar un altre cop a la ubicació del client. I com ja s'ha mencionat, aquests desplaçaments generen una certa contaminació deguda als gasos que emeten els mitjans de transport o l'energia que consumeixen.

15. Pressupost

En aquest apartat es té la intenció de recollir els costos que ha suposat la realització d'aquest projecte i el que costaria contractar aquest servei.

Degut a que no s'han produït desplaçaments ni s'ha hagut de comprar cap tipus de material extra, no hi ha hagut costos indirectes.

Tampoc s'ha precisat contractar cap pla de pagament per desenvolupar l'aplicació ni cap de llicència.

Per tant, l'únic factor a tenir en compte per estimar el pressupost és el nombre d'hores invertides i el salari d'un enginyer junior. Es considera que la remuneració per assumir el cost del consum energètic s'inclou com a part d'aquest sou.

Com ja s'ha mencionat en el punt anterior, s'aproxima que s'ha treballat una mitja de 6 hores diàries durant 80 dies. Tenint en compte que el salari d'un enginyer junior s'estima en 25 €/h, el cost total del projecte serien 12000 €.

Conclusions

Per a poder fer una correcta valoració dels resultats obtinguts i extreure unes conclusions en la finalització del projecte s'han de tenir en compte els objectius marcats a l'inici d'aquest. Aquests objectius eren comprendre i aprendre a utilitzar els llenguatges i programes necessaris per al desenvolupament d'una aplicació mòbil, així com obtenir una aplicació de troubleshooting útil per als clients de l'empresa que els hi serveixi per a resoldre possibles problemes amb els variadors de freqüència.

Personalment considero que el resultat ha superat, sense cap mena de dubte, les expectatives marcades en un inici. L'aplicació obtinguda funciona de forma correcta, té un disseny i una interacció agradable per l'usuari, i compleix amb les funcionalitats que s'havien especificat al començament del projecte. Encara més, a aquesta se li han pogut afegir diversos aspectes i seccions que en un primer moment no s'havien contemplat i que han fet créixer l'aplicació i que sigui molt més completa.

Tot i que, com s'ha expressat en l'apartat de possibles millores, encara hi ha alguns aspectes pendents que ajudarien a fer un salt de qualitat a l'aplicació, però aquesta és una versió més que decent i digna de ser comercialitzada.

S'ha adquirit un gran coneixement en el funcionament del framework Angular i en la seva metodologia per separar i connectar el front-end i el back-end en el desenvolupament d'aplicacions web. Així com també s'han après nous llenguatges de molta utilitat en l'actualitat com són, sobretot, HTML i TypeScript. Sense deixar de banda Ionic Framework, el protagonista i responsable de l'obtenció de l'aplicació. És una eina molt versàtil, fàcil d'aprendre i utilitzar, i amb la qual es poden obtenir grans resultats. Encara queda una gran part d'aquest framework pendent d'explorar i la qual espero poder descobrir en un futur.

El procés per arribar a aquest punt no ha estat fàcil, a causa de la inexperiència en el món de la programació i el desconeixement absolut dels llenguatges i programes utilitzats hi ha hagut moments de frustració, en els quals les coses no acabaven de sortir o s'havia de dur a terme una recerca intensa de com solucionar els problemes que anaven sorgint, i de com incloure i programar certes parts de l'aplicació.

En conclusió, després d'analitzar i veure amb perspectiva el camí recorregut, comparant els objectius amb els resultats obtinguts, s'extreu una valoració positiva. Personalment, estic orgullós del producte final que s'ha obtingut en acabar el projecte i en la meua aportació per ajudar a l'empresa Tecnotrans Bonfiglioli a créixer tecnològicament. Un bon indicador de la feina ben feta ha estat rebre l'aprovació i satisfacció dels responsables del projecte a l'empresa amb el meu treball realitzat.

Agraïments

M'agradaria començar per donar les gràcies a l'empresa Tecnotrans Bonfiglioli i al meu tutor, Jordi Juliench, per haver apostat per mi a l'hora d'escollir-me per a l'oferta de pràctiques. Gràcies a ells i a l'equip de treball del que formo part, encapçalat pel Jordi Ortuño, he pogut aprofitar el projecte en el qual estic a la feina per a desenvolupar el meu Treball Final de Grau. Sempre han estat molt atents amb mi, han dipositat una gran confiança en la meva persona i sobretot, m'han donat responsabilitats que feien que em sentís valuós amb les meves aportacions.

També m'agradaria agrair a la meva família el suport que m'ha donat durant la realització d'aquest projecte i durant els quatre anys del grau. Hi ha hagut moments difícils, de situacions dures i complicades, i el seu suport sempre ha sigut incondicional i de gran ajuda.

Gràcies a tots.

Bibliografía

- [1] ALVARADO, I. *Principales características de los tipos de aplicaciones móviles*. 10/01/2019. [<https://ceroideas.es/principales-caracteristicas-de-los-tipos-de-aplicaciones-moviles/>, 02/03/2020].
- [2] RankMyApp. *Conozca los tipos de aplicaciones y sepa cuándo usarlos*. 03/10/2018. [<https://www.rankmyapp.com/es/mercado-es/conozca-los-tipos-de-aplicativos-e-sepa-cuando-usarlos/>, 02/03/2020] .
- [3] PIMIENTA, P. *Tipos de aplicaciones móviles y sus características*. 05/05/2014 [<https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>, 02/03/2020].
- [4] VIDAL, M. *¿Qué son las Progressive Web Apps? ¿Por qué son tan importantes?* 05/11/2019 [<https://www.iebschool.com/blog/progressive-web-apps-analitica-usabilidad/>, 02/03/2020].
- [5] RAMÍREZ, I. *¿Qué es un Aplicación Web Progresiva o PWA?* 03/07/2018 [<https://www.xataka.com/basics/que-es-una-aplicacion-web-progresiva-o-pwa>, 02/03/2020].
- [6] WIKIPEDIA. *Progressive Web Apps*. [https://es.wikipedia.org/wiki/Progressive_Web_Apps, 02/03/2020].
- [7] IONIC FRAMEWORK. [<https://ionicframework.com/>, 03/03/2020].
- [8] WORLD WIDE WEB CONSORTIUM. HTML. [<https://www.w3.org/html/>, 10/05/2020].
- [9] WIKIPEDIA. HTML. [<https://es.wikipedia.org/wiki/HTML>, 10/05/2020].
- [10] KINDPNG. *Logo HTML*. [https://www.kindpng.com/imgv/iJbTR_java-html-language-logo-png-transparent-png/, 10/05/2020].
- [11] WORLD WIDE WEB CONSORTIUM. CSS. [<https://www.w3.org/Style/CSS/>, 10/05/2020].
- [12] ADAWEB. *Logo CSS*. [<https://www.adaweb.es/prefers-color-scheme-css-media-queries/css-logo-adaweb/>, 10/05/2020].
- [13] SASS. [<https://sass-lang.com/>, 10/05/2020].

- [14] CODINGPOTIONS. *Tutorial Sass. Todo sobre el preprocesador de CSS.* 09/052019. [https://codingpotions.com/tutorial-sass, 10/05/2020].
- [15] WIKIPEDIA. *JavaScript.* [https://es.wikipedia.org/wiki/JavaScript, 10/05/2020].
- [16] ECMA INTERNATIONAL. *Standard ECMA 262. ECMAScript 2019 Language Specification.* 06/2019 [https://www.ecma-international.org/publications/standards/Ecma-262.htm, 26/05/2020].
- [17] WIKIPEDIA. *JavaScript.* [https://es.wikipedia.org/wiki/JavaScript#/media/Archivo:Unofficial_JavaScript_logo_2.svg, 10/05/2020].
- [18] JAVILAZARO. *Principios de JavaScript.* [https://www.javilazaro.es/programacion/principios-javascript/, 10/05/2020].
- [19] TYPESCRIPT. [https://www.typescriptlang.org/, 10/05/2020].
- [20] WORLD VECTOR LOGO. *Logo TypeScript.* [https://worldvectorlogo.com/es/logo/typescript, 10/05/2020].
- [21] ANGULAR. [https://angular.io/, 11/05/2020].
- [22] FREEPNG. *Angularjs.* [https://www.freepng.es/png-62y1b9/, 11/05/2020].
- [23] ANGULAR. *Introduction to Angular concepts.* [https://angular.io/guide/architecture, 11/05/2020].
- [24] APACHE CORDOVA. [https://cordova.apache.org/, 11/05/2020].
- [25] WIKIPEDIA. *Apache Cordova.* [https://en.wikipedia.org/wiki/Apache_Cordova#/media/File:Apache_Cordova_Logo.svg, 11/05/2020].
- [26] CLIPARTMAX. *Ionic Logo.* [https://www.clipartmax.com/middle/m2H7H7Z5K9A0Z5Z5_ionic-logo-ionic-framework/, 11/05/2020].
- [27] CARVAJAL, D. *Creando nuestra primera aplicación multiplataforma con Ionic 4.* [https://dcarvajal7.es/creando-nuestra-primera-aplicacion-multiplataforma-con-ionic-4/, 11/05/2020].
- [28] NODEJS. [https://nodejs.org/en/, 12/05/2020].

- [29] NPM. [<https://www.npmjs.com/>, 12/05/2020].
- [30] FREEBIESUPPLY. *Node.js Logo*. [<https://freebiesupply.com/logos/node-js-logo/>, 06/06/2020].
- [31] WIKIPEDIA. *Npm*. [<https://es.wikipedia.org/wiki/Npm#/media/Archivo:Npm-logo.svg>, 06/06/2020].
- [32] IONIC FRAMEWORK. DOCS. *Ionic CLI*. [<https://ionicframework.com/docs/cli>, 12/05/06].
- [33] VISUAL STUDIO CODE. [<https://code.visualstudio.com/>, 12/05/2020].
- [34] ESPARTERO, P. *Tips para utilizar Visual Studio Code (VSCode)*. 11/02/2020. [<https://blog.aitana.es/2020/02/11/tips-utilizar-visual-studio-code-vscode/>, 06/06/2020].
- [35] CAPACITOR. [<https://capacitor.ionicframework.com/>, 12/05/2020].
- [36] IONIC ZONE. *Capacitor*. 03/2018. [<https://ionic.zone/capacitor>, 06/06/2020].
- [37] FIREBASE. [<https://firebase.google.com/>, 30/05/2020].
- [38] FIREBASE. *Representación de la marca de Firebase*. [<https://firebase.google.com/brand-guidelines/?hl=es-419>, 06/06/2020].
- [39] IONICONS. [<https://ionicframework.com/docs/v3/ionicons/>, 01/06/2020].
- [40] IONIC CREATOR. [<https://docs.usecreator.com/>, 12/05/2020].
- [41] ANGULAR. *User input*. [<https://angular.io/guide/user-input>, 22/05/2020].
- [42] IONIC FRAMEWORK. DOCS. V3 API *Navcontroller*. [<https://ionicframework.com/docs/v3/api/navigation/NavController/>, 22/05/2020].
- [43] IONIC FRAMEWORK. DOS. V3. API. *Navparams*. [<https://ionicframework.com/docs/v3/api/navigation/NavParams/>, 22/05/2020].
- [44] IONIC FRAMEWORK. DOCS. V3. NATIVE. *Call Number*. [<https://ionicframework.com/docs/v3/native/call-number/>, 22/05/2020].
- [45] IONIC FRAMEWORK. DOCS. V3. DEVELOPER RESOURCES. *Using ngx-translate*. [<https://ionicframework.com/docs/v3/developer-resources/ng2-translate/>, 22/05/2020].
- [46] GITHUB. *Ngx-translate*. [<https://github.com/ngx-translate/core>, 22/05/2020]

- [47] STATCOUNTER. *Mobile Operating System Market Share Worldwide. Apr 2019 -Apr 2020.* [<https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201904-202004>, 31/05/2020].
- [48] NET MARKET SHARE. *Operating System Market Share.* [<https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Mobile%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-07%22%2C%22dateEnd%22%3A%222020-04%22%2C%22hiddenSeries%22%3A%7B%7D%2C%22tableOrder%22%3A%5B%5B2%2C%22desc%22%5D%5D%2C%22segments%22%3A%22-1000%22%7D>, 32/05/2020].
- [49] IONIC FRAMEWORK. DOCS. DEVELOPING. *Android Development.* [<https://ionicframework.com/docs/developing/android>, 31/05/2020].
- [50] ANDROID STUDIO. [<https://developer.android.com/studio>, 31/05/2020].
- [51] ANDROID STUDIO. GUÍA DEL USUARIO. *Cómo ejecutar apps en Android Emulator.* [<https://developer.android.com/studio/run/emulator>, 31/05/2020].
- [52] ANDROID STUDIO. GUÍA DEL USUARIO. *Cómo compilar y ejecutar una app.* [<https://developer.android.com/studio/run>, 01/06/2020].