

# Creació i implementació d'un sistema de control de pas i accés per a dispositius mòbils

Pol Valletbó Montfort

**Resum**— Aquest paper mostra tot el procés de desenvolupament d'un sistema de control de pas i accés utilitzant dispositius mòbils. Es detalla la metodologia utilitzada, així com les diferents tasques realitzades per superar cada una de les fases del desenvolupament. El paper fa una petita explicació de tecnologies com AngularJS, cordova, NodeJS, entre d'altres, que han estat les tecnologies emprades per a la implementació del sistema. Per altra banda, es mostren els problemes que han sorgit durant el transcurs del projecte, juntament amb la solució que s'ha escollit per poder avançar. Finalment es discuteixen els resultats obtinguts, i es mostren imatges de l'aplicació resultant. En definitiva, el paper pretén exposar la feina feta per a poder tirar endavant un projecte de desenvolupament de software.

**Paraules clau**—Desenvolupament, dispositiu mòbil, accés, ubicació, multi plataforma, AngularJS, NodeJS, software.

**Abstract**—This paper shows the whole development process of an access and location control system, using mobile devices. It details the methodology used, as well as every necessary task to achieve each phase of the process. This paper gives a brief explanation of technologies as AngularJS, cordova, NodeJS, among others. These technologies have been used to implement the system. On the other hand, the paper shows the several problems that have appeared during the project among the decision taken to overcome these problems. Finally, the paper discusses the obtained results and shows images of the resultant application. In short, the paper wants to show the work done throughout this project to achieve the execution of a software development project.

**Index Terms**—Development, mobile device, acces, location, multi platform, AngularJS, NodeJS, Software.

---

## 1 ESTAT DE L'ART

Cada vegada més, la paraula smartcity està agafant més protagonisme en l'àmbit de les TIC. El concepte d'smartcity pretén utilitzar la tecnologia actual per facilitar la vida quotidiana de les persones, dins la seva ciutat. Actualment la gran majoria de les persones porten a la seva butxaca un dispositiu connectat constantment a Internet i amb una quantitat d'aplicacions enorme, per realitzar qualsevol tipus de tasca. El concepte d'smartcity vol que el ciutadà pugui interactuar amb la ciutat utilitzant el seu dispositiu mòbil, una manera d'aconseguir aquesta interacció és mitjançant tecnologies com els codis QR [1] o NFC [2].

Aquestes dues tecnologies han aparegut a l'escena dels dispositius mòbils recentment. Tot i que de diferent manera, tant els QR com els tags NFC permeten emmagatzemar informació per ser llegida posteriorment amb un

lector. Qualsevol dispositiu mòbil actual disposa d'una càmera i una capacitat de processament suficient per a poder llegir la informació amagada dins un codi QR. Pel que fa als tags NFC, només els dispositius més moderns disposen d'un lector per a aquesta tecnologia. El fet de poder emmagatzemar informació que posteriorment podrà ser llegida pels usuaris, és una combinació perfecta per a crear aplicacions compatibles amb el concepte d'smartcity.

Aplicacions com SmartChrono ja utilitzen aquestes tecnologies per poder ajudar a l'usuari per a poder realitzar tasques de manera més senzilla que si no disposés del dispositiu mòbil. El que pretén aquest projecte és poder utilitzar els codis QR o tags NFC, per a què els usuaris puguin demostrar que han estat a una ubicació en un instant determinat de temps. També vol permetre a l'usuari autenticar-se davant una porta, utilitzant també les tecnologies QR i NFC.

- 
- E-mail de contacte: [pol.valletbo@e-campus.uab.cat](mailto:pol.valletbo@e-campus.uab.cat)
  - Menció realitzada: *Tecnologies de la Informació*.
  - Treball tutoritzat per: *Jaume Pujol Capdevila (Departament d'Enginyeria de la informació i de les comunicacions.)*
  - Curs 2013/14

## 2 PROBLEMA INICIAL

El problema que el projecte pretén solucionar, és el problema del control de pas i d'accés. En moltes situacions és important saber la posició on es troba una persona. Per exemple, en una cursa d'orientació cada participant ha de passar per uns punts de control i deixar alguna prova per demostrar que ha estat físicament en aquella posició. La manera típica és firmant un paper que hi ha a cada punt de control. Aquesta alternativa té alguns desavantatges, com per exemple el fet que la comprovació per saber si cada corredor ha passat per cada un dels punts de control, és lenta i farragosa, s'han de reunir tots els papers de control i anar mirant si hi ha la firma del corredor en cada un d'ells. Això es podria solucionar utilitzant una aplicació mòbil que registrés el pas de cada un dels corredors pels punts de control, així seria instantani saber si un corredor s'ha deixat algun punt de control. Una altra situació on un control de pas automatitzat podria ser interessant és per exemple, la gestió de la informació dels estudiants d'un institut. Una manera fàcil de passar llista seria que cada usuari utilitzés el seu dispositiu mòbil per demostrar que ha estat a classe. Això facilitaria la feina al professor, ja que no hauria de passar llista a cada classe i permetria també extreure estadístiques sobre l'assistència dels alumnes d'una manera molt fàcil.

Per altra banda, el control d'accés és un altre problema recurrent en situacions molt diverses. El control d'accés es podria veure com una extensió del control de pas afegint la decisió de permetre l'accés o no, segons quin usuari sigui i on es trobi en el moment de la petició. Seguint l'exemple de la gestió d'un institut, automatitzar el control d'accés a les aules podria ser útil per als professors, ja que podrien accedir a totes les aules a les quals estiguin autoritzats, sense haver de carregar amb totes les claus de les diferents portes.

## 3 OBJECTIUS

Com a resultat d'aquest projecte es vol crear un sistema que doni solució als problemes anteriorment descrits. Per fer-ho cal marcar des de l'inici uns objectius a assolir per tal de poder planificar correctament el treball a realitzar durant el transcurs del projecte. A continuació es llistaran els diferents objectius que el projecte pretén assolir.

- Crear un sistema capaç de controlar el pas i accés de persones mitjançant un dispositiu mòbil.
- Crear un sistema prou genèric per a poder resoldre el problema de localització i accés en entorns diversos.
- Aconseguir que l'usuari prefereixi utilitzar aquest sistema d'accés abans que el mètode clàssic, les claus.
- Generar una experiència d'usuari satisfactòria.
- Dotar a l'organització que implementi el sistema d'informació extra sobre els seus usuaris.

Per assolir els diferents objectius el sistema haurà de complir determinats requisits tant funcionals com no funcionals. Aquests requisits seran més específics, per tant, permetran basar la planificació basant-se en ells.

## 3.1 Requisits funcionals

### 3.1.1 Essencials

- El sistema ha de ser capaç de registrar totes les localitzacions enviades pels usuaris.
- Un usuari ha de poder veure una llista de les localitzacions que ha enviat.
- El sistema ha de poder contestar a peticions d'accés, permetent així l'accés a aquell usuari si realment està autoritzat.
- De la mateixa manera que amb les localitzacions, un usuari haurà de poder veure les peticions d'accés que ha fet.

### 3.1.2 Opcionals

- Dos usuaris que es trobin han de poder registrar la seva trobada.
- L'usuari ha de poder publicar una localització a xarxes socials com Facebook o Twitter.

## 3.2 Requisits no funcionals

### 3.2.1 Essencials

- L'aplicació ha de ser suportada pels sistemes operatius ios i Android.
- Les comunicacions han de ser xifrades.
- Registrar una localització ha de ser tan fàcil com escanejar un codi QR o tag NFC.

### 3.2.2 Opcionals

- Fer el sistema compatible amb el nou dispositiu de Google, Google glass.

Aquests són els diferents requisits que haurà de complir el sistema per tal de satisfer els objectius llistats anteriorment.

## 4 MODEL FÍSIC DEL SISTEMA

Existeixen moltes opcions per poder solucionar els dos problemes anteriorment definits, com per exemple el sistema de geolocalització gps. Actualment la majoria de dispositius incorporen aquesta tecnologia, tot i així aquesta tecnologia té alguns inconvenients que amb la realització d'aquest projecte es volen evitar. El gps és una tecnologia cara pel que fa al consum de dades i bateria que requereix. Un altre problema del gps és que es pot modificar intencionadament la posició on es troba l'usuari, això seria un error crític pel que fa al control d'accés. Amb aquest projecte es vol aconseguir un sistema senzill, simple d'utilitzar i que arribi a la quantitat més gran d'usuaris possible. És per això que s'optarà per tecnologies com NFC o codis QR per poder solucionar els problemes plantejats.

Aquest paper proposa un sistema prou genèric per a poder solucionar el problema de control de pas i accés en la gran majoria de situacions en què aquests problemes apareixen. Per fer-ho, la solució proposada consta de tres elements: aplicació per a dispositius mòbil, servidor encarregat de gestionar i codis QR o tags NFC per transme-

tre la informació de la posició a l'aplicació mòbil. Aquests tres elements es poden combinar de diferents maneres per donar solució al mateix problema. Seguidament es discutiran dues d'aquestes combinacions que afecten el disseny físic del sistema. La primera es basarà amb un punt de control actiu i la segona mostrarà com solucionar el problema utilitzant un punt de control estàtic. El punt de control és l'element que coneix la posició on està, per tant quan un usuari interactua amb aquest element el sistema sabrà que l'usuari ha estat físicament davant d'aquell punt, ja sigui perquè el punt de control li passa la informació (estàtic) o perquè l'usuari envia la seva identificació i el punt la processa juntament amb la posició que ja coneix (actiu).

## 4.1 Control de pas

### 4.1.1 Punt de control actiu

En aquest cas l'usuari s'hauria d'autenticar davant el punt de control, això es podria fer mostrant un codi QR amb la pantalla del dispositiu. Seria el punt de control que, sabent la posició on es troba juntament amb la identificació de l'usuari, es connectaria amb el servidor per confirmar que aquell usuari ha estat físicament davant seu. Un cop es realitzés aquesta comunicació, el servidor actualitzaria les dades d'aquell usuari per afegir que ha estat en aquella ubicació concreta. A partir de llavors l'usuari podria comprovar que a la seva llista d'ubicacions se li ha afegit la que acaba de realitzar. La primera figura resumeix l'explicació d'aquest model de control de pas.

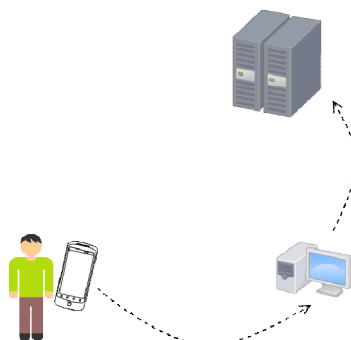


Fig: 1

### 4.1.2 Punt de control estàtic

En aquest model, el punt de control seria un simple codi QR que guardaria informació sobre la posició on es troba. L'aplicació mòbil hauria de ser capaç d'extreure aquesta informació i enviar-la cap al servidor perquè, com abans, es poguessin actualitzar les dades d'aquell usuari. En la figura número dos es pot veure com seria la comunicació de la nova ubicació entre els tres elements.

La principal diferència entre els dos esquemes és el cost d'implementació. Proveir cada punt de control d'ubicació amb un dispositiu capaç de llegir les dades que el dispositiu de l'usuari és molt més costós que deixar un codi QR o tag NFC enganxat en el punt que es vulgui controlar el pas. A més a més, si el punt de control no està vigilat, seria arriscat deixar un dispositiu electrònic, ja

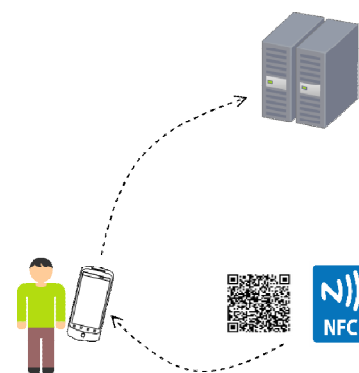


Fig: 2

que seria susceptible a robatoris. En canvi, reemplaçar un tag NFC o codi QR malmès o robat tindria un cost molt menor.

## 4.2 Control d'accés

Pel que fa al control d'accés, també hi ha diferents models que poden satisfer el mateix problema. El control d'accés és un problema crític, ja que si es compromet la identitat d'un usuari, l'individu que ha efectuat l'atac podria tenir accés a recintes restringits. Per tot això, és important decidir a consciència quin és el millor model a seguir per solucionar aquest problema.

Una decisió important és la de triar qui és el portador de la clau. Es pot dissenyar un sistema en què l'usuari tingui una clau a la seva aplicació, la mostri al sistema que hi hagi a la porta i sigui la porta qui enviï la clau cap al servidor i esperi una posterior resposta. En canvi, un altre mètode seria que la clau la mostrés la mateixa porta i l'aplicació fos l'encarregada de comunicar-se amb el servidor.

### 4.2.1 Aplicació portadora clau

Aquesta opció és la més semblant a la realitat, l'usuari porta la clau que mostrarà al sistema perquè aquest validi la clau i decideixi si aquest usuari té accés al recinte. Aquest model de control d'accés seria molt semblant al control de pas amb punt de control actiu, l'usuari mostra la seva identificació al dispositiu que posteriorment processarà la petició. L'única diferència seria que el dispositiu de la porta esperaria una resposta confirmant l'accés, o no, de l'usuari per aquella porta.

L'avantatge principal d'aquest mecanisme és la seva semblança a la vida quotidiana. Les persones estan acostumades a obrir portes a partir d'una clau que elles posseeixen, la clau interactua amb la porta i aquesta s'obre. Aquest avantatge alhora es pot convertir en un inconvenient. Si un tercer aconsegueix comprometre el dispositiu d'un usuari del sistema i aconsegueix la seva clau, aquest tercer podria accedir als recintes on l'usuari pogués entrar. Aquest mecanisme és el que es fa servir a la majoria d'events com concerts, conferències tecnològiques, etc. On l'usuari mostra el seu codi QR a l'entrada, aquest codi es comprova i l'organització et deixa passar.

#### 4.2.2 Porta mostra clau

Un altre model capaç de resoldre el problema seria el contrari, en lloc de ser l'usuari el portador de la clau, seria la mateixa porta que li mostraria una clau amagada dins un codi QR o mitjançant NFC i l'aplicació de l'usuari seria l'encarregada de llegir aquesta clau i enviar-la, juntament amb les credencials de l'usuari, cap al servidor. El servidor seria l'encarregat d'acceptar l'accés d'aquest usuari i enviar la resposta a la mateixa aplicació i a la porta perquè aquesta s'obris.

Aquest model és una extensió del mecanisme de control de pas amb punt de control estàtic, el dispositiu de la porta actuaria com un simple codi QR amb dades que ha de llegir l'usuari. Aquest model soluciona el principal problema de l'anterior sistema de control d'accés, el fet que la clau guardada per l'usuari fos susceptible de ser compromesa. En aquest cas la porta demanaria una clau diferent cada vegada que un usuari es trobés davant, per tant un tercer que hagués aconseguit interceptar una clau no podria utilitzar-la, ja que la clau per accedir seria diferent en cada petició d'accés. En aquest cas el principal perill resideix en la generació i comunicació de la clau. Per implementar aquest sistema caldrà un sistema de generació de claus que serveixi una clau per a cada petició de les diferents portes i una comunicació amb el dispositiu de la porta xifrada. La figura número tres mostra les diferents comunicacions necessàries implementar aquest sistema.

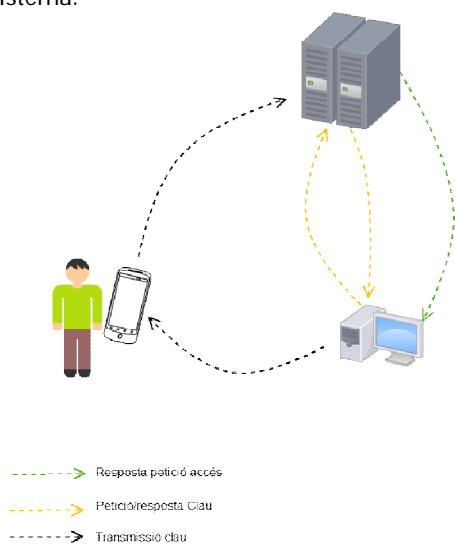


Fig: 3

## 5 TECNOLOGIES

Actualment el món del desenvolupament de software disposa d'una gran quantitat de llenguatges i tecnologies per donar solució a les diferents tipologies de problema que es vulgui solucionar. És tal la quantitat existent de tecnologies, que moltes són útils per solucionar un mateix problema. És per això que existeixen diverses opcions per a realitzar la implementació d'aquest sistema, cal decidir quina opció va més acord amb els objectius i requisits esmentats en l'apartat anterior.

A continuació es mostraran algunes de les tecnologies actuals útils per a la realització del projecte. Per fer-ho, es classificaran les tecnologies en banda del client i banda del servidor, front end i back end respectivament.

### 5.1 Front end

En aquest projecte la banda del client serà clarament l'aplicació per a dispositius mòbils que interactuarà amb la resta de sistema. La manera clàssica de desenvolupar una aplicació mòbil és utilitzant els SDK's que proporcionen les empreses responsables de cada sistema operatiu. Aquests SDK's disposen d'una àmplia documentació i una gran comunitat al darrere. Utilitzar aquest mètode és la manera on es poden aconseguir aplicacions amb un major rendiment, a més a més permet aprofitar els diferents components del dispositiu (gps, càmera, etc...) al màxim. El principal inconvenient d'utilitzar aquest mètode per implementar una aplicació és que cada sistema operatiu utilitza un llenguatge de programació diferent. És per això que si es vol dissenyar una aplicació multi plataforma, el cost serà molt elevat, ja que s'haurà d'implementar tantes vegades com sistemes operatius vulguis arribar.

Darrerament està augmentant l'ús de tecnologies que permeten el desenvolupament multi plataforma. La majoria d'aquestes tecnologies transformen llenguatge Html, css i javascript a codi nadiu de la plataforma que es desitgi. Apache Cordova [3] és una d'aquestes tecnologies, permet generar aplicacions natives tant per Android, com per ios, Blackberry, entre d'altres. Crear aplicacions utilitzant aquest mètode permet utilitzar qualsevol eina útil pel desenvolupament web, per tant, qualsevol dels frameworks existents que ajuden a dissenyar i estructurar pàgines web. AngularJS és un framework basat en javascript que ajuda a organitzar el codi seguint el model de desenvolupament web, model vista controlador. Escollir aquesta opció per desenvolupar una aplicació també té problemes. Aquesta tècnica de desenvolupament no permet aprofitar al màxim les novetats que van afegint en cada un dels sistemes operatius. Un altre problema d'utilitzar aquesta tecnologia és el rendiment. El rendiment que ofereix aquest tipus de solució no és el mateix que utilitzant el llenguatge nadiu de cada sistema operatiu, és per això que depenent del tipus d'aplicació que es vulgui construir, no es podrà fer utilitzant aquest mètode. Aquesta baixada de rendiment es deu a la traducció que ha de fer Cordova per passar del codi web al llenguatge del sistema operatiu.

Un altre tema que cal tenir en compte pel que fa a la banda del client, és el disseny de la interfície gràfica que tindrà l'aplicació. Depenent de l'opció escollida per desenvolupar l'aplicació, nativa o multi plataforma, la creació de la interfície gràfica serà diferent. Pel que fa al desenvolupament nadiu, cada sistema operatiu disposa d'elements propis per generar la interfície d'usuari. En canvi, l'opció multi plataforma genera elements gràfics a partir de CSS i Javascript. De la mateixa manera és possible utilitzar qualsevol dels frameworks existents, que proveeixen als desenvolupadors d'elements dissenyats per experts a partir de CSS i Javascript.

## 5.2 Back end

De la mateixa manera que per la banda de client, pel que fa a la banda de servidor també existeixen moltes tecnologies per poder gestionar el trànsit de dades entre l'aplicació, el servidor i el sistema gestor de base de dades. L'opció més típica és la d'utilitzar la pila LAMP. Aquesta pila de tecnologies consta de: Linux com a sistema operatiu poder fer córrer el servidor, Apache com a servidor per atendre i respondre les diferents peticions web, MySQL com a sistema de base de dades encarregat de la persistència de les mateixes, i per últim, Php com a llenguatge de programació de la banda del servidor. Aquest conjunt de tecnologies és el més utilitzat típicament.

Recentment està començant a prendre protagonisme una altra pila de tecnologies, aquesta és la pila MEAN. MongoDB amb a sistema gestor de base de dades no relacional, Express com a framework per gestionar de manera més senzilla el servidor, AngularJS com a framework Javascript per gestionar la part del client, i NodeJS com a servidor web creat per ser programat amb Javascript. Aquesta nova pila permet al desenvolupador utilitzar el mateix llenguatge per programar cada un dels elements del sistema, tant la part del client com la del servidor. MongoDB trenca amb el clàssic model de taules i columnes dels sistemes SQL. Mongo utilitza el concepte de col·leccions i documents, un esquema molt més flexible, ja que no obliga a tots els documents d'una col·lecció a tenir els mateixos atributs. Els documents d'aquest esquema són arxius JSON, format molt utilitzat per intercanviar dades per la xarxa. Això fa que la base de dades d'aquesta pila emmagatzemi les dades en el mateix format que li arriben, sense necessitar una traducció prèvia.

## 6 MODEL FINAL

Els dos apartats anteriors han proposat diverses alternatives per donar solució al problema exposat anteriorment, el que vol mostrar aquest apartat és l'elecció definitiva sobre disseny i tecnologies que s'han utilitzat per implementar el sistema. De la mateixa manera que abans, s'organitzarà en tres apartats on s'exposaran les raons per haver escollit cada una de les alternatives.

### 6.1 Disseny físic

Valorant els dos mecanismes, tant per al sistema de control de pas com d'accés, s'ha optat per implementar el que utilitzava un punt de control estàtic. Bàsicament s'ha optat per aquest model pel simple fet que els costos són considerablement més baixos si es desitja proveir al sistema de molts punts de control. L'elecció del sistema de control d'accés ve lligada amb l'opció escollida pel control de pas. S'ha optat perquè el sigui la porta qui mostri la clau, encara que no sigui el més intuïtiu per als usuaris. Això s'ha decidit perquè es volia mantenir el mateix esquema per a solucionar els dos problemes, en aquest cas només faria falta estendre el mecanisme perquè el servidor retorni la resposta de la petició d'accés.

### 6.2 Front end

Finalment l'aplicació per a dispositius mòbil ha estat implementada amb l'eina multi plataforma. Aquesta ha estat l'opció escollida per una raó principal, poder assolir l'objectiu d'aconseguir una aplicació pels dos sistemes operatius principals. Un altre dels motius que han pesat més per acabar decidint utilitzar el desenvolupament web ha sigut el fet de poder conèixer noves maneres de desenvolupar una aplicació mòbil. A continuació es podran veure les tecnologies web concretes que s'han utilitzat.

- AngularJS [4]: Framework creat per Google, útil per a poder crear pàgines web modernes seguint el model "model vista controlador", facilitant moltíssim el tractament del DOM. Aquest framework és l'encarregat de gestionar tota la informació que passa a través de l'aplicació, tant la que arriba des del servidor, com la que la mateixa aplicació ha d'enviar. Javascript és el llenguatge necessari per poder desenvolupar una aplicació utilitzant aquest framework.
- Ionicframework [5]: Aquest framework proveeix als desenvolupadors de les eines necessàries per poder crear una interfície d'usuari moderna per a dispositius mòbil. Actualment encara està en fase beta, però s'ha optat per aquesta opció degut a que es complementa a la perfecció amb AngularJS i Cordova. Aquest framework ha estat pensat per a ser utilitzat amb aquestes dues tecnologies, per tant és una de les millors opcions si s'opta pel desenvolupament multi plataforma utilitzant AngularJS.

### 6.3 Back end

Pel que fa a la banda de servidor, no hi ha una raó principal per haver optat entre un sistema o un altre, ja que amb qualsevol de les dues tecnologies es podia obtenir un resultat molt semblant. Finalment s'ha optat per la pila MEAN descrita anteriorment. El fet de tractar-se de tecnologies relativament noves ha pesat a l'hora de fer l'elecció. Un avantatge d'utilitzar aquestes tecnologies és el fet de poder implementar, tant la banda del client com la banda del servidor, amb el mateix llenguatge de programació, fet que facilita el desenvolupament. De la mateixa manera que a l'apartat anterior, a continuació es mostraran la llista definitiva de tecnologies seleccionades.

- MongoDB [6]: Tecnologia encarregada de la persistència de tota la informació necessària perquè el sistema pugui funcionar. Mongo emmagatzema les dades en format JSON, aquest fet permet guardar les dades directament com les rep el servidor, sense fer cap canvi. Un altre punt positiu és la flexibilitat i velocitat que ofereix aquest sistema. Per últim, novament es tracta d'una tecnologia que està agafant força actualment, això fa que sigui interessant descobrir com funciona i saber utilitzar-la per si s'acaba imposant com a sistema majoritari en el desenvolupament web.
- Express [7]: Framework que facilita molt la creació d'un servidor web basat en NodeJS. Express estalvia la configuració de més baix nivell neces-

sària perquè el servidor funcionés. També facilita moltíssim la gestió de les diferents peticions que pot rebre el servidor, d'una manera molt senzilla és possible crear un servidor seguint completament el model REST.

- NodeJS [8]: Aquesta tecnologia és la base de tot el back end. Node és el servidor que gestiona la interacció entre l'aplicació mòbil i les dades emmagatzemades a la base de dades. També és l'encarregat de processar tota la informació i enviar-la cap a l'aplicació perquè aquesta pugui aprofitar-la. Node és l'encarregat de rebre totes les peticions d'accés i ubicacions, posteriorment processar-les i amb l'output generat, enviar les dades resultants on toqui.
- Mongoose [9]: Mòdul de Node que permet gestionar de manera senzilla els accessos, tractament de dades, etc... amb la base de dades no relacional MongoDB. Mongoose s'ha utilitzat pels diferents processos del servidor que necessiten interacció amb la base de dades.

La quarta figura mostra una visió global d'on se situa, cada una de les tecnologies llistades, dins el sistema.

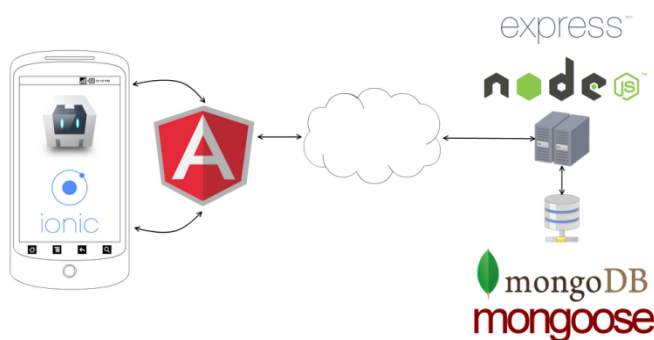


Fig: 4

## 7 PLANIFICACIÓ

Aquest apartat vol mostrar quina ha estat la planificació seguida per poder tirar endavant el projecte. Aquest projecte consta de 300 hores a repartir entre la part tècnica del projecte, disseny del sistema, implementació i validació, però també cal tenir en compte el temps necessari per realitzar tota la documentació necessària per tal de poder mostrar quina ha estat la feina feta. Per mostrar aquesta planificació, primer s'explicarà la metodologia escollida per poder dur a terme les diferents fases de la planificació, a continuació es llistaran les diferents tasques fetes a cada una de les etapes.

Pel que fa al model utilitzat per tirar endavant el projecte s'ha optat per una barreja entre el model de desenvolupament en cascada i el model iteratiu, intentant aprofitar els avantatges de cada un d'ells. La part del procés de desenvolupament en cascada constarà de dues etapes, la primera serà la fase de decisió de requisits i la segona etapa del model cascada continuarà totes les etapes restants. Aquestes etapes seguiran l'esquema iteratiu, s'organitzaran en iteracions controlades on a cada iteració es repetiran les diferents etapes. Cada una d'aquestes iteracions constarà de tres etapes: disseny, implementació i

proves. El que es pretén amb aquest model és assolir una milestone al final de cada una de les iteracions. Pròximament es mostrarà una llista detallada amb cada una de les tasques dins de cada etapa del procés de desenvolupament.

### 7.1 Definició de requisits

Aquesta és la primera etapa del procés de creació de software en aquest projecte. Aquesta etapa pretén, a partir de la recerca sobre l'estat de l'art en processos d'automatització del control de pas i accés de les persones, generar una sèrie de requisits que el sistema resultant haurà de complir per poder assolir els diferents objectius marcats. Com que aquesta etapa forma part de la part del model cascada, els requisits decidits un cop finalitzada l'etapa no podran ser modificats. A continuació es llistaran cada una de les tasques que conformen aquesta primera etapa.

- Estudi dels problemes a solucionar.
- Recerca sobre l'estat de l'art.
- Definició dels requisits funcionals.
- Definició dels requisits no funcionals.
- Classificar els requisits segons nivell de prioritat.

### 7.2 Disseny

L'etapa de disseny és la primera que es realitzarà a l'inici de cada una de les diferents iteracions del projecte. L'objectiu d'aquesta etapa és pensar quina és la millor manera d'implementar les diferents tasques necessàries per complir amb els objectius de la iteració. Per trobar aquesta manera cal: dissenyar models, desenvolupar diagrames de flux, definir les funcionalitats que ha d'haver de tenir cada mòdul del sistema, etc... En definitiva, l'etapa de disseny és la que marcarà el camí a l'etapa d'implementació, perquè aquesta última pugui avançar amb normalitat. A continuació es llistaran les tasques d'aquesta etapa classificades segons a la iteració que pertanyin.

#### 7.2.1 Primera iteració

- Prototip interfície d'usuari.
- API servidor.

#### 7.2.2 Segona iteració

- Esquema base de dades.
- Mecanisme control ubicació.
- Esquema dispositiu porta.
- Definició de rols.

#### 7.2.3 Tercera iteració

- Esquema mecanisme control accés.

### 7.3 Implementació

Aquesta és la fase crítica de cada una de les iteracions. Durant aquesta etapa s'han de codificar els diferents models dissenyats en l'etapa anterior, aconseguint així, que el sistema prengui forma. Es considera una etapa crítica, ja que durant aquesta fase és probable que apareguin problemes que provoquin canvis a la planificació inicial. Tot i haver realitzat els dissenys el millor possible, a l'hora de la implementació normalment apareixen errors. És important considerar aquest risc a l'hora de fer la planificació temporal del projecte. També es considera una etapa crítica, ja que el resultat final del projecte estarà directament relacionat amb la feina feta durant aquesta etapa, per tant, és necessari fer la feina ben feta en aquesta fase.

Aquesta etapa també s'anirà repetint en cada una de les iteracions. A continuació es podrà veure totes les tasques d'aquesta etapa agrupades per iteració.

#### 7.3.1 Primera iteració

- Interfície d'usuari.
- Servidor preparat per rebre i contestar peticions.

#### 7.3.2 Segona iteració

- Comunicació client-servidor.
- Creació i emmagatzematge d'ubicacions.
- Incorporació lector QR a l'aplicació mòbil.
- CORS a nivell de client i servidor.

#### 7.3.3 Tercera iteració

- Software dispositiu porta.
- Acoblament dels diferents mòduls.
- Creació i emmagatzematge d'accessos.

### 7.4 Validació

Aquesta és l'etapa final del procés iteratiu. La feina d'aquesta etapa és validar el correcte funcionament de les diferents implementacions fetes en l'etapa anterior. El procés de validació o test és una part fonamental en qualsevol projecte de software, provar és l'única manera de garantir que el software fet és de qualitat. Aquesta fase podria haver estat exclosa del cicle iteratiu i fer-se al final, un cop les diferents parts del sistema estiguessin acabades, aquesta opció es va descartar ràpidament pel fet que provant, a part de trobar errades, poden sorgir idees per a millorar la implementació del sistema, per tant l'opció d'anar fent proves a mesura que avança la implementació s'adapta millor a les necessitats d'aquest projecte.

El mecanisme principal utilitzat per a la realització de la validació dels diferents components, ha estat la prova manual tant a l'aplicació, com al servidor. Aquesta validació permet reproduir les accions que realitzaria un usuari i d'aquesta manera trobar errors de funcionalitat, visuals, de respostes, etc...

#### 7.4.1 Primera iteració

- Proves de resposta interfície d'usuari.
- Test de resposta a la API del servidor.

#### 7.4.2 Segona iteració

- Testejar comunicació entre client i servidor.
- Lector QR.
- Proves sobre l'emmagatzematge d'ubicacions.

#### 7.4.3 Tercera iteració

- Test sobre el software resident a les diferents portes.
- Proves sobre l'emmagatzematge d'accessos i ubicacions.
- Proves de sistema.

Un cop realitzades les tres primeres iteracions es va decidir afegir una quarta. Aquesta última més enfocada a revisar i modificar la feina feta en les anteriors iteracions, és per això que no es va seguir la mateixa metodologia per a aquesta última iteració. Bàsicament aquesta última iteració ha servit per deixar l'aplicació preparada per ser entregada.

Un dels objectius de cada una d'aquestes iteracions és la consecució d'una milestone al final de cada una de les diferents iteracions. Cada una d'aquestes milestones significa un avanç important sobre la totalitat del projecte. A continuació es mostrarà una llista de les tres milestones que es volen assolir amb la realització de les tres iteracions detallades prèviament.

1. Crear interfície d'usuari i funcionament intern de l'aplicació mòbil. Crear el servidor preparat per rebre peticions i respondre a la segona iteració. Implementar reconeixement de codis QR i tags NFC.
2. Implementar les comunicacions entre client i servidor. Disseny i implementació del dispositiu a la porta d'accés.
3. Implementació de l'aplicació web per a l'administrador del sistema.

## 8 PROBLEMES

En projectes de qualsevol tipus, l'aparició de problemes és un fet completament freqüent, cal saber-ho i actuar en conseqüència. Una bona manera de superar els problemes és amb una bona planificació prèvia. Partint d'una mala planificació inicial, un problema inesperat pot ser catastròfic per la finalització d'un projecte. Aquest apartat pretén mostrar els problemes apareguts durant el transcurs del projecte, juntament amb les decisions preses per encarrilar de nou el projecte minimitzant així, l'impacte del problema sobre la planificació del projecte.

Un dels problemes recurrents en el món del desenvolupament software és la configuració de l'entorn de treball. En determinades empreses aquest problema pot causar que un treballador perdi dies per aconseguir tenir tot l'entorn de treball en marxa. Cal tenir-ho present i marcar un dia de la planificació com a dia per configurar l'entorn. La primera tecnologia multi plataforma que es volia utilitzar era Phonegap [10], igual que Cordova permet generar una aplicació nativa a partir de llenguatge de programació web. El problema va sorgir a l'hora de crear i instal·lar l'aplicació en un dispositiu mòbil, l'aplicació no es podia generar per uns problemes de configuració desconeguts. Aquest fet no va afectar la planificació temporal, ja que estava previst que es poguessin donar situacions semblants, però sí que va afectar d'una altra manera a la planificació inicial, Phonegap va haver de ser substituït per Cordova. Aquestes dues tecnologies estan relacionades, Phonegap és una distribució de Cordova amb alguna funcionalitat extra, però en essència són el mateix. Per tant, aquest problema es va poder solucionar sense gaire impacte al projecte, es podria classificar com a un problema de poc impacte.

El següent problema va aparèixer durant la fase d'implementació de la segona iteració. El model inicial de sistema contemplava la utilització, tant de codis QR, com de la tecnologia NFC. El fet és que en el moment d'investigar l'accés de l'aplicació a les funcionalitats internes del mòbil, com la càmera, GPS, etc... es va veure que poder accedir a aquests recursos depenia dels plugins que oferia Cordova. NFC és una tecnologia recentment implemen-



tada en alguns dispositius mòbils, això fa que el plugin existent per accedir a aquesta tecnologia encara no estigui adoptat per la gran majoria de comunitat i per tant, la documentació del mateix sigui pobre. Es va creure que intentar implementar aquesta tecnologia al sistema portaria més problemes que avantatges, no valdria la pena l'esforç necessari per portar-ho a terme amb el benefici resultant si s'hagués implementat. Una altra raó que dona suport a la decisió final és el fet que els dispositius d'Apple no disposen d'aquesta tecnologia, per tant s'estaria incomplint un dels requisits principals. La majoria de dispositius Android tampoc disposen d'aquesta tecnologia encara, per tant la implementació d'aquesta tecnologia hauria de ser un afegit, en cap cas una característica essencial. Per tant, la solució va ser eliminar aquesta tecnologia del sistema. L'adaptació del sistema per a utilitzar NFC serà present a la secció dedicada a possibles millores del sistema en una implementació futura.

Durant el desenvolupament de la part del servidor va aparèixer un error a priori poc important, però que finalment va fer modificar la planificació temporal més del previst. Per defecte els servidors no comparteixen recursos quan aquests són demanats per orígens desconeguts, restricció pensada per evitar certs problemes de seguretat que podria comportar la compartició d'informació amb orígens desconeguts. En el cas d'una pàgina web típica, el mateix servidor és qui serveix la pàgina web que l'usuari veurà, per tant quan aquest usuari fa una petició de certs recursos mitjançant la web, el servidor coneix l'origen i permet aquest intercanvi de recursos. En el cas d'aquest projecte el servidor només actua de font d'informació per a l'aplicació. L'aplicació ja disposa de totes les vistes, només ha de fer peticions al servidor per omplir de dades aquestes vistes, aquestes dades són enviades pel servidor utilitzant el format JSON. Per tant, quan l'aplicació fa la petició de dades el servidor desconeix aquell origen i en conseqüència li nega l'accés. Les sigles CORS [11] provenen de Cross Origin Resource Sharing i el que precisament volen dir és això, compartició de recursos amb orígens "creuats" o desconeguts. Tant al client com en el servidor s'ha d'afegir unes línies de codi que obliguin al servidor servir qualsevol recurs sense importar quin sigui l'origen de la petició. Això permet que el servidor actuï com una API, accedint a una direcció determinada des de l'aplicació el servidor respondrà amb les dades demanades. Aquest problema va ser més difícil de resoldre de l'esperat, bàsicament degut al desconeixement personal sobre APIS i el món de l'aplicació mòbil connectada a un servidor.

Per últim, la funcionalitat de modificar la imatge de perfil de l'usuari ha estat una de les més complicades de resoldre, a causa de problemes amb els permisos de les imatges una vegada arriben al servidor. Per tractar les imatges un cop estan al servidor, s'utilitza el sistema de fitxers de NodeJS. La solució ha passat per utilitzar una funció d'aquest sistema que permet modificar els permisos d'arxius, personalment desconeguda anteriorment.

## 9 RESULTATS

Un cop realitzades les diferents iteracions i superats els problemes exposats anteriorment, s'ha obtingut una aplicació resultant. Aquest apartat discutirà sobre si aquest sistema resultant compleix, o no, amb els requisits, per tant assoleix els objectius proposats a l'inici d'aquest projecte. Per poder veure clarament la consecució d'aquests objectius es mostraran en forma de taula tots els requisits i objectius. Es marcaran els que s'han pogut assolir durant el projecte i els que finalment s'han hagut de deixar per futures millores del sistema. D'aquests darreres, es donarà una justificació pel motiu que ha causat aquesta no consecució. Per altra banda, aquest apartat mostrarà les funcionalitats finals disponibles per cada un dels elements que s'han implementat: aplicació mòbil, servei administració web i servidor per gestionar les dades. Per últim, aquesta última secció vol mostrar les habilitats, tant tècniques com personals, adquirides gràcies a tot treball realitzat durant el transcurs d'aquest projecte.

### 9.1 Consecució requisits

Pel que fa als requisits funcionals, s'han pogut assolir els diferents requisits marcats com a essencials. Per altra banda, l'únic requisit opcional que s'ha complert és el de poder generar informes segons el rol d'usuari. El rol administrador des de la web podrà veure un informe amb totes les ubicacions i accessos de l'usuari desitjat. Pel que fa a la implementació de l'event trobada i la compartició de localitzacions i/o accessos a les xarxes socials, no ha pogut ser implementat per les modificacions que ha anat patint la planificació a causa dels problemes que han anat sorgint. En qualsevol cas, el sistema està preparat per poder implementar aquests dos requisits, no seria gaire costós implementar-ho en una futura versió. Aquesta informació es pot veure resumida en la figura número cinc.

Requisits funcionals		
Essencial	L'aplicació ha de ser capaç de registrar totes les localitzacions enviades pels usuaris.	✓
Essencial	Un usuari ha de poder veure una llista de les localitzacions que ha enviat.	✓
Essencial	El sistema ha de poder contestar a peticions d'accés, permetent així l'accés.	✓
Essencial	Un usuari poder veure les peticions d'accés que ha realitzat.	✓
Opcional	Dos usuaris han de poder registrar l'event de trobada.	✗
Opcional	L'usuari ha de poder compartir una localització a les xarxes locals.	✗

Fig: 5

De la mateixa manera que amb els requisits funcionals, la totalitat de requisits essencials ha pogut ser assolida al final de les quatre iteracions de desenvolupament. Pel que fa a l'únic requisit no funcional, no ha pogut ser assolit principalment per l'actual dificultat de disposar del nou dispositiu de Google. Tot i no haver pogut ser implementat, s'ha tingut present en tot moment. El fet que la porta sigui l'encarregada de mostrar el codi QR es va decidir en gran part per permetre acoblar les ulleres de Google al sistema. En la sisena figura es pot veure aquesta informació.



Requisits no funcionals		
Essencial	L'aplicació ha de ser suportada pels sistemes operatius Android i iOS.	✓
Essencial	Les comunicacions han de ser xifrades.	✓
Essencial	Registrar una localització ha de ser tan fàcil com escanejar un codi QR.	✓
Opcional	Fer l'aplicació compatible amb el nou dispositiu de Google, Google glass.	✗

Fig: 6

A causa de l'estreta relació entre els requisits funcionals i els objectius, es pot afirmar que no hi ha cap objectiu que no s'hagi assolit. També es pot afirmar que els tres primers objectius s'han assolit correctament. Pel que fa als altres dos objectius no es podrà saber si s'han complert o no fins que els usuaris de l'aplicació donin feedback sobre el sistema. Caldria obrir el sistema a un petit grup d'usuaris i posteriorment contestessin a un qüestionari que reflectís el seu nivell de satisfacció envers el sistema.

## 9.2 Sistema resultant

El sistema resultant, un cop finalitzat tot el procés de desenvolupament, consta de tres parts clarament diferenciades: aplicació mòbil, aplicació web per a administració del sistema i servidor encarregat de gestionar tant l'aplicació mòbil com el servei web. Aquest apartat pretén mostrar el resultat final de cada un d'aquests tres elements. Per fer-ho s'exposaran les funcionalitats que estan totalment implementades i preparades per funcionar de forma correcta. A l'annex es mostraran imatges de com a quedat el sistema gràficament.

### 9.2.1 Aplicació mòbil

L'aplicació mòbil final està preparada per funcionar per a qualsevol dispositiu Android. El framework Ionic s'encarrega de mostrar la interfície gràfica de forma correcta per a les diferents mides de pantalles als diferents dispositius Android. El sistema ha estat testejat en emulador per provar diferents models de dispositiu mòbil. Pel que fa al testeig en un dispositiu real, s'ha provat íntegrament en un Nexus 5. Pel que fa al sistema operatiu iOS, només faria falta modificar les parts de l'aplicació que accedeixen als recursos interns del dispositiu, que pel cas d'aquesta aplicació només es tractaria del lector de codis QR. No caldria modificar cap altra part de codi, per tant el canvi seria trivial. S'ha decidit no provar-ho, ja que per a poder compilar una aplicació per a iOS és necessari disposar d'una clau de desenvolupador registrat. A continuació es llistaran les diferents funcionalitats que permet utilitzar el sistema final:

- Compartir ubicacions amb la resta d'usuaris del mateix grup.
- Demanar accés a recintes protegits pel sistema.
- Veure el llistat d'ubicacions i accessos propis i dels usuaris d'un mateix grup.
- Modificació de la imatge de perfil.
- Modificació de la contrasenya pròpia.

### 9.2.2 Web administració

L'usuari o usuaris que tinguin permisos d'administració al sistema podran accedir a l'aplicació web que permet gestionar diferents aspectes del sistema. S'ha decidit realitzar tota la gestió del sistema via web, en lloc de fer-ho

des de la mateixa aplicació, ja que seria una feina incòmoda de realitzar en la pantalla d'un dispositiu mòbil. De totes maneres i si l'usuari ho desitja, podria accedir al servei web mitjançant el navegador del seu dispositiu. Cal dir que aquest servei no està optimitzat per ser utilitzat en un dispositiu mòbil. Aquest servei permet realitzar als administradors les següents accions:

- Eliminar usuaris.
- Crear usuaris nous.
- Crear nous grups.
- Crear noves portes.
- Llistat complet tant d'ubicacions com d'accessos de l'usuari desitjat.
- Permetre i revocar accés a les diferents portes.

Aquest servei per als administradors compleix amb l'objectiu de proveir a l'empresa sobre informació extra dels seus usuaris, ja que els administradors poden veure en tot moment les posicions on han estat els seus usuaris i les peticions d'accés que han realitzat.

### 9.2.3 Servidor

Aquest servidor és l'encarregat de gestionar completament el sistema, totes les peticions de l'aplicació mòbil i l'aplicatiu web, van a parar a ell. El servidor és capaç de processar aquestes peticions, connectar-se a la base de dades si és necessari i finalment respondre a les peticions utilitzant el format establert. El servidor és l'encarregat d'enviar tokens [12] de sessió als diferents usuaris que s'identifiquin correctament, per posteriorment comprovar si cada petició conté el token correcte per poder accedir a les dades del sistema. El servidor per si sol no té cap funcionalitat accessible per a l'usuari final, sinó que el servidor és l'encarregat de fer possible el bon funcionament de les diverses funcionalitats que ofereixen tant l'aplicació mòbil, com el servei web. Tot aquest sistema no té cap sentit si aquest servidor està inactiu, això és degut al fet que l'aplicació necessita connexió constant per omplir amb les dades requerides per cada una de les vistes de l'aplicació.

A l'apèndix es pot veure la llista de les diferents peticions que el servidor està preparat per rebre. De la mateixa manera, es detallarà el resultat que l'aplicació obtindrà un cop realitzada aquella petició.

## 10 MILLORES FUTURES

La vida d'un projecte de desenvolupament de software no té fi, sempre es poden aplicar millores per transformar l'aplicació, web, etc... en un millor servei de cara als usuaris. Aquest projecte no és cap excepció, tant la part de l'aplicació mòbil, el servei web d'administració i el servidor encarregat de gestionar el sistema, poden rebre actualitzacions per tal d'oferir un millor servei a l'usuari final.

A continuació es llistaran les millores més immediates, que es realitzarien en la primera nova iteració del projecte. Algunes d'aquestes millores entraven dins la planificació inicial del projecte, però per culpa de problemes que han causat una modificació temporal a aquesta planificació, algunes funcionalitats han quedat fora d'aquesta primera versió del projecte. Altres en canvi són millores

que han anat sorgint durant el transcurs del projecte. S'organitzaran en millores d'aplicació, pàgina d'administració i servidor.

### 10.1 Aplicació mòbil

- Implementació de la tecnologia NFC: Aquest component és un dels que estava planificat per a implementar dins el projecte actual. Com s'ha explicat a l'apartat de problemes trobats, es va decidir no implementar aquesta tecnologia, ja que hauria comportat una sèrie de problemes que no compensarien amb els beneficis afegits d'utilitzar NFC tenint implementat un sistema de codis QR. És per això que en una futura iteració, disposant de més temps, la incorporació de la tecnologia NFC seria una de les primeres tasques a realitzar.
- Event trobada: De la mateixa manera que amb l'NFC, aquesta tasca formava part dels requisits opcionals del sistema. Com que no s'ha pogut implementar en aquesta versió, seria una de les primeres tasques a realitzar en futures iteracions.
- Remodelació interfície gràfica: Aquesta és una de les millores típiques actualitzacions de qualsevol projecte de software. En el cas d'aquest projecte concret, s'ha utilitzat ionicFramework per realitzar aquesta interfície gràfica. Aquest framework encara es troba en fase Beta, per tant, arribaran moltes millores de cara al futur i seria interessant aplicar-les per poder aconseguir una experiència d'usuari més satisfactòria.

### 10.2 Web administració

- Mostra d'estadístiques: Una funcionalitat interessant a implementar en una futura versió, seria la inclusió d'estadístiques genèriques sobre les ubicacions i accessos que afegixen els usuaris. Mostrar aquestes estadístiques de forma gràfica facilitaria la lectura de dades a l'administrador del sistema.

### 10.3 Servidor

Com s'ha explicat anteriorment, el servidor no té funcionalitats pròpies que pugui oferir a l'usuari final, sinó que el servidor permet que les funcionalitats, tant de l'aplicació mòbil com del servei d'administració, puguin funcionar correctament. Per tant les millores en la part de servidor estaran directament relacionades amb les millores a implementar en l'aplicació i servei d'administració.

## 11 CONCLUSIONS

Un cop finalitzat aquest projecte es poden treure conclusions tant de la feina realitzada, com del que ha suposat personalment treballar en projecte de desenvolupament de software utilitzant tecnologies noves fins al moment.

Pel que fa a la part tècnica del projecte, s'ha pogut elaborar un sistema capaç de resoldre els problemes exposats a l'estudi de viabilitat inicial. Tot i que no s'han pogut assolir la totalitat de requisits, tots els essencials han estat implementats. S'ha aconseguit un sistema compost de tres

elements capaços de treballar conjuntament per aconseguir que l'usuari pugui utilitzar el servei d'una forma molt senzilla.

Un dels punts forts d'aquest projecte és la utilització de tecnologies en el món del desenvolupament de software. Personalment estic satisfet per haver estat capaç d'aplicar aquest ventall de tecnologies, aconseguint un sistema capaç de resoldre els problemes plantejats a l'inici del projecte.

Per últim voldria destacar l'aprenentatge personal que ha suposat la realització d'un projecte de desenvolupament de software, des de zero, passant per totes les fases de creació. El fet d'haver pogut tirar endavant aquest projecte m'ha fet veure que puc encarar nous projectes, sense por a fracassar per utilitzar tecnologies anteriorment desconegudes.

Sens dubte ha estat una experiència molt enriquidora que m'ha permès pujar un esglaó en el meu nivell de desenvolupament de software, de la mateixa manera, m'ha endinsat en la gran comunitat del desenvolupament, anteriorment desconeguda per mi.

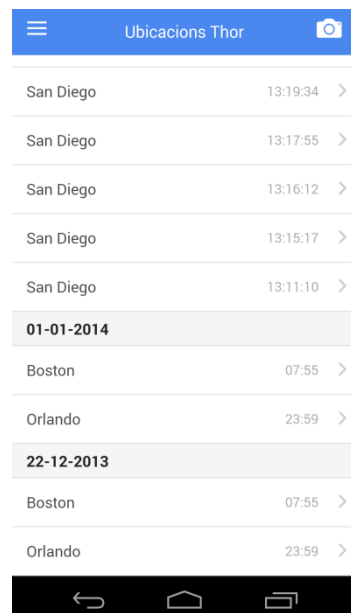
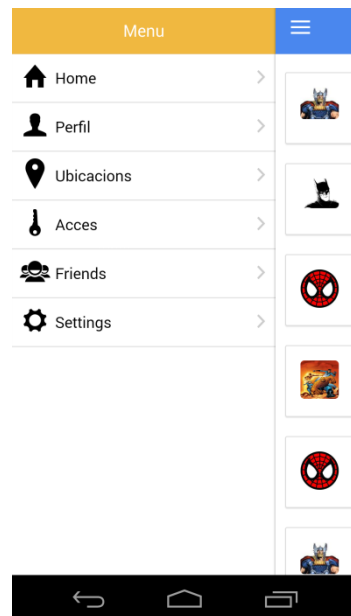
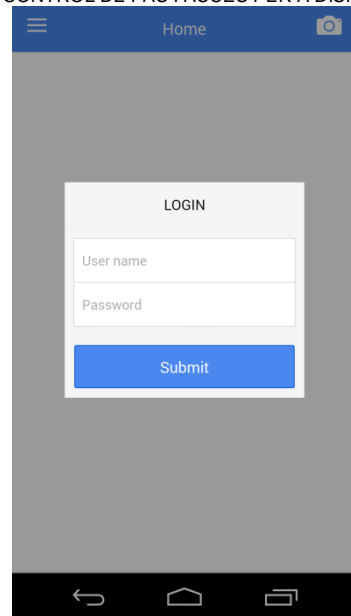
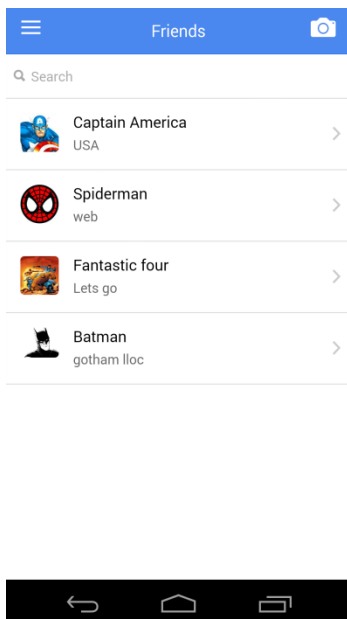
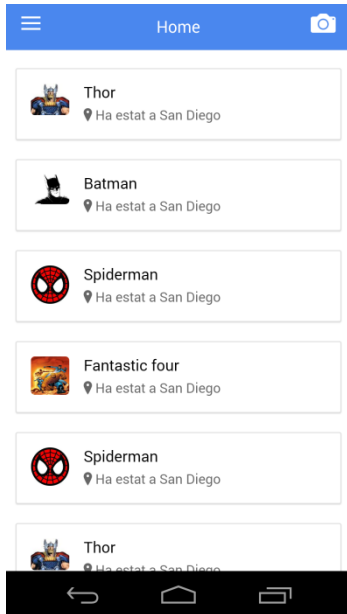
## BIBLIOGRAFIA

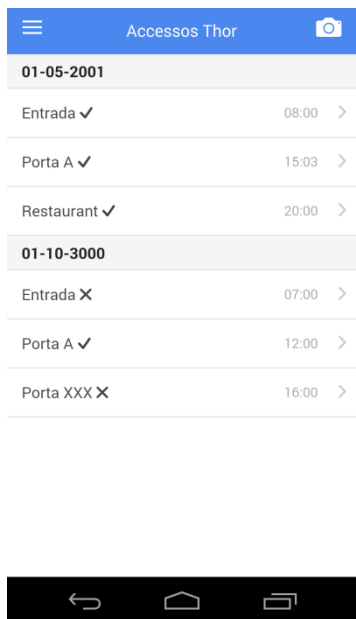
- [1] Denso Wave Incorporated, QR Essentials [Online]. Disponible a: <http://www.qrcode.com/en/> accedit per últim cop: 20/05/2014.
- [2] Vedat Coskun, Bursa Ozdenizci, Kerem Ok. A survey on near field communication technology. Springer Science+Business media New york. 1 December 2012.
- [3] Apache Cordova [Online]. Disponible a: <http://cordova.apache.org/> accedit per últim cop a: 25/06/2014.
- [4] AngularJS API [Online]. Disponible a <https://docs.angularjs.org/api>.
- [5] Ionic Framework [Online]. Disponible a: <http://ionicframework.com/docs/> accedit per últim cop: 29/05/2014.
- [6] Kristina Chodorow, Michael Dirolf. MongoDB, the definitive guide. United states of America, O'Reilly. 2010.
- [7] Hage Yaapa. Express web application development. United Kingdom, Packt publishing Ltd. 2013.
- [8] NodeJS [Online]. Disponible a: <http://nodejs.org/> accedit per últim cop a: 20/06/2014.
- [9] Mongoose module [Online]. Disponible a: <https://www.npmjs.org/package/mongoose> accedit per a últim cop a: 18/06/2014.
- [10] Thomas Myer. Beginning Phonegap. United states of America, John Wiley & Sons Inc. 2012.
- [11] Anne Van Kesteren. Cross-Origin Resource Sharing (CORS) [Online]. Disponible a: <http://www.w3.org/TR/cors/> accedit per últim cop: 23/04/2014
- [12] Json web token module [Online]. Disponible a: <https://www.npmjs.org/package/jwt-simple> accedit per últim cop a: 29/05/2014.

## APÈNDIX

### A1. APLICACIÓ MÒBIL

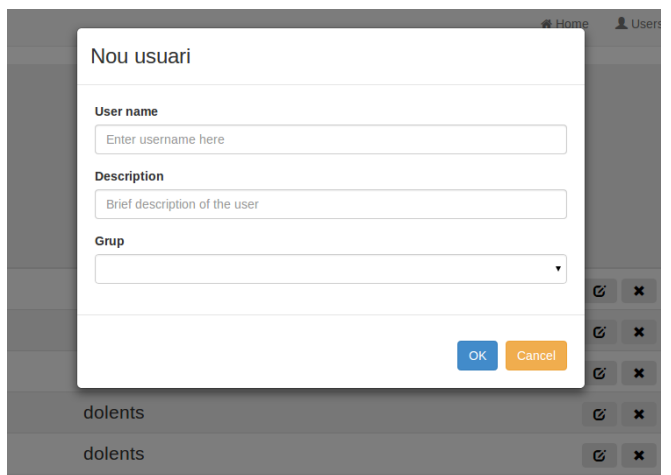
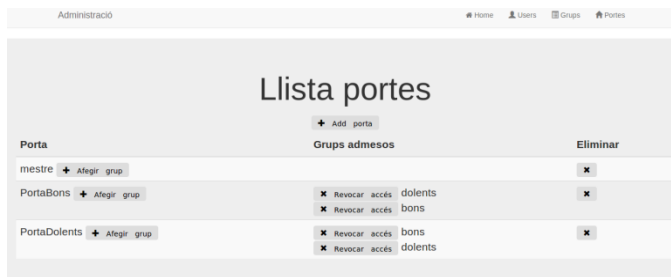
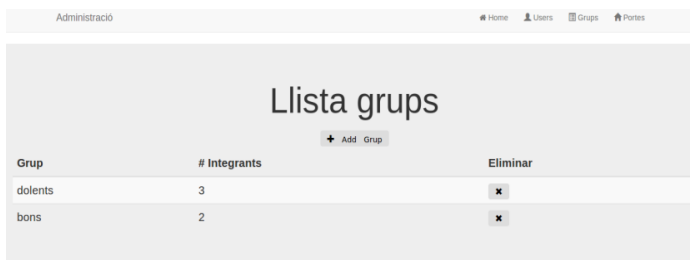
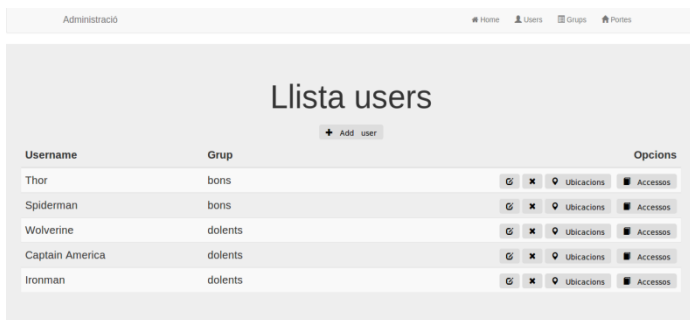
La primera secció de l'apèndix mostrarà diverses captures de pantalla de l'aplicació resultant.





## A2. WEB ADMINISTRACIÓ

A continuació es mostraran algunes captures de pantalla de l'aplicació web que permet administrar tot el sistema.



## A3. RUTES SERVIDOR

A continuació es mostrarà una llista de les diferents peticions que el servidor està preparat per rebre. De la mateixa manera es detallarà el resultat que l'aplicació obtindrà un cop demanat per aquella ruta:

- GET /users: Retorna la llista de dades de cada un dels usuaris.
- GET /users/:NomUsuari: Retorna totes les dades referents a un usuari.
- POST /users/new: Permet crear un nou usuari al sistema.
- DELETE /users/delete/: NomUsuari: Permet eliminar l'usuari seleccionat.
- GET /ubicacions/:NomUsuari: Retorna la llista d'ubicacions de l'usuari escollit.
- POST /ubicacions/nova: Permet afegir una ubicació nova al llistat de l'usuari que faci la petició.
- GET /accessos/:NomUsuari: Retorna el llistat d'accessos de l'usuari demanat.
- POST /accessos/nou: Permet afegir una petició d'accés nova a l'usuari que hagi fet la petició.
- GET /images/avatar/:NomUsuari: Retorna la imatge de perfil de l'usuari desitjat.
- GET /images/ubicacio/:lloc: Retorna l'imatge de la ubicació seleccionada.
- GET /images/acces/:lloc: Retorna l'imatge de la porta seleccionada.
- GET /clau/:idPorta : Els dispositius de les portes accediran a aquesta ruta per demanar la clau que mostraran a l'usuari que es trobi davant.

- GET /novetats: La vista principal de l'aplicació accedirà a aquesta ruta per mostrar a l'usuari les noves interaccions que han realitzat els usuaris del seu mateix grup.
- POST /login: Última ruta, permet identificar als usuaris mitjançant el seu nom d'usuari únic i la contrasenya.
- GET /admin/users: Llista d'usuaris accessible des de l'administració.
- GET /admin/grups: Retorna la llista de grups existents.
- POST /admin/grups/nou: Permet crear un grup nou.
- GET /admin/portes: Retorna llista de portes.
- POST /admin/portes/nova: Permet crear una nova porta.
- PUT /admin/portes/nouGrup: Permet garantir accés a un nou grup.
- DELETE /admin/portes/:idPorta: Eliminar porta marcada.
- DELETE /admin/:porta/grup/:nomGrup: Permet revocar l'accés d'un grup a una determinada porta.
- DELETE /admin/grups/:nomGrup: Permet eliminar grup.
- GET /porta: Retorna la pàgina web que permetrà a l'usuari demanar una clau per a la porta seleccionada.
- POST /loginAdmin: Ruta per identificar administradors.
- PUT /modificaPassword: Permet canviar la contrasenya de l'usuari.
- PUT /modificaEstat: Ruta que modifica estat de l'usuari.
- POST /modificaAvatar: Permet modificar la imatge de perfil de l'usuari.

Cal remarcar que a les rutes que comencen per /admin, només hi tindran accés els usuaris identificats com a administradors. Les altres rutes seran accedides des de l'aplicació mòbil per poder omplir les seves vistes amb les dades de retorn.