



AparCar

Aplicació per a la zona vermella de
Cardedeu

Alumne: Martí Alonso Garcia

Institut: INS Arquitecte Manuel Raspall

Àmbit: Tecnològic

Tutor: Albert Navarro

Classe: 2ⁿ Batx. C

Curs: 2019/2020

Agraïments

En primer lloc voldria donar les gràcies a les següents persones per ajudar-me en aquest treball de recerca.

A Pat Vidiella, responsable de la zona vermella de Cardedeu, per donar-me informació sobre l'actual sistema de zona vermella.

A Enric Olivé, alcalde de Cardedeu, per atendre a la proposta d'aquest projecte.

A Albert Navarro, tutor d'aquest treball, per guiar-me durant la realització d'aquest.

Al meu pare pels consells sobre com millorar les aplicacions.

Índex

1. Introducció.....	6
2. Descripció general.....	8
2.1. Servei web.....	8
2.2. Base de dades.....	9
2.3. App usuaris.....	9
2.4. App revisors.....	10
2.5. App comerços.....	10
3. Detalls tècnics.....	11
3.1. Servei web.....	11
3.1.1. Protocol.....	11
3.2. Base de dades.....	15
3.2.1 Estructura.....	15
3.3. App usuaris.....	18
3.3.1. Regex.....	18
3.4. App web comerços.....	20
3.4.1. Protocol HTTP.....	20
3.4.2. Codi QR.....	22
3.5. Seguretat.....	24
3.5.1. TLS.....	24
3.5.2. Sistema de tokens.....	26
3.5.3. Emmagatzematge de contrasenyes.....	28
4. Funcionament.....	30
4.1. App usuaris.....	30
4.1.1 Inici de sessió i registre.....	30
4.1.2. Menú de navegació.....	31
4.1.3. Estacionament.....	31
4.1.4. Vehicles.....	32
4.1.5. Mapa.....	33
4.1.6. Temps extra.....	34
4.1.7. Configuració.....	34
4.1.8. Notificacions.....	35
4.2. App revisors.....	36
4.3. App comerços.....	37

4.4. Servei web.....	38
5. Conclusions.....	29
6. Bibliografia.....	40
7. Annex.....	41
8. Glossari.....	51

1. Introducció

L'objectiu principal d'aquest treball de recerca és identificar i dissenyar totes les parts necessàries d'una aplicació mòbil per estacionar a la zona vermella de Cardedeu. Per aconseguir la total funcionalitat de l'aplicació no només és necessari la creació de l'aplicació per l'usuari sinó la creació també d'una aplicació per als revisors, una aplicació pels comerços i un servidor amb una base de dades que connectarà aquestes tres aplicacions. També s'ha de tenir en compte la seguretat, un factor molt important avui en dia en aplicacions digitals.

El segon objectiu d'aquest treball és presentar-li aquesta aplicació a l'Ajuntament de Cardedeu i que aquesta accepti la implementació d'aquest sistema.

La idea inicial d'AparCar és una aplicació Android per estacionar a la zona vermella de Cardedeu, és a dir, que en comptes d'utilitzar el disc per indicar l'hora d'arribada, utilitzar aquesta aplicació per indicar l'hora amb un sol clic.

A més a més, amb aquesta aplicació també hi haurà la possibilitat d'ampliar el temps d'estacionament utilitzant codis proporcionats per botigues, i d'aquesta manera impulsar el comerç local.

Després de fer recerca sobre aplicacions mòbils similars a AparCar, no he trobat cap aplicació que ofereixi la possibilitat d'indicar l'estacionament a zones vermelles, o zones d'aparcament gratuït limitat.

Totes les aplicacions d'aparcament a Google Play Store, com per exemple «[El Parking](#)», «[smou](#)» o «[Aparcar App Calafell](#)», són aplicacions per zones d'aparcament de pagament com zones blaves o zones verdes.

Per poder desenvolupar aquesta aplicació de manera fàcil i eficient he utilitzat diferents eines informàtiques. Cal mencionar que totes les eines utilitzades són comercialment gratuïtes i tothom les pot utilitzar.

- [IntelliJ IDEA Community](#). Eina principal per al desenvolupament de les aplicacions Android, l'aplicació web per a comerços i el servei web.
- [GIMP](#). Eina per a la creació de gràfics com el logotip de l'aplicació.

- [FileZilla](#). Eina per penjar fitxers al servidor extern.
- [PuTTY](#). Eina per comunicar-se amb el servidor extern.
- [PuTTYgen](#). Eina per a la creació de claus per a connexions segures.
- [Notepad++](#). Eina per a l'edició de fitxers de text de l'aplicació web per a comerços.

Per la part d'eines de desenvolupament no hi ha hagut cap cost, però l'allotjament del servidor extern i del domini (aparc.cat) sí que han tingut un cost econòmic.

El preu del servidor és de 3,62 € al mes. Si multipliquem aquest preu pels tres mesos que s'ha contractat, el preu total del servidor ha sigut de **10,86 €**.

El preu del domini és de **18,15 €** a l'any i només s'ha contractat un any.

El cost total, doncs, és de **29,01 €**.

Malgrat que la part visible d'AparCar només sigui les aplicacions Android i web, convé ressaltar les tecnologies que fan possible el correcte funcionament com ara Regex, el codi QR, TLS i els algorismes de hash.

Finalment, cal destacar que el resultat d'aquest desenvolupament ha estat satisfactori i l'app d'AparCar és completament funcional. També afegir que estic en procés de presentar-li aquest sistema a l'Ajuntament de Cardedeu.

2. Descripció general

AparCar consisteix de cinc parts clau: l'aplicació mòbil d'usuaris, l'aplicació mòbil de revisors, l'aplicació web de comerços, la base de dades i el servei web.

Com es pot observar en la figura 1, tots els tres clients i la base de dades es comuniquen amb el servei web. Aquest servei web i la base de dades s'executen en un servidor extern que està encès les 24 hores (també conegut com el núvol).

Aquestes cinc parts estan agrupades en dos grups: servidor i clients.

Aquests dos grups es diferencien en la manera de operar-les. Per una

banda, els usuaris, els revisors i els comerços tenen cadascú la seva app que operen ells mateixos. Per l'altra banda, el servei web i la base de dades no necessiten ser operades i donen servei a tots els clients. També es diferencien en la quantitat d'instàncies que hi ha de cada grup. De servidor només hi ha un, però instàncies de l'app d'usuaris hi haurà moltes, seran idèntiques i es comunicaran amb el servei web simultàniament.

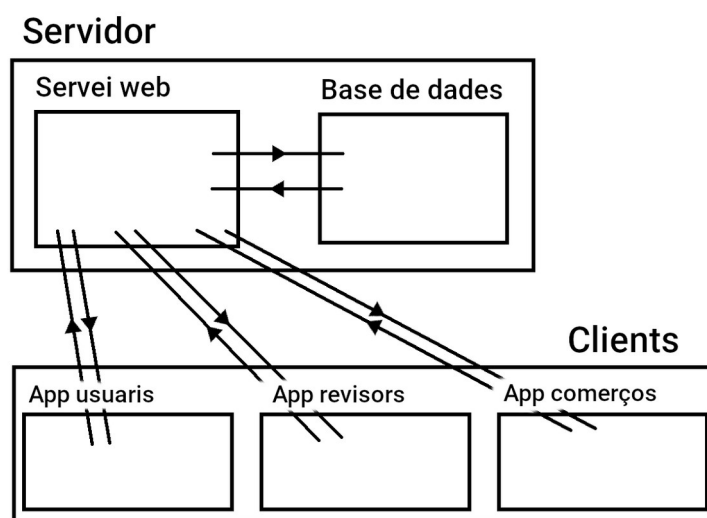


Figura 1: Diagrama simplificat de les 5 parts

2.1. Servei web

El servei web és un programa *Java*¹ que sempre està esperant que algun dels clients es connecti amb ell i li demani dades, com el llistat de vehicles, o que realitzi una acció. Aquest doncs, es connectarà amb la base de dades, si és necessari, i li retornarà al client les dades que havia demanat.

És el component més important de tots, ja que és indispensable perquè tots els altres components funcionin.

¹ Java: Llenguatge de programació molt popular. https://www.java.com/es/about/whatis_java.jsp

2.2. Base de dades

Una base de dades és un grup de dades organitzades de manera que sistemes informàtics puguin llegir, afegir i actualitzar aquestes dades de forma ràpida i segura. El sistema de base de dades que utilitzarà AparCar és MySQL.

MySQL és un sistema de gestió de base de dades que utilitza un llenguatge especial per realitzar consultes i donar instruccions a tals bases de dades. Gràcies al fet que és de codi lliure, MySQL és un dels sistemes de base de dades més populars d'Internet i amb força documentació.

2.3. App usuaris

L'aplicació mòbil per a usuaris és l'aplicació que utilitzaran els usuaris per indicar que han estacionat. Els usuaris també podran escanejar els codis que els hi donin les botigues per poder afegir temps extra a l'estacionament.

Aquesta és l'única aplicació amb la qual els usuaris interaccionaran directament. Això vol dir que ha de ser perfecta i no ha d'haver-hi cap error, i per tant s'ha de revisar el seu correcte funcionament mil·limètricament. També podem dir que aquesta aplicació ha de ser robusta, és a dir, que l'usuari només pugui utilitzar l'aplicació d'una manera, la manera correcta, i que si per qualsevol cosa hi ha un error, que l'aplicació no es pengi i que aparegui un missatge amb instruccions clares de què ha de fer l'usuari en aquest cas.

Un altre aspecte a tenir en compte és el disseny. El disseny ha de ser simple i intuïtiu perquè un usuari que no sàpiga com funciona l'aplicació la pugui utilitzar sense problemes.

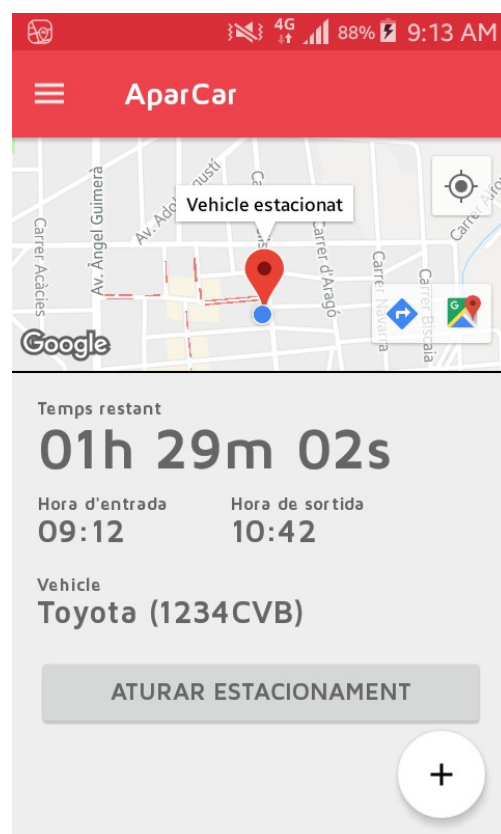


Figura 2: Captura de l'app usuaris

2.4. App revisors

L'aplicació mòbil per a revisors és l'aplicació que utilitzaran els agents de vigilància de la zona vermella per verificar si el vehicle en qüestió ha utilitzat l'aplicació mòbil per estacionar.

Com que AparCar conviuria amb el mètode tradicional de disc de la zona vermella, l'aplicació mòbil per als revisors també ha de ser molt simple i fàcil d'utilitzar perquè la implementació d'AparCar tingui un efecte mínim en l'actual vigilància de la zona vermella.

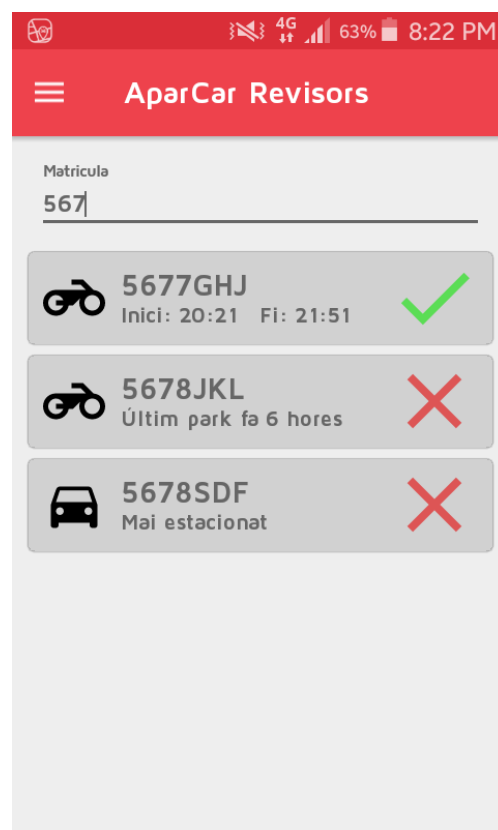


Figura 3: Captura de l'app revisors

2.5. App comerços

L'aplicació per a comerços és una aplicació web, això vol dir que per accedir-hi s'ha d'utilitzar un navegador web com Chrome o Firefox.

Els comerços hi accedeixen a través del domini aparc.cat, on apareix una pantalla d'inici de sessió (figura 4). Un cop iniciada la sessió podran generar codis (figura 5) que posteriorment podran donar als clients perquè puguin afegir més temps d'estacionament.



Figura 4: Captura pàgina web aparc.cat, pàgina d'inici de sessió

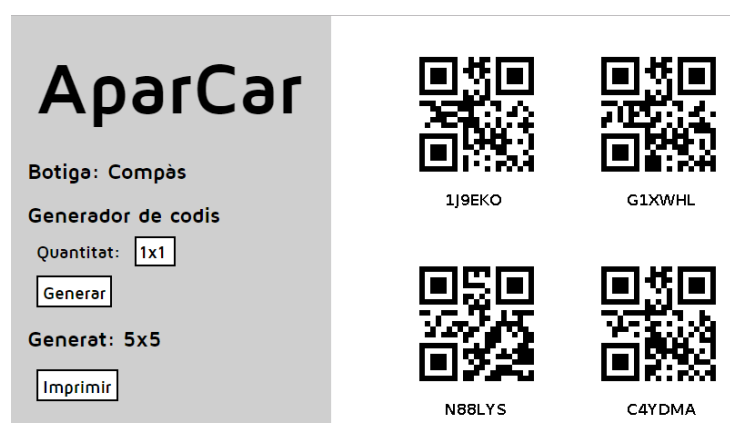


Figura 5: Captura pàgina web aparc.cat, pàgina de generació de codis

3. Detalls tècnics

Hi ha una sèrie de tecnologies que normalment no són visibles per als usuaris però que tenen un paper crucial per al funcionament del sistema en general. Al realitzar qualsevol acció des de l'app d'usuaris les dades es verifiquen amb *Regex*², s'envien al servei web amb un protocol, passen per un sistema de *tokens*³ i s'emmagatzemen en una base de dades amb una estructura definida. Tots aquests passos són necessaris per mantenir les dades completament segures.

3.1. Servei web

Per poder connectar totes les parts d'aquesta aplicació necessitem un servidor central que contingui una base de dades i que pugui “parlar” amb totes les parts, és a dir, que faci d'intermediari entre les aplicacions i la base de dades. La raó per la qual he decidit utilitzar aquest programa intermediari és perquè si l'aplicació d'usuari s'ha de connectar directament amb la base de dades, l'aplicació necessita la contrasenya de la base de dades. Això és un problema perquè qualsevol informació que estigui dins de l'aplicació d'usuaris pot ser fàcilment extreta. D'aquesta manera també podem limitar les operacions que es facin a la base de dades i implementar sistemes de seguretat.

3.1.1. Protocol

Perquè una aplicació es pugui comunicar amb el servidor correctament i de manera eficient s'ha d'establir un protocol de comunicacions. El protocol que utilitza AparCar és relativament senzill.

El missatge del client ha de consistir de “(funció):(access token):(dades)” o “(funció):(access token)”. El missatge del servei web ha de consistir de “(resultat):(dades)” o “(resultat)”.

Les figures 6, 7 i 8 descriuen les funcions enviades per l'app d'usuaris, l'app de revisors i els resultats enviats pel servei web en casos especials, respectivament.

² Regex: Expressions regulars en anglès, defineix un patró de recerca. (Apartat 3.3.1 Regex)

³ Token: Clau en anglès, contrasenya generada per màquina. (Apartat 3.5.2 Sistema de tokens)

Enviades pel client (app usuari):

Funció	Descripció	Dades requerides	Possibles resultats
version	Obté informació sobre la versió més recent, i quina és la mínima versió requerida perquè l'app funcioni.	Cap, no requereix access token	version:(última versió): (versió requerida)
register	Crea un nou compte.	La funció register té un protocol especial perquè el client encara no té l'access token, el qual és "register:(nom d'usuari): (contrasenya):(nom)"	registerok usernamealreadyregistered
login	Inicia sessió per obtenir el refresh token.	La funció login té un protocol especial perquè el client encara no té l'access token, el qual és "login:(nom d'usuari): (contrasenya)"	loginok:(token):(nom) wronglogin
signout	Tanca la sessió.	Cap	signoutok
gettoken	Obté l'access token.	La funció gettoken té un protocol especial perquè el client encara no té l'access token, el qual és "gettoken:(refresh token)"	accesstoken:(token) invalidrefresh token
startpark	Indica al servidor que ha estacionat.	Matrícula del vehicle, latitud, longitud	startparkok licensealready park wronglicense useralready park

stoppark	Indica al servidor que vol finalitzar l'estacionament.	Cap	stopparkok noactivepark
getpark	Llegeix la informació sobre l'estacionament actual.	Cap	park:(usuari):(matrícula): (data d'inici):(latitud): (longitud):(extratime) noactivepark
getextratime	Llegeix la quantitat de temps extra.	Cap	extratime:(temps extra)
addtime	Introdueix un codi per afegir temps extra d'estacionament.	Codi	addtimeok:(codi): (botiga):(duració) invalidcode usedcode expiredcode
usetime	Indica al servidor que vol utilitzar temps extra.	Quantitat de temps extra	usetimeok noactivepark noextratime invalidquantity alreadyextratime
getvehicles	Llegeix la informació sobre tots els vehicles.	Cap	vehicles:(nom del vehicle):(matrícula): (tipus de vehicle): ... novehicles
addvehicle	Afegeix un vehicle.	Nom del vehicle, matrícula i tipus de vehicle	addvehicleok invalidlicense licensealreadyadded invalidvehicletype
report	Reporta un error o suggeriment.	Tipus de report (error o suggeriment) i missatge	reportok invalidreporttype

Figura 6: Taula de funcions i resultats utilitzats per l'app usuaris

Enviades pel client (app revisors):

Funció	Descripció	Dades requerides	Possibles resultats
getlicense	Busca informació del vehicle i l'estacionament de les coincidències de la matrícula.	Matrícula del vehicle	licenses:(matrícula):(tipus de vehicle):(data de l'últim estacionament):(data d'inici de l'estacionament):(latitud):(longitud):(extratime): ... novehiclefound

*Figura 7: Taula de funcions i resultats utilitzats per l'app revisors***Resultats enviats pel servidor en casos especials:**

Resultat	Descripció
invalidaccesstoken	L'access token és incorrecte (o ha expirat).
badsyntax	El client ha enviat un missatge que no correspon amb el protocol.
unknownfunction	El client ha enviat una funció que no existeix.
nopermission	El client amb un compte d'usuari ha enviat una funció exclusiva per a revisors.
databaseerror	La base de dades ha donat un error.

Figura 8: Taula de resultats especials

3.2. Base de dades

En MySQL les dades s'organitzen en taules. Cada taula té unes columnes que determinen el tipus de dades i cada fila representa una entrada de dades.

3.2.1 Estructura

Les figures 9, 10, 11, 12, 13 i 14 descriuen l'estructura de les taules que hi ha a la base de dades d'AparCar.

Taula users

Conté informació sobre els usuaris: el nom d'usuari, la contrasenya, la *salt*⁴, el nom real de l'usuari, el tipus de compte (usuari o revisor), la data del registre, l'actual refresh token i el nombre de minuts de temps extra.

	username (Text)	password (Text)	salt (Text)	name (Text)
Entrada 1 Exemple	marti@mail.com	contrasenya	5dfR81Ac	Marti Alonso
Entrada 2 Exemple	revisor1@aj.cat	contrasenya1	rT3VsDp9	Pepito Garcia
	usertype (Text)	registertime (Número)	token (Text)	extratime (Número)
Entrada 1 Exemple	USER	1559450474640	gHj6WrY54...	120
Entrada 2 Exemple	INSPECTOR	1559451514460	ui7gH5B8y...	0

Figura 9: Taula de l'estructura de la taula users

⁴ Salt: Sal en anglès, complement de la contrasenya abans de ser processada per *hash*⁵. (Apartat 3.5.3 Emmagatzematge de contrasenyes.)

⁵ Hash: Algoritme de processament de dades d'una sola direcció. (Apartat 3.5.3 Emmagatzematge de contrasenyes.)

Taula vehicles

Conté informació sobre els vehicles dels usuaris (nom del vehicle, matrícula, tipus de vehicle i data de l'últim cop que va ser estacionat).

	username (Text)	name (Text)	license (Text)
Entrada 1 Exemple	marti@mail.com	Seat 2007	1234BCG
Entrada 2 Exemple	marti@mail.com	GasGas	3456GHJ
	type (Text)	lasttime (Número)	
Entrada 1 Exemple	CAR	1559451514660	
Entrada 2 Exemple	MOTORCYCLE	1559452254480	

Figura 10: Taula de l'estructura de la taula vehicles

Taula parks

Conté els estacionaments actuals amb el nom d'usuari, la matrícula del vehicle estacionat, la data d'inici d'estacionament, la posició en format latitud i longitud, i el nombre de minuts de temps extra afegit.

	username (Text)	license (Text)	starttime (Número)
Entrada 1 Exemple	marti@mail.com	1234BCG	1559451474640
Entrada 2 Exemple	manolo@mail.com	6789CVB	1559452514460
	latitude (Número decimal)	longitude (Número decimal)	extratime (Número)
Entrada 1 Exemple	41,638680	2,361521	30
Entrada 2 Exemple	41,637798	2,360766	0

Figura 11: Taula de l'estructura de la taula parks

Taula history

És el registre dels estacionaments finalitzats i té la mateixa estructura que la taula parks.

	username (Text)	license (Text)	starttime (Número)
Entrada 1 Exemple	pep@mail.com	4568ABC	1559431434940
Entrada 2 Exemple	jordi@mail.com	6789CVB	1559424678460
	latitude (Número decimal)	longitude (Número decimal)	extratime (Número)
Entrada 1 Exemple	41,638680	2,361521	20
Entrada 2 Exemple	41,637798	2,360766	35

Figura 12: Taula de l'estructura de la taula history

Taula timetickets

Conté els codis que els usuaris poden utilitzar per ampliar el temps d'estacionament amb la botiga on ha estat generat, la quantitat de minuts extres que conté, la data d'expiració, si ha estat utilitzat o no i qui l'ha utilitzat.

	code (Text)	shop (Text)	duration (Número)
Entrada 1 Exemple	GFTHD8	compàs	30
Entrada 2 Exemple	YH2VK5	novetats	30
	expire	used (Booleà)	usedby (Text)
Entrada 1 Exemple	1559462514460	true	marti
Entrada 2 Exemple	1559461474640	false	null

Figura 13: Taula de l'estructura de la taula timetickets

Taula shops

Conté els noms d'usuari de les botigues que poden generar codis per ampliar el temps d'estacionament amb la seva contrasenya, la *salt* i el nom real de la botiga.

	shop (Text)	password (Text)	salt (Text)	name (Text)
Entrada 1 Exemple	compas	contrasenya	5dfR81Ac	Compàs
Entrada 2 Exemple	novetats	contrasenya1	rT3VsDp9	Novetats

Figura 14: Taula de l'estructura de la taula shops

3.3. App usuaris

Com ja he mencionat prèviament, l'aplicació per a usuaris ha de ser robusta, és a dir, que l'usuari només la pugui utilitzar d'una manera, la manera correcta. És per això que l'app d'usuaris ha de validar el format de les dades introduïes per l'usuari.

3.3.1. Regex

Regex o regular expression (expressions regulars en anglès) és una cadena de caràcters que defineixen un patró de recerca. Regex s'utilitza per buscar un caràcter o una cadena de caràcters dins d'un text amb una sèrie de normes o per verificar que un text té un format específic. Dins d'AparCar he utilitzat regex per verificar el correu electrònic a l'hora de registrar-se i per verificar les matrícules dels vehicles.

Regex per a validar el correu electrònic

El format d'una adreça electrònica és el següent: aaaa@aaaa.aaa

I el regex que coincideix amb aquest format és: **(\S+)@(?=.\+\.+)(\S+)**

Aquest regex es pot dividir en tres grups:

- **(\S+)**
Aquest patró busca per una cadena de qualsevol caràcter que no sigui un espai i amb una llargada mínima d'1.
- **@**
Aquest grup és el més senzill. Simplement busca una vegada el símbol @.
- **(?=.\+\.+)(\S+)**
Aquest patró busca per una cadena de qualsevol caràcter que no sigui un espai i que contingui un punt amb caràcters davant i darrere.

Agrupant aquests tres grups definim que el text ha d'estar compost per una cadena de caràcters que no tingui espais, seguit pel símbol @ i seguit per una cadena de caràcters que no tingui espais i que contingui un punt amb caràcters davant i darrere. Aquesta definició correspon al format d'una adreça de correu electrònic.

Regex per a validar les matrícules dels vehicles

A Espanya hi ha tres sistemes de matrícules, provincial numèric (figura 15), provincial alfanumèric (figura 16) i nacional (figura 17).

Definim:

0 -> Número

X -> Qualsevol lletra

A -> Qualsevol lletra (excepte Ñ, Q i R)

B -> Qualsevol lletra (excepte A, E, I, O, U, Ñ i Q)



Figura 15: Matricula sistema provincial numèric



Figura 16: Matricula sistema provincial alfanumèric



Figura 17: Matricula sistema nacional

- **Sistema provincial numèric:**

Els formats d'aquest sistema són: X000000, XX000000, XXX000000

I el regex que coincideix amb aquests formats és: **[A-Z]{1,3}[0-9]{6}**

El patró [A-Z]{1,3} busca entre un i tres cops qualsevol lletra majúscula.

El patró [0-9]{6} busca sis números seguits del 0 al 9.

- **Sistema provincial alfanumèric:**

Els formats d'aquest sistema són: X0000A, XX0000A, X0000AA, XX0000AA

I el regex que coincideix és: **[A-Z]{1,2}[0-9]{4}(?!.*[ÑQR])[A-Z]{1,2}**

El patró [A-Z]{1,2} busca entre un i dos cops qualsevol lletra majúscula.

El patró [0-9]{4} busca quatre números seguits del 0 al 9.

El patró (?!.*[ÑQR])[A-Z]{1,2} busca entre un i dos cops qualsevol lletra majúscula excepte Ñ, Q i R.

- **Sistema nacional:**

El format d'aquest sistema és: 0000 BBB

I el regex que coincideix és: **[0-9]{4}(?!.*[AEIOUÑQ])[A-Z]{3}**

El patró [0-9]{4} busca 4 números seguits del 0 al 9.

El patró (?!.*[AEIOUÑQ])[A-Z]{3} busca tres cops qualsevol lletra majúscula excepte A, E, I, O, U, Ñ i Q.

Si la matrícula introduïda per l'usuari coincideix amb qualsevol d'aquests tres sistemes, la matrícula és marcada com a vàlida i s'introdueix a la base de dades.

3.4. App web comerços

L'aplicació per a comerços és una pàgina web, així doncs el client serà un navegador web. Igual que l'app d'usuaris i l'app de revisors, els navegadors web també es comuniquen amb un protocol amb el servei web. Aquest protocol és el protocol HTTP.

Dins d'aquesta pàgina web les botigues poden generar codis de temps extra. Per facilitar als usuaris el procés d'introduir un codi, juntament amb el codi aleatori de 6 lletres o números hi ha un codi QR que conté exactament el mateix codi.

3.4.1. Protocol HTTP

HTTP (HyperText Transfer Protocol, protocol de transferència d'hipertext en anglès) és el protocol estàndard de comunicació entre un servidor de pàgina web i un navegador web.

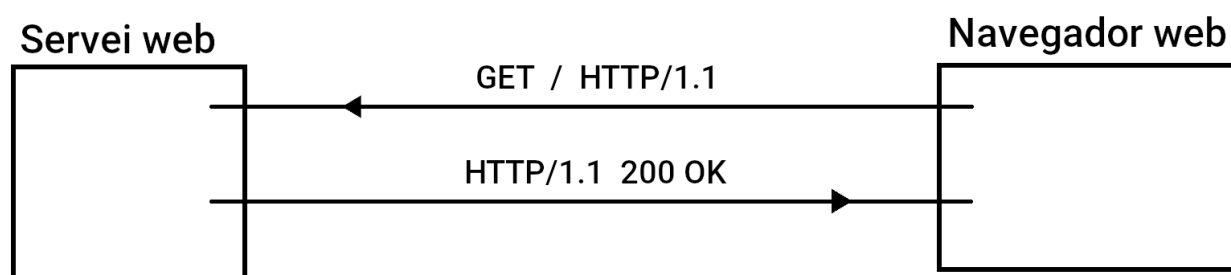


Figura 18: Diagrama protocol HTTP

La figura 18 mostra una versió molt simplificada del protocol HTTP.

El "GET / HTTP/1.1" és la sol·licitud i el "HTTP/1.1 200 OK" és la resposta.

Sol·licitud

La sol·licitud la podem dividir en tres parts:

- **Mètode (GET):**

Hi ha diversos mètodes però els més comuns, i els únics que utilitzarem a AparCar, són el GET i el POST. El mètode GET sol·licita un recurs i el mètode POST envia dades que poden modificar o crear el recurs que també sol·licita.

- **Recurs/fitxer (/):**

Indica el recurs que se sol·licita.

- **Versió del protocol (HTTP/1.1):**

Indica la versió del protocol HTTP que està utilitzant el navegador web.

Dins la sol·licitud també s'inclouen altres dades com: quin navegador web és, quin tipus de format accepta pel recurs sol·licitat, les *Cookies*⁶ que té guardades d'aquesta pàgina web, entre d'altres.

En cas que el mètode de la sol·licitud sigui POST, també s'inclouen les dades que el navegador web vol enviar.

Resposta

La resposta la podem dividir en dues parts.

- **Versió del protocol** (HTTP/1.1):

Indica que el servidor suporta la versió del protocol que el navegador web està utilitzant.

- **Codi de resposta** (200 OK):

Indica l'estat de la sol·licitud. De codis de resposta hi ha molts però els més comuns, i els únics que utilitzarem a AparCar, són:

200 OK: La sol·licitud és vàlida.

404 Not Found: El recurs sol·licitat no s'ha trobat dins del servidor.

403 Forbidden: L'usuari no té permís per veure el recurs sol·licitat.

303 See Other: Utilitzat per a realitzar redireccions.

500 Internal Server Error: Hi ha hagut un error en processar la sol·licitud.

Com en la sol·licitud, en la resposta també s'inclouen altres dades com: quant pesa el recurs sol·licitat, l'idioma del recurs, si el navegador web ha de guardar el recurs en la memòria cau o no, les *Cookies* que el navegador ha de guardar, entre d'altres.

I finalment també s'inclou el recurs sol·licitat en cas que la sol·licitud sigui vàlida.

⁶ Cookies: Galetes en anglès, dades enviades per pàgines web i guardades en l'ordinador de l'usuari mentre navega per la web.

3.4.2. Codi QR

El codi QR (Quick Response Code, codi de resposta ràpida en anglès) és un codi de barres bidimensional inventat el 1994 per la indústria de l'automòbil del Japó. Normalment s'utilitza en la indústria per a identificar objectes de manera ràpida i eficient. S'ha popularitzat entre les aplicacions tecnològiques per a representar enllaços d'Internet o codis alfanumèrics.

Un gran avantatge d'aquest codi és la capacitat que té de correcció d'error, és a dir, que si un tros de la imatge està trencada, el lector podrà llegir igualment el contingut del codi. Aquest procés de correcció d'error es fa mitjançant un algoritme anomenat *Reed-Solomon*⁷.

Com funciona el codi QR?

La figura 19 és un codi QR que conté "AparCar". Està compost per petits quadrats blancs i negres. A primera vista es poden identificar



Figura 20: Codi QR amb patró fixe i format

tres quadrats grans. Formen part d'un patró fix (zona vermella en la figura 20) que serveix perquè el lector pugui identificar la mida, l'orientació i l'angle de la imatge. Un cop el lector pot normalitzar la imatge i llegir el contingut, s'ha d'identificar la informació sobre el format del codi QR.

Els dos quadrats verds indiquen el nivell de correcció d'error, en aquest cas, nivell baix (figura 21).

Els quadrats de color lila són més informació sobre la correcció d'error, i els tres quadrats blaus indiquen el tipus de màscara.

Hi ha vuit tipus de màscares (figura 22), en aquest cas la màscara és « $(i + j) \% 2 = 0$ » que vol dir que cada quadrat va intercalant entre 0 i 1. La màscara s'aplica només a la zona de dades, tal com indica la figura 23.

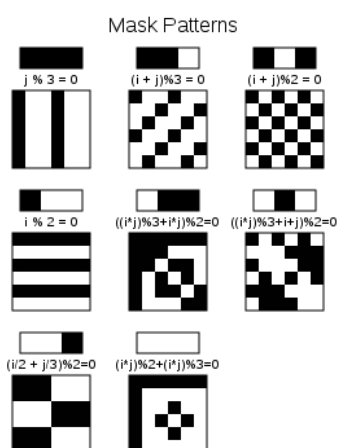


Figura 22: Tipus de màscares



Figura 19: Codi QR d'exemple



Figura 21: Nivells de correcció



Figura 23: Codi QR amb màscara

⁷ Reed-Solomon: Codis de correcció d'errors utilitzats en CDs, DVDs, codis QR, etc.

https://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction

Funciona de tal manera que si en la màscara el quadrat és negre, el mateix quadrat en les dades s'inverteix, és a dir, si el quadrat era negre, passa a ser blanc, i si era blanc, passa a ser negre.

El codi quedaria tal com indica la figura 24 després d'aplicar la màscara.



Figura 24: Codi QR amb màscara aplicada

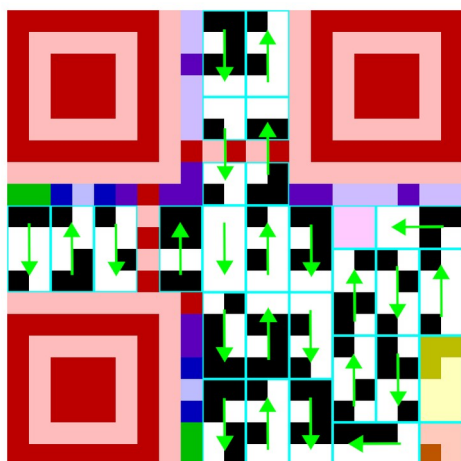


Figura 25: Codi QR amb blocs de dades

Un cop aplicada la màscara s'identifica el tipus de codificació de les dades (zona taronja en la figura 25)

En aquest cas, l'indicador 0100 vol dir que és de codificació per bytes. Aquesta codificació utilitza 8 bits per caràcter i la codificació ISO/IEC 8859-1 (variant d'ASCII⁸) per passar de números a lletres.

Així doncs, les dades es divideixen en blocs de 8 quadrats i segueixen l'ordre que indica la figura 25.

Aquests blocs de dades es llegeixen de forma binària (quadrat negre = 1, quadrat blanc = 0). S'ha de tenir en compte l'ordre en el qual es llegeixen els quadrats dins dels blocs de dades seguint la figura 26, on 1 representa el bit més significatiu i 8 representa el bit menys significatiu.

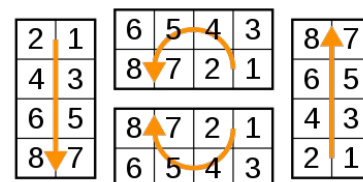


Figura 26: Ordre dels bits dels blocs de dades

El primer bloc indica la quantitat de blocs o caràcters del codi, en aquest cas llegim 00000111, que en decimal és 7. Els següents 7 blocs, doncs, contenen els caràcters que componen el contingut del codi:

- Bloc 1: 01000001 -> 65 -> A
- Bloc 2: 01110000 -> 112 -> p
- Bloc 3: 01100001 -> 97 -> a
- Bloc 4: 01110010 -> 114 -> r
- Bloc 5: 01000011 -> 67 -> C
- Bloc 6: 01100001 -> 97 -> a
- Bloc 7: 01110010 -> 114 -> r

⁸ ASCII: Codi de caràcters basat en l'alfabet llatí. <http://www.asciitable.com/>

El resultat és “AparCar”.

Després dels 7 blocs hi ha un bloc de 4 quadrats blancs (zona rosa en la figura 25) que indica el final del contingut. Els blocs restants són dades per a la correcció d'errors.

Com es pot observar, llegir un codi QR no és una tasca senzilla, i això que no he tingut en compte l'algoritme per a la correcció d'errors. És per això que a l'aplicació AparCar he utilitzat una llibreria per a la generació i lectura de codis QR.

Per a la generació he utilitzat [ZXing](#), una llibreria de codi lliure per a Java.

I per a la lectura he utilitzat [Barcode Scanner](#), una llibreria de codi lliure basada en ZXing per a Android.

3.5. Seguretat

La seguretat d'una aplicació és un factor molt important a tenir en compte.

Quan algun dispositiu envia un missatge a un altre dispositiu per Internet, tal missatge és visible per tothom i és per això que és important encriptar els missatges perquè només el receptor pugui llegir el missatge original.

També s'ha de preveure el factor humà, les contrasenyes no són suficientment segures i per això he implementat un sistema de tokens.

De la mateixa manera, s'ha de preveure possibles fuges de seguretat i emmagatzemar les contrasenyes de tal manera que mai estiguin en compromís.

3.5.1. TLS

El sistema de seguretat per a transmissions per Internet més utilitzat és TLS (Transport Layer Security, seguretat de capa de transport en anglès). Aquest sistema assegura que els missatges enviats entre un client i un servidor estan completament encriptats i ningú excepte ells poden veure el contingut original dels missatges.

Com funciona aquest sistema?

TLS utilitza una combinació de dos sistemes d'enciptació: Enciptació de clau simètrica i enciptació de clau pública

En l'enciptació de clau simètrica només hi ha una clau. Aquesta clau pot xifrar i desxifrar missatges. Si només s'utilitza aquesta enciptació, el servidor i el client han de saber la clau prèviament, però no es pot enviar per Internet perquè qualsevol la podria veure.

L'enciptació de clau pública consisteix a crear dues claus: una clau pública visible per a tothom i una clau privada que només la té el servidor. Amb la clau pública només es pot xifrar els missatges i amb la clau privada es desxifren els missatges. D'aquesta manera, un client pot xifrar un missatge amb la clau pública i el servidor pot desxifrar-lo amb la clau privada.

Com TLS utilitza aquests dos sistemes per crear una connexió totalment segura:

1. El client es posa en contacte amb el servidor i li demana la clau pública.
2. El servidor envia al client la clau pública.
3. El client crea una clau simètrica, la xifra utilitzant la clau pública i li envia al servidor.
4. El servidor desxifra el missatge, que és la clau simètrica creada pel client.
5. Ara el servidor i el client tenen la mateixa clau simètrica sense que ningú altre la sàpiga i es poden enviar missatges xifrats.

Les aplicacions d'Android d'AparCar utilitzen TLS per comunicar-se amb el servei web i l'aplicació web per a comerços utilitza HTTPS, que és HTTP (el protocol per a pàgines web) sobre una connexió segura TLS.

3.5.2. Sistema de tokens

Un sistema de tokens és essencial per a la manipulació de dades de manera segura. La figura 27 mostra com l'aplicació d'usuaris i de revisors es comuniquen amb el servidor mitjançant un sistema de tokens.

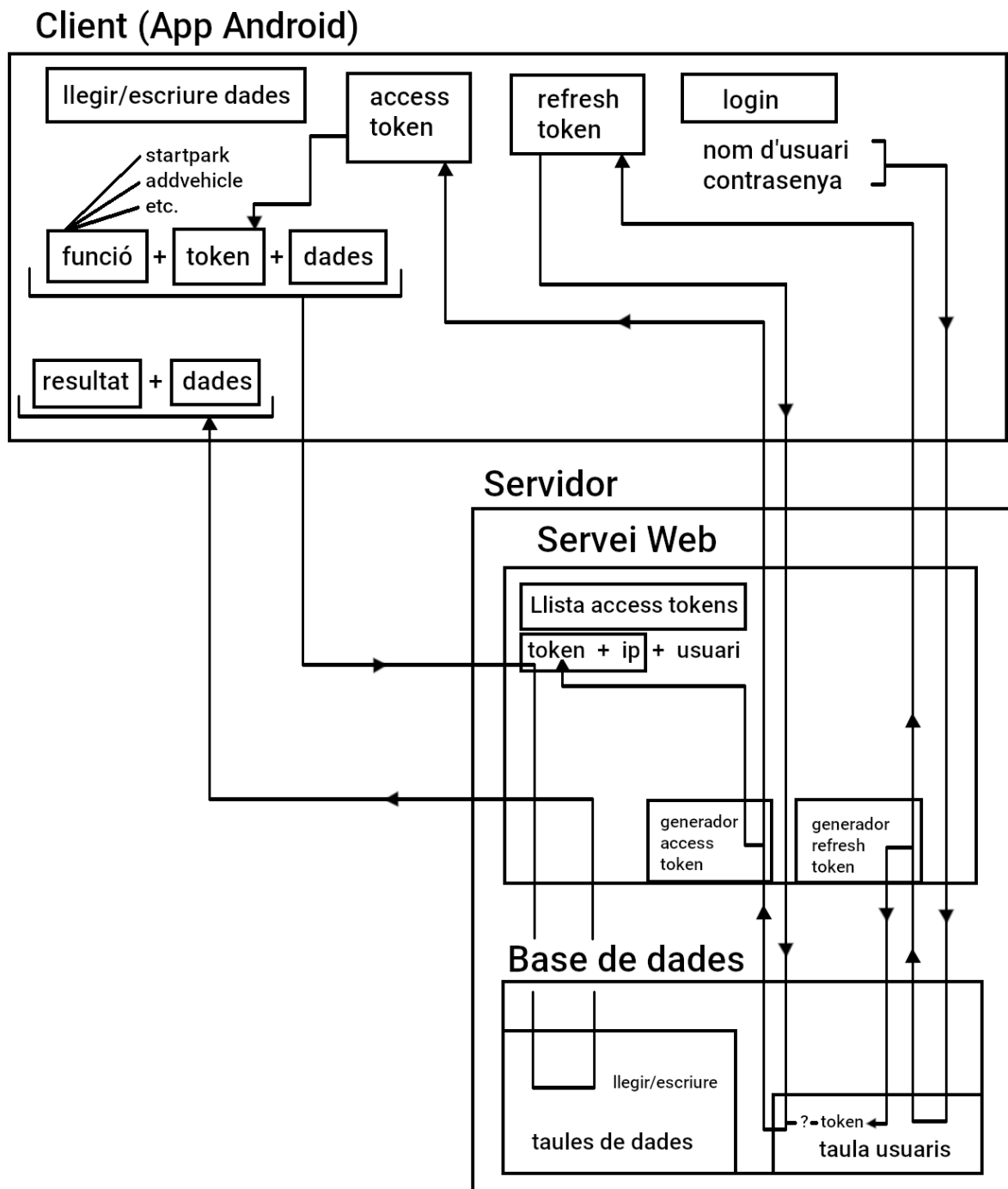


Figura 27: Diagrama del sistema de tokens

Com es pot observar, hi ha dos tipus de tokens: refresh token i access token.

- **Refresh token:**

Aquest és un tipus de token que no caduca o té una durada molt llarga. El servidor proporciona aquest token quan un usuari inicia sessió amb usuari i contrasenya. Amb aquest token el client pot demanar l'access token al servidor.

- **Access token:**

Aquest és un tipus de token que té una durada curta. S'utilitza per autoritzar el client per realitzar accions (llegir/escriure dades) en el servidor.

Una bona pràctica de seguretat és desfer-se de la contrasenya com més aviat millor un cop l'usuari l'hagi introduït a l'aplicació. El refresh token s'utilitza com a substitució de la contrasenya. Aquest token s'ha de guardar de manera segura perquè és el que proporcionarà els access tokens. Si algú no autoritzat obtingués un access token, el temps d'exposició seria mínim gràcies al fet que els access tokens tenen una durada curta. També, quan algú intenta realitzar una acció amb aquest token des d'una IP diferent de la que es va assignar a aquest token, el token caduca automàticament, negant l'accés a qualsevol que intenti entrar de manera maliciosa.

Un altre avantatge dels tokens és que són generats per màquina, fent-los més segurs que contrasenyes inventades per humans.

Com el client es comunica amb el servidor amb aquests tokens

Per començar el client proporciona el nom d'usuari i contrasenya al servidor per aconseguir el refresh token. El servidor li passa les dades a la base de dades i comprova que la informació sigui correcte. Si és així, el servidor genera un refresh token que el guarda a la base de dades i li envia al client. Un cop el client té el refresh token, li demana al servidor l'access token proporcionant-li el refresh token. El servidor comprova a la base de dades que el refresh token és correcte i genera un access token que es guarda en el servidor internament juntament amb la IP del client i el nom d'usuari. Aquest access token també s'envia al client. Un cop el client el rep, ja pot començar a llegir i escriure dades. Per fer-ho, el client enviarà la funció, el token i les dades. El servidor, doncs, comprovarà que el token i la IP són correctes, executarà la funció a la base de dades i retornarà un resultat al client.

3.5.3. Emmagatzematge de contrasenyes

Guardar les contrasenyes dels usuaris tal com són a la base de dades és molt mala idea. Si la base de dades tingués una fuga de seguretat, podríem posar en compromís molts comptes d'usuari d'altres serveis webs, ja que molts usuaris reutilitzen la mateixa contrasenya.

Per resoldre aquest problema s'utilitzen algorismes de hash. Aquests algorismes són d'una sola direcció, això vol dir que poden processar un text i donar-te un resultat però amb aquest resultat no es pot tornar al text original, tal com indica la figura 28.

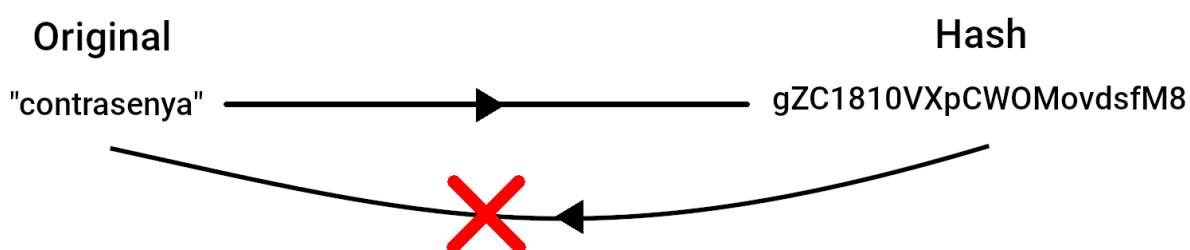


Figura 28: Exemple algoritme de hash

També tenen altres propietats que els fan molt adients per a l'emmagatzematge de contrasenyes. Si al text original se li modifica només un caràcter, el hash canviarà completament. Són molt complexes i les operacions matemàtiques utilitzades necessiten molts recursos computacionals, fent que aquests algorismes tinguin una durada i no es puguin obtenir milions de resultats per segon. Si algú volgués obtenir l'original des del hash, hauria de processar cada combinació possible amb aquest algoritme i comparar cada resultat amb el hash. Com que aquest procés té una durada, fer-ho per bilions de combinacions pot acabar sent una eternitat.

Així doncs, les contrasenyes es guarden a la base de dades després d'haver estat processades per aquests algorismes de hash. A l'hora de verificar si la contrasenya és la correcta, es processa també la contrasenya en qüestió i es compara el resultat amb el resultat guardat a la base de dades.

Però tots els problemes no s'acaben aquí, existeixen les anomenades «rainbow tables». Són bases de dades que contenen bilions de contrasenyes freqüentment utilitzades i el

seu respectiu hash ja processat. Molt probablement es podria trobar aquest hash en aquestes bases de dades i així obtenir la contrasenya original. Per solucionar aquest problema s'afegeix a la contrasenya el que se'n diu salt (sal en anglès) abans de ser processada per l'algoritme de hash. Aquesta salt és una simple cadena de caràcters aleatoris que fa que el hash canviï completament, deixant les «rainbow tables» fora de joc. S'emmagatzema conjuntament amb el hash per quan es vulgui verificar la contrasenya poder afegir la salt també a la contrasenya en qüestió, processar-la amb l'algoritme de hash i comparar-la amb el hash guardat a la base de dades.

AparCar utilitza l'última tecnologia en l'algoritme de hash: [Argon2](#)

Aquest algoritme permet regular el nombre d'iteracions aplicades al text original, i d'aquesta manera regular més o menys quant de temps és necessari per a processar les contrasenyes.

4. Funcionament

4.1. App usuaris

4.1.1 Inici de sessió i registre

El funcionament de l'app d'usuaris és molt senzill.

Per començar, els usuaris s'hauran de registrar.

Per registrar-se, hauran d'introduir el seu nom, el seu correu electrònic, una contrasenya i donar-li clic a **REGISTRAR-SE**. La contrasenya ha de tenir un mínim de 8 caràcters.

En el cas que ja estiguin registrats, podran iniciar sessió donant-li clic a **Iniciar sessió**.

En la pantalla d'iniciar sessió els usuaris hauran d'introduir el seu correu electrònic, la seva contrasenya i donar-li clic a **INICIAR SESSIÓ**.

Per dirigir-se a la pantalla de registre només cal donar-li clic a **Crear un compte**.

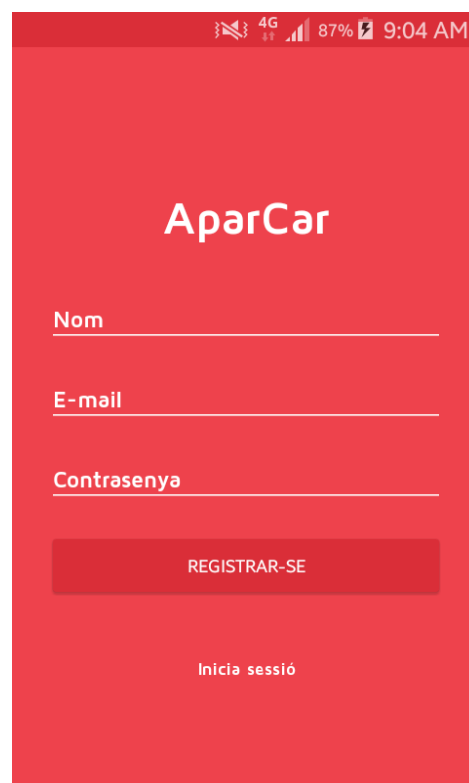
Aquesta és una captura de pantalla de la pantalla de registre de l'aplicació AparCar. La interfície té un fons vermell. A la part superior, hi ha el logotip 'AparCar' en blanc. Sota, hi ha tres camps de text blancs amb etiquetes 'Nom', 'E-mail' i 'Contrasenya' en vermell. A la dreta de cada etiqueta hi ha una línia blanca per a l'entrada de text. Al centre, hi ha un botó vermell amb el text 'REGISTRAR-SE' en blanc. Just a sota del botó, hi ha un enllaç 'Inicia sessió' en blanc.

Figura 29: Captura app usuaris, pantalla de registre

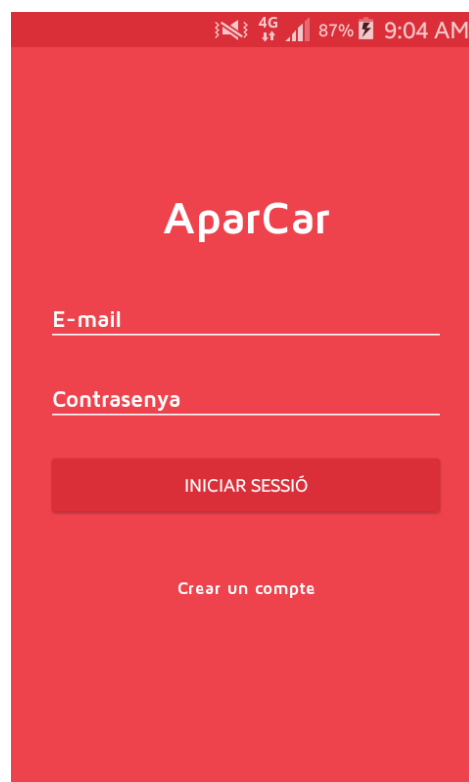
Aquesta és una captura de pantalla de la pantalla d'inici de sessió de l'aplicació AparCar. La interfície té un fons vermell. A la part superior, hi ha el logotip 'AparCar' en blanc. Sota, hi ha dos camps de text blancs amb etiquetes 'E-mail' i 'Contrasenya' en vermell. A la dreta de cada etiqueta hi ha una línia blanca per a l'entrada de text. Al centre, hi ha un botó vermell amb el text 'INICIAR SESSIÓ' en blanc. Just a sota del botó, hi ha un enllaç 'Crear un compte' en blanc.

Figura 30: Captura app usuaris, pantalla d'inici de sessió

4.1.2. Menú de navegació

Un cop iniciada la sessió arribarem a la pantalla inicial. Al principi, la pantalla inicial sempre serà la pantalla d'estacionament però podem anar a les altres pantalles mitjançant el menú de navegació.

Per obrir el menú de navegació només cal donar-li clic a les tres barres horitzontals que trobarem a dalt a l'esquerra. Des d'aquest menú podrem navegar fins a les pantalles d'estacionament, vehicles, mapa, temps extra i configuració.

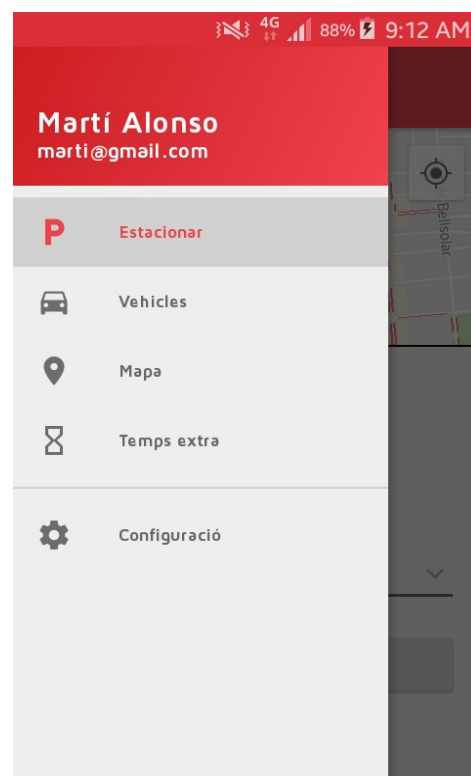


Figura 31: Captura app usuari, menú de navegació

4.1.3. Estacionament

La pantalla d'estacionament la podem dividir en dues parts.

A la part superior hi ha un mapa de Cardedeu amb les àrees de zona vermella marcades.

A la part inferior hi ha el panell d'estacionament. Aquest panell ens indica el temps d'estacionament, l'hora d'entrada i l'hora de sortida. També conté un desplegable on podem seleccionar el vehicle que volem estacionar. Per estacionar només cal donar-li clic a **INICIAR ESTACIONAMENT**.

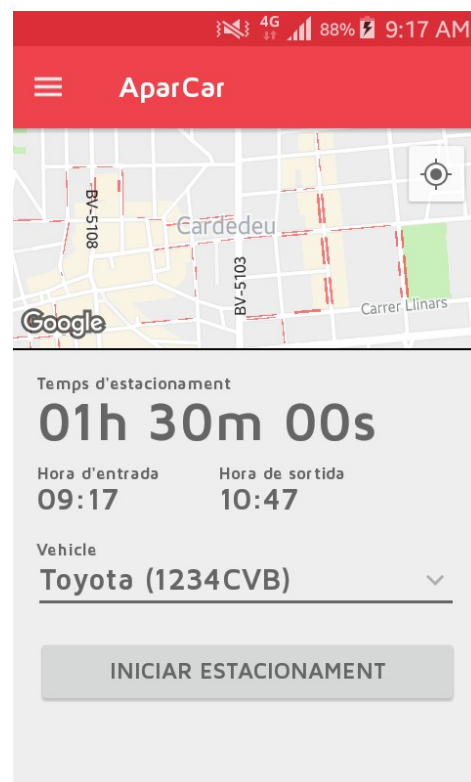


Figura 32: Captura app usuari, pantalla d'estacionament

Un cop hem indicat que hem estacionat, en el mapa ens apareixerà un marcador amb la posició del nostre vehicle.

Per indicar que ja no estem estacionats a la zona vermella només cal donar-li clic a **ATURAR ESTACIONAMENT**.

Donant-li clic al botó rodó amb el símbol + podem ampliar el temps d'estacionament seleccionant quant temps extra volem afegir a l'estacionament actual.

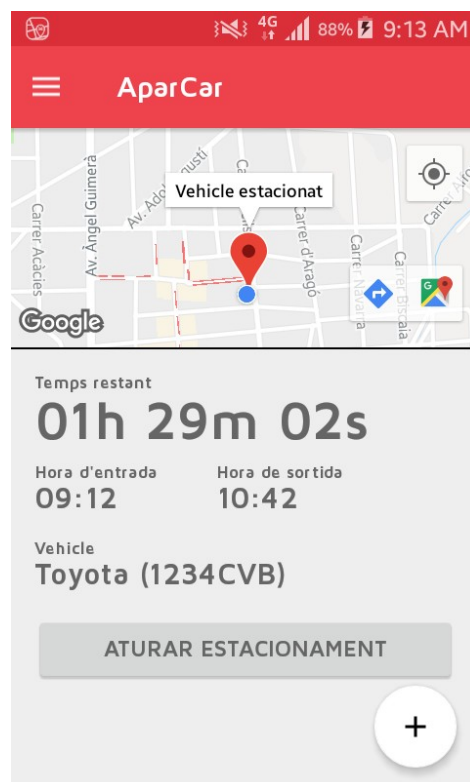


Figura 33: Captura app usuari, pantalla d'estacionament un cop estacionats

4.1.4. Vehicles

La pantalla de vehicles mostra una llista de tots els vehicles que hem afegit a l'aplicació. Cada element de la llista ens indica quin tipus de vehicle és, quin és el seu nom i quina és la seva matrícula.

Per afegir un vehicle haurem de donar-li clic al botó rodó amb el símbol + que ens portarà a la pantalla per afegir vehicles.

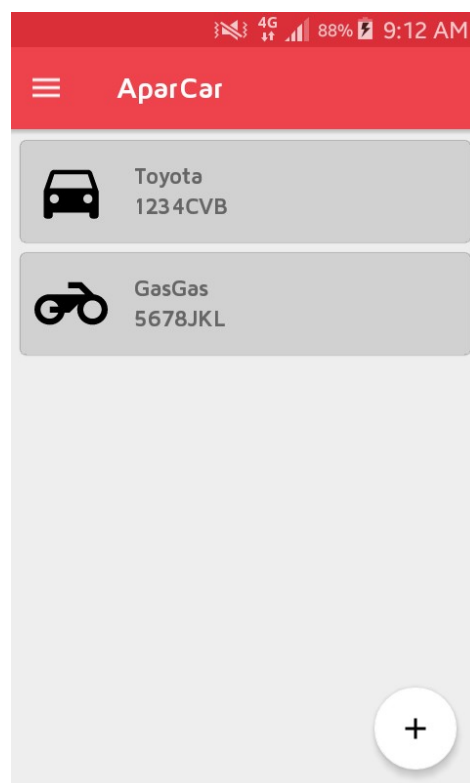


Figura 34: Captura app usuari, pantalla de vehicles

Aquí els usuaris hauran d'indicar el tipus de vehicle (cotxe o moto), el nom del vehicle i la seva matrícula. El nom del vehicle pot ser qualsevol nom i només és representatiu per als usuaris. Serveix perquè els usuaris puguin identificar el vehicle en qüestió dins de l'app sense la matrícula.

Per confirmar aquestes dades hauran de donar-li clic a **AFEGIR VEHICLE**.

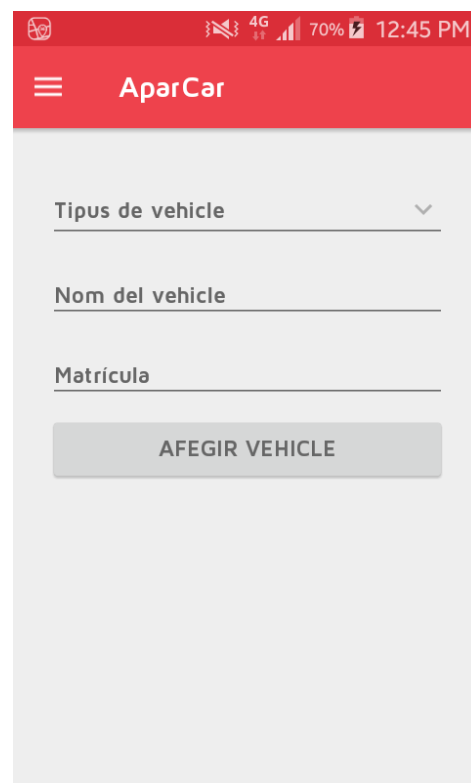


Figura 35: Captura app usuaris, pantalla per afegir un vehicle

4.1.5. Mapa

La pantalla del mapa mostra a pantalla completa un mapa de Cardedeu. Tal com en la pantalla d'estacionament, les àrees de zona vermella estan marcades de color vermell i hi ha un marcador amb la posició del vehicle estacionat.

El botó de la part superior mou el mapa de tal manera que la posició actual queda en el centre i els dos botons de la part inferior obren l'aplicació de Google Maps amb la posició del vehicle estacionat com a destí.

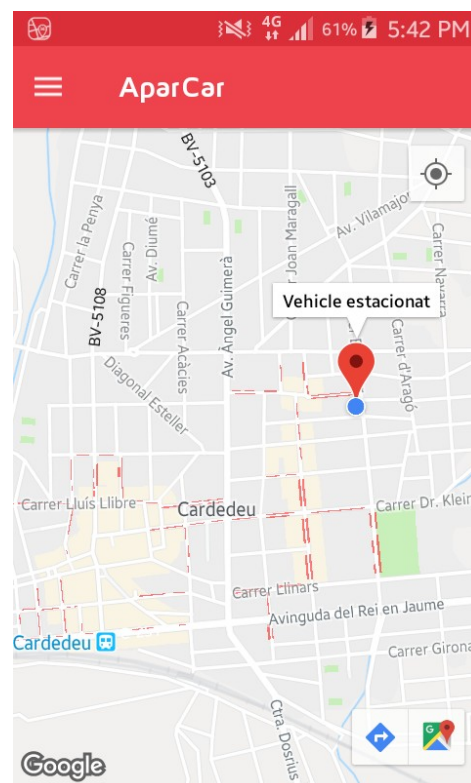


Figura 36: Captura app usuaris, pantalla del mapa

4.1.6. Temps extra

La pantalla de temps extra mostra el temps extra total acumulat i l'historial de codis de temps extra introduïts a l'aplicació. Cada element de l'historial ens indica el codi, de quina botiga és el codi i el temps extra que contenia aquest codi.

Per introduir un nou codi haurem de donar-li clic al botó rodó amb el símbol + que ens portarà a la pantalla amb el lector de codis QR.



Figura 37: Captura app usuàries, pantalla de temps extra

4.1.7. Configuració

La pantalla de configuració està dividida en tres categories: notificacions, app i compte.

En la categoria de notificacions l'usuari pot decidir si vol rebre notificacions o no. També pot escollir quan vol rebre una notificació d'alerta del temps restant d'estacionament.

En la categoria d'app l'usuari pot reportar un error o fer un suggeriment. Donant-li clic a qualsevol dels dos, apareixerà una petita finestra on podrà escriure una breu explicació de l'error o suggeriment.

En la categoria de compte l'usuari pot tancar la sessió actual i iniciar sessió amb un altre compte.

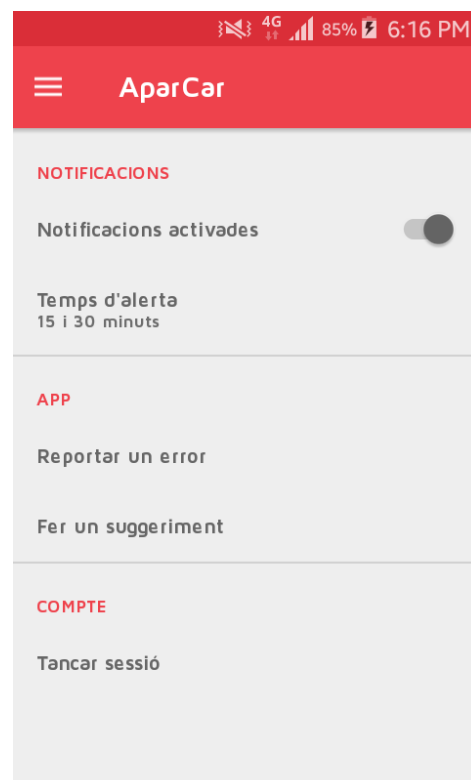


Figura 38: Captura app usuàries, pantalla de configuració

4.1.8. Notificacions

Un cop l'usuari hagi estacionat, l'aplicació mostrarà una notificació constant que cada minut es va actualitzant automàticament amb el temps d'estacionament restant.

Depenen de com ho hagi especificat l'usuari, també rebrà una notificació d'alerta quan quedin 15 i/o 30 minuts.

A més a més, l'usuari rebrà una notificació quan el temps d'estacionament s'hagi esgotat completament.

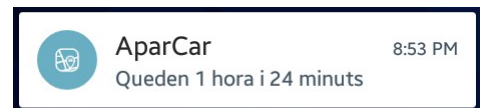


Figura 39: Captura app usuari, notificació constant

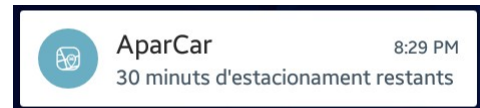


Figura 40: Captura app usuari, notificació de 30 minuts restants

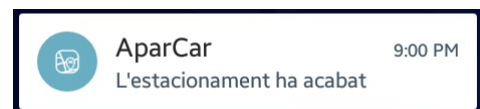


Figura 41: Captura app usuari, notificació de fi d'estacionament

4.2. App revisors

El funcionament de l'app de revisors també és molt senzill. Després d'identificar-se utilitzant una pantalla d'inici de sessió idèntica a la de l'app d'usuaris, s'arriba a la pantalla de verificació.

En aquesta pantalla els revisors hauran d'introduir la matrícula del vehicle en qüestió. A mesura que es vagi escrivint, aniran apareixent resultats que coincideixen amb la matrícula.

El vistiplau verd significa que el vehicle està actualment estacionat. En aquest cas el resultat també ens mostra quina és l'hora d'entrada i quina és l'hora de sortida.

Per contra, la creu vermella significa que el vehicle no està estacionat, i el resultat mostra quan fa de l'últim estacionament.

Si li donem clic a un resultat amb el vistiplau, apareixerà una pantalla amb el mapa de Cardedeu. En aquest mapa estarà marcada la posició del vehicle en qüestió.



Figura 42: Captura app revisors, pantalla de verificació

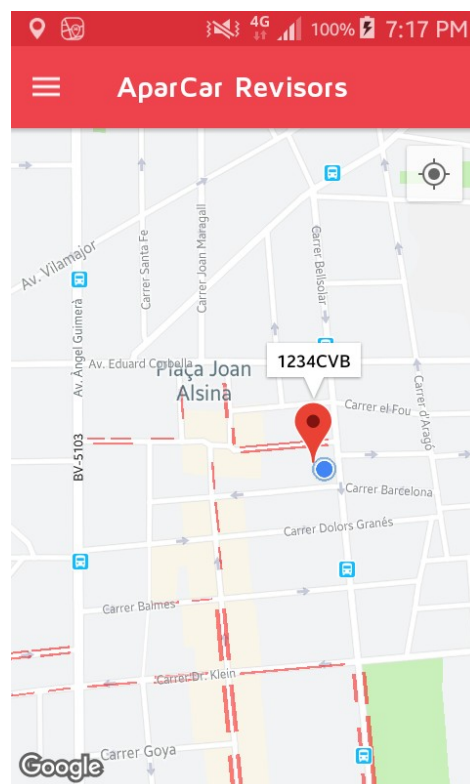


Figura 43: Captura app revisors, pantalla del mapa

4.3. App comerços

Per accedir a l'app de comerços les botigues hauran d'introduir l'adreça aparc.cat al seu navegador d'Internet de preferència.

Si és el primer cop que hi entren els apareixerà un simple formulari per iniciar sessió (figura 44).

Després d'iniciar sessió, accediran a la pàgina principal. Aquí és on podran generar els codis. Per fer-ho simplement s'ha de seleccionar la quantitat de codis que es volen generar utilitzant un desplegable i donar-li clic a **Generar**.

Un cop generats, es poden imprimir fàcilment només fent clic a **Imprimir**.



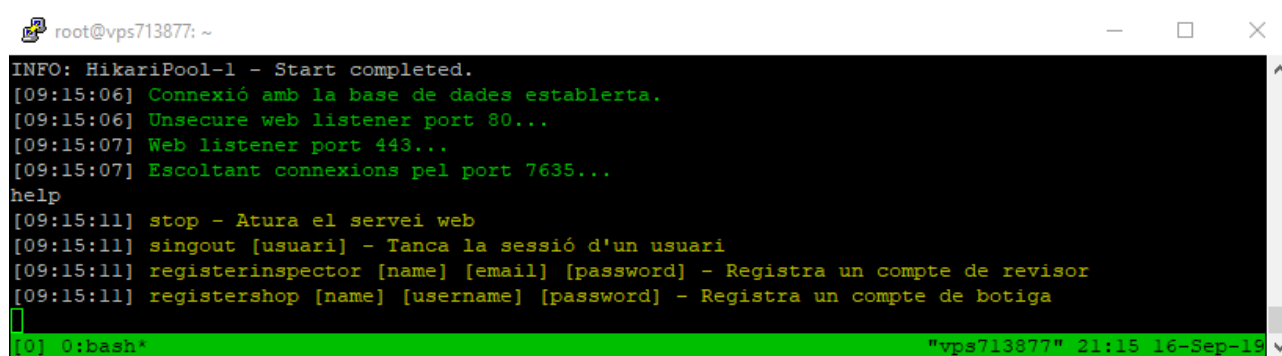
Figura 44: Captura app comerços, formulari d'inici de sessió



Figura 45: Captura app comerços, generació de codis

4.4. Servei web

Encara que el servei web funcioni de manera completament autònoma té una sèrie de comandaments per a casos especials.



```
root@vps713877: ~  
INFO: HikariPool-1 - Start completed.  
[09:15:06] Connexió amb la base de dades establerta.  
[09:15:06] Unsecure web listener port 80...  
[09:15:07] Web listener port 443...  
[09:15:07] Escoltant connexions pel port 7635...  
help  
[09:15:11] stop - Atura el servei web  
[09:15:11] singout [usuari] - Tanca la sessió d'un usuari  
[09:15:11] registerinspector [name] [email] [password] - Registra un compte de revisor  
[09:15:11] registershop [name] [username] [password] - Registra un compte de botiga  
[0] 0:bash* "vps713877" 21:15 16-Sep-19
```

Figura 46: Captura de l'eina PuTTY, connexió amb el servei web

El comandament **help** ens mostrarà una llista dels comandaments que podem utilitzar.

- **stop**

Aquest comandament atura el servei web. S'utilitza en el cas que es vulgui fer manteniment en el servei web en si o en el servidor sencer.

- **singout**

Aquest comandament tanca la sessió de l'usuari especificat. S'utilitza en el cas que es detectin irregularitats en el comportament del compte i es vulgui tancar la sessió d'aquest compte per seguretat.

- **registerinspector**

Aquest comandament registra un compte de revisor. S'utilitza per crear els comptes que utilitzaran els revisors, ja que aquests comptes no es poden registrar des de l'app.

- **registershop**

Aquest comandament registra un compte de botiga. De la mateixa manera que per als comptes de revisors, s'utilitza aquest comandament, ja que aquests comptes no es poden registrar des de la web.

5. Conclusions

Així doncs, aquest projecte ha estat completament satisfactori. He estat capaç d'implementar tots els sistemes tecnològics necessaris; identificar i dissenyar les parts marcades en l'objectiu principal, i tenir sempre en compte la seguretat.

Per un costat, està el resultat que és un sistema amb una aplicació mòbil per estacionar a la zona vermella de Cardedeu completament funcional. Per l'altre, estan tots els coneixements adquirits sobre les tecnologies necessàries per a la creació d'un sistema llest per ser utilitzat en condicions reals.

Per acabar, cal mencionar que l'objectiu secundari de presentar-li aquest projecte a l'Ajuntament de Cardedeu continua en procés i estic pendent d'una entrevista amb l'alcalde.

6. Bibliografia

- ESEPE STUDIO; 16/08/2005; ¿Qué es MySQL?
<https://www.esepestudio.com/noticias/que-es-mysql>
- JONNY FOX; 23/06/2017; Regex tutorial — A quick cheatsheet by examples
<https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285>
- ROBOMATICS; 30/03/2013; How to Decode a QR Code by Hand
<https://www.youtube.com/watch?v=KA8hDldvfv0>
- SEBASTIAN PEYROTT; 07/10/2015; Refresh Tokens: When to Use Them and How They Interact with JWTs
<https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/>
- TOM SCOTT; 20/11/2013; How NOT to Store Passwords! - Computerphile
<https://www.youtube.com/watch?v=8ZtlnCIXe1Q>
- WIKIPEDIA; 13/09/2019; Matrículas automovilísticas de España
https://es.wikipedia.org/wiki/Matr%C3%ADculas_automovil%C3%ADsticas_de_Espa%C3%B1a
- WIKIPEDIA; 13/09/2019; Hypertext Transfer Protocol
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- WIKIPEDIA; 17/09/2019; QR code
https://en.wikipedia.org/wiki/QR_code
- ZEUS KERRAVALA; 09/11/2018; What is Transport Layer Security (TLS)?
<https://www.networkworld.com/article/2303073/lan-wan-what-is-transport-layer-security-protocol.html>

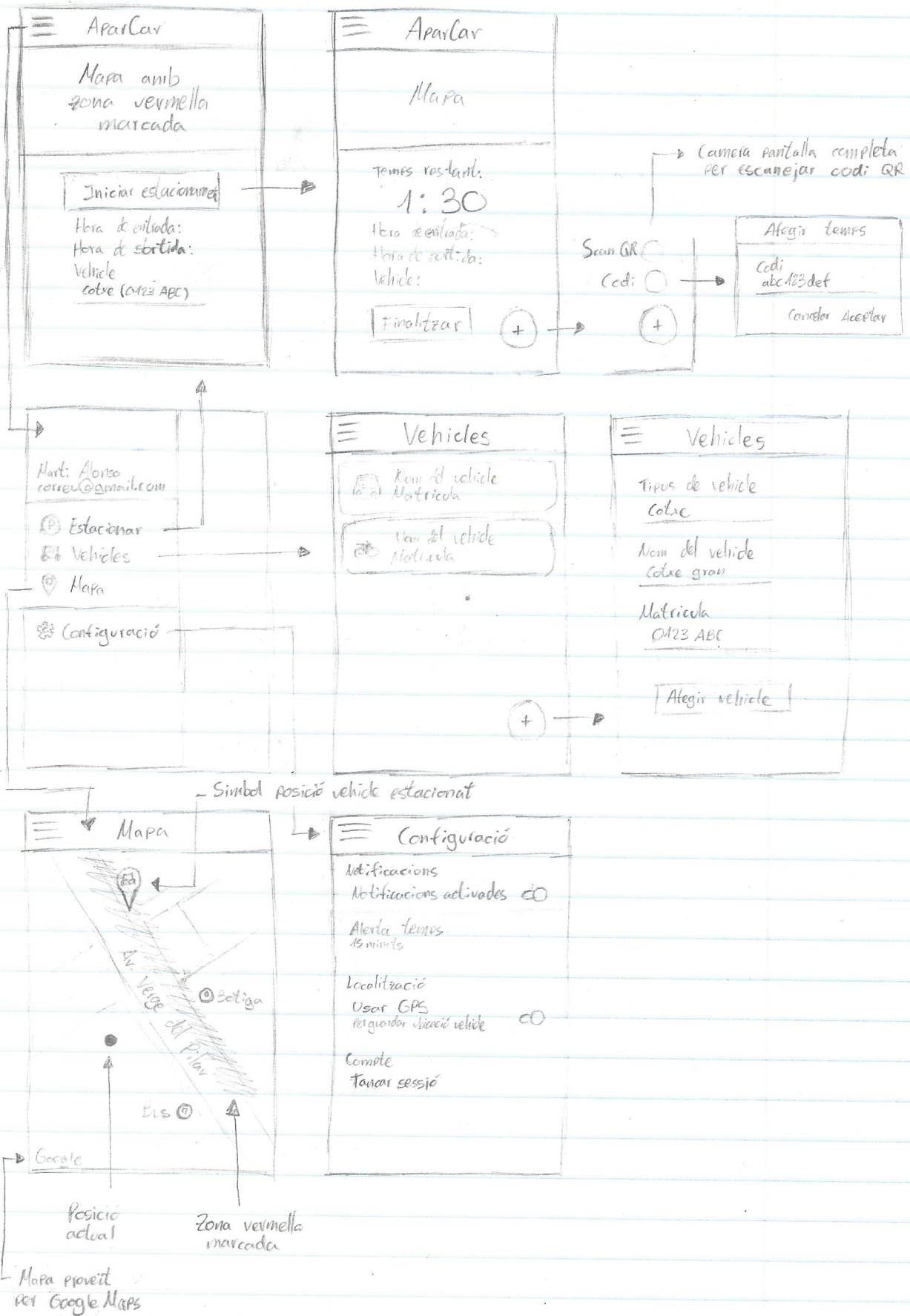
7. Annex

7. Annex.....	41
7.1. Esbós disseny inicial de l'app.....	42
7.2. Instància de proposta a l'Ajuntament.....	44
7.3. Document adjunt a la proposta.....	46

7.1. Esbós disseny inicial de l'app

Esbós del disseny inicial de totes les pantalles de l'app d'usuaris. Aquest disseny ha anat canviant a mesura que s'ha desenvolupat l'aplicació.

AparCar (Disseny inicial)



7.2. Instància de proposta a l'Ajuntament

Instància presentada el dia 13 d'agost del 2019 proposant la implementació d'aquest sistema a la zona vermella de Cardedeu i sol·licitant una entrevista amb l'alcalde per a una presentació més detallada.



Ajuntament de Cardedeu

Plaça de Sant Joan, 1 T 938 444 004
08440 Cardedeu F 938 711 477

www.cardedeu.cat
NIF P-0804500-G

cardedeu@cardedeu.cat

Nom i cognoms <i>Marti Alonso Garcia</i>		Document d'identitat <i>47996391Z</i>
Adreça <i>Bellsolar 85 2º 2ª ESC A</i>		Codi Postal - Població <i>08440 Cardedeu</i>
Telèfon <i>658027414</i>	Correu electrònic <i>martialonso11@gmail.com</i>	
Actuant en representació de		Document d'identitat
Adreça		Codi Postal - Població
Telèfon	Correu electrònic	

EXPOS

Una aplicació per a telèfons mòbils per a la zona vermella de Cardedeu.
És un projecte per facilitar als residents l'estacionament a la zona vermella i per impulsar el comerç local.

SOL·LICITO

Una entrevista amb el Senyor Alcalde per presentar-li aquesta proposta més detalladament.

Cardedeu. Data: *13/08/19*

Signatura

Al dors del document trobareu informació sobre Protecció de Dades del vostre interès

7.3. Document adjunt a la proposta

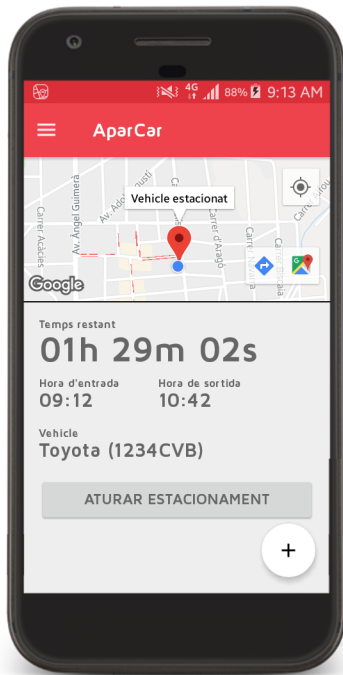
Document de presentació d'aquest projecte que va ser adjuntat a la instància quan va ser presentada a l'Ajuntament. Conté una breu descripció del funcionament de les aplicacions mòbils.



AparCar

Aplicació per a la
zona vermella de
Cardedeu

Amb aquesta aplicació podem oferir ventatjes als residents i treballadors locals afectats per la zona vermella i **implusar el comerç** recompensant les compres locals amb temps d'estacionament extra.



▲ Disseny simple i eficaç

Amb un simple click, estacionem a la zona vermella de manera senzilla i sense complicacions.

▲ Notificacions

Rep notificacions quan l'estacionament està a punt de finalitzar.

Multiples vehicles ▲

Suport de multiples vehicles per estacionar en qualsevol moment.





Impulsem el comerç

Recompensar el comerç local amb temps d'estacionament extra mitjançant codis que poden oferir les botigues als seus clients.



GPERI6

Vigilància

Amb una app també molt senzilla per als revisors de la zona vermella.





Contacte

Martí Alonso Garcia

658 02 74 14

martialonso11@gmail.com

8. Glossari

App: Abreujament d'application (aplicació en anglès), programa informàtic en un dispositiu mòbil.

Android: Sistema operatiu per a mòbils creat per Google. Més del 85% dels mòbils en el mercat utilitzen Android. Les aplicacions es desenvolupen amb el llenguatge de programació Java i XML.

Java: Un dels llenguatges de programació més populars. És un llenguatge creat per Oracle que pot córrer en gairebé qualsevol dispositiu. En Android, Java s'utilitza per programar la lògica de l'aplicació.

XML: Acrònim d'eXtensible Markup Language (llenguatge de marques extensible en anglès), llenguatge que permet l'organització i marcatge d'elements en documents. En Android, XML s'utilitza per al disseny de la interfície d'usuari.

Token: Clau en anglès, té la mateixa funció que una contrasenya però és generat per un ordinador i l'usuari no ha d'interaccionar amb ell.

Cookies: Galetes en anglès, dades enviades per pàgines web i guardades en l'ordinador de l'usuari mentre navega per la web.

IP: Acrònim d'Internet Protocol (Protocol d'Internet en anglès), fa referència a l'adreça d'un dispositiu connectat a Internet.

