Smart Parking per una Smart City

Treball fi de Màster Interuniversitari en Seguretat de les TIC

Estudiant: *Oriol Dols Rodríguez*
Consultors: *Josep Maria Gastó Heras*
Jordi Castellà-Roca
Alexandre Viejo

Índex

Resum

Tothom avui dia utilitza els dispositius mòbils per a moltes i variades activitats quotidianes, buscar informació d'un local a prop on poder dinar, mirar els horaris dels transports públics de la ciutat actualitzats gairebé al moment, buscar el taxi lliure més proper de la zona on estàs o inclús bescanviar cupons. En definitiva, el telèfon mòbil és una eina de gran utilitat i ens serveix per millorar la nostra qualitat de vida gràcies a les moltes aplicacions de serveis que existeixen i que ens faciliten en una petita mesura el nostre dia a dia.

Per altra banda, cada cop a les grans ciutats és molt complicat trobar places d'aparcament lliures fàcilment, a les hores puntes es provoquen embussos, accidents a causa de la poca atenció que es té de la carretera al estar buscant aparcament. En definitiva, un estrès pels ciutadans i un augment de la contaminació degut als embussos.

Aprofitant les avantatges i la llibertat que ens ofereixen els dispositius mòbils, s'ha ideat una aplicació que només pintant un número identificador a cada plaça d'aparcament d'una ciutat permeti visualitzar totes les places d'aparcament d'una ciutat i poder anar-hi directament sense necessitat de perdre temps buscant. Així quan s'arribi a la plaça escollida només caldrà introduir el número i formalitzar el pagament.
A banda de la evident avantatge de no perdre temps i millorar la circulació de la ciutat, l'usuari final, en cas de tenir permisos avançats(per exemple al comprar l'aplicació en comptes de descarregar-se-la gratuïtament) podrà alliberar la plaça en el moment que desitgi i els minuts restants els podrà fer servir gratuïtament al proper aparcament.
Un altre avantatge, ara respecte l'administrador dels aparcaments és que podrà tenir un control més clar sobre quines són les àrees on s'aparca més i per tant on potser fa falta ampliar o les zones deficitàries ja que poca gent hi aparca. A més a més, el cos de seguretat encarregat de vigilar l'ús correcte dels aparcaments rebran informes dels propis usuaris indicant incidències, ajudant a controlar els malfactors.

1. – Introducció

Una SmartCity és una ciutat que a través de les tecnologies de la informació i les comunicacions permet que la seva infraestructura i els seus serveis públics siguin més interactius i més accessibles per al ciutadà.

És el paradigma d'un món digital en el que tot està connectat. Des de dispositius mòbils fins a objectes sense connectivitat avui dia, com poden ser els edificis, vehicles, aparcaments o els electrodomèstics. És a dir, qualsevol cosa que pugui ser controlable i/o se'n pugui extreure'n informació.
És en definitiva, es tracta d'una plataforma digital que permet millorar l'economia, el benestar dels ciutadans i permet estar en contacte directe amb tots els elements que formen una ciutat.

Un dels problemes que tenen avui dia les ciutats és el gran volum de trànsit i els ajuntaments han d'invertir molts diners en crear un model de gestió urbanística que ajudi a disminuir del centre de la ciutat gran part d'aquest volum de trànsit, sobretot el provocat a l'hora de buscar aparcament. Alguns exemples d'aquestes inversions són la creació de diferents zones d'aparcament de pagament (zona blava, verda i taronja) per a intentar millorar la mobilitat i facilitar l'aparcament.
Tanmateix, aquestes zones no ajuden a disminuir del tot el trànsit al centre de la ciutat, provocant congestions degut a la gent que circula buscant aparcament. A més a més, aquesta circulació densa fa que augmentin els accidents provocats per estar més pendent de buscar aparcament que no pas de la calçada.

A banda dels problemes viaris esmentats, el mètode de pagament i el preu per l'ús d'aquests aparcaments és ineficient i excessiu per als ciutadans. Així com ho és el sistema de detecció d'infraccions.

Degut a aquesta problemàtica, s'ha decidit desenvolupar una aplicació de Smart Parking que abasti la gestió de tots els aparcaments, tant públics(ajuntaments) com privats(empreses de pàrquing) de tota la ciutat, afegint el control de la gestió dels infractors directament sobre el ciutadà, ajudant així al cos o empresa encarregada d'aquesta tasca actualment.

L'aplicació permet veure en temps real l'estat de tots els pàrquings de la ciutat per així estalviar temps a l'hora de buscar aparcament, provocant que la gent deixi de buscar-ne pel centre de la ciutat per anar directament on hi ha places lliures. Els aparcaments només caldrà identificar-los afegint un ròtol o una pintura per a poder introduir aquest codi identificador al mòbil alhora de fer el pagament.

A part, el sistema de pagament serà acord als minuts que estiguis ocupant la plaça i, permeten ampliar el temps que calgui. També hi haurà una bossa de minuts o crèdit restant amb els

minuts que has pagat però no has utilitzat, per a que els puguis utilitzar durant el teu pròxim aparcament.

Addicionalment, i com a control d'incidències, s'ha creat un sistema d'avís d'incidències en el qual els propis ciutadans poden avisar d'un ús incorrecte de les places d'aparcament, ajudant així tant als cossos de seguretat vial com a la pròpia educació de la ciutadania.

1.1. – Motivació

Des de l'empresa Staitec va sorgir la possibilitat de realitzar una aplicació prototip per tal de aconseguir una millora en la gestió dels aparcaments d'una ciutat i, indirectament millorar el trànsit d'aquesta, aprofitant el poder de les tecnologies de la informació i la comunicació[1] . Gràcies a les TIC els usuaris poden comprovar en temps real l'estat dels aparcaments de cada zona de la ciutat. Mentre que a l'administració encarregada dels aparcaments li permet gestionar tota aquesta informació digitalment. És a dir, permet recollir informació de les zones amb més incidències reportades o les zones que cal reforçar, en definitiva, permet extreure'n informació per a poder millorar el servei.

Amb la visió de les avantatges de les TIC en ment, s'ha buscat la millor manera de contribuir directament en la vida quotidiana dels ciutadans. Per aquest motiu, s'ha buscat com implementar l'après durant el transcurs del màster(evitar injeccions SQL, gestió de rols, gestió de la informació i el coneixement) però que alhora pogués servir per a millorar la qualitat de vida de la ciutadania. Tant pel fet de que s'evitin estressos innecessaris, com per a la qualitat de la informació que les administracions en poden extreure sobre l'ús dels aparcaments a la seva ciutat.

A més a més, també hi ha el repte de veure si la pròpia ciutadania és capaç de gestionar-se ella mateixa alhora d'informar a les autoritats sobre infraccions d'aparcament. Ja que el sentit d'aquest projecte depèn, en part, del seu correcte ús per part dels ciutadans.

Ja existeixen solucions d'aparcament intel·ligent a moltes ciutats, la majoria utilitzant sensors mentre que aquesta te avantatge sobre la resta de mecanismes ja que l'únic que es necessita és que s'identifiqui a la pròpia plaça fixa el nom d'aquell aparcament en concret. Per tant és un sistema de baix cost i avui dia es essencial trobar mesures austeres i al mateix temps útils.

Es una gran motivació veure com amb una petita idea i un petit desenvolupament es pot aconseguir millorar el rendiment de les ciutats i donar un servei al ciutadà.

1.2. – Objectius

Per a resoldre la problemàtica detectada s'ha desenvolupat una aplicació mòbil que utilitzant les TIC permeti:

- Autentificar-se i autenticar-se mitjançant diferents rols que donen accés a diferents funcionalitats dins l'aplicació.
- Gestionar múltiples vehicles per cada usuari registrat.
- Comprovar l'estat dels aparcaments en temps real i poder filtrar quins tipus d'aparcament vols visualitzar.
- Reportar incidències per part dels usuaris sobre mals usos de les places de pàrquing directament a l'òrgan encarregat.
- Pagament segur.
- Trobar el camí més curt cap a una determinada localització.
- Guardar un històric de les últimes places comprades.
- Creació de zones on s'agrupen els aparcaments per a poder extreure'n informació més ràpidament per part de l'òrgan gestor dels aparcaments

1.3. – Planificació

El desnvolupament d'aquesta aplicació es separarà en 4 parts:

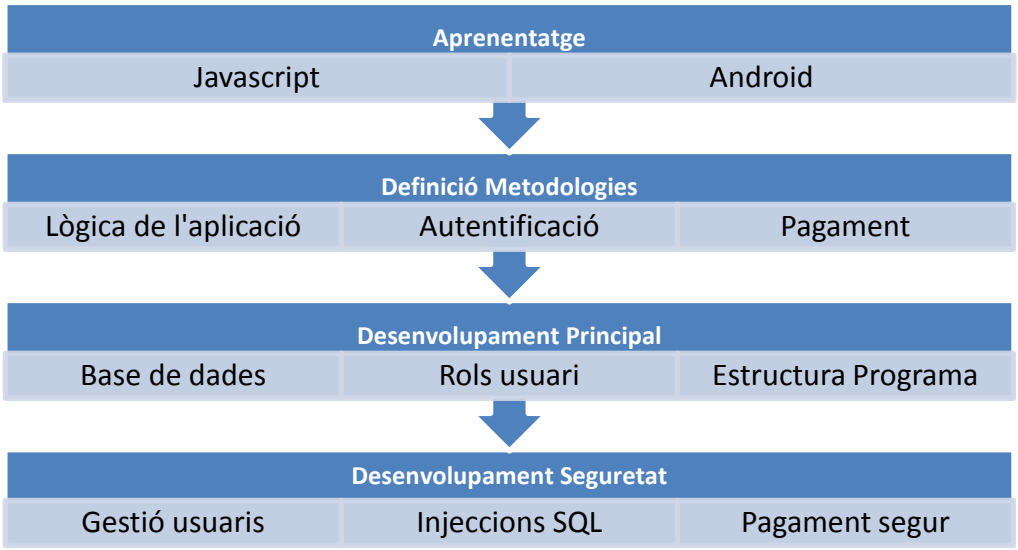
- Aprenentatge bàsic per a desenvolupar una aplicació per Android (15 dies).
- Definició del projecte i estudi de les metodologies a desenvolupar (15 dies):
  - Definició del funcionament de l'aplicació
  - Definició de la metodologia de pagament segura a desenvolupar
  - Definició de la gestió de rols
- Desenvolupament de l'aplicació:(2 mesos):
  - Desenvolupament de l'estructura de l'aplicacio
  - Desenvolupament de les metodologies de seguretat escollides
  - Desenvolupament de la gestió de rols
- Fase final de te steig i depuració exhaustiva d'errors (15 dies).

Durant la primera quinzena del primer mes es va buscar informació, tutorials i guies per aprendre a desenvolupar una aplicació per Android mitjançant HTML5, a través de la plataforma PhoneGap, així com de coneixements bàsics d'Android.

La segona quinzena va estar dedicada a definir com s'organitzarà l'aplicació, és a dir, es defineix quin és, a gran escala, el funcionament de l'aplicació, com es vol gestionar l'autentificació mitjançant rols i quins mecanismes de pagament es poden implementar.

Un cop definit l'esquelet de l'aplicació, es comença a desenvolupar l'estructura de l'aplicació, la base de dades, les connexions amb el servidor, el desenvolupament de l'autentificació mitjançant rols, la metodologia de pagament i aplicar les mesures de seguretat estudiades (protecció d'injeccions a la BD, gestió segur de rols i pagament xifrat)

Finalment, l'últim més s'ha dedicat a realitzar el joc de proves per comprovar que l'aplicació funciona correctament i té una tolerància d'errors acceptable.



Planificació Desenvolupament Aplicació

1.4. – Organització

Al capítol 2 es troben les tecnologies que s'han utilitzat per a realitzar aquest projecte i quin ha estat el motiu d'escollir-les.

Al capítol 3 es troben els actors que formen el projecte, la seva lògica i el disseny de la base de dades com dels seus elements de seguretat.

Al capítol 4 hi ha la implementació més rellevant de cada mòdul.

Al capítol 5 es troben els jocs de proves, sobre quin entorn s'han realitzat i quines proves s'han fet. Així com els resultats obtinguts.

Al capítol 6 es troben les conclusions finals del projecte, què s'ha aconseguit respecte al planificat i quins són els resultats. A més a més hi ha l'anàlisi de com s'encararà el projecte d'aquí en endavant.

2. –Tecnologies

Per a realitzar aquest projecte s'han utilitzat diverses tecnologies per a que l'aplicació sigui multi plataforma i, per tant, accessible a més gent. S'ha buscat obtenir un rendiment i una aparença semblant al que s'aconseguiria a través d'un desenvolupament amb codi natiu, per tal de que l'experiència de l'usuari sigui satisfactòria a la vegada que no noti que es tracta d'una aplicació inicial i no nativa.

Per oferir un servei associat amb les tecnologies de la informació i la comunicació s'ha dotat a l'aplicació de metodologies per a obtenir informació directe de la ciutat a la vegada que permet realitzar pagaments de forma segura.

2.1 – Phonegap

Phonegap[2][3][4] és un framework o plataforma WORA (*Write-Once-Run-Anywhere*) que permet dissenyar, provar i utilitzar aplicacions per a que siguin compatibles en diferents dispositius mòbils, configuració de pantalles i navegadors. El resultat d'aquesta plataforma són aplicacions híbrides, que si bé no donen el mateix resultat en temes de rendiment que les aplicacions natives, ofereixen un gran rendiment.

Aquesta plataforma va ser creada per *Nitiobi*[5] amb llicencia de codi lliure. Aquesta companyia va ser comprada per Adobe, passant així el control de PhoneGap a un dels gegants del software, ja que estava interessat en la evolució del HTML5.

Les principals raons per escollir desenvolupar aquesta aplicació mitjançant PhoneGap i no amb codi natiu són les següents:

- Multi plataforma
  Permet desenvolupar aplicacions per a dispositius iOS, Android, BlackBerry OS, Windows Phone 7, WebOS (Palm), Symbian o Bada.

- Ràpid
  Permet desenvolupar aplicacions no gaire complicades el més ràpidament possible, sempre i quan sigui una aplicació que no necessiti de molts recursos.

- Codi únic

A excepció d'algunes petites modificacions que diferencien les funcionalitats de cada dispositiu, amb un sol desenvolupament de codi l'aplicació és multi plataforma.

- API
  Phonegap proporciona una API que permet accedir a característiques i funcionalitats natives dels dispositius mòbils utilitzant JavaScript

- Empaquetat
  El desenvolupament es fa mitjançant HTML5, CSS3 i Javascript, però el producte final és un arxiu binari permeten que es pugui distribuir a qualsevol Marketplace.

- Còmode
  Està disponible en forma de plugins per a desenvolupar-ho al Eclipse o al XCode, així com l'opció d'un SDK.

- Frameworks addicionals
  Phonegap sol utilitzar frameworks com ara JQuery o Sencha Touch que faciliten que la visualització i el comportament siguin com una aplicació nativa.

2.2. – HTML5

El HTML5[6] és una nova versió del llenguatge universal HTML. La primera versió formava el llenguatge de programació bàsic i utilitzats per totes les planes web. Bàsicament, el codi HTML està format per un conjunt d'etiquetes que permeten donar format i mostrar diferents tipus de contingut, ja siguin imatges o textos. Tanmateix, no és capaç d'interactuar amb variables o bases de dades.

Les millores més rellevants d'aquesta nova versió són les següents:

- Simplificació i noves maneres d'especificar alguns paràmetres o peces de codi
- Permet reproduir contingut multimèdia sense necessitat de plugins
- Animacions sense necessitat d'Adobe Flash
- Emmagatzematge de dades per banda del client permeten que les aplicacions siguin més ràpides.
- Efectes i nova versió de CSS per a una millor interfície.
- Geolocalització que permetrà als llocs web conèixer la ubicació física des d'on s'accedeix.
- Tipografies no estàndards.

S'ha escollit aquest llenguatge pel fet de ser el pilar sobre el que treballa PhoneGap, juntament amb el Javascript i el CSS3.

2.3. – Javascript

El Javascript[7] és un llenguatge de programació del costat del client. Això significa que és el navegador o l'aplicació la que suporta tota la carrega de processament. Degut a la seva gran compatibilitat amb la majoria de navegadors és el llenguatge de costat del client més utilitzat. Quan es parla de llenguatge pel costat del client ens referim a llenguatges que poden ser tractats directament pel navegador i no necessiten cap pretractament, fent que sigui més ràpida l'execució que no pas quan treballen al costat del servidor.

És un llenguatge de programació que permet fer els programes amb rapidesa. A més a més, permet afegir efectes especials a les pàgines creant continguts dinàmics i també permet crear pàgines interactives.

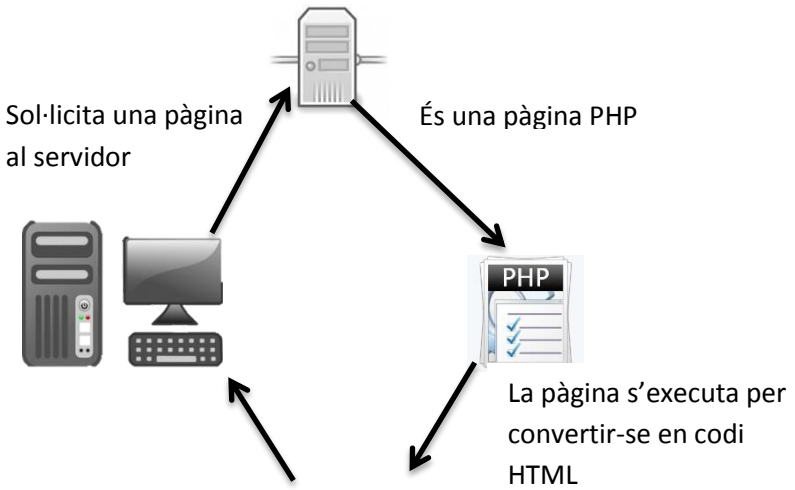
S'ha utilitzat el llenguatge Javascript ja que és el llenguatge principal que utilitza PhoneGap, apart de ser molt versàtil i intuïtiu.

2.4. – PHP

Mentre que Javascript és un llenguatge de programació per part del client, el PHP[8] treballa per part del servidor. És un llenguatge de programació independent de plataforma, gratuït, ràpid i amb una gran llibreria de funcions i documentació.

Es parla d'un llenguatge del costat del servidor quan s'executa al servidor web, just abans que s'enviï la pàgina a través d'internet cap al client. Les pàgines que s'executen al servidor poden realitzar accessos a base de dades, connexions de xarxa i d'altres tasques que després es passarà cap al client. Aquest l'únic que rebrà serà una pàgina amb el codi HTML resultant de l'execució del PHP, sent compatible amb tots els navegadors.

En el següent dibuix es pot observar l'esquema del funcionament de les pàgines PHP:



La pàgina HTML s'envia al client.

S'ha escollit el PHP per a consultes al servidor ja que és gratuït i ens permet treballar amb un servidor MySQL.

2.5. – MySQL

"*Structured Query Language*"(SQL) és un llenguatge estàndard de comunicació amb les bases de dades. És a dir, es tracta d'un llenguatge normalitzat que permet treballar amb qualsevol altre tipus de llenguatge(PHP o ASP) en combinació amb qualsevol tipus de base de dades(MS Access, SQL Server, MySQL[9], entre d'altres).

Les principals raons per escollir MySQL respecte han estat les següents:

- Gran velocitat de realització d'operacions. El millor en quant a rendiment.
- Baix cost en requeriments per a l'elaboració de la base de dades.
- Facilitat de configuració i instal·lació.
- Suporta gran quantitat de Sistemes Operatius

2.6. – CSS3

El CSS (Cascading Style Sheet) és un llenguatge per definir l'estil o l'aparença de les pàgines web, escrites amb codi HTML o dels documents XML. Es va crear per separar el contingut de l'estil, permeten als dissenyadors mantenir un control molt més precís sobre l'aparença de les pàgines.

La nova versió del CSS té com a novetat més important, de cara als desenvolupadors de web, és la incorporació de nous mecanismes per mantindré un major control sobre l'estil amb el que es mostren els elements de les pàgines, sense haver de recórrer a trucs o *"hacks"*, els quals sovint complicaven el codi.

Aquesta tecnologia serveix per a millorar la interfície gràfica de l'aplicació per tal d'aconseguir un efecte més agradable per l'usuari final.

S'ha escollit aquesta tecnologia ja que és el llenguatge predefinit per establir l'aparença a través de PhoneGap.

2.7. – Frameworks i API's externes

Les tecnologies abans esmentades serveixen per donar forma i vida a l'aplicació, però per a dotar-la de més funcionalitats i millorar la interfície i la interacció del dispositiu amb l'usuari són necessaris llibreries externes. Per tal de millorar la interfície i el nivell d'interacció s'ha escollit el Sencha Touch[10], que permet dotar l'aplicació de certs mecanismes per a que treballi de la mateixa manera que ho faria una aplicació nativa.

Després, per al posicionament dels aparcaments i de la ubicació de l'usuari al mapa s'ha escollit l'API de Google[11] degut a que aporta una gran facilitat d'ús i moltes característiques amb les que dotar al mapa. En el mateix àmbit, degut a que en una mateixa secció de mapa pot ser possible que es carreguin múltiples marcadors d'aparcament, s'ha escollit l'API també de Google anomenada MarkerClusterer[12].

S'han utilitzat els frameworks de Sencha Touch i les api's de google pel posicionament al mapa així com pel càlcul del camí més curt. També s'ha utilitzat l'api de Geonames[13] per a obtenir la localització mitjançant les coordenades. L'utilització de PayPal[14] s'ha escollit ja que permet formalitzar els pagaments de forma senzilla i segura, permeten abstreure's de la manera en que es tracten els diners i de com es tracten els temes de privadesa.

Addicionalment, per tal d'obrir l'enllaç de connexió amb PayPal al fer el pagament, s'ha utilitzat un navegador molt lleuger anomenat Childbrowser[15], per tal de consumir el mínim de recursos possibles i degut a que l'execució d'un navegador estàndard des de dins de l'aplicació provoca que aquesta es tanqui.

2.7.1. – Sencha Touch

Sencha Touch és un framework pel desenvolupament d'aplicacions mòbils que utilitza CSS3 per a poder personalitzar fàcilment la interfície de l'aplicació, aconseguint un acabat semblant al que tindria una aplicació nativa. A més a més, al estar dissenyat específicament per dispositius tàctils, disposa dels anomenats "Touch Events, que ens permet interactuar amb el mòbil aprofitant tots els seus esdeveniments propis, com ara el *tap, double tap, rotate*, entre d'altres.

Una de les funcionalitats més útils i que ha suposat escollir Sencha d'altres frameworks ha estat la possibilitat d'utilitzar un gestor de base de dades on carregar i emmagatzemar els registres necessaris per a poder treballar localment. Aquesta funcionalitat està dividida en dues parts, el *Sencha Store* i el *Sencha Model*, que formarien una petita base dades interna amb els camps que s'hagin especificat, respectivament.

S'ha decidit escollir aquest framework respecte d'altres com JQuery a causa dels següents motius:

- JQuery per a mòbils està en fase beta. Per tant no és gaire recomanable realitzar un aplicació quan el producte sobre el qual ha de treballar encara no està del tot desplegat.
- JQuery utilitza molt de codi HTML, mentre que amb Sencha gairebé tot es programa directament amb Javascript i que a més, suporta desenvolupament orientat a objectes.
- Sencha està dissenyat en estructura de Model-Vista-Controlador (MVC). Fent més eficient i podent desenvolupar amb més bones pràctiques de programació.
- Rendiment gairebé com una aplicació nativa a través de desenvolupament amb Sencha.

2.7.1.1 . Sencha Store i Sencha Model

Aquestes dues funcionalitats ens permeten emmagatzemar i manipular les dades guardades.

El *Model* està format per un conjunt de tipus de registre. Per exemple, el *Model* d'una taula d'usuaris contindria un camp per nom d'usuari, un per contrasenya i un per data de creació. És a dir, el *Model* defineix l'estructura de les dades a les que es vol accedir.

Per altra banda, el *Store* és un objecte que emmagatzema conjunts de dades i que permet a l'aplicació accedir a determinats camps. Seguint amb l'exemple anterior, el *Store* d'usuaris contindria totes les dades definides al *Model* d'usuaris.

El *Store* necessita d'una crida a servidor per a recuperar les dades. Aquestes dades cal que mantinguin el format especificat al *Model* per a que es pugui accedir-hi a posteriori.
Aquesta crida es realitza mitjançant dues eines:

- *Proxy:* indica al *Store* on i com carregar les dades. Normalment amb dades amb format JSON.
- *Reader*: Indica a l'aplicació on ha de buscar les dades.

Les principals característiques(veure Figura 1) d'aquest model de gestió de base de dades és el següent:

- Proporcionar un emmagatzematge a la banda del client.
- Permet recuperar dades tant de base de dades com de dades locals.
- El *Model* especificat serveix per determinar quines dades es vol guardar.
- Un *Store* és un conjunt d'instàncies del *Model*.
- El *Store* permet ordenar, filtrar, i realitzar consultes a les instàncies del *Model*.

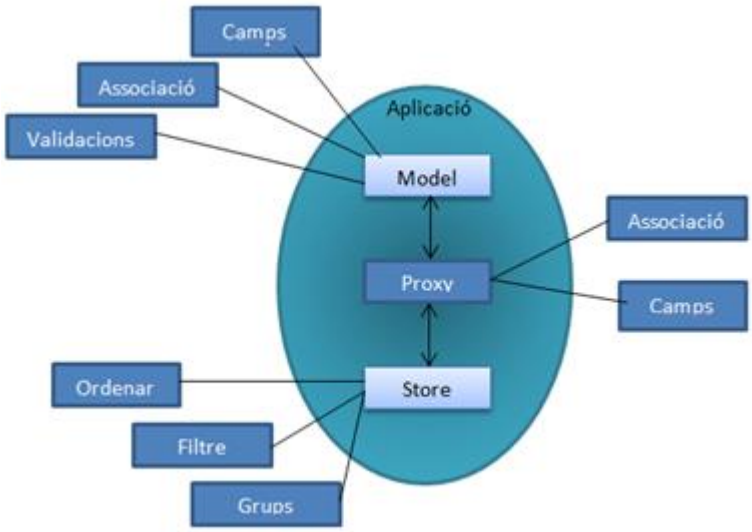


Figura 1 – Característiques Sencha Store-Model

En aquesta figura es poden veure les principals característiques i funcionalitats del gestor de base de dades de Sencha.

2.7.2. – Google Maps

L'API Javascript de Google Maps permet incloure el mapa de Google a les pàgines web o, a qualsevol aplicació que faci servir Javascript. L'última versió de l'API, està dissenyada explícitament per a ser més ràpida i més enfocada als dispositius mòbils, sense obviar que seguirà servint pels navegadors tradicionals.

Aquesta API proporciona nombroses funcionalitats per tractar els mapes, de la mateixa manera que ho permet fer des de la plana oficial de Google Maps, i permeten afegir contingut al mapa en forma de marcadors.
S'ha escollit aquesta API ja que és un servei gratuït, potent i de fàcil ús.

Es van considerar altres opcions com OpenStreetMap[16] però, Google Maps disposa de més funcionalitats integrades i és l'opció més utilitzada i amb uns fòrums de respostes i ajuda bastant ampli. A més a més, l'estil de l'OpenStreetMap sembla estar més encarat a mostratge de rutes més que per a mostrar informació. Es pot comparar les diferències entre ambdues plataformes a la següent imatge(veure figura 2).

Figura 2 – Diferències entre els dos mapes

En aquesta imatge es poden observar les diferencies a l'hora del mostratge entre OpenStreetMap(esquerra) i Google Maps(dreta).

2.7.2.1 – Càlcul de distàncies

Les imatges que retorna la API de Google Maps són de dues dimensions. Per tal de representar la Terra en una superfície plana de dues dimensions, l'API de Google Maps utilitza una projecció[17].

Dins de les projeccions en 2D, les aparences a vegades enganyen. A causa de que la projecció cartogràfica requereix d'una distorsió, els càlculs geomètrics Euclidians[18] no són molts cops aplicables. Per exemple, la distància més curta entre dos punts a una esfera no és una línia recta. Sinó que partint d'un *gran cercle*[19], la distància més curta entre els dos punts és l'arc de cercle màxim que divideix l'esfera en dos hemisferis idèntics.

Degut a aquestes diferències de càlculs geomètrics dins d'una esfera (o de la seva projecció), és necessari disposar de càlculs geomètrics esfèrics[20] per a poder obtenir distàncies, direccions i àrees.

Aquestes funcionalitats es troben dins la API `google.maps.geometry.spherical` la qual proporciona mètodes estàtics per computar valors escalars a partir de coordenades esfèriques(longitud i latitud).

S'ha escollit aquesta perquè no cal carregar cap altre llibreria o realitzar cap càlcul matemàtic per tal de trobar la distància entre dos punts a una esfera. Evitant així problemes de compatibilitat al moment de mostrar les dades al mapa.

2.7.2.2 – Càlcul de rutes

Degut a que la utilització de la llibreria de Google Maps del càlcul per tal de trobar les distàncies més properes entre dos punts dins d'una esfera, s'ha decidit aprofitar-la per tal de calcular la direcció més curta per anar d'una destinació a una altra.

Cal destacar que no és una funcionalitat rellevant dins de l'aplicació, ja que de la seva utilització no se'n treu gaire profit. Però s'ha decidit integrar-la per tal d'aportar a l'usuari una manera ràpida i fàcil d'integrar mínimament dues de les eines que s'han cregut més essencials al moment de buscar una aparcament: el GPS i el visionat dels aparcaments.

Introduint aquesta llibreria permet introduir una funcionalitat que disposa d'un gran potencial. Ja que permet desenvolupar un sistema que integri completament el GPS amb l'aplicació, permetent buscar un aparcament proper al carrer al qual es vol desplaçar, mentre el GPS serveix de guia.

2.7.3. – MarkerClusterer

En aquesta imatge (veure figura 3) es pot observar com es posicionen al mapa els diferents marcadors. Quan n'hi ha pocs no hi ha problema de visualització, ara bé, si es comencen a posar més i més marcadors o si es volgués visualitzar tots els marcadors d'aparcament de diverses ciutats provocaria que no es pogués distingir entre diferents aparcaments provocant confusió.

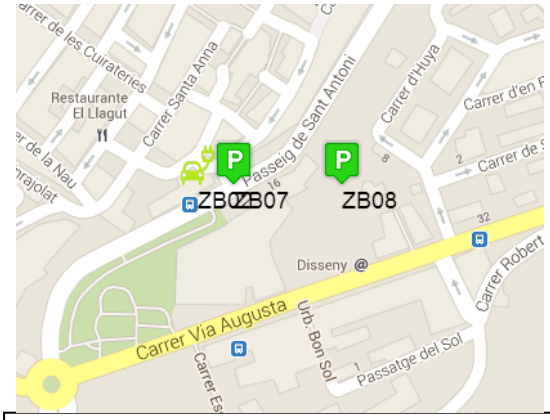


Figura 3 – Masses marcadors provoquen confusió

En aquesta figura es pot observar com en cas que hi hagi molts aparcaments a una zona determinada pot provocar confusió ja que no es pot identifica correctament la posició exacte de l'aparcament.

Per aquest motiu, prevenint el cas que s'arribessin a posar tots els aparcaments de la ciutat al mapa, s'ha decidit utilitzar una llibreria de Google que permet agrupar els marcadors que es troben dins d'una quadricula d'un tamany definit per codi, que a més es millora al agrupar-se primer per les àrees definides a la base de dades (veure Figura 4). És a dir, cada aparcament pertany a una àrea i, dins d'aquesta àrea, s'aprofita aquesta llibreria per anar agrupant els aparcaments. A més a més, això permet agrupar o mostrar els aparcaments de la zona on es troba l'usuari.

Figura 4 – Beneficis d'agrupar marcadors

En aquesta figura es pot observar com s'agrupen els aparcaments en àrees. Cada cop que s'amplia o s'allunya es tornen a calcular aquests grups per mostrar o amagar els aparcaments. Els aparcaments que estiguin sols dins d'una àrea no s'agrupen fins que no s'amplia o s'allunya el mapa. Els llocs propers a l'usuari no s'agrupen mentre no canviï la distància al mapa.

2.7.4. – Geonames

Geonames és un "*webserice*" (servei accessible a través de la web) sota llicencia de *Creative Commons*[21] que integra dades geogràfiques tals com noms de ciutats en diferents idiomes, elevació, població, localització de la ciutat a través de coordenades, entre d'altres serveis. Les dades són afegides, editades i corregides pels propis usuaris fent que sempre estigui en continu canvi i millora.

Aquesta base de dades conté més de 8 milions de noms geogràfics que corresponen a més de 6.5 milions de llocs existents. Aquests nomes estan organitzats en 9 categories i 645 sub-categories. A més a més, les dades tals com la latitud, longitud, altitud, població o el codi postal estan disponibles en diversos idiomes per a cada ubicació.

Aquestes coordenades geogràfiques es basen en el sistema WGS84[22] (Sistema Geodètic Mundial de l'any 1984).

D'entre les opcions que es van avaluar per cobrir aquesta necessitat va ser l'esmentada Geonames i el Google Geocoder.

Les avantatges d'utilitzar Geonames és que està basat en dades lliures, és de lliure col·laboració i, per tant, en cas de dades errònies és fàcilment modificable. A més a més, disposa de millor cobertura de països tot i que retorna menys dades que el Google Geocoder. Però retorna les dades essencials per a que funcioni l'aplicació.

S'ha escollit aquest webservice degut a la bona documentació que té de totes les seves funcionalitats, es gratuïta i de fàcil ús. És tant senzill com cridar a la rutina que vols executar amb els paràmetres corresponents i recuperar la resposta i codificar-la per a que l'entengui el codi de l'aplicació.

2.7.5. – PayPal

Alhora de buscar un mètode de pagament segur i que complís amb la normativa per tal que no hi haguessin problemes amb temes de confidencialitat, fraus i que estigui d'acord amb l'estàndard, es va escollir utilitzar un sistema de pagament ja desenvolupat i adaptar-lo a les necessitats de l'aplicació. A més a més, les altres opcions que es van contemplar, com ara Google Checkout, Amazon i mitjançant SMS es van descartar degut als motius següents:

	PayPal	**Google Checkout**	**Amazon**	**SMS**
Qui pot pagarme?	Qualsevol amb una targeta de crèdit	Usuaris registrats	Usuaris registrats	Qualsevol usuari amb mòbil
Mètodes de pagament acceptats	Tots els grans bancs i targetes de crèdit, xecs online i PayPal	Targetes de crèdit més usades.	Targetes de crèdit i dèbit.	Cobrament a través de la factura telefònica.
Carros de compra permesos	Més de 300	55	1	0
Atenció al client.	Atenció telefònica, online i managers dedicats.	Online	Online	Ha de ser facilitat per la pròpia empresa que rep els pagaments.

Es pot observar els diferents avantatges i inconvenients de cadascuna de les diferents tecnologies per a fer pagaments online. Es va estudiar utilitzar NFC però el poc desplegament entre els mòbils va fer que es descartés aquesta opció.

D'entre les representades a la taula es va escollir PayPal degut a la major llibertat de formes de pagament, respecte Google Checkout i Amazon. Tot i que el servei de SMS és el que més usuaris comprèn, la poca integració en aplicacions mòbil, el fet que el cobrament es realitza amb la factura telefònica i que no desprèn una imatge de seriositat, va fer que es descartés.

Figura 5 – Diferents mètodes de pagament

En aquesta figura es poden veure algunes de les diferents propostes que hi ha al mercat per a realitzar pagaments via mòbil. En aquest cas es va estudiar de fer servir Amazon, Google Checkout, PayPal i SMS.

Per aquest motiu s'ha utilitzat PayPal quan s'han de realitzar els pagaments, ja que ofereix:

- Accés instantani als diners tan bon punt ha finalitzat la transacció
- No cal que els clients que hagin de registrar-se a PayPal. Poden pagar mitjançant la seva targeta de crèdit per exemple.
- La protecció contra el frau i la seguretat tant dels compradors com de l'empresa es manté sota la seva responsabilitat. A més a més, ha de complir els requeriments de l'estàndard PCI[23].
- No hi ha cobrament extra per cada transacció.

A l'escollir PayPal es va veure que disposa de diferents API's per a diferents perfils:

- Els SDK per a dispositius iOS i Android, que proporcionen una llibreria nativa que simplifica el control d'acceptar pagaments via targes de crèdit i pagaments via PayPal mitjançant el teu dispositiu. Aquesta llibreria presenta una interfície d'usuari senzilla, una funcionalitat[24] que permet escanejar la tarja de crèdit i pagar automàticament.
- La REST API, que permet crear aplicacions de PayPal.

- L'API clàssica, que ens permet afegir un pagament i transacció mitjançant PayPal. Aquesta API cobreix tots els aspectes de les transaccions, des de notificacions, subscripcions, pagaments paral·lels, permisos i funcionalitats per tractar reemborsaments.

Un cop analitzades les tres tipologies d'API, s'ha escollit la API Clàssica, ja que és la que s'adapta més bé a les nostres necessitats.

2.7.6. – Childbrowser

Degut a que el desenvolupament amb Phonegap es fa mitjançant amb codi HTML/Javascript, l'aplicació funciona amb el navegador per defecte del dispositiu on s'executi. Al Safari en cas de un IPhone o "Internet" a Android. Per tant, no es podrà obrir una nova finestra, és a dir, si és crida window.open("http://www.google.com"); senzillament se substituirà l'aplicació per la nova plana web.
Per aquest motiu, ja que no es vol que es finalitzi l'aplicació a l'usuari per a mostrar un enllaç, s'ha decidit utilitzar una llibreria desenvolupada per *Jesse MacFadyn* i que es troba al repositori de plugins oficials per Phonegap.

Aquesta llibreria s'anomena Childbrowser i permet que l'usuari obri un nou navegador sense haver de tancar l'aplicació i així es pot redireccionar per a que puguin obrir la sessió a PayPal.
A més a més, disposa dels esdeveniments necessaris per a controlar en tot moment el funcionament del navegador, permeten que un cop es tanqui o es vagi enrere, torni a la mateixa pantalla des d'on s'ha cridat l'aplicació.

3. –Arquitectura i disseny

La aplicació està estructurada en finestres on cadascuna controla i realitza una funció concreta, mentre que la finestra principal és l'encarregada de gestionar les variables globals de l'aplicació i d'inicialitzar la resta de finestres.

Inicialment hi h un fitxer amb codi HTML que és l'encarregat de inicialitzar totes les llibreries necessàries i de cridar al CSS que dona forma a l'aplicació. Un cop s'hagi cridat, s'executa automàticament l'aplicació. La qual primer inicialitza totes les variables globals i comprovarà si l'usuari té alguna sessió oberta.
En cas que no disposi de cap sessió oberta, obre la finestra d'iniciar sessió, des d'on si no s'està donat d'alta es pot accedir a la finestra de registrar-se. L'opció de registre guarda el resum de la contrasenya conjuntament amb el resum d'un salt aleatori de 32 caràcters alfanumèrics.
A més a més, el correu electrònic de l'usuari, per tal d'aplicar la Llei Orgànica 15/1999 de Protecció de Dades de Caràcter Personal, se'n guarda el seu resum mitjançant l'algoritme MD5. Amb això s'aconsegueix emmascarar el correu electrònic fent que sigui inintel·ligible i illegible.

En cas que en tingui una sessió oberta, ja sigui perquè hagi iniciat sessió o perquè s'hagi registrat, s'obre la finestra principal de l'aplicació.
Des de la finestra principal, un cop dins de l'aplicació, es pot accedir a tots els aparcaments de la ciutat, agrupats per zones indicant quantes places lliures queden a cada zona.

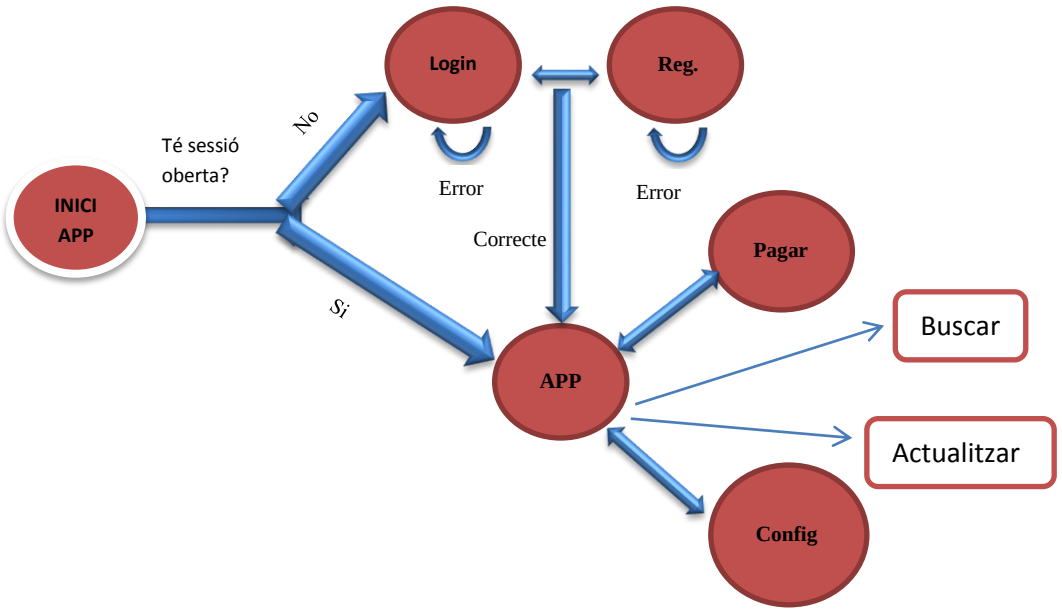
Quan es localitzi un aparcament lliure al mapa es pot procedir a fer el pagament per a disposar d'aquella plaça durant els minuts que es sol·licitin. Per fer aquest pagament s'obre una nova finestra mitjançant el plugin de Childbrowser que permet tornar a l'aplicació tant bon punt es finalitzi la transacció, ja sigui perquè s'ha cancel·lat com perquè ha finalitzat amb èxit.

Es disposa d'una funcionalitat extra per als usuaris amb permisos avançats per tal de que tant bon punt desitgin alliberar la plaça d'aparcament, els minuts restants se'ls hi sumin al seu crèdit personal. Aquest crèdit s'utilitza a la seva pròxima compra d'un aparcament, restant així del preu total els minuts que disposen de crèdit.

Per tal de que les crides a BD siguin més ràpides i senzilles, s'utilitza una de les funcionalitats que ofereix Sencha Touch, anomenat *Store*, permetent encapsular, a la banda del client, en *cache,* un model dels objectes. Aquest model s'haurà definit prèviament, especificant quins camps es vol obtenir de la base de dades. Aquesta consulta a la base de dades es realitza des d'un fitxer PHP que es crida des de l'esmentat Store. Aquesta funcionalitat permet filtrar, ordenar i fer peticions sense haver d'accedir tota l'estona a la base de dades. Així només caldrà accedir-hi en els moments en que sigui necessari actualitzar les dades.

3.1. – Disseny Lògic de l'aplicació

A la següent figura es pot observar el disseny lògic i la interacció entre les diferents finestres i funcionalitats.



Quan un usuari quan engega l'aplicació es verifica si existeix una sessió oberta amb l'identificador d'aquell usuari i si en cas afirmatiu s'obre la pantalla principal de l'aplicació.

Per altra banda, si no ha iniciat sessió, ja sigui perquè no ha entrat mai o perquè l'ha tancat, s'obre la pantalla per a poder iniciar la sessió. En el cas no s'introdueixen bé les dades no es permet l'accés. Cada cop que l'usuari s'equivoqui o introdueixi erròniament una dada, se l'avisa per tal que pugui corregir l'error. Quan hagi introduït les dades correctament es validen que existeixin i es permet l'accés en cas siguin correctes. Un cop validades, s'obre la pantalla principal.

Per a poder registrar-se cal prémer el botó corresponent a la finestra d'inici sessió. S'obre una nova finestra que permet introduir les dades obligatòries per a fer el registre(usuari, correu, contrasenya -dues vegades-, descripció i matricula d'un vehicle). El flag de pagament indica quin tipus de permisos disposarà aquest usuari.

Si les dades introduïdes són vàlides(correu electrònic amb un format vàlid i les contrasenyes són idèntiques), s'obrirà la pantalla principal.

Des de la pantalla principal es pot accedir a qualsevol funcionalitat (permesa pel nostre rol) de l'aplicació.

L'usuari pot accedir a la funció de pagar la plaça d'aparcament per a poder estacionar el seu vehicle. També pot buscar una certa direcció per a que li calculi el recorregut més curt. Configurar totes les opcions de visualització i demés de l'aplicació. I finalment, actualitzar les dades dels aparcaments mostrats al mapa.

3.2. – Elements i Actors

Els elements o finestres encarregats de gestionar les diferents característiques i funcionalitats de l'aplicació són les següents:

- **Fitxer HTML**
  És l'encarregat d'inicialitzar llibreries i cridar CSS.
  En aquest fitxer es troba, a més de les rutines encarregades d'inicialitzar tots els components necessaris, tots els esdeveniments externs a l'aplicació que es vol gestionar. Es controlen els esdeveniments que s'executaran en el moment que es perdi la connexió a internet i quan es recupera la connexió perduda.
  En el cas que l'aplicació perdi la connexió, és a dir, que no es pugui connectar a internet, s'obre una nova finestra per avisar a l'usuari que no es disposa de connexió a internet, permeten a l'usuari comprovar si hi ha connexió.
  Automàticament, tant bon punt es disposi de connexió, es reobre l'aplicació. Els possibles problemes produïts per aquesta pèrdua de connexió dependran del moment en que es perdi l'esmentada connexió. En cas de perdre's abans de fer cap consulta a la base de dades o de fer la transferència mitjançant PayPal, no s'haurà formalitzat la transferència però tampoc es disposa de la plaça. En canvi, si es perd just en el moment que s'està fent la transferència, es podria perdre la informació. L'explicació més detallada sobre aquest error es realitza al punt 5.3. – Tolerància als errors i la solució trobada es detalla al punt 6.1. –Treball Futur.

- **Fitxer APP**
  Aquest fitxer es llençat automàticament després de que es carreguin totes les llibreries i classes. S'especifiquen i s'inicialitzen totes les constants necessàries per a realitzar les operacions de càlcul d'àrees, posicionament i tipus d'usuari.
  A més a més, abans d'iniciar l'execució del programa en si, es verifica si l'usuari ha iniciat alguna sessió amb anterioritat i en cas que la tingui oberta, s'obre directament la pantalla principal. Si en canvi l'usuari no te cap sessió oberta, ja sigui perquè la va tancar o perquè no està registrat, s'obrirà una finestra que permet a l'usuari iniciar sessió o obrir una altra finestra si s'ha de registrar.

- **RegisterPanel**
  Aquesta finestra permet registrar-se a un nou usuari. Les dades que es demanen són el nom de l'usuari (o un pseudònim), el seu correu electrònic, una contrasenya

per accedir a l'aplicació, una descripció del seu primer vehicle i la matricula d'aquest. Si l'usuari escull la versió de pagament(en aquest cas activant el flag) es permetrà accés a les funcionalitats avançades.
S'ha decidit emmascarar correu electrònic per a fer visible i entenedor a l'usuari que el seu correu es guardarà xifrat i ningú podria aconseguir-ne l'accés per a extreure'n les dades personals.

Per a un correcte control d'errors es demana a l'usuari que introdueixi dues vegades la contrasenya i es verifica que aquestes siguin idèntiques. En cas contrari s'informa i es demana que ho torni a intentar.
A més a més, es verifica que el correu electrònic introduït contingui un format vàlid, extraient el domini (exemple, gmail, hotmail, entre d'altres) i comprovant si es possible establir una connexió es suposa que el correu electrònic es vàlid.
En cas que no sigui un correu vàlid, no es permet el registre i es demana que ho torni a provar.
Tampoc es permet el registre d'un usuari amb un correu electrònic que ja estigui introduït a la base de dades.

Quan el registre sigui satisfactori, s'obrirà la finestra principal, carregant totes les dades abans esmentades.

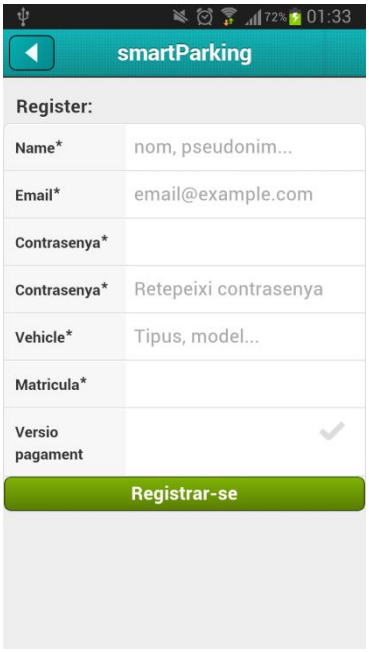


Figura 6 – Pantalla de registre

Els camps indispensables per registrar-se són els marcats amb un *. Es demana repetir la contrasenya per tal de que l'usuari verifiqui que l'ha introduït correctament.
El correu electrònic, es guarda xifrat per a protegir-lo segons l'establer per la LOPD. També es podria demanar consentiment per guardar-lo sense xifrar, però no s'ha cregut necessari degut a que només s'utilitza com a verificació per comprovar si existeix un determinat usuari.

- **LoginPanel**
  Aquesta finestra permet l'inici de sessió per a usuaris registrats. Per a que els usuaris no registrats puguin accedir a l'aplicació s'ha posat un botó a aquesta finestra per a que puguin accedir al registre.

  Per a iniciar sessió s'ha d'introduir el correu electrònic i la contrasenya. Per a verificar aquesta autentificació primer de tot es torna a calcular el resum del correu electrònic introduït i es comprova que coincideixi amb algun a la taula d'usuaris de la base de dades. Si s'ha trobat una coincidència, es recupera el resum de la contrasenya emmagatzemada i es torna a calcular el resum de la nova contrasenya utilitzant com a salt el resum de la contrasenya guardada. Si el nou resum coincideix amb el vell es que es tracta de la mateixa contrasenya i, per tant, es mira a quin grup d'usuaris pertany per a saber a quines funcionalitats té accés. Finalment es permet iniciar sessió.

- **InitialPanel**
  És la finestra principal. El primer que realitza aquesta finestra es calcular a quina ciutat es troba l'usuari. Aquest valor s'obté de realitzar una crida al webservice de Geonames passant per paràmetre la latitud i la longitud de la localització de l'usuari.
  Al disposar de la ciutat, es fa una consulta a la base de dades, mitjançant el gestor de que disposa Sencha (els anomenats *Store* i *Models*) per tal de poder guardar en memòria interna els resultats i, així, no haver d'accedir cada cop. Només cal accedir-hi quan es necessiti actualitzar les dades, per exemple al comprar una plaça d'aparcament, quan venci el temps de la plaça o quan l'usuari vulgui recarregar les dades del mapa.

  Es busca únicament les dades de la ciutat on es troba l'usuari per tal de no sobrecarregar de dades irrellevants l'aplicació, fent que funcioni més fluidament. Quan s'hagi localitzat la ciutat on es troba l'usuari i s'hagin baixat els seus aparcaments, es calcula les diferents àrees d'aparcaments i s'agrupen per proximitat, a excepció de l'àrea on es trobi l'usuari que es mostra tota.
  Aquest càlcul per agrupar els aparcaments per àrees es realitza obtenint totes les àrees amb les que s'ha dividit prèviament la ciutat i buscant quina és la més pròxima.

  Tant bon punt es disposi dels aparcaments de la ciutat on es troba l'usuari es visualitza al mapa les localitzacions d'aquests, indicant si estan o no lliures. Per tal de no omplir el mapa amb tots els marcadors de tots els aparcaments de la ciutat, es calcula de quina àrea són aquells aparcaments i en cas que sigui de la mateixa

àrea on es troba l'usuari s'agrupen per separat dels de la resta de zones. A més, els grups indiquen quantes places lliures queden dins d'aquella àrea (veure figura 7).

Figura 7 - Diferenciació ocupació zones

En aquesta figura es pot observar com s'agrupen els aparcaments en àrees. Cada cop que s'amplia o s'allunya es tornen a calcular aquests grups per mostrar o amagar els aparcaments.
Un grup amb el símbol **4+** indica que més del 60% dels aparcaments estan lliures.
El grup amb el símbol **2+** indica que hi ha entre el 25% i el 60% de places lliures.
El grup amb el símbol **2-** indica que hi ha menys del 25% de places lliures.

El mapa és el component predominant a aquesta pantalla, ocupant-la tota a excepció de dues barres de navegació. Una a la part inferior que dóna accés a diverses funcionalitats, entre elles la de realitzar l'aparcament. També hi ha una barra a la part superior que informa del nom de l'aplicació, un botó que dóna accés a la configuració i un altre que permet tancar l'aplicació finalitzant la sessió.

És a dir, des de la pantalla principal es pot accedir a totes les opcions que permet l'aplicació. Es pot configurar quins aparcaments veure, modificar les dades d'un vehicle, entre d'altres. També es pot buscar el camí més curt a una destinació.
Aquesta funcionalitat demana que s'introdueixi una destinació i calcula la distància més curta, pel que fa a quilòmetres, des del punt on es troba l'usuari. Cal especificar carrer i ciutat per a que calculi correctament la destinació.
Finalment, amb el botó de re-posicionar, es torna a calcular la posició relativa de l'usuari i a quina ciutat es troba i carregant els aparcaments.

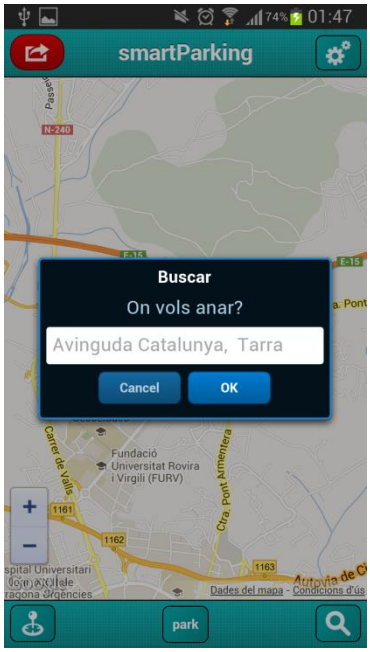


Figura 8 – On vols anar?

En aquesta figura es pot observar el missatge per introduir la localització cap on es vol calcular la ruta. Cal introduir una direcció precisa.

Figura 9 – Recorregut fins destinació

En aquesta figura es pot observar el missatge per introduir la localització cap on es vol calcular la ruta. Cal introduir una direcció precisa.

- **ParkingPanel**

  Des d'aquesta finestra s'accedeix a realitzar el pagament per una plaça d'aparcament. Cal introduir la plaça d'aparcament, seleccionar un vehicle d'entre llista de disponibles per l'usuari en qüestió i quants minuts es desitja aparcar.

  Mentre no s'introdueixin totes les dades no es permet realitzar la compra. Quan s'hagi introduït totes les dades i es premi pagar, es mostra el navegador Childbrowser amb l'adreça de PayPal on fer el pagament. Si no es disposa de compte a PayPal, es pot realitzar la transferència mitjançant el número de targeta de crèdit. En cas que es disposi de compte a PayPal només cal iniciar sessió i validar el pagament. Tal i com mostra la imatge (veure figura 10), en aquest pas, just abans d'iniciar sessió per a validar la compra, es veu la informació relativa a la plaça comprada(preu, minuts i identificador plaça).

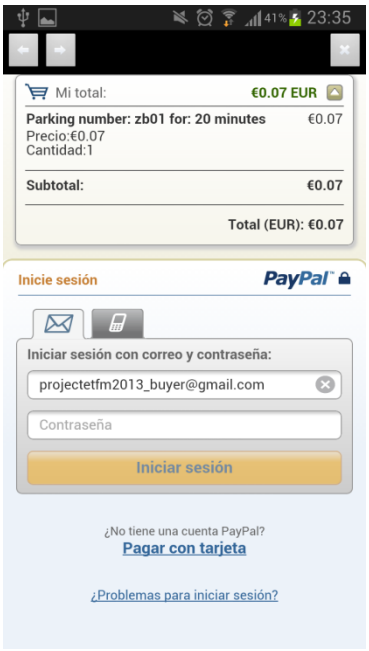


Figura 10 – Finestra per a realitzar el pagament

Aquesta finestra permet a l'usuari iniciar sessió a PayPal en cas que en tingui, ja sigui mitjançant el seu correu electrònic i contrasenya o, de forma més segura, a través del número del seu telèfon mòbil i un codi PIN. També hi ha l'opció per a que usuaris no registrats puguin formalitzar el pagament utilitzant el seu número de targeta de crèdit.

Es pot comprovar com tot i pagar per una duració de20 minuts, el preu equival al que costarien 4. Això es deu a que l'usuari disposa de 16 minuts de crèdit. El preu al minut és de: 0.0183 arrodonit a dues xifres decimals. S'arrodoneix a dues xifres decimals ja que PayPal no n'admet més de dues.

Al pagar sempre es demanarà usuari i contrasenya abans de poder realitzar el pagament.

Disposant de compte a PayPal es permet l'accés mitjançant el número de mòbil i un codi PIN. Aquest número de telèfon cal que s'hagi donat d'alta al compte de PayPal. El PIN és un codi de 48 dígits que ajuda a mantenir la seguretat del compte de l'usuari i permet confirmar els pagaments més ràpidament i de forma més còmode.
Tant bon punt s'hagi formalitzat el pagament, s'informa a l'usuari i quan aquest tanqui el navegador s'obre la finestra principal amb les dades de les places d'aparcament actualitzades.

Com a funcionalitat extra dins de la pantalla de l'aparcament es troba l'opció de canviar d'usuari abans de realitzar el pagament i de modificar la matricula del vehicle seleccionat. Amb la funcionalitat del canvi d'usuari es permet donar llibertat a l'usuari per a que en cas que aquell dia s'estigui conduint el vehicle d'un altre familiar, podrà iniciar sessió amb aquell usuari. Cal tenir en compte que la

matricula amb la que es formalitza el pagament ha de coincidir amb la del vehicle utilitzat per a que no es puguin reportar incidències.
La funcionalitat de modificar la matricula permet arreglar qualsevol error que hagi pogut tenir l'usuari al fer el registre, com per exemple, canviar la descripció del vehicle si ha patit qualsevol canvi des de que es té l'aplicació i, a més, fer-ho ràpidament al moment de comprar la plaça.

- **SettingsPanel**
  Des d'aquesta finestra es pot accedir a la configuració de l'aplicació.
  Les opcions de que es disposa estan classificades dins de diferents pestanyes per tal de tenir-ho tot ordenat i que sigui més còmode i senzill per a l'usuari.
  A la primera pestanya es permet donar d'alta un nou vehicle per afegir al llistat de l'usuari.
  La segona pestanya serveix per a modificar el filtre de visualització de places d'aparcament. És a dir, seleccionar quines determinades tipologies d'aparcament es vol veure: zones blaves per a vehicles elèctrics o minusvàlids, zones blaves per la resta de vehicles.
  La tercera pestanya mostra l'històric d'aparcaments de l'usuari i la posició al mapa dels que hi ha actius, per a que pugui trobar el seu cotxe amb facilitat.
  La quarta pestanya permet denunciar una infracció directament a l'òrgan encarregat de la seguretat i el correcte funcionament dels aparcaments. Per indicar una infracció cal introduir la matricula de l'infractor, l'identificador de l'aparcament i escollir d'entre un llistat d'incidències (mal aparcat, aparcament no lliure). Degut a que gran part de l'èxit de l'aplicació recau en el civisme de la ciutadania, s'ha decidit que cal penalitzar als usuaris que en fan un mal ús.
  La cinquena pestanya permet visualitzar els aparcaments actius i alliberar-los quan es vulgui marxar de l'aparcament. En cas que es tracti d'un usuari amb permisos avançats, els minuts restants es sumaran com a crèdit per a la pròxima compra.

- **ChangePlateView**
  Aquesta pantalla és accessible des de la finestra de pagament. Des d'aquí es permet modificar la descripció i/o la matricula del vehicle seleccionat (veure figura 11).
  S'ha decidit fer aquesta funcionalitat a la pantalla de l'aparcament en comptes de a la configuració degut a que les dades del vehicle normalment no variaran però en cas de necessitat, és més ràpid comprovar un error en la matricula i modificar-lo des de la pròpia finestra de pagament, ja que és allà on s'escull entre els diferents vehicles dels que es disposen.

Figura 11 – Modificació informació sobre un vehicle

Aquest formulari permet modificar la informació d'un vehicle. Les dades introduïdes substituiran a les que es troben guardades a la taula Vehicles de la base de dades al registre pertanyent a la matricula escollida.
El camp de la matricula és un desplegable amb tots els vehicles que l'usuari ha registrat a l'aplicació.

- **ChangeLoginView**
  Abans de realitzar la compra de la plaça d'aparcament, tenim l'opció de canviar d'usuari. Amb aquesta funcionalitat es vol donar a l'usuari l'opció de no haver de registrar tots els vehicles als que té accés si no vol. Evitant així omplir molts vehicles que no usa periòdicament. Així, una família amb diferents vehicles només cal que registrin el vehicle que més utilitzen. Amb això es pretén que el dia en que s'agafi el vehicle d'un altre membre de la família, no calgui que registrin un nou vehicle, sinó que senzillament hauran d'iniciar sessió amb l'altre usuari i realitzar el pagament. Així s'aconsegueix fer més ràpid els tràmits ja que no cal que vagin a la finestra de configuració per afegir un nou vehicle ni cal que tanqui l'aplicació.
  Després de canviar d'usuari es torna a obrir la finestra del pagament per a poder seguir amb la compra còmodament.

3.3. – Disseny dels mecanismes de seguretat

A l'hora de desenvolupar un sistema d' d'autentificació i autorització s'ha analitzat que PHP no disposa de cap estàndard clar que permeti implementar, d'una manera transparent i única, els elements de seguretat adequats, sinó que senzillament disposa de simples mecanismes per a utilitzar autenticació HTTP. Tanmateix, existeixen petits frameworks que proporcionen els mecanismes de seguretat necessaris per a realitzar els mecanismes d'entrada i de gestor de rols .

Tot i aquestes petites aplicacions, s'ha decidit desenvolupar un mecanisme d'autentificació i autorització propi per tal de poder controlar exactament quins són els mecanismes de seguretat, poder-lo modificar fàcilment per agregar noves funcionalitats a mesura que són necessàries i per tant, que no estigui sobrecarregat amb extensions innecessàries, agilitzant així tot el procés. Però més que res, per tal d'aprendre els mecanismes de seguretat per a autenticar i autoritzar usuaris, tot i que aquests frameworks disponibles per PHP possiblement utilitzin tècniques més provades i validades.

3.3.1. – Emmagatzematge segur de les dades

Les formes més comuns amb les quals es pot emmagatzemar una contrasenya són les següents:

- Sense codificar, amb text pla.
- Codificada, utilitzant per exemple la rutina en base64.
- Funcions resum o hash simples sense salt o amb salts preestablerts.
- Funcions resum o hash amb salts aleatoris i codificats.

Evidentment, cal descartar la primera opció per francament insegura. La segona opció, tot i guardar la contrasenya codificada, la utilització de funcions estàndard és ràpidament identificable i pot resoldre's de manera trivial.
La tercera via, tot i ser més segura que les dues primeres ja que qualsevol funció de resum és irreversible, és vulnerable a atacs de força bruta, generador de col·lisions, atac per diccionari i atac amb taules pre-calculades.

3.3.1.1. – Que és el Hash?

La funció Hash o resum és una funció computable mitjançant un algoritme $H: U \rightarrow M$

$$x \quad \rightarrow h(x) .$$

Aquest algoritme rep com a paràmetre d'entrada un conjunt de caràcters, essencialment cadenes de text, que converteix a una altra cadena de text de longitud fixa. Un exemple de funció hash o resum seria la següent imatge(veure Figura 12), on com a paràmetre d'entrada hi hauria la cadena de text "*Hola Món"* i de la qual s'obté el seu resum.

Figura 12 – Funcionament d'una funció Hash o Resum
Les funcions de resum donat una cadena de text, retornen una cadena de text de format il·legible i inintel·ligible.

Les funcions de resum són usades en múltiples camps. S'utilitzen en els següents casos:

- Construcció d'estructures de dades. El seu ús en diferents estructures de dades fan més òptimes les cerques. Per exemple les taules de hash.
- Per a protegir la integritat. Com a firma digital o suma de verificació.
- Eina per a l'autentificació i el control d'accés. Per exemple per a protegir les contrasenyes
- Eina per a la identificació i la ràpida comparació de dades. Per exemple per detectar la signatura dels virus.

El salt[25] és una cadena de bits aleatoris que s'utilitzen concatenats amb una de les entrades de la funció de resum. La finalitat del salt radica en devaluar la utilitat dels mètodes d'atac criptogràfics basats en salts precalculats i fer més difícil l'obtenció d'una contrasenya per part d'un atacant. A l'haver generat una cadena de text xifrada amb la unió de la contrasenya i el salt de N bits, l'atacant haurà de calcular el resum de cada paraula juntament amb totes les possibles combinacions de salts fins a obtenir una coincidència.
Suposant que només s'utilitzessin contrasenyes basades en paraules del diccionari i sabent que només en fa servir una de 50.000, un atacant hauria de fer $2^n \times 50000$ combinacions, on N es el nombre de bits del salt en qüestió.

3.3.1.2. – Atacs criptogràfics

Un atac de força bruta consisteix a calcular el codi de resum de totes les possibles combinacions dels valors d'entrada i comparar el resultat amb el resum de la contrasenya atacada, fins que es troba una semblança.
Tot i ser un atac limitat per la quantitat de recursos i d'intents de que es disposi, la poca mentalització dels usuaris envers la seguretat de les seves contrasenyes(entre les vint-i-cinc contrasenyes més usades[26] a internet hi ha 'password', '1234567' o 'abcd123') fa que sigui un atac factible amb pocs intents.

Una col·lisió[27] es produeix quan, almenys, dos o més valors diferents produeixen el mateix resum. Aquesta condició es pot trobar normalment en algoritmes poc segurs, com pot ser el MD5. Tanmateix, la viabilitat del generador de col·lisions recau no només en trobar-ne una, sinó que el valor sigui semànticament equivalent.

Els atacs per diccionari es comporten igual que els atacs per força bruta exceptuant que utilitzen una llista de paraules preestablertes en comptes de buscar combinacions. Degut a que molts usuaris utilitzen contrasenyes basades en paraules, com per exemple, '*papallona*' o '*password*' l'eficàcia d'aquest mètode incrementa. A més a més, existeixen versions que permeten trobar variacions de la paraula, substituint, per exemple, majúscules per minúscules i lletres per números. Per exemple: '*pApA110nA*' o '*Pa$$w0rD*'

Les taules precalculades o taules "*Rainbow*" contenen codis de resum ja calculats per a una sèrie de combinacions, permeten comparar directament si el resum de la contrasenya està dins de les combinacions de la taula. Amb aquest mètode no necessita tants de recursos ni de temps com per exemple l'atac per força bruta.

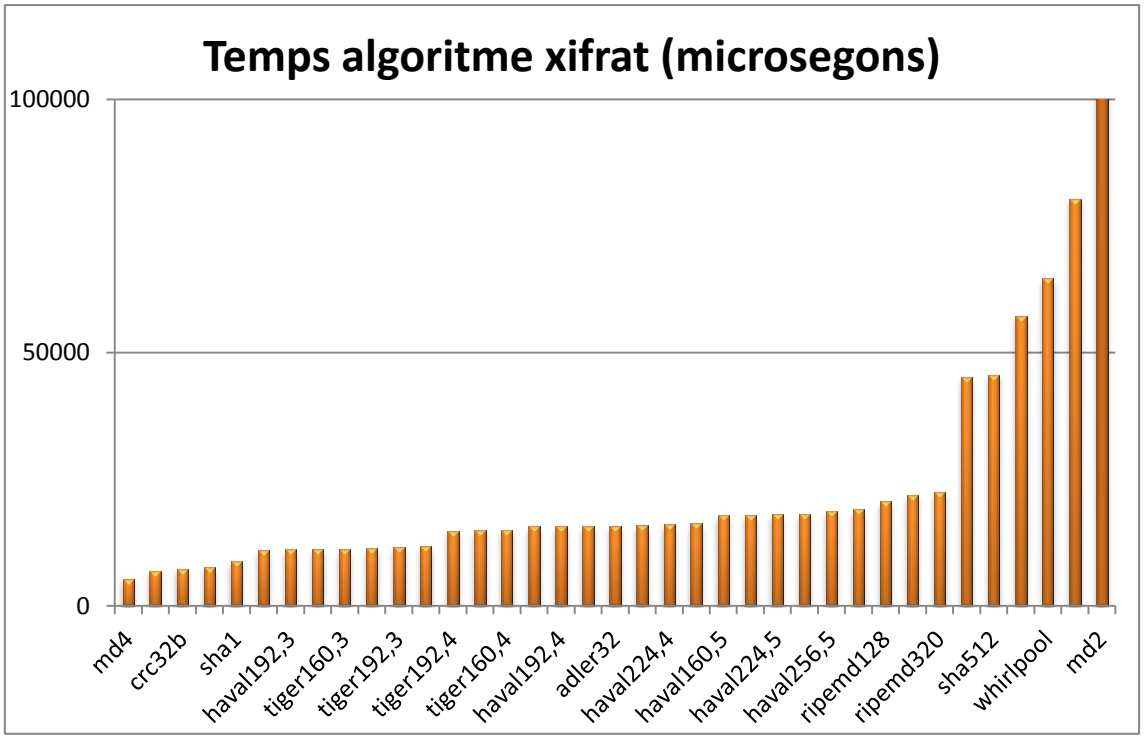
3.3.1.3. – Metodologia seguretat desenvolupada

Quan un nou usuari es vol registrar, per millorar la seguretat de l'aplicació es comprova que el correu electrònic tingui un format vàlid, evitant així que pugui ser objecte d'atacs i així es pot assegurar que no s'omple de registres brossa el servidor.
En la mateixa línia, quan l'usuari es registri, es demana que introdueixi dues vegades la contrasenya, per tal d'evitar confusions. La contrasenya no es guarda en text pla, sinó que se'n guarda el seu resum mitjançant l'algoritme SHA256 juntament amb un salt aleatori de caràcters ASCII. Aquest salt aleatori primer es converteix en base 64 per tal de que sigui interpretable per a una rutina de generació de resums, obtenint un salt de 32 caràcters alfanumèrics semblant a la següent cadena de caràcters:
831ce23ce8f534de507ced88a5db1640.

S'ha decidit a utilitzar un salt aleatori codificat mitjançant un algoritme de resum per tal d'evitar problemes de col·lisions i per a que un usuari maliciós no pugui fer ús de taules *Rainbow*.

Per tal d'escollir un mecanisme de xifrat s'ha avaluat tant el temps necessari per xifrar com la seguretat per tal de trobar quina era la opció més viable a nivell tècnic sense comprometre la seguretat.

Es pot veure a la següent gràfica el temps aproximat de cada algoritme per a calcular el resum d'una cadena de text de 1024000 bytes de dades aleatòries.



Taula de referència

Xifrat	Microsegons
md4	5307,912
md5	6890,058
crc32b	7298,946
crc32	7561,922
sha1	8886,098
tiger128,3	11054,992
haval192,3	11132,955
haval224,3	11160,135
tiger160,3	11162,996
haval160,3	11242,151
haval256,3	11327,981
tiger192,3	11630,058
haval128,3	11880,874
tiger192,4	14776,945
tiger128,4	14871,12
tiger160,4	14946,937
haval160,4	15661,954
haval192,4	15717,029
haval256,4	15759,944
adler32	15796,184
haval128,4	15887,022
haval224,4	16047,954
ripemd256	16245,126
haval160,5	17818,927
haval128,5	17887,115
haval224,5	18085,002
haval192,5	18135,07
haval256,5	18678,903
sha256	19020,08
ripemd128	20671,844
ripemd160	21853,923
ripemd320	22425,889
sha384	45102,119
sha512	45655,965
gost	57237,148
whirlpool	64682,96
snefru	80352,783
md2	705397,844

Veient aquesta gràfica es pot extreure que tant important és escollir un algoritme de resum segur com que no repercuteixi en el rendiment del sistema o aplicació. Cal analitzar què ex vol xifrar i quina criticitat té per tal d'establir un bon mecanisme de codificat.
Analitzant aquesta gràfica es poden descartar tant el MD2 i el MD4 degut a que ja no es consideren segurs avui dia.

En aquest cas s'ha decidit pel mètode *Crypt* de PHP, el qual es un algoritme d'un sol sentit, per tant no hi ha funció de desencriptar. Per comprovar que la contrasenya introduïda es correcta cal tornar a xifrar la contrasenya i comparar-la amb la que hi ha a la base de dades. Aquesta funció utilitza dos paràmetres d'entrada, la contrasenya subministrada i alguns caràcters addicionals anomenats 'salt', amb els quals s'afegeix una major complexitat a la funció. El resultat es presenta com una cadena codificada en base64.

Aquest mètode retorna el resum d'una cadena de text utilitzant l'algoritme basat en el *DES*[28] estàndard d'Unix o d'algoritmes alternatius que pugin estar disponibles al sistema, tals com:

- **CRYPT_STD_DES:** Resum estàndard basat en el DES amb un "salt" de dos caràcters de l'alfabet "./0-9A-Za-z". La utilització de caràcters invàlids provocarà que falli.

- **CRYPT_EXT_DES**: Resum extens basat en DES. El "salt" és una cadena de text de 9 caràcters que consisteix en un guió seguit de 4 Bytes del comptatge d'iteracions i 4 Bytes del salt

- **CRYPT_MD5:** Resum MD5 amb un "salt" de dotze caràcters començant amb **1.**

- **CRYPT_BLOWFISH:** Resum amb Blowfish amb un salt semblant al següent: **"$2a$", "$2x$"** o **"$2y$",** un paràmetre de cost de dos dígits, "$", i 22 caràcters de l'alfabet "./0-9A-Za-z". Utilitzar caràcters fora d'aquest rang en el salt causarà que crypt() torni una cadena de longitud zero. El paràmetre de cost de dos dígits és el logaritme en base 2 del comptador de la iteració de l'algoritme hash basat en Blowfish subjacent, y ha d'estar en el rang 04-31.

- **CRYPT_SHA256:** Hash SHA-256 amb un "salt" de setze caràcters prefixat amb **5**. Si la cadena de text del "salt" s'inicia amb '**rounds=<N>$**', el valor numèric de N s'utilitza per indicar quantes vegades el bucle del hash s'ha d'executar, de manera similar al paràmetre de cost a Blowfish. El nombre de rondes por defecte són 5000, el mínim és de 1000 i el màxim de 999,999,999.

- **CRYPT_SHA512:** Hash SHA-512 amb un "salt" de setze caràcters prefixat amb **5**. Si la cadena de text del "salt" s'inicia amb '**rounds=<N>$**', el valor numèric de N

s'utilitza per indicar quantes vegades el bucle del hash s'ha d'executar, de manera similar al paràmetre de cost a Blowfish. El nombre de rondes por defecte són 5000, el mínim és de 1000 i el màxim de 999,999,999.

Per xifrar la contrasenya s'ha decidit utilitzar un codificador en base 64 que inicialitza una llavor per a crear un salt aleatori cada vegada. D'aquest salt se'n treu el seu resum mitjançant l'algoritme MD5 i juntament amb la contrasenya i uns caràcters delimitadors ens permeten emmagatzemar el seu resum utilitzant la funció de hash SHA-256.

Utilitzant aquest algoritme, si s'introduís una contrasenya del tipus: UOC2013_TFM ens retornaria quelcom semblant al següent:
$6$831ce23ce8f534de$M0bZGghrSys0e5ScgDVnljR9.W64MsSn68Hw7ljy1g8pbjRqVVfwITLXQ2Z69aTj/87ThOXMap0tLZggrHp7M1.

3.3.2. – Injeccions SQL

Per tal d'evitar que un usuari malintencionat pugi injectar codi SQL maliciós al formulari d'entrada per a iniciar sessió o registrar-se, s'ha afegit, dins del codi PHP que fa la crida a la base de dades, les sentències de seguretat per evitar aquests atacs.

Utilitzant la sentència "*mysql_real_escape_string*" permet obviar els caràcters especials dins d'un string per tal de poder fer-ho servir de forma segura a la sentència SQL.

Per exemple, si s'introduis dins d'una consulta a base de dades per provar de fer l'inici de sessió mitjançant usuari i contrasenya, la següent instrucció:

$pwd = 'OR '1'='1'. Al fer el la petició a la base de dades ens quedaria així:

```
"SELECT * FROM Users WHERE email='".$email."' AND password='".$pwd."'";
```

I per tant,a l executar la sentència quedaria el següent:
"SELECT * FROM Users WHERE email='odols@uoc.edu' AND password=''OR '1'='1'";
Produint que la base de dades interpretés que si el password és '' o bé si és 1 igual a 1, el qual sempre serà cert, i ens permetrà l'entrada a l'aplicació.

Afegint la sentencia mysql_real_escape_string s'obvien els caràcters especials, com per exemple les cometes simples fent que la consulta quedi de la següent manera:
"SELECT * FROM Users WHERE email='odols@uoc.edu' AND password='OR 1=1'";

3.3.3. – Autenticació i autorització

A diferència de Java o ASP.NET, a PHP no hi ha un estàndard tan clar d'autenticació i autorització i per tant s'ha decidit implementar un petit gestor d'autentificació basat en rols que donen accés a diverses funcionalitats.

S'ha creat dins de la base de dades una taula que contindrà tots els possibles rols que pot tenir un usuari. Aquest rol vindrà determinat pel tipus de registre que s'esculli. Es parteix que per defecte existeixen dos tipus de rols, el d'usuari bàsic i el d'usuari avançat.
Quan un usuari registrat vulgui iniciar sessió es verificarà el grup al qual pertany per tal de que pugui accedir únicament a les seves funcionalitats.

Per a validar l'usuari a l'iniciar una sessió es busca quin és el resum del correu electrònic introduït i es busca si hi ha cap coincidència a la taula d'usuaris de la base de dades i en cas afirmatiu, se'n recupera el resum de la contrasenya que s'ha emmagatzemat durant el registre.

En cas que existeixi, es comprova la contrasenya. Per a fer aquesta verificació s'ha de seguir diferents passos perquè no es pot descodificar una funció resum ni tampoc es pot tornar a calcular ja que al utilitzar un salt aleatori, les dues contrasenyes no coincidirien. Per tant, es recupera la contrasenya emmagatzemada a la taula d'usuaris de la base de dades. Disposant d'aquest salt es torna a calcular el seu resum amb l'algoritme SHA256 i si els seus resums coincideixen es valida l'accés.

3.4. – Disseny estructural de la base de dades

Quan l'usuari inicia sessió, abans de mostrar els aparcaments calcula mitjançant el webservice de GeoNames a quina ciutat es troba l'usuari. Al obtenir aquesta ciutat es comprova a la BD tots els aparcaments relacionats amb aquella ciutat i es guarden al gestor de base de dades del Sencha Touch, poden així accedir-hi posteriorment de forma més ràpida.
Els aparcaments que no es trobin a prop de l'usuari s'agruparan en zones definides a la base de dades, ja que cada aparcament pertany a una zona.

Dins de la base de dades no s'emmagatzemen dades rellevants que vulnerin la Llei de Protecció de Dades. Com que el correu electrònic està considerat una dada de caràcter personal per la Llei Orgànica 15/199 de Protecció de Dades de Caràcter Personal[29] s'ha decidit xifrar amb l'algoritme MD5 aquesta dada per a que no vulneri la llei.

La base de dades la formen les següents taules:

- **Ciutats**
  En aquesta taula s'emmagatzemen les ciutats que volen tenir una SmartCity.

- **Companies**
  Totes les companyies patrocinadores dels aparcaments.

- **CompaniesCities**
  Per tal de lligar les companyies i les ciutats s'ha decidit crear una taula que identificarà a quina ciutat està cada companyia. Cada companyia pot estar a N ciutats i una ciutat pot tenir N companyies.

- **Area**
  Dins de cada àrea d'una ciutat hi hauran definits els aparcaments d'aquella ciutat, formant així una quadrícula que ens servirà per delimitar a quina àrea es troba l'usuari i mostrar els pàrquings que té al voltant mentre que s'agrupen els de la resta d'àrees.

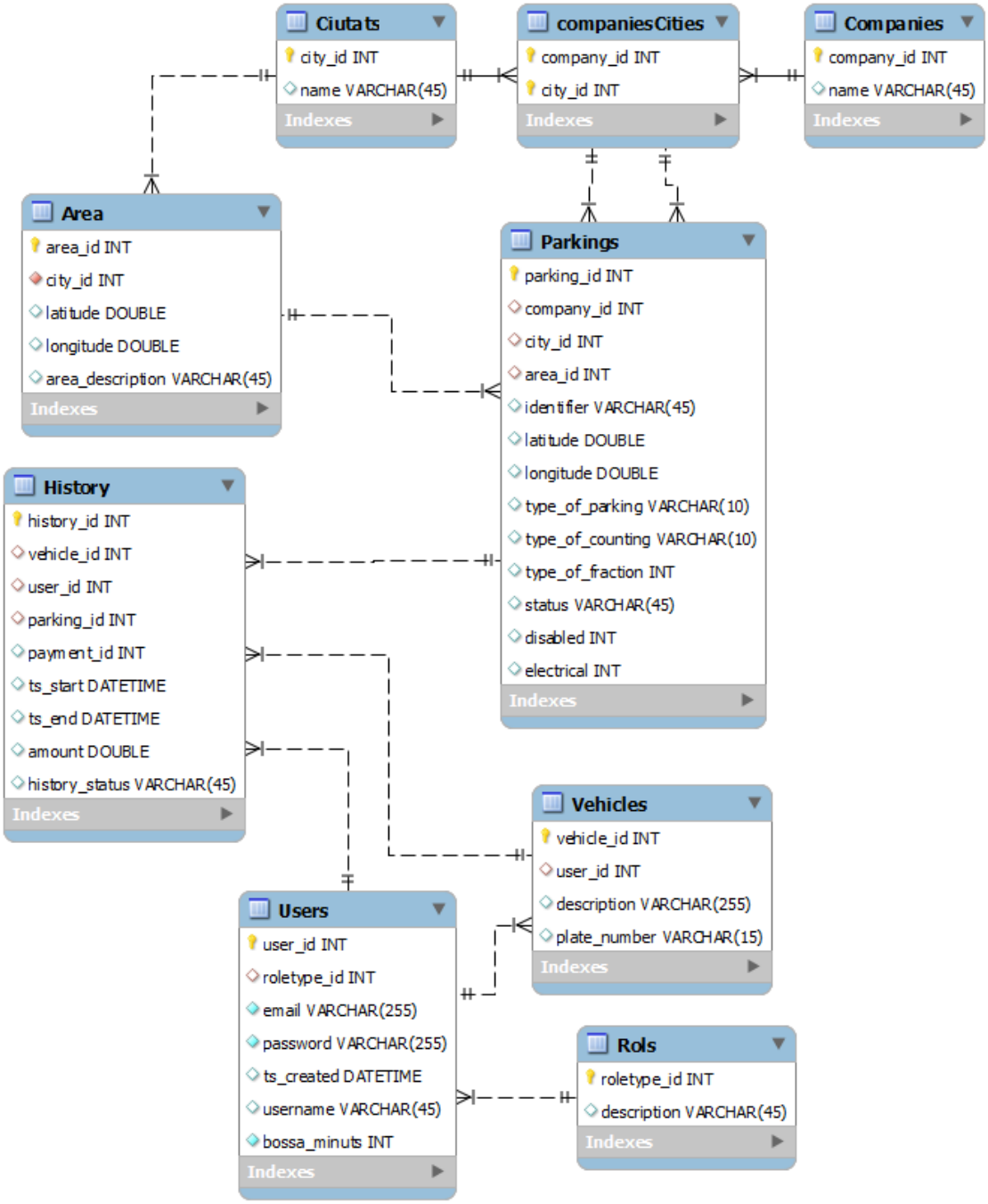
- **Parkings**
  Un aparcament formarà part d'una àrea dins d'una ciutat i només podrà ser d'una companyia en concret. Es guarda la latitud i la longitud de la plaça de l'aparcament per tal de poder-lo situar al mapa.
  La tipologia de l'aparcament indica si es tracta de zona blava, zona verda, aparcament general, etc.
  Es va definir que els aparcaments podrien diferenciar dos tipus de comptador. Un semblant al que utilitzen els aparcaments de les zones blaves, que es paga pels minuts que es vol estar i el mecanisme que utilitzen els aparcaments privats, pagar el preu del temps que s'hi ha estat quan es marxa.
  El tipus de comptador servirà per diferenciar el temps de cobrament de la plaça d'aparcament, així un de tipus fraccionat comptarà el preu per minut, havent d'introduir quants minuts es vol aparcar.
  En canvi un aparcament que vol utilitzar un comptador creixent de temps, per cada minut que passi anirà cobrant fins que s'alliberi la plaça. Tant bon punt s'ha alliberat la plaça es formalitzarà el pagament.
  Tanmateix, aquesta segona funcionalitat, tot i que dissenyada no es va poder implementar ja que es va analitzar els problemes que suposava que la gent pagués al finalitzar l'estança.
  L'estat ens indicarà si l'aparcament esta deshabilitat, lliure o ocupat.
  Els flags de *disabled* i *electrical* indiquen si es tracta d'una plaça per a minusvàlids i/o vehicles elèctrics.
  Un aparcament pertany a una àrea, la qual pot tenir N aparcaments. Cada aparcament pertany a 1 companyia de 1 ciutat.

- **History**
  Dins d'aquesta taula es guarda l'històric de compres de cada usuari. També serveix per saber quan una plaça està a punt de vèncer.

- **Users**
  Taula principal juntament amb la de vehicles. Es guarda la informació bàsica dels usuaris registrats, el rol que han escollit i el nombre de minuts disponibles que els hi resten de crèdit.

- **Vehicles**
  Cada usuari pot tenir N vehicles lligats. Aquí es guarda la matricula i una descripció del vehicle, per identificar quin dels vehicles de la família escollim.

- **Rols**
  En aquesta taula s'emmagatzemen els dos tipus de grups que formen l'aplicació. Els que tenen permisos avançats i els que són usuaris bàsics.
  Un usuari pot tenir N rols.



4. – Implementació

El desenvolupament de les classes s'ha fet ordenant-los per blocs i per cada bloc els seus mètodes, fent que l'estructuració de codi sigui més senzilla d'entendre. Els blocs especificats a continuació descriuen els mètodes més rellevants que s'han desenvolupat per a fer l'aplicació.

4.1 – Registre

Dins d'aquest bloc s'ha desenvolupat el mètode d'emmagatzematge de la contrasenya en forma segura. Per tal de desenvolupar aquest bloc s'ha creat un mètode *generatePasswordHash* que donat una contrasenya retorna el seu resum.

La lògica d'aquest mètode és la següent:

- Es calcula un salt aleatori de 32 caràcters alfanumèrics del qual se'n busca el seu resum amb l'algoritme MD5.
- Un cop calculat aquest salt, es busca el resum de la contrasenya introduïda per paràmetre juntament amb el salt calculat. S'utilitza l'algoritme de resum SHA256.
- Es retorna el resum de la contrasenya per a que pugui ser guardat a la base de dades.

4.2 – Inici Sessió

Aquest bloc s'ha desenvolupat per a que donat un usuari i una contrasenya es verifiqui si existeix a la base de dades i n'aconsegueixi el seus permisos. Per a realitzar aquest desenvolupament s'ha dissenyat un mètode anomenat *checkLoginData* que rep per paràmetre una contrasenya i un usuari.

La lògica d'aquest mètode és la següent:

- Primer es busca mitjançant l'usuari passat per paràmetre si existeix el seu resum amb MD5 a la base de dades.
- Si no existeix s'informa de l'error.
- Si existeix se n'extreu el resum emmagatzemat i es torna a calcular el resum de la nova contrasenya utilitzant la recuperada de base de dades com el seu salt.
- Si el nou resum de la contrasenya és el mateix que el de la contrasenya introduïda, és que l'usuari és vàlid i se n'aconsegueixen els seus permisos.
- Si no s'informa de l'error.

4.3 – Crontab

Per a realitzar l'actualització de les places a punt de vèncer s'ha desenvolupat una rutina que s'executa cada cinc minuts i que verifica a la taula d'Històrics hi ha places a punt de vèncer i de quin usuari. Si es troba que un usuari té una plaça comprada que està a punt d'acabar el seu període de validesa, se l'informa mitjançant una finestra emergent. Aquesta alerta només sortirà quan restin menys de cinc minuts per a que finalitzi la plaça.

Quan una plaça d'aparcament ha vençut, s'actualitza les dades corresponents a l'històric de l'usuari i es torna a situar la plaça d com a lliure.
En aquest cas, quan a un usuari se li venç una plaça d'aparcament i no ha retirat el vehicle, es considera una incidència de la qual els usuaris podran informar a l'òrgan encarregat de velar del bon ús dels aparcaments.

4. 4 – Càlcul àrees

Per a realitzar el càlcul d'àrees s'ha creat, primer de tot, una taula a la base de dades on hi ha la relació de a quina àrea d'una ciutat hi va un aparcament.

Per tant, primer de tot es calcula a quina ciutat es troba l'usuari, mitjançant una crida al webservice Geonames, que retorna una cadena de text que es tracta per tal d'extreure'n la ciutat. Amb la ciutat es busca a la base de dades totes les àrees d'aquesta. Un cop es disposa de totes les àrees es busca la més propera a la localització de l'usuari utilitzant els mètodes de càlculs geomètrics esfèrics de Google.

Al obtenir l'àrea més propera, es suposa que està dins d'aquella àrea i es comença a buscar tots els aparcaments de la ciutat.

Tots els aparcaments que estiguin dins de l'àrea on es troba l'usuari s'agrupen junts i la resta dels aparcaments es van agrupant per les seves respectives zones. En cas que en una zona només hi hagi un aparcament, no s'agruparà fins que no s'allunyi el mapa. Com més apropi el mapa l'usuari, més aparcaments aniran mostrant-se.

Els indicadors de les àrees aporten informació respecte al nombre de places lliures que queden dins d'aquella plaça.

4.5 – Càlcul ocupació

El càlcul de l'ocupació dels aparcaments es realitza en el moment en que s'agrupen.
Primer de tot es calcula quants aparcaments hi ha per agrupar i quants estan lliures.

Després es divideix el nombre d'aparcaments lliures del total i es comprova quin percentatge retorna. Amb aquest valor es pinta l'indicador del grup d'un color o d'un altre, per tal d'indicar de forma gràfica el nombre de places lliures restants.

Cada cop que l'usuari allunyi o apropi el mapa es tornen a recalcular els grups i es tornen a realitzar els càlculs d'ocupació. Així, es veu fàcilment el nombre de places lliures a la ciutat(si s'allunya el mapa) i el nombre de places lliures per cada zona(si s'apropa el mapa).

5. – Joc de Proves

El primer cop que un usuari accedeix a l'aplicació ha de prémer el botó per Registrar-se. Un cop allà es demanen les dades bàsiques(correu electrònic, nom usuari, contrasenya i les dades d'un vehicle).

Quan es fa un inici de sessió, un cop donat d'alta a l'aplicació, es torna a xifrar el correu electrònic introduït i es verifica que sigui el mateix guardat a la base de dades.

Per tal de comprovar el correcte funcionament de les parts de PayPal, de la bossa de minuts i de l'actualització del venciment de les places de pàrquing, s'han fet diverses proves descrites a continuació utilitzant diversos entorns de proves. S'ha utilitzat diversos entorns per tal de provar primer el correcte funcionament bàsic de l'aplicació i segon que totes les característiques extres funcionen correctament.

5.1. – Entorn de proves

La base de l'entorn de treball ha estat utilitzar el WAMP[30] per tal de crear el servidor i la base de dades per a poder treballar. Un cop establert el servidor i la base de dades, s'han utilitzat dos dispositius per a la realització de les proves.

Primer s'ha fet servir el navegador de l'ordinador per a poder comprovar immediatament l'evolució del desenvolupament, anar fent petites proves, en definitiva, la primera línia de depuració per tal de que la lògica de transicions a l'aplicació funcionés correctament i la correcta visualització de la interfície.

Un cop les proves bàsiques estaven fetes, s'ha fet servir dos dispositius mòbils Android, un dispositiu de gamma baixa amb la versió 4.0.4 d'Android i un altre de gamma mitja amb la versió 4.1.2. S'han fet servir aquests dos mòbils com a exemple de les velocitats de processament en dues categories diferents, per tal de reduir al màxim el temps d'espera durant les consultes a la base de dades. Aquesta diferenciació de rendiment ens servia per depurar i optimitzar també el codi.
Les proves amb els dispositius mòbils han servit per avaluar el correcte funcionament de les característiques que no funcionaven amb el navegador web a causa de compatibilitat amb les llibreries.
Totes aquestes proves s'han realitzat mitjançant la xarxa Wifi, degut a que la connexió amb el servidor és a través de WAMP i es necessita estar connectat a la mateixa xarxa LAN per a poder accedir al servidor.

Taula comparativa dels dos entorns de proves mòbils:

	Sonny Ericsson Neo V	**Samsung Galaxy S3**
Processador	Single-Core 1000 Mhz	Quad-Core 1400 Mhz
Memòria RAM	512MB LPDDR2	2GB LPDDR2
Memòria Interna	1GB (320MB accessible al usuari)	16GB
Memòria Externa	Fins 32GB	Fins 64GB
Resolució Pantalla	480 x 854 píxels	720 x 1280 píxels
Tamany Pantalla	3,7"	4,8"
Sistema Operatiu	Android 4.0.4 Ice Cream Sandwich ICS	Android 4.1.2 Jelly Bean
Xarxa	3G HSUPA	4G LTE

5.2. – Comprovació funcionament PayPal

Per avaluar la lògica i que les transaccions es realitzen correctament, s'ha creat un compte d'empresa a PayPal que permet fer proves en un entorn sandbox[31]. Aquest compte disposa d'un venedor i d'un comprador per anar testejant.
Entrant al compte de desenvolupadors es pot veure els dos comptes creats i les últimes transferències.

Email address	Type	Country
projectetfm2013_buyer@gmail.com	PERSONAL	ES
projectetfm2013-facilitator@gmail.com	BUSINESS	US

Es pot comprovar a la següent imatge com coincideixen les transferències rebudes per l'empresa com les realitzades pel comprador.

To	Subject	Date
projectetfm2013-facilitator@gmail.com	Notification of payment received	25 May 2013
projectetfm2013_buyer@gmail.com	Recibo de su pago a Oriol Dols's Test Store	25 May 2013
projectetfm2013_buyer@gmail.com	Recibo de su pago a Oriol Dols's Test Store	25 May 2013
projectetfm2013-facilitator@gmail.com	Notification of payment received	25 May 2013
projectetfm2013_buyer@gmail.com	Recibo de su pago a Oriol Dols's Test Store	25 May 2013
projectetfm2013-facilitator@gmail.com	Notification of payment received	25 May 2013
projectetfm2013_buyer@gmail.com	Recibo de su pago a Oriol Dols's Test Store	25 May 2013
projectetfm2013-facilitator@gmail.com	Notification of payment received	25 May 2013
projectetfm2013-facilitator@gmail.com	Notification of payment received	25 May 2013
projectetfm2013_buyer@gmail.com	Recibo de su pago a Oriol Dols's Test Store	25 May 2013

La lògica de funcionament del pagament mitjançant PayPal és el següent:

1. Un cop a la finstra d'aparcament, s'introdueeix l'identificador de l'aparcament que es troba a la pròpia zona i procedim a pagar.
2. Si l'usuari té permisos i no té la bossa de minuts buida, primer es resta els minuts introduits dels minuts de la bossa i es calcula el nou preu amb el resultant. La bossa s'actualitza tant bon punt es confirmi el pagament. Si la resta ha donat negativa és que ha demanat més minuts dels que li quedaven a la bossa i, per tant, s'obrirà la finestra per a fer la transferència (veure figura 13) a PayPal dels minuts resultants.
3. Si l'usuari no té permisos o bé té la bossa de minuts buida s'obrirà la finestra per a fer la transferència a PayPal amb els minuts que ha introduït al formulari d'aparcament.
4. Un cop introduit el nostre usuari i contrasenya de PayPal demana per confirmar el pagament i si es vol es pot dexiar escrita una nota al venedor(figura 14) i si la transferència ha anat bé informa a l'usuari amb un missatge(figura 15) i ja es pot tancar la fienstra, mostrant així la pantalla principal de l'aplicació actualitzada.
5. En cas de no tenir compte a PayPal es pot o donar-se d'alta o pagar via tarja (figura 16).

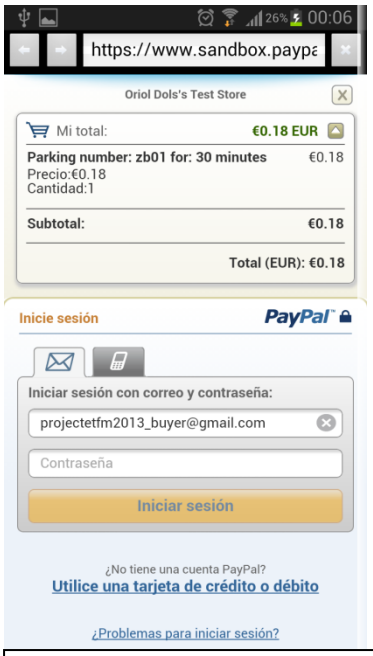


Figura 13 –Iniciar Sessió amb PayPal

En aquesta imatge es pot observar la primera pantalla que s'obrirà al realitzar el pagament. Primer de tot demana estar donat d'alta al servei PayPal. Tanmateix, també es permet pagar mitjançant una targeta de crèdit qualsevol. Es pot comprovar exactament què es paga i quant.

Figura 14 - Confirmar pagament

En aquesta imatge es pot observar el moment en que es realitza la transferència. Tan bon punt es premi *Continuar* es realitzarà el pagament

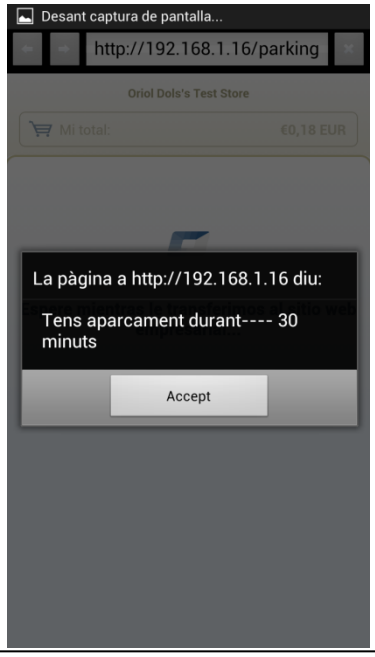


Figura 15 – Finestra de pagament confirmat

En cas que hagi anat tot bé mostrarà el missatge indicant els minuts dels que es disposa. Que han de coincidir amb els minuts introduïts.

Figura 16 - Pagament amb targeta crèdit

En cas de no disposar de compte a PayPal, es permet el pagament mitjançant una tar

Un cop realitzat el pagament a la pantalla principal s'actualitza el mapa i ens canviarà la icona de l'aparcament indicant-nos que aquella plaça està ocupada. A partir d'aquí, cada cinc minuts es comprova automàticament quan de temps queda per a que venci la plaça i quan restin menys de 5 minuts ens avisa. A més a més, des de la pantalla de Settings es pot alliberar les places que hàgim comprat. Al alliberar, si som usuaris amb permisos avançats, els minuts restants s'afegiran a la nostra bossa de minuts. En cas contrari senzillament s'allibera la plaça.

5.3. – Tolerància als errors

En cas que un usuari introdueixi un número negatiu a la durada de l'estacionament se li retorna un missatge d'error conforme no s'ha pogut realitzar(veure figura 17). El mateix que si introdueix una plaça d'aparcament invàlida.

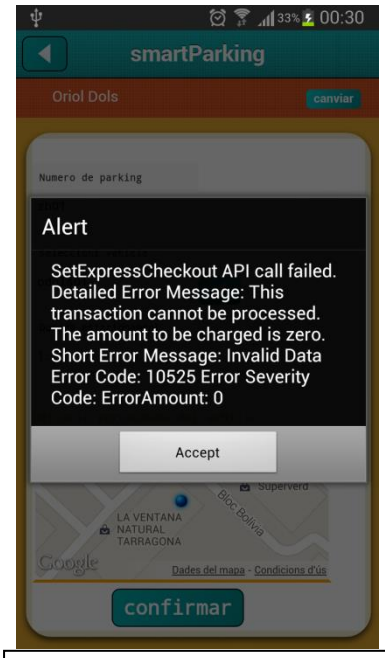


Figura 17 – Missatge d'error retornat per PayPal

En aquesta imatge es pot observar el missatge que retorna la crida a PayPal on s'informa de que les dades referents a la quantitat introduïda són invalides. Ja que no es pot cobrar 0€.

Quan es perd la connexió, per exemple degut a poca cobertura, l'aplicació es posa en mode espera alertant al usuari que no disposa de de connexió a internet mostrant una finestra especial. Tant bon punt es recuperi la connexió es tornarà a obrir l'aplicació a la pantalla principal. Resolent així qualsevol error en la lògica de l'aplicació.

En cas que s'estigui a mitja transferència via PayPal, si s'ha perdut la connexió abans de confirmar el pagament, evidentment, no es perden els diners però si que s'ha de tornar a llogar la plaça. Mentre que si s'ha perdut la comunicació durant els segons que passen entre confirmar i actualitzar la base de dades, es podria arribar a perdre la informació.

5.4. – Temps de resposta

El temps d'arrencada de l'aplicació no és del tot ràpid ja que ha d'aconseguir la latitud i la longitud d'on es troba l'usuari, enviar una consulta al webservice de Geonames per tal de saber a quina ciutat es troba i a partir d'aquí agafar de la base de dades els registres de la ciutat en qüestió.

Un cop aconseguits els registres cal realitzar el càlcul per agrupar per àrees i trobar a quina àrea es troba l'usuari. Això ho aconseguim calculant la distancia més curta amb cada àrea.

A la següent taula es poden observar els temps de resposta per cada esdeveniment rellevant a l'aplicació..

	Sony Ericsson Neo V	**Samsung Galaxy S3**
Arrencada Aplicació (des de que cliques la icona fins que comença a carregar)	6 segons	4 segons
Arrencada Aplicació (des de que es clica el botó fins que es mostra el mapa amb dades)	12 segons	8 segons
Registre (temps des de que es confirma el registre fins a obrir el mapa amb dades)	10 segons	9 segons
Inici Sessió (temps des de que es confirma l'inici de sessió fins a obrir el mapa amb dades)	10 segons	9 segons
Tancament Sessió (des de que es clica el botó fins a mostrar escriptori mòbil)	2 segons	2 segons
Càlcul ciutat	1 segons	1 segons
Actualitzar Dades Aparcaments (des de que es clica el botó fins a que recarrega el mapa amb dades)	4 segons	5 segons
Realitzar pagament (des de que es valida el pagament fins a que finalitza la transacció i s'avisa l'usuari)	9 segons fins a iniciar sessió a Paypal. 3 segons des de iniciar sessió fins a boto confirmar. 7 segons des de confirmar fins a pagat. 4 segons bossa minuts.	9 segons fins a iniciar sessió a Paypal. 3 segons des de iniciar sessió fins a boto confirmar. 7 segons des de confirmar fins a pagat. 2 segons bossa minuts.
Alliberar plaça (des de que es	7 segons	5 segons

clica la plaça a alliberar fins a actualitzar dades al mapa)		
Offline (temps que tarda l'aplicació en detectar que s'ha perdut la connexió)	2 segons	1 segons
Online (temps que tarda l'aplicació en detectar que s'ha recobrat la connexió).	2 segons	1 segons
Distància més curta entre dos punts (des de que s'introdueix una destinació fins que es calcula el recorregut)	5 segons	3 segons
Càlcul àrees aparcaments (des de que es calcula on pertany cada aparcament fins que surt agrupat)	6 segons	4 segons

6. – Conclusions

L'aplicació un cop acabat el desenvolupament és capaç de gestionar els rols de cada usuari que entra a l'aplicació així com fer una correcte autentificació i autenticació i, depenen del rol té l'usuari que inicia sessió, se li permeten accedir a diferents funcionalitats, com ara la bossa de minuts. Aquesta bossa o crèdit només està disponible per usuaris amb permisos avançats. Per cada usuari registrat es pot guardar N vehicles, els quals pot modificar en qualsevol moment.

A més a més, s'ha aconseguit agrupar tots els aparcaments per zones, per tal de millorar la visualització del mapa. També es pot filtrar per amagar els tipus d'aparcament que no volem veure(minusvàlids, zones blaves, elèctrics, entre d'altres). A l'estar els aparcaments agrupats en zones o àrees, és més fàcil extrapolar la informació per a generar estadístiques.

Qualsevol usuari registrat pot reportar sobre les infraccions que detecti per a que les autoritats competents pugin analitzar i actuar com escaigui. Amb això s'ha pretès atorgar a la ciutadania una gestió directe del bon ús dels aparcaments, permeten informar de mals usos per part d'altres conductors.

El desenvolupament del pagament s'ha hagut de desenvolupar a través de PayPal degut a les necessitats de complir l'estàndard del PCI i per les dificultats d'introduir un mecanisme antifrau i que gestioni la confidencialitat del pagament d'acord a les normatives.
Així PayPal ens aporta aquesta capa d'abstracció, sense caldre preocupar-se sobre la seguretat de la transacció econòmica. El que ha calgut fer és modificar les crides de PayPal per a que s'adaptin a les necessitats de l'aplicació.
Per cada plaça comprada es guarda un històric amb el que es va calculant el temps restant per informar l'usuari quan s'està a punt d'arribar a la data de venciment.

6.1. –Treball Futur

Cal destacar que aquesta aplicació és un prototip per a poder comprovar com una aplicació d'aquestes característiques podria adaptar-se a una ciutat. Ja que cal tenir en compte que gran part de la seva raó de ser és la capacitat de la gent de ser civilitzats i pagar realment quan utilitzen un aparcament.

Per aquest motiu, la pròxima fita seria desenvolupar l'aplicació directament sobre llenguatge natiu, tant per Android com per iOS per tal d'aprofitar la velocitat de processament del dispositiu i no haver de dependre del rendiment del navegador.

A més a més, s'intentarà aprofitar al màxim el motor de cerca de Google que ja s'utilitza a l'aplicació i que permet trobar el camí més curt cap a una destinació. Amb ell s'aconseguiria que introduint una localització al mapa et mostres el camí fins allà, indicant com si d'un GPS es tractés i que et mostrés les places lliures a prop d'aquella destinació. Això seria una bona

funcionalitat per a poder aprofitar al màxim diferents funcionalitats com són el GPS i l'aparcament.

Com a punt bàsic a millorar serà arreglar les comunicacions de l'aplicació amb la base de dades per tal de resoldre els possibles problemes de desconnexió puntual que afectin a l'emmagatzematge de les dades, repercutint directament sobre el sistema. Això s'arreglaria utilitzant l'emmagatzematge puntual a una base de dades SQLite. Així s'evitaria perdre informació en cas de desconnexió, podent recuperar-la i actualitzar-la a la base de dades general en quan es disposés de connexió.

Com a punt de millora respecte a la seguretat implantada en aquest projecte és la de modificar la fórmula d'entrada al registrar un nou usuari demanant permisos per emmagatzemar sense xifrar el correu electrònic de l'usuari, protegint-lo tal i com especifica la Llei de Protecció de Dades. Amb això es pretén aconseguir que un usuari que hagi oblidat la contrasenya pugui demanar que s'enviï al correu especificat a la base de dades una nova contrasenya aleatòria, que després es podrà modificar des de dins l'aplicació.

Bibliografia

[1] Tecnologies de la Informació i la Comunicació. http://ca.wikipedia.org/wiki/Tecnologies_de_la_Informaci%C3%B3_i_la_Comunicaci%C3%B3/ (visitada Abril 2013).

[2] Phonegap. http://phonegap.com/ (visitada Maig 2013).

[3] Thomas Myer . Beginning Phonegap. John Wiley & Sons, Inc. ISBN: 978-1-118-15665-0, 2012.

[4] Andrew Lunny. Beginner's Guide. PACKT Publishing . ISBN 978-1-849515-36-8, 2011.

[5] Nitobi Software. http://www.crunchbase.com/company/nitobi-software (visitada Maig 2013).

[6] HTML5. http://en.wikipedia.org/wiki/HTML5 (visitada Maig 2013)

[7] Flanagan, David, and Shafer, Dan. JavaScript: The Definitive Guide. O'Reilly & Associates, 1998.

[8] Newman, C. Sams' teach yourself PHP 5.0 in 10 minutes. Sams Publishing, Indianapolis, IN, USA, 2005.

[9] Ian Gilfillan. La Biblia MySQL. Anaya Multimedia.

[10] John E.Clarck, Bryan P. Johnson. Sencha Touch Mobile Javacript Framework. PACKT Publishing. ISBN 978-1-84951-510-8, 2012

[11] Google API Javascript V3. https://developers.google.com/maps/documentation/javascript/reference (visitada Abril 2013).

[12] Google Maps API. https://developers.google.com/maps/articles/toomanymarkers (visitada Maig 2013).

[13] Geonames. http://www.geonames.org/ (visitada Maig 2013)

[14] Developers PayPal. https://developer.paypal.com/webapps/developer/index (visitada Maig 2013).

[15] Llibreries PhoneGap. https://github.com/phonegap-build (visitada Maig 2013).

[16] Open Street Map. http://www.openstreetmap.org/ (visitada Abril 2013).

[17] Projecció Esfera en mapa 2D. https://developers.google.com/maps/documentation/javascript/maptypes#Projections (visitada Maig 2013).

[18] Geometria Euclidiana. http://es.wikipedia.org/wiki/Geometr%C3%ADa_eucl%C3%ADdea (visitada Maig 2013).

[19] Gran Cercle. http://en.wikipedia.org/wiki/Great_circle (visitada Maig 2013)

[20] Geometria Esfèrica. http://en.wikipedia.org/wiki/Spherical_geometry (visitada Maig 2013).

[21] Creative Commons. http://creativecommons.org/ (visitada Maig 2013).

[22] Sistema Geodèsic Mundial 1984. http://es.wikipedia.org/wiki/WGS_84 (visitada Maig 2013).

[23] Estpandard en Seguretat PCI. https://www.pcisecuritystandards.org/ (visitada Maig 2013).

[24] Tecnologia pagament *card.io*. https://www.card.io/ (visitada Abril 2013).

[25] Criptografia: Salt. http://en.wikipedia.org/wiki/Salt_(cryptography) (visitada Maig 2013).

[26] Contrasenyes més usades. http://www.prweb.com/releases/2012/10/prweb10046001.htm (visitada Maig 2013).

[27] Criptografia: Col·lisions. http://es.wikipedia.org/wiki/Colisi%C3%B3n_%28hash%29 (visitada Maig 2013).

[28] Criptografia: Estàndard del Xifratge de Dades. http://en.wikipedia.org/wiki/Data_Encryption_Standard (visitada Maig 2013).

[29] Llei Orgànica 15/1999 de Protecció de Dades de Caràcter Personal. http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html (visitada Maig 2013).

[30] Entorn de desenvolupament web. http://www.wampserver.com/en/ (visitada Maig 2013).

[31] Entorn de proves aïllat o Sandbox. http://en.wikipedia.org/wiki/Sandbox_(software_development) (visitada Maig 2013).