



**Universitat de les  
Illes Balears**

Escola Politècnica Superior

**Memòria del Treball de Fi de Grau**

# Logística de Turisme Sobre SmartPhones

Ignasi Coll Bagur

**Grau d'Enginyeria Informàtica**

Any acadèmic 2017-18

DNI de l'alumne: 41747375-Y

Treball tutelat per Miquel Mascaró Portells

Departament de Matemàtiques i Informàtica

|                                                                                                                                                                                                         |       |    |       |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----|-------|----|
| S'autoritza la Universitat a incloure aquest treball en el Repositori Institucional per a la seva consulta en accés obert i difusió en línia, amb finalitats exclusivament acadèmiques i d'investigació | Autor |    | Tutor |    |
|                                                                                                                                                                                                         | Sí    | No | Sí    | No |
|                                                                                                                                                                                                         | X     |    | X     |    |

Paraules clau del treball:

Turisme, Viatges, Web/App,



Informe del director.

El doctor Miquel Mascaró Portells, tutor del Treball de Fi de Grau d'Ignasi Coll Bagur, que du per títol Logística de Turisme Sobre Smartphones; fa constar:

Que el projecte està acabat i que els resultats són aptes per a la seva presentació.

Palma 19 de Juny de 2018.

Miquel Mascaró Portells





Universitat de les  
Illes Balears



Treball Fi de Grau

Grau en Informàtica

Logística de turisme sobre  
SmartPhones

Ignasi Coll Bagur

**Tutor / Supervisor**

Miquel Mascaró Portells

Escola Politècnica Superior  
Universitat de les Illes Balears  
Palma, 6 de juny de 2018

# Sumari

|                                                |    |
|------------------------------------------------|----|
| Agraïments.....                                | 9  |
| Acrònims.....                                  | 10 |
| Resum.....                                     | 11 |
| 1. Introducció.....                            | 12 |
| 1.1. Context.....                              | 12 |
| 1.2. Motivacions i objectius del projecte..... | 13 |
| 2. Estat de l'art.....                         | 15 |
| 3. Arquitectura.....                           | 18 |
| 3.1. Esquema.....                              | 18 |
| 3.1.1. Estat inicial.....                      | 18 |
| 3.1.2. Telèfon Mòbil.....                      | 20 |
| 3.1.3. Web Service Propi.....                  | 21 |
| 3.1.4. Base de dades.....                      | 21 |
| 3.1.5. Web Service Extern (API).....           | 21 |
| 3.2. Requisits Funcionals.....                 | 21 |
| 4. Implementació.....                          | 24 |
| 4.1. Telèfon Mòbil (Cordova).....              | 25 |
| 4.2. Web Service Propi (PHP).....              | 29 |
| 4.3. Base de Dades (MySQL).....                | 30 |
| 4.4. Web Service Extern – API.....             | 31 |
| 4.4.1. Google Maps.....                        | 32 |
| 4.4.2. Redsys.....                             | 33 |

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| 4.5. Desenvolupament de l'aplicació .....                              | 34        |
| 4.5.1. IDE de Programació .....                                        | 34        |
| 4.5.2. Llenguatge de Programació.....                                  | 35        |
| 4.5.3. Estructura.....                                                 | 37        |
| 4.5.4. Problemàtiques de desenvolupament i ús de les APIs externes.... | 40        |
| 5. Metodologia de desenvolupament .....                                | 43        |
| 6. Demostració i resultats.....                                        | 48        |
| 6.1. Inici de l'aplicació.....                                         | 48        |
| 6.2. Plana 'Mi Reserva' .....                                          | 50        |
| 6.3. Plana 'Servicios' .....                                           | 52        |
| 6.3.1. Llocs d'interès prop de la ubicació del turista.....            | 52        |
| 6.3.2. Excursions.....                                                 | 54        |
| 7. Conclusions .....                                                   | 58        |
| <b>Referències</b> .....                                               | <b>61</b> |

# Llista d'imatges

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Imatge 1: Representació bàsica de l'arquitectura de l'aplicació .....                     | 19 |
| Imatge 2: Tecnologia utilitzada per dur a terme l'arquitectura de l'aplicació .....       | 24 |
| Imatge 3: Estructura de l'aplicació realitzada.....                                       | 38 |
| Imatge 4: Estructura de carpetes del projecte .....                                       | 39 |
| Imatge 5: Cicle de vida de l'aplicació.....                                               | 45 |
| Imatge 6: Plana de càrrega amb decisió de redirecció al login o a la plana principal..... | 49 |
| Imatge 7: Plana d'informació de la reserva de l'usuari.....                               | 51 |
| Imatge 8: Plana d'informació de vols .....                                                | 52 |
| Imatge 9: Plana de serveis prop de la ubicació del client.....                            | 53 |
| Imatge 10: Llistat de les excursions pròximes al client.....                              | 54 |
| Imatge 11: Botó de compra de l'excursió.....                                              | 55 |
| Imatge 12: Informació detallada de l'excursió elegida .....                               | 55 |
| Imatge 13: Formulari de compra d'una excursió.....                                        | 56 |
| Imatge 14: Plana de pagament d'una excursió .....                                         | 56 |



# Agraïments

Vull agrair a totes les persones que han estat al meu costat durant tot el camí que ha suposat aquesta etapa d'estudis, cada un ha aportat el seu gra d'arena que ha fet possible l'escriptura d'aquestes línies.

Al meu tutor, en Miquel Mascaró, per a donar-me les pautes necessàries per la realització d'aquesta memòria i tots els seus consells per perfeccionar-la fins a obtenir un gran Treball de Fi de Grau.

A tots els amics que he fet en aquesta illa, com els que he tingut tota la vida, que m'han recolzat en les bones i les dolentes i he pogut comptar sempre amb ells.

A n'Azahara, qui ha revisat part d'aquest TFG i m'ha ajudat per millorar la redacció d'aquest, apart de fer que el meu dia a dia sigui més feliç gràcies a la seva companyia.

Al meu germà, en Xavier, que m'ha ajudat als inicis d'aquesta carrera amb tots els meus dubtes de forma desinteressada. Gràcies a tu m'he convertit en un gran Enginyer i sé que en un futur farem història junts.

Als meus pares, en Toni i na Charo, que no els hi he d'agrair tan sols el seu suport durant aquesta dura carrera, sinó tot el que han fet per jo durant la meva vida. Eu estat un exemple a seguir per mi i gràcies a vosaltres dos he aconseguit els valors i l'educació que han fet que sigui qui som. Podria escriure milions de paraules, però ho resumiré donant les gràcies per aquests 22 fantàstics anys que he passat al vostre costat, esper que siguin molts més.

# Acrònims

|             |                                        |
|-------------|----------------------------------------|
| <b>SEO</b>  | Search Engine Optimization             |
| <b>TFG</b>  | Treball de Fi de Grau                  |
| <b>API</b>  | Application Programming Interfaces     |
| <b>WS</b>   | Web Service                            |
| <b>BD</b>   | Base de Dades                          |
| <b>CORS</b> | Intercanvi de Recursos d'Origen Creuat |
| <b>IDE</b>  | Integrated Development Enviroment      |
| <b>SDK</b>  | Software Development Kit               |
| <b>APK</b>  | Android Application Package            |
| <b>TPV</b>  | Terminal Punto de Venta                |

# Resum

El Treball de fi de Grau exposat es basa en una aplicació per ‘SmartPhone’ (tant per Android com per iOS) en la qual els clients que han reservat un paquet de vacances amb l’agència de viatges, poden conèixer tota la seva informació sobre la reserva, tals com l’hotel al qual s’allotjaran, quin tipus d’habitació tenen, qui ocuparà cada habitació, les hores i el lloc dels seus transports... A més a més, l’aplicació inclou una part en la qual surt un mapa amb la ubicació de l’hotel i tots els llocs d’interès que hi ha al voltant. Es poden filtrar dits llocs per cada tipus (restaurants, gasolineres, lloguer de cotxes, ...) i conèixer la seva informació de contacte.

Finalment, també es podran reservar totes les excursions que promociona l’agència de viatges que es trobin a prop de l’estança del client, proporcionant una sèrie de fotografies, explicacions i una plataforma de pagament segura.

**Paraules clau: Android, iOS, Viatges, WebApp**

# 1. Introducció

El present Treball de fi de Grau sorgeix de la feina que vaig efectuar per una empresa anomenada “Leomar Viajes” mentre cobria una plaça de Pràctiques en Empresa. Aquestes pràctiques es varen realitzar en finalitzar el curs 2016-2017 durant les vacances d’estiu i, entre altres coses, es va portar a terme la feina exposada a continuació.

Durant la meua estança com a programador dintre de l’empresa, va sorgir un nou projecte intern, on se’m va encomanar realitzar una aplicació per ‘SmartPhones’ amb la qual tots els clients que contractessin les seves estances vocacionals amb “Leomar Viajes”, poguessin tenir un control de totes les seves activitats programades, tals com, l’hotel en el qual s’allotjarien, les habitacions contractades, l’horari dels vols, l’hora de recollida pels transports contractats i un llarg etcètera que es podrà veure en el llarg d’aquest document.

## 1.1. Context

L’organització per la qual vaig ser contractat, es trobava en un moment on ja disposava d’un servei web pel qual es venien els diferents vols i/o paquets turístics, i també, amb una intranet on tots els treballadors hi feien feina depenent dels seus rols. A més a més, tota la informació estava

## 1. Introducció

emmagatzemada dins una base de dades, i mitjançant aquesta, es comunicaven les aplicacions descrites anteriorment.

Cal destacar que es tracta d'una empresa enfocada en el món del turisme oferint preus competitius en el mercat de la venda de vols, hotels, excursions, etc., el sector més competitiu actualment en el món de la Internet, on tenen un gran pes les optimitzacions a nivell [SEO](#) i existeix una gran competència donat el gran nombre d'organitzacions que ofereixen aquest servei.

### 1.2. Motivacions i objectius del projecte

Dintre d'aquest context és el moment quan l'empresa va voler ampliar el seu mercat i que, en paral·lel de tot el que s'ha esmentat, es comencés amb el desenvolupament d'una aplicació per a telefonia mòbil amb la qual els turistes poguessin accedir a ella mitjançant el seu número de reserva. La realització d'aquesta aplicació aportaria a l'empresa diversos beneficis:

- Satisfacció per part dels clients en disposar aquests d'una plataforma que faciliti l'organització del seu viatge. El que es pot traduir al fet que tornin a contractar amb l'empresa futurs paquets vocacionals.
- Avantatge competitiva en aportar un servei únic, que tan sols poden utilitzar els clients que contractin amb la nostra agència.

Per tal de poder assolir tots aquests beneficis, des del principi del projecte, es varen establir uns objectius o requisits que hauria de contenir la futura aplicació:

- Permetre entrar al sistema mitjançant el número de reserva que se li aporta als clients quan aquests reserven un vol.
- Tenir una plana web on s'agrupa informació del seu interès durant tota la seva estança: Dates dels vols a collir i tots els transports contractats, hotel i habitacions reservades amb els membres que ocuparan cada una de les habitacions i les dades de contacte de l'hotel.

## 1. Introducció

- Contenir dins l'aplicació un mapa on se centri directament a l'hotel contractat pel client i on apareguin llocs d'interès prop d'aquest, com per exemple, restaurants, gasolineres, bars, miradors...
- Oferir la possibilitat als clients de llistar totes les excursions que es poden fer prop de la seva estança i poder contractar qualsevol d'aquestes excursions mitjançant una plataforma de pagaments segura.
- Comptar amb la disponibilitat d'una pàgina de contacte amb la qual es pugui comunicar amb l'organització mitjançant un formulari i amb el número de telèfon de la seu més propera i dels caps dels guies de les excursions.

## 2. Estat de l'art

Cal esmentar que la creació d'una aplicació de turisme no és un fet nou, pel que podem mencionar una gran llista d'aplicacions que han treballat en el mateix context, i, per tant, ofereixen uns serveis similars al presentat [TFG](#):

- **Aena [1]:** Aquesta pàgina web o aplicació per telèfon és una de les més emprades a l'hora d'emprendre un viatge si el destí o aeroport d'origen resideix a Espanya. Això es deu al fet que gràcies a aquesta aplicació es podria veure tota la informació del teu vol, si ha sofert retards i diu els llocs d'interès que et pots trobar a cada un dels aeroports del país.
- **Mi Nube [2]:** Aquesta plataforma va començar com una App que permetia descobrir llocs per viatjar. Això ha anat evolucionant i actualment permet un gran nombre de funcionalitats, d'entre elles, poder organitzar el teu propi viatge i descobrir llocs d'interès prop de la ubicació de l'usuari final, així com restaurants, monuments, etc.
- **Tripadvisor [3]:** Aquesta aplicació permet veure tots els hotels i restaurants en una zona determinada i saber les opinions de milers de turistes que ja han visitat aquests llocs, aportant algunes ressenyes i valoracions per qualificacions per tal de recomanar o desaconsellar dit

## 2. Estat de l'art

lloc. A més a més, també permet la reserva dels restaurants i els hotels dintre de la mateixa aplicació.

- **AroundMe [4]:** Una altra aplicació que permet trobar tots els restaurants, monuments, hotels, tendes, pàrquings, hospitals i, bàsicament qualsevol lloc interessant que puguis trobar prop de les coordenades de qui l'utilitzi. Mitjançant l'aplicació, també pots visualitzar tota la informació relacionada amb el lloc en concret i una ubicació exacta en un mapa.
- **Prava [5]:** En aquest cas estem parlant d'una App que serveix per organitzar un viatge en grup. Cada un dels integrants del viatge poden descarregar l'aplicació i compartir la informació que té sobre el destí. Alhora disposa d'una agenda per ordenar les visites que es volen fer diàriament i un mòdul per controlar i dividir les despeses. A més a més, es poden explorar alguns llocs de cada una de les zones que han visitat altres usuaris de l'aplicació per si es vol cercar idees de visites i afegir-les a l'itinerari en qüestió.
- **Mtrip [6]:** Aquesta App ofereix uns serveis bastant similars a les anteriors. En aquest cas s'ha d'introduir el lloc de destí juntament amb les dates de l'estança. A continuació, es trien el tipus de llocs que s'interessen visitar (museus, monuments, parcs, etc). Seguidament l'App et proposa un itinerari amb l'opció de personalització, permetent eliminar o afegir algunes activitats.
- **Google Trips [7]:** L'empresa Google també ha ficat mà en aquest sector mitjançant la seva aplicació oficial de viatges. A través aquesta, es pot trobar una guia turística personalitzada, podent accedir a tota la informació sense necessitat de connexió a internet.
- **Google Maps [8]:** Avui en dia gairebé tothom coneix aquesta gran plataforma de Google. Mitjançant aquesta es pot arribar a veure pràcticament tots els llocs d'interès d'una zona determinada i actuar com a GPS marcant el camí per arribar des d'un punt fins a qualsevol altre.



## 2. Estat de l'art

- **Local Scope** [\[9\]](#): Aquesta plataforma de pagament i tan sols disponible per telèfons amb el sistema operatiu iOS, permet trobar llocs d'interès d'acord amb la ubicació actual. A més a més, per a cada un d'aquests llocs, proporciona ressenyes escrites per altres viatgers, fotografies i vídeos que s'hagin gravat al voltant de la ubicació introduïda.
- **Izi.TRAVEL** [\[10\]](#): De nou es presenta una aplicació que actua com a guia de viatges, trobant tots els llocs d'interès al voltant d'una ubicació posicionada damunt un mapa. El que fa diferent aquesta aplicació és que conté milers d'àudios per tal de poder fer una ruta guiada als principals museus i galeries d'art sense necessitat de connexió a internet si descarregues els àudio-guides abans de visitar el lloc en qüestió.

Com s'ha esmentat anteriorment, el món dels viatges és un sector molt competitiu en l'actualitat, per això tots els antecedents presentats a la llista anterior no són més que un percentatge molt petit de les aplicacions d'aquest sector que ens podem trobar actualment a Internet.

Així mateix, a l'aplicació d'aquest [TFG](#) s'ha emprat moltes de les tècniques que ja s'han utilitzat amb anterioritat, fusionant-les i adaptant-les a l'organització pròpia, juntament amb unes funcionalitats pròpies per oferir un producte únic. Totes les funcionalitats i objectius es poden veure al [capítol 1.2 de la introducció](#).

# 3. Arquitectura

Per fer possible tots els objectius marcats per l'organització, a l'hora de realitzar l'aplicació, s'ha dut a terme la modelització que s'exposa durant tot aquest punt.

## 3.1. Esquema

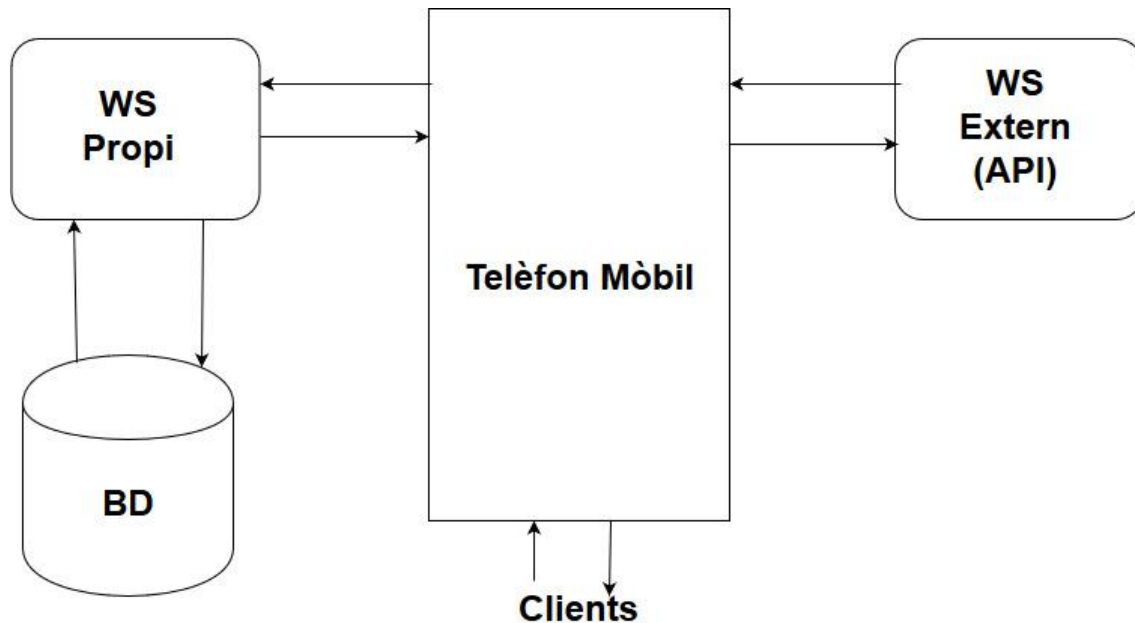
A l'esquema següent, es pot veure l'arquitectura que es farà servir per realitzar el projecte. Aquesta està principalment integrada per un telèfon mòbil pel qual es connectaran els clients i un *Web Service* propi per consultar totes les dades dels clients emmagatzemades dintre d'una base de dades. A més a més, els clients també es podran connectar a algunes [API](#) externes per tal d'oferir alguns serveis proporcionats per altres organitzacions (veure Imatge 1).

### 3.1.1. Estat inicial

En el moment que va sorgir la idea del TFG, ens trobàvem en un estat inicial en la que teníem una plana web de l'organització mitjançant la qual es realitzaven les reserves dels paquets de viatges. A partir d'aquestes reserves, es guardaven totes les dades dels clients que contractaven amb nosaltres dintre d'una base de dades, mitjançant la qual quedava registrat totes les activitats i reserves que havien realitzat.

### 3. Arquitectura

Es va pensar, doncs, a fer el desenvolupament d'una aplicació per tots els clients que ja havien contactat amb la nostra organització, ja que es va pensar que era una bona forma de fidelitzar els clients, aportant-los un valor afegit a la seva mitjançant un servei únic.



Imatge 1: Representació bàsica de l'arquitectura de l'aplicació

El punt de partida va ser la base de dades de l'organització mitjançant la qual es podia realitzar un servei per compactar totes les dades dels clients dintre d'una aplicació, on poguessin tenir-les a mà en tot moment.

Així mateix, per tal de poder oferir les dades d'una forma amigable i comprensible per l'usuari, necessàvem un *Web Service* propi que fes totes les consultes necessàries a aquest per demanar la informació a la base de dades de forma estructurada, i retornar-les al programa client.

Llavors, a la part del programa client, s'havia de construir una interfície amigable i comprensible per l'usuari, per tal que l'usuari final pogués navegar entre les diferents seccions de l'aplicació amb la finalitat d'efectuar consultes a aquest *Web Service* quan fos necessari, per obtenir les dades i pintar-les

### 3. Arquitectura

dintre d'aquesta. A més a més, a part d'accedir al nostre WS per agafar les dades dels clients i els seus moviments, havia de ser capaç de fer consultes a unes API externes, per obtenir la informació d'aquestes i pintar-les de nou dintre de l'aplicació. D'aquesta forma, amb les dades que ens proporcionaven altres organitzacions, podíem donar un valor afegit a l'usuari i un motiu més per utilitzar la nostra aplicació.

Les API comentades anteriorment, són les de Google Maps i Redsys. Aquesta primera s'ha utilitzat per tal d'obtenir un mapa del món i pintar damunt aquest els marcadors dels llocs d'interès que hi ha en les zones que oferim serveis. Per altra banda, Redsys és una plataforma de pagament segur per tal d'oferir als usuaris diferents excursions prop de la seva ubicació i donar opcions de compra per afegir-les a la reserva del client en qüestió. Les dos apis són comentades a fons a la secció d'implementació, concretament els [punts 4.4.1](#) per l'API de Google Maps i la [secció 4.4.2](#) per l'API de Redsys.

#### 3.1.2. Telèfon Mòbil

A l'hora de crear una aplicació, els clients accediran a ella mitjançant els seus *SmartPhones*. Aquests telèfons s'executaran amb els diferents sistemes operatius depenent del dispositiu: Android, iOS i Windows Phone. Cal dir que s'han d'englobar tots aquests possibles dispositius, ja que els clients que han contractat amb l'empresa han de poder accedir al producte únic que oferim, i no es poden veure limitats per telèfon que posseeixen.

El dispositiu en qüestió s'ha de poder connectar mitjançant un programa client al WS propi i als externs per poder obtenir tota la informació i dades i pintar-les d'una forma amigable i comprensible per l'usuari. Aquesta transferència de dades s'ha de realitzar mitjançant *http protocol*, passant uns paràmetres a través d'un format de text lleuger apte per l'intercanvi de dades.

### 3. Arquitectura

#### 3.1.3. Web Service Propi

Per aquesta aplicació també s'ha de disposar d'un *Web Service* propi, allotjat als servidors de l'organització per tal de poder accedir a totes les dades dels clients que han contractat un paquet turístic a partir del nostre servei web. Aquest [WS](#) ha de ser capaç de connectar-se amb la base de dades de l'organització, per tal de poder recollir i llegir totes les dades i després tornar-les al programa client, qui farà ús d'aquesta informació per pintar-la d'una forma senzilla i entenedora per l'usuari final.

#### 3.1.4. Base de dades

Per altra part, també es farà ús d'una [BD](#) per tal de tenir tota la informació dels usuaris que s'han d'imprimir per la pantalla. En tenir tantes dades dels nostres clients, necessitem garantir la integritat d'aquestes. Per aquest motiu és més interessant que aquesta base de dades segueixi un model relacional, amb una estructura sòlida relacionant els diferents elements que la componen.

#### 3.1.5. Web Service Extern (API)

L'aplicació també podrà accedir a Web Service extern a l'organització per tal de comunicar-se amb serveis externs i poder-los incloure dintre de l'aplicació sense necessitat d'haver-hi d'escriure el codi per aquesta funcionalitat concreta. [Al capítol 4 d'implementació](#) es veuran totes les API específiques que s'han fet servir.

## 3.2. Requisits Funcionals

El funcionament de l'aplicació es duu a terme de la forma que s'explicarà a continuació. Cal recordar, que en cap moment es realitza la compra de viatges mitjançant l'aplicació, sinó que es porta a terme mitjançant la plana web. Una vegada el client realitza la compra d'un vol, aquest rebrà un correu electrònic que contindrà tota la informació del que ha contractat en aquell moment, així com les dades que s'han introduït. A més a més, aquest correu tindrà un

### 3. Arquitectura

enllaç cap a l'aplicació a totes les plataformes per la qual s'ha desenvolupat, moment en el qual es podrà començar a fer ús de l'aplicació desenvolupada.

L'usuari final, una vegada hagi descarregat l'aplicació a la tenda pertinent (de forma gratuïta), obtindrà la informació i introduirà el número de reserva i la data d'entrada que se li ha proporcionat al correu electrònic. D'aquesta manera tindrà accés a l'aplicació. Serà en aquest moment, quan el dispositiu mòbil realitzi una petició al Web Service intern de l'organització, per tal d'accedir-hi a tota la informació del paquet que ha contractat el client. El Web Service de l'organització, recollirà tota aquesta informació de la base de dades, que la web haurà gravat prèviament. Aquesta informació es guardarà a la sessió de l'usuari a la part del client, per tal de no haver de demanar les dades constantment al Web Service i estalviar dades mòbil al client. A part de tenir aquest avantatge esmentada, les dades que s'han guardat a la memòria del telèfon es destruiran automàticament un cop es tanqui l'aplicació, pel que no es requerirà molta memòria i es prevé del robatori de la informació del client en cas d'un atac d'un pirata informàtic. A més a més, en haver de buscar entre un volum de dades gran, es baixarà d'una forma notable el temps de resposta de l'aplicació, ja que s'estalvia l'enviament de dades a través de la xarxa i les pot pintar directament des de la memòria interna del telèfon.

A partir d'aquest moment, l'usuari podrà accedir a tota la informació que s'ha recollit, i navegant entre les diferents planes de l'aplicació es pintaran quan sigui necessari, i a més a més, es podrà realitzar en algun moment una petició a una API externa si es requereix d'ella.

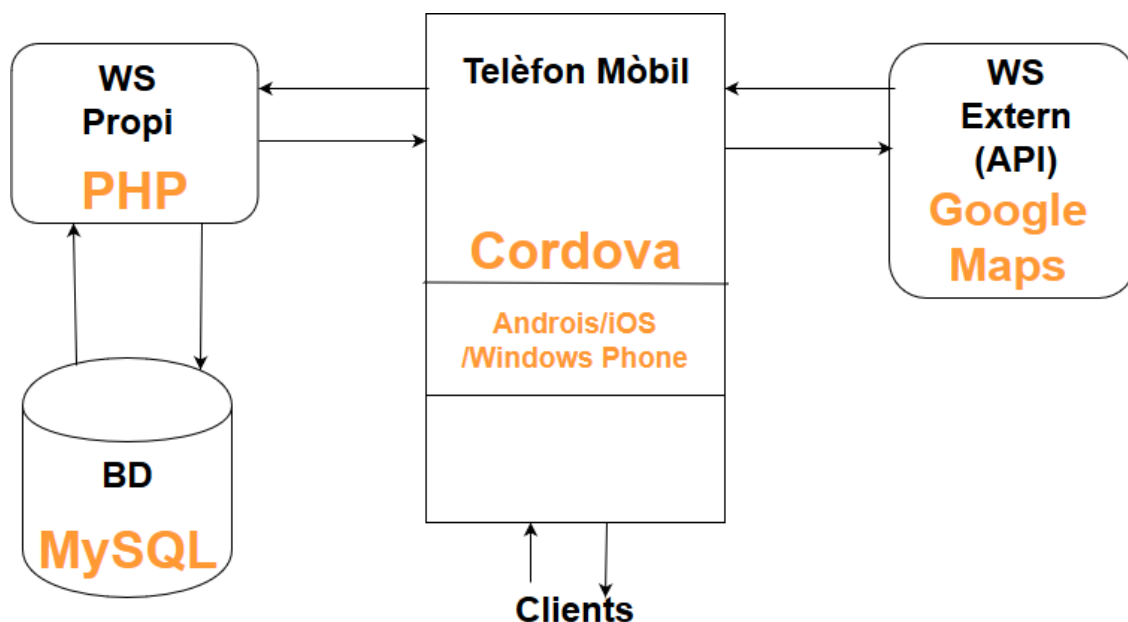
A més a més, per tal de facilitar l'autenticació de l'usuari amb l'aplicació, les credencials inicials a la pàgina d'inici de sessió tant sols s'hauran d'introduir un cop. D'aquesta forma, quan s'executi l'aplicació de nou, les dades d'autenticació estaran guardades i no serà necessari tornar-les a introduir, i

### 3. Arquitectura

automàticament accedirà al Web Service per tal d'actualitzar la informació de la reserva de l'usuari.

# 4. Implementació

Per tal de dur a terme tot el que s'ha exposat als capítols anteriors, es va haver de triar unes tecnologies específiques. A continuació, es presenta l'esquema de l'arquitectura amb les tecnologies concretes que es varen triar. A més, durant aquest capítol entrarem molt més a fons amb qüestions tècniques.



Imatge 2: Tecnologia utilitzada per dur a terme l'arquitectura de l'aplicació



### 4.1. Telèfon Mòbil (Cordova)

Com s'ha esmentat anteriorment, el dispositiu que utilitza el client no ha de ser un factor determinant per tal de poder utilitzar l'aplicació. D'aquesta forma, l'aplicació s'havia d'adaptar als diferents sistemes operatius mòbils, per tal de garantir que tots els clients puguin beneficiar-se de l'aplicació.

Arribat en aquest moment, s'havia de prendre una decisió pel desenvolupament de l'aplicació: el desenvolupament natiu per a cada una de les plataformes o el desenvolupament d'una aplicació multi plataforma.

Per una banda, el desenvolupament natiu d'una aplicació ens aporta un millor rendiment i ens assegura una bona interfície d'usuari, ja que s'ha desenvolupat i optimitzat de forma específica per la plataforma en concret. A més a més, utilitzant el desenvolupament natiu podrem accedir al 100% de les funcionalitats permeses i a les característiques del dispositiu si fos necessari. El principal inconvenient d'aquesta opció seria el temps de desenvolupament, ja que hauríem de desenvolupar una aplicació per a cada un dels sistemes operatius que volíem abastar: Android, iOS i Windows Phone.

Per l'altre costat, ens trobàvem amb la decisió de dur a terme el desenvolupament mitjançant una aplicació híbrida, és a dir, una aplicació que s'adapta als diferents sistemes operatius. Aquesta opció ens soluciona el principal inconvenient que ens aporta una aplicació nativa, però a canvi hem de pagar per un menor rendiment. En aquest tipus d'aplicació, l'execució és una mica més lenta, gasta més memòria que les aplicacions natives i consumeixen més recursos, com per exemple, una despesa més elevada de la bateria del dispositiu. Així mateix, aquestes aplicacions híbrides solen tenir una gran quantitat de *plugins*, pel que no és difícil poder accedir a les característiques del sistema pel qual s'executa, encara que no es garanteix poder accedir al 100%.

#### 4. Implementació

Així mateix, si ens centrem tan sols dintre de l'espectre de possibilitats de les aplicacions multi plataforma, podem trobar una gran quantitat d'eines que ens permeten realitzar una *app* híbrida. Alguns exemples que permeten fer això són: Cordova [\[11\]](#), Ionic [\[12\]](#), Unity [\[13\]](#) i Xamarin [\[14\]](#). Aquests *frameworks* mencionats són els més populars que existeixen actualment, encara que en podem trobar molts més. La similitud entre tots aquests serveis és la seva finalitat: permetre la creació d'una *app* híbrida.

Arribats a aquest punt, es va decidir triar la branca del desenvolupament d'una *app* híbrida, ja que el cost de desenvolupament era molt menor que desenvolupar una aplicació per a cada un dels sistemes operatius. A més a més, tan sols es tenien les nocions bàsiques de programació amb Android i una experiència nul·la amb la programació d'aplicacions pels sistemes operatius iOS i Windows Phone. Llavors, calia prendre una decisió amb per elegir el framework que s'utilitzaria.

Les aplicacions que utilitzen Cordova es programen com una aplicació web normal, utilitzant els llenguatges HTML5, JavaScript i CSS. En aquest punt, es tenia soltesa programant amb aquests 3 llenguatges de programació per una experiència prèvia de desenvolupament de diverses planes web. A més a més, disposa d'una comunitat de programadors bastant estesa, amb una quantitat infinita de *plugins* per accedir als recursos i característiques del dispositiu.

La programació mitjançant Ionic Framework és una mica diferent, ja que està més enfocada a la programació dirigida per components. En si mateix, també es desenvolupa una plana web, però amb una sintaxi pròpia destinada a aquest paradigma de programació. A més a més, per la interacció a la part client utilitza el *framework* AngularJS, una tecnologia relativament recent amb la qual encara no s'estava familiaritzat. Cal destacar que aquest servei utilitza per davall el sistema de Cordova mencionat anteriorment.

## 4. Implementació

Unity és un altre *framework* que s'utilitza per crear aplicacions híbrides, amb una programació basada amb C#, un llenguatge en el qual tampoc es tenia experiència. Addicionalment, aquest es va crear inicialment per facilitar la programació de videojocs, pel que és un entorn bastant centrat en un tipus de creació concreta, la qual s'allunyava dels nostres objectius principals

Xamarin és la solució adoptada per Microsoft, el qual utilitza una programació basada també amb C#, integrant la programació de .NET a les aplicacions per a telèfon mòbil.

Davant totes aquestes opcions, ens vam decantar per realitzar el desenvolupament de l'aplicació mitjançant l'eina Cordova. Això es deu al fet que utilitzava uns llenguatges de programació amb els quals s'està bastant familiaritzat, pel que la programació es realitzaria molt més ràpidament que amb les altres opcions. A més a més, totes les opcions que oferien la programació amb C# varen ser descartades immediatament per la baixa experiència amb el llenguatge i principalment per l'absència de compatibilitat amb l'API de Google Maps que s'havia d'utilitzar ([veure secció 4.4.1](#)).

Una vegada presa la decisió, ens trobàvem amb una eina de codi obert que permet fer una programació utilitzant HTML5, JavaScript i CSS3, evitant haver de programar amb el llenguatge natiu per a cada una de les plataformes.

Aquest fet ens proporcionava un avantatge molt gran: programant l'aplicació tan sols un cop ens assegurava tenir un executable per Android, iOS, Windows Phone i Web, cosa que feia que ens estalviés un quantitat elevada de temps per tal de dur el desenvolupament de l'aplicació.

A més a més, ja es tenia soltesa amb la programació Web, cosa que aportava encara més facilitat, mentre que es tenia una experiència nul·la amb la programació d'aplicacions per el sistema operatiu iOS, Windows Phone, C# i AngularJS.

#### 4. Implementació

De totes maneres, s'havia de tenir en compte que l'aplicació podria ser executada a diferents dispositius, tals com SmartPhones, Tablets i Ordinadors de sobretaula, així que havia de tenir una interfície amigable per els 3 tipus de dispositius diferents. Per aquesta raó, al programar-se com una web normal, s'havia de desenvolupar una Web/App *responsive*, és a dir, que s'adapti depenent del dispositiu que faixi ús de l'aplicació. Per tal de fer això possible, estalviar molt més temps i facilitar la programació, es va decidir comprar un *Theme* que ens donés una primera estructura. Després de fer una recerca, es va decidir comprar-lo mitjançant la pàgina web ThemeForest [15], més concretament un anomenat MyMobile. Es pot veure aquest tema, totes les funcionalitats i una demostració online mitjançant el seu anunci de compra dintre de la seva plana de compra [16]

Cal destacar que d'aquesta forma ens donava molta flexibilitat, ja que ens permetia utilitzar totes les llibreries de JavaScript i tots els *plugins* que es poden instal·lar propis de Cordova.

Com a llibreries pròpies de JavaScript cal destacar l'ús de JQuery, un *framework* que ens proporciona molta facilitat de programació a la part client, sobretot per les consultes als WebServices, tant propis com externs. A més a més, també ens ha servit per fer la Web/App multiidioma. Això es deu a jquery-localize [17], que permet assignar uns identificadors a cada una de les parts del nostre HTML, i segons l'idioma que seleccioni l'usuari, se li aporta amb l'idioma desitjat. Cal dir que la traducció no és automàtica, sinó que depenen d'uns fitxers JSON (un per cada idioma), que utilitza com a clau l'identificador esmentat anteriorment i com a valor el text amb l'idioma corresponent. Tenir aquesta traducció a diversos idiomes ens aporta arribar a un públic molt més ampli, ja que la majoria dels clients provenen de l'estranger. Gràcies a això, es reforça la nostra idea d'arribar a la major part possible dels nostres clients, fent que l'idioma no sigui un condicionant a l'hora d'utilitzar el nostre producte.

## 4. Implementació

A més a més, també es fa ús de les fonts de Google, d'aquesta forma, s'afegeixen tipografies més amigables per l'usuari. Aquestes han estat la Roboto [18] i la Satisfy [19], que venien per defecte amb el Tema que es va adquirir.

Cal afegir que s'ha fet ús d'un altre *framework*, anomenat *Font Awesome* [20]. En aquest cas, l'únic del què s'ha aprofitat d'aquest ha estat les icones. Font Awesome ofereix un gran repertori d'icones que es poden afegir de forma molt senzilla a l'HTML, mitjançant l'ús de classes de CSS.

Com a *plugins* propis de Cordova cal destacar l'ús de *inAppBrowser* [21], que permet viatjar entre diverses webs reals sense necessitat d'haver de sortir de l'aplicació. Això ens és útil, ja que es proporciona un enllaç de l'hotel on ha contractat el client, pel que si vol veure més informació o contactar amb l'hotel no farà falta sortir de l'aplicació. A més a més, l'aplicació proporciona una plataforma de pagament segura per tal de poder comprar les excursions, fent ús de la plataforma Redsys [22], que ha d'anar a una web externa per tal d'efectuar el pagament.

### 4.2. Web Service Propi (PHP)

La nostra aplicació no podria funcionar tan sols amb la programació de la part client, sinó que també necessita una part que s'executi al servidor per tal de poder comunicar-se amb la Base de Dades i obtenir tota la informació necessària per el client.

Existeix un gran nombre de llenguatges que ens permet crear un Web Service, però en aquest cas s'ha elegit el PHP, un llenguatge de programació obert amb una corba d'aprenentatge bastant baixa, i amb una connexió a la base de dades relativament senzilla de realitzar.

Es va triar aquest llenguatge de programació, ja que es tenia una experiència prèvia amb aquest llenguatge, on ja es dominava la connexió amb la base de dades, que era bàsicament pel que s'utilitzaria.

## 4. Implementació

Per tal de programar-ho, es va tenir en compte que les peticions podrien provenir des de qualsevol adreça IP, pel que es varen afegir els encapçalats corresponents per tal de permetre aquesta connexió i no tenir problemes de control d'accés [CORS](#).

A més a més, aquest Web Service funciona mitjançant el pas de paràmetres amb el format JSON, ja que quan es fa la petició s'envia les dades que s'ha introduït al *login*, sempre mitjançant el mètode POST. Aquest WS, rebrà tots els paràmetres que s'envien i farà les peticions a la base de dades segons aquests. Simplement, després d'haver fet totes les seleccions, es recullen els resultats d'aquest i es construeix un altre JSON, que serà el que es retornarà al programa client, qui serà l'encarregat de *desparsetjar-lo* per obtenir tota la informació que s'ha sol·licitat.

La construcció de la resposta en JSON amb PHP es pot realitzar fent servir sempre el mateix patró, ja que tan sols es fa un *array* de resposta amb tots els resultats obtinguts, aplicant-li una clau a cada una d'elles. Finalment s'ha d'aplicar una funció que realitza una conversió al format JSON passant *l'array* construït fins aleshores.

Cal destacar que s'ha utilitzat el JSON com a llenguatge de comunicació entre les diferents entitats perquè resulta molt més lleuger que XML, pel que la resposta és una mica més ràpida. A més a més, es molt més fàcil *parsetjar* aquest llenguatge, ja que amb una sola instrucció de Javascript és possible realitzar-ho, mentre que per XML necessita un *parsetjador* específic. La facilitat de programació per recuperar les dades també es fa molt més ràpida, ja que l'accés és pràcticament immediat si s'ha creat un bon JSON a la part servidor, cosa que amb XML no ho es.

### 4.3. Base de Dades (MySQL)

També s'ha mencionat a capítols anteriors que es necessitava una base de dades relacional per tal de poder estructurar i recuperar tota la informació

## 4. Implementació

de l'usuari de forma eficient, garantint la integritat de les dades. Ara bé, dintre d'aquest grup de dades relacionals, existeix un gran número per triar.

De totes maneres, l'elecció de quina base de dades utilitzar no va tenir discussió. Cal recordar que també s'ha dit anteriorment que primer de tot els clients contracten un paquet de vacances mitjançant la Web de la companyia, i una vegada se li ha proporcionat les dades d'accés, pot accedir a l'aplicació. Com que el procés de reserva es realitza per web, i aquesta està muntada d'una forma on es guarden les dades dins una base de dades MySQL, tota la informació dels clients estarà emmagatzemada dintre d'aquesta.

Com es pot suposar, com que totes les dades dels clients es troben dins una base de dades MySQL, serà damunt aquesta per la qual s'han de realitzar les consultes, pel que no és necessari crear-ne un altre.

Així mateix, fer una migració de tota la base de dades a un altre motor no tindria sentit, tenint en compte que aquesta es va actualitzant constantment. Aquest fet comportaria l'obtenció d'una informació desactualitzada en algunes ocasions. A més a més, es tenia una gran experiència amb aquest tipus de base de dades, pel que la programació d'alguna modificació o implementació nova damunt d'aquesta no comportaria molts problemes.

### 4.4. Web Service Extern – API

Per tal d'oferir més serveis per la nostra aplicació, hem necessitat fer ús d'unes determinades API's externes a la nostra organització. D'aquesta forma, hem pogut oferir més serveis pels nostres clients, cosa que farà que es sentin més còmodes i satisfets amb el nostre producte final. La satisfacció del client final és essencial, ja que pot diferir a futures contractacions amb nosaltres en un futur. En aquesta ocasió, s'ha fet servir la famosa API de Google Maps i l'API de Redsys per tal d'oferir una passarel·la de pagament.

### 4.4.1. Google Maps

L'API de Google Maps ens ha servit de molta ajuda, ja que teníem molt clar en els objectius del projecte que volíem oferir una informació valuosa pensat amb el que li podria interessar a l'usuari.

Per això mateix, el que més li pot interessar als turistes a l'hora d'emprendre un viatge és trobar tots els llocs d'interès que es troben prop de la seva ubicació. Per aquesta raó, es varen marcar diversos punts d'interès a l'illa de Mallorca, distribuïts en diferents categories: Bars, restaurants, gasolineres, miradors, parcs d'atraccions, platges i un llarg etcètera. Així mateix, també s'havien de poder distingir uns dels altres de forma visual, pel que també es varen realitzar diferents dissenys gràfics que es situarien segons la seva categoria damunt el mapa.

Com que l'API de Google Maps és molt potent, també es va poder afegir finestres modals, que, al fer 'click' damunt qualsevol dels marcadors, sortís una finestra informativa amb la informació del lloc en concret, tal com el títol del lloc, número de telèfon, direcció, etc.

Tot això és possible, ja que a la base de dades es va afegir una taula, on s'emmagatzema la informació de tots els llocs (la icona que és posiciona damunt el mapa i la informació d'interès segons la categoria del lloc). L'avantatge que té això és que qualsevol lloc nou que s'afegeixi a la base de dades es veuria reflectida al mapa sense necessitat d'haver d'actualitzar l'aplicació, ja que tots els punts es col·loquen de forma dinàmica dependent de la informació que es troba a la base de dades.

A més a més, per tal de personalitzar-ho el màxim possible per l'usuari, aquest mapa s'inicialitza centrant-se a la ubicació de l'hotel que ha contractat, amb un zoom bastant elevat per tal de remarcar els llocs propers a la zona d'allotjament del client. A més a més, es va afegir interacció per JavaScript perquè l'usuari final pugui filtrar segons el tipus de lloc que li



## 4. Implementació

interessi visitar en el moment en el qual accedeix a l'aplicació. D'aquesta forma, amb l'ajuda d'unes icones i uns 'checkbox' pot fer aparèixer o eliminar els punts del mapa que li convingui. Per acabar, també es proporciona un botó per centrar el mapa al lloc de l'hotel fàcilment.

### 4.4.2. Redsys

L'altre API externa de la qual feim ús a l'aplicació és Redsys, per tal de poder fer els pagaments *online* de forma segura pels clients.

Com s'ha dit en tot moment en el present TFG, els usuaris no fan la contractació dels paquets de viatges mitjançant l'aplicació, però si poden contractar altres serveis a partir d'aquesta. Aquests serveis són totes les excursions que s'ofereixen prop de l'allotjament del client, tals com algunes rutes guiades, visites a parcs aquàtics, etc.

Per això mateix, l'aplicació proporciona un llistat de totes les excursions possibles segons la ubicació del client, i permet fer la contractació d'aquestes. Per això hi ha una secció a l'aplicació on l'usuari té un formulari pel qual introdueix les seves dades personals i el nombre d'assistents que volen participar en l'esdeveniment pel qual s'han interessat. Aquest formulari calcula el preu depenent de l'excursió en concret, els participants i l'edat de cada un d'ells. En el moment que volen comprar, simplement es connecta a la passarel·la de pagament Redsys mitjançant el plugin *InAppBrowser* de Cordova, que s'ha esmentat anteriorment per tal de no haver de sortir de l'aplicació. Com que l'aplicació calcula el preu, simplement es connecta a aquesta pàgina de pagament segur connectada directament al compte del banc de l'organització, i permet pagar l'import mitjançant diverses formes de pagament: PayPal, transferència bancària, etc.

### 4.5. Desenvolupament de l'aplicació

Per tal de dur a terme tot el que s'ha explicat durant l'apartat 4, és necessari explicar totes les eines que s'han utilitzat per fer-ho possible

#### 4.5.1. IDE de Programació

Per programar totes les funcionalitats de la pràctica, necessitàvem un [IDE](#) de programació, és a dir, una sistema informàtic per tal de facilitar-nos el desenvolupament de l'aplicació.

Existeixen una infinitat de IDE de programació, pel que teníem moltes opcions possibles. En el nostre cas, es va elegir l'entorn de programació NetBeans [\[23\]](#), ja que es tracta d'un IDE multi llenguatge i multi plataforma. Gràcies a aquestes particularitats, podíem tenir un intèrpret de tots els llenguatges que volíem utilitzar: HTML, JavaScript, CSS i PHP. A més a més, també es podia afegir un *plugin* d'Android, pel que ens era útil en algunes ocasions per fer modificacions del Manifest, per exemple, per canviar la icona de llançament de l'aplicació.

Així i tot, també es va utilitzar, en menor mesura, l'IDE d' Android Studio [\[24\]](#). Gràcies a aquest IDE podíem gestionar de forma molt senzilla l'[SDK](#) d' Android per afegir o eliminar algunes dependències. A més a més, ens era necessari per a generar l'[APK](#) de l'aplicació i poder-la signar correctament.

Per altra banda, també va ésser necessari l'ús de XCode [\[25\]](#), l'IDE propi de iOS. Les raons pel seu ús també recauen per tal de fer el signament de l'aplicació correctament i treure el format propi de les aplicacions iOS. D'aquesta manera es realitzaven les proves damunt un dispositiu real per assegurar el funcionament òptim.

### 4.5.2. Llenguatge de Programació

Pel desenvolupament de l'aplicació, s'han utilitzat diversos llenguatges de programació depenent de la necessitat de la funcionalitat que es volia utilitzar.

Tota la part visual s'ha portat a terme mitjançant els llenguatges HTML5 i CSS3. Amb el llenguatge d'etiquetes HTML, es creen i ordenen tots els components que ha d'incloure cada una de les planes, mentre que s'utilitzava el llenguatge CSS3 per tal d'aplicar els estils damunt aquests components i fer-ho amigable per l'usuari. Cal destacar que amb la compra del Theme esmentat anteriorment, molt d'aquest codi ja es trobava fet, pel que s'havia d'aplicar la correspondència entre els components i les classes de css. Així i tot, també es varen realitzar modificacions damunt aquests arxius per tal d'adaptar-ho exactament amb els objectius visuals. Per aquesta part visual, també s'ha fet ús del Framework Font Awesome, per tal de ser capaços de pintar unes icones predeterminades. Aquest *framework*, molt utilitzat per tota la comunitat desenvolupadora de planes web, és de gran utilitat per crear les metàfores d'algunes accions mitjançant les icones o per fer molt més amigable i comprensible la interfície gràfica d'usuari.

Per dur a terme totes les tasques d'interacció a la part client de l'aplicació, es va fer ús del llenguatge Javascript. Gràcies a aquest, s'ha programat tota la interacció entre els diferents components de la plana web amb l'ajuda del *framework* JQuery.

Mitjançant aquest, es recupera d'una forma més senzilla les diferents entrades d'un formulari i realitzar les comunicacions AJAX amb el Web Service propi de l'organització. Aquests formularis que desemboquen a una trucada AJAX els podem trobar a la plana de *login*, de contacte i de compra d'una excursió.

A més a més, s'utilitza la memòria de client de Javascript de SessionStorage (es destrueix a l'acabar la sessió). Amb aquesta memòria, podem guardar tota la informació que recollim al WebService durant l'inici de l'aplicació i

#### 4. Implementació

utilitzar-la entre les diferents planes. Així, podem utilitzar aquests resultats de forma molt més ràpida, ja que no s'ha d'esperar al fet que es realitzi la comunicació al Web Service en tot moment i esperar a rebre totes les dades. La memòria LocalStorage (no s'elimina mai mentre no es faci manualment), també s'utilitza per saber si s'ha realitzat el *login* amb anterioritat i evitar haver d'iniciar sessió cada vegada que es vulgui accedir a l'aplicació. Així mateix, també es guarda l'idioma per defecte de l'usuari. D'aquesta manera es millora la usabilitat de l'aplicació, evitant haver d'omplir formularis i canviar l'idioma constantment.

Una altra forma d'interacció mitjançant JavaScript que s'ha implementat ha estat la funcionalitat multi idioma que s'ofereix. Com s'ha dit anteriorment, això es realitza mitjançant el *framework* JQuery-Localize, que funciona mitjançant identificadors. Segons l'identificador de cada un dels components i de l'idioma seleccionat, s'introdueix el text i es fa visible a la vista de l'usuari. Aquests textos es troben dintre d'un fitxer JSON, on la clau és l'identificador del component i el valor és el text que ha d'incloure. Per aquesta raó, es necessiten 2 fitxers JSON, un per a cada un dels idiomes i la consulta a un o altre es realitzarà mitjançant el valor a la variable LocalStorage. Aquesta es podrà canviar fàcilment per l'usuari mitjançant unes banderes que es troben dintre del menú.

Finalment, el llenguatge JavaScript s'ha utilitzat per realitzar tota la part dels serveis que estiguin prop de l'estància de l'hotel del client. Això es realitza mitjançant l'API de Google Maps de JavaScript, per la qual es va demanar una clau única d'accés. A més a més, es va definir una interacció mitjançant el *click* a cada un dels punts marcats damunt el mapa, per la qual fa obrir una finestra modal per a obtenir més informació sobre el lloc en concret.

Com a llenguatge a la part servidor, s'ha utilitzat el llenguatge de programació PHP. Per a la programació del BackEnd també teníem moltes

## 4. Implementació

opcions a elegir, tals com JSP [26], Python [27], Ruby [28]. D'aquests llenguatges tan sols es tenia experiència de programació amb JSP, encara que s'havia treballat en major mesura amb PHP. Així mateix es va decidir utilitzar la programació amb PHP sense fer ús de cap *framework*.

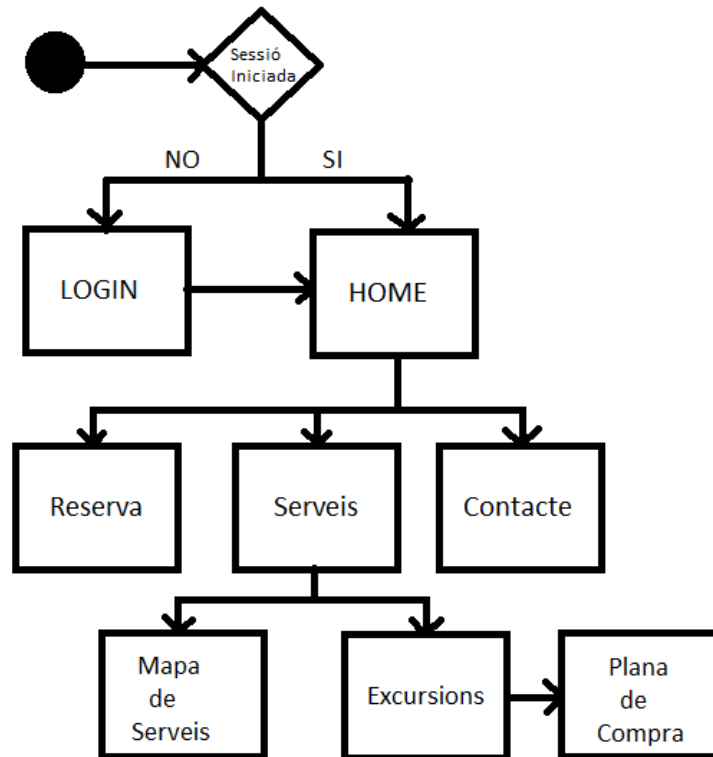
Aquesta programació amb PHP havia de servir com a Web Service per recollir la informació de la base de dades MySQL i retornar-la al programa client, pel que aquest llenguatge era l'òptim a causa de la facilitat de programació que aporta en aquest camp. A més, permet construir una resposta JSON, el llenguatge d'intercanvi de dades que es volia utilitzar degut a la seva rapidesa i lleugeresa.

Finalment, cal fer menció al llenguatge que s'ha utilitzat pel sistema gestor de base de dades MySQL. Aquest ha estat el llenguatge SQL, que s'ha fet servir per fer totes les consultes de selecció dintre de la lògica del programa PHP.

### 4.5.3. Estructura

El desenvolupament de l'aplicació s'ha realitzat seguint l'estructura lògica del seu funcionament. D'aquesta forma, l'esquema de la figura inferior determina el seu funcionament lògic quan s'inicia l'aplicació a qualsevol dels dispositius.

#### 4. Implementació



Imatge 3: Estructura de l'aplicació realitzada

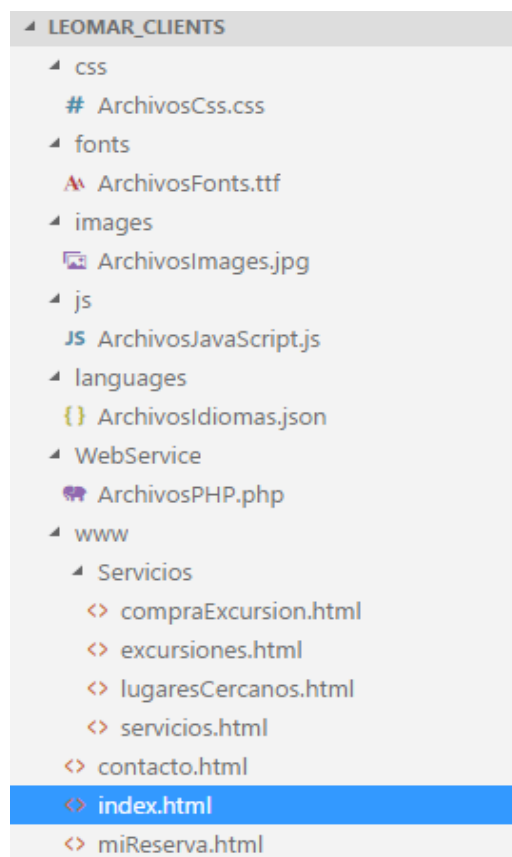
Com es pot observar, l'aplicació comença amb un triangle de decisió, en el qual es mira si la variable guardada dintre de LocalStorage està inicialitzada. Si és així, es fa la consulta al *web service* intern per obtenir totes les dades i es fa una redirecció cap a la plana principal. En cas contrari, s'haurà de passar per una plana de *login*, on l'usuari haurà d'introduir les dades necessàries per identificar-lo, recollir totes les dades i redirigir-lo posteriorment a la *home*. Durant aquest procés de decisió, surt una animació per tal de no mostrar una pantalla en blanc, que podria desembocar a una mala experiència d'usuari, el qual podria pensar que l'aplicació no funciona o que tarda molt a iniciar-se. Aquest és un fet molt important, ja que un estudi ha demostrat que el 53% dels usuaris abandonen aquelles Planes Web o aplicacions que tarden més de 3 segons sense respondre a cap estímul.

Una vegada l'usuari ha arribat a la pàgina principal, podrà accedir a totes les altres planes mitjançant el menú que es troba a la part inferior de la

#### 4. Implementació

pantalla. Totes aquestes planes són accessibles en tot moment, excepte la pàgina de compra, que es pot accedir a ella quan ha seleccionat una de les excursions i ha omplert el formulari corresponent per calcular el preu final de la compra.

Per tal de dur a terme tota aquesta estructura web, s'ha volgut fer una representació gràfica de tots aquests fitxers, pel que es mostra l'estructura de les carpetes dintre del projecte. Aquesta estructura ens permetia tenir tot el codi organitzat i accessible en tot moment. A més a més, aquest tipus d'aplicació pot no deixar de créixer mai, ja que sempre es poden afegir noves funcionalitats. Amb aquesta estructura i un codi ben comentat, qualsevol persona podria reprendre el projecte en qualsevol moment després de veure el funcionament d'aquest.



Imatge 4: Estructura de carpetes del projecte

### 4.5.4. Problemàtiques de desenvolupament i ús de les APIs externes

Per al desenvolupament d'aquesta aplicació han sorgit alguns problemes, exposats durant aquest punt.

La principal problemàtica que vàrem tenir va ser amb la implementació de la passarel·la de pagament de Redsys. En un primer moment, es va realitzar la implementació correctament, però quan es va provar més endavant fallava contínuament. Després de revisar nombroses vegades, es va trucar a la companyia i ens va informar que l'API havia deixat de funcionar per l'encriptació SHA-1 per passar a fer els pagaments més segurs mitjançant l'algoritme SHA-256.

Els canvis entre aquests 2 algoritmes d'encriptació recauen que el sistema de SHA-1 s'utilitza un nombre menor de bits a l'hora de realitzar les transaccions, pel que és molt més susceptible de rebre atacs de diversos pirates informàtics.

En contractar un servei d'una companyia com Redsys, aquest canvi d'algoritme és molt encertat, ja que els sistemes que utilitzen l'encriptació mitjançant codis HASH per SHA-1 ja s'han vist compromesos diversos cops, la primera vegada durant l'any 2005, després que es comprometés la seguretat de l'algoritme MD5

Seguidament, ens varen enviar tota la documentació necessària per dur a terme aquest canvi, pel que es varen realitzar totes les modificacions pertinents perquè tornés a funcionar.

Altres problemàtiques varen sortir a l'hora d'implementar el *web service* propi de l'organització, ja que al trobar-se amb una base de dades tan gran



#### 4. Implementació

amb la qual no s'havia treballat mai, s'havia d'estudiar prèviament tots els passos que s'havien de fer per realitzar les consultes. Per tal de fer-les, es va haver de fer consultes considerablement llargues per obtenir tanta informació sobre la reserva relacionant un gran número de taules. De totes maneres, aquestes consultes valien la pena, ja que oferim una informació adaptada a l'usuari que l'utilitza.

Per fer ús de l'API de Google Maps, es va fer servir la seva documentació oficial que ofereix a la seva plana web, centrant-nos amb l'API de JavaScript. Prèviament, es va demanar un *Access Token* (clau necessària per fer ús de l'API), mitjançant la qual es realitzaven totes les consultes damunt l'API. Així mateix, l'adaptació no va ser trivial, ja que es varen incloure moltes funcionalitats per interactuar amb el mapa (marcadors personalitzats, centrar el mapa a la ubicació del client, poder marcar i desmarcar els punts del mapa, etc).

L'ús de l'API de Redsys és un tema més complicat perquè no és d'ús obert, sinó que s'han de fer algunes accions burocràtiques. Per tal de donar-te d'alta, s'ha de parlar directament amb l'entitat bancària de la qual es forma part. Durant aquesta comunicació, es realitzen una sèrie de preguntes per tal de saber si al banc li serà rentable adaptar aquest pagament per [TPV](#) virtual. Cal dir que, si formes part d'una organització, aquest procés s'agilitza molt. Una vegada s'hagi acceptat la proposta, el banc et proporciona unes claus per tal de poder fer ús d'aquest sistema, sempre en una fase de proves. Durant aquesta fase de proves, es proporcionen uns números de targeta falsos, pel que després de la implementació, has de fer proves per tal d'assegurar el seu correcte funcionament. Una vegada passades aquestes proves i el banc s'ha assegurat de què la implementació és correcta, es proporcionen les dades reals per tal de rebre els futurs pagaments per aquest sistema.

En el moment de la meua incorporació, les primeres accions burocràtiques ja s'havien realitzat, pel que el meu treball va consistir en realitzar

#### 4. Implementació

l'implementació d'aquesta plataforma de pagament i assegurar-ne el correcte funcionament. Actualment, la organització Leomar Viajes ja pot utilitzar aquesta passarel·la de pagament correctament, rebent tots els ingressos de la compra dels clients.

# 5. Metodologia de desenvolupament

Per tal de poder desenvolupar tot el que s'ha explicat en el present TFG, es va seguir una metodologia per poder gestionar el projecte i obtenir un rendiment major.

Cal dir que no es va aplicar al peu de la lletra cap dels sistemes coneguts actualment, sinó que es va fer una adaptació d'un sistema àgil a les necessitats de l'organització en aquell moment.

El primer que ens vàrem qüestionar era la forma de desenvolupament, ja que ens podríem decantar per realitzar una Gestió Tradicional del projecte, o bé implantar un sistema àgil. En aquest cas no va haver molta discussió, ja que el desenvolupament havia de ser creat exclusivament per jo i dirigit pel meu cap. Això significava realitzar el projecte amb 2 persones, un equip molt petit de desenvolupament, on les gestions àgils són molt més eficients que una tradicional amb un sistema de cascada.

Una vegada presa aquesta decisió, havíem d'elegir un sistema de desenvolupament que es coneixen dintre dels sistemes àgils, que són molt

## 5. Metodologia de desenvolupament

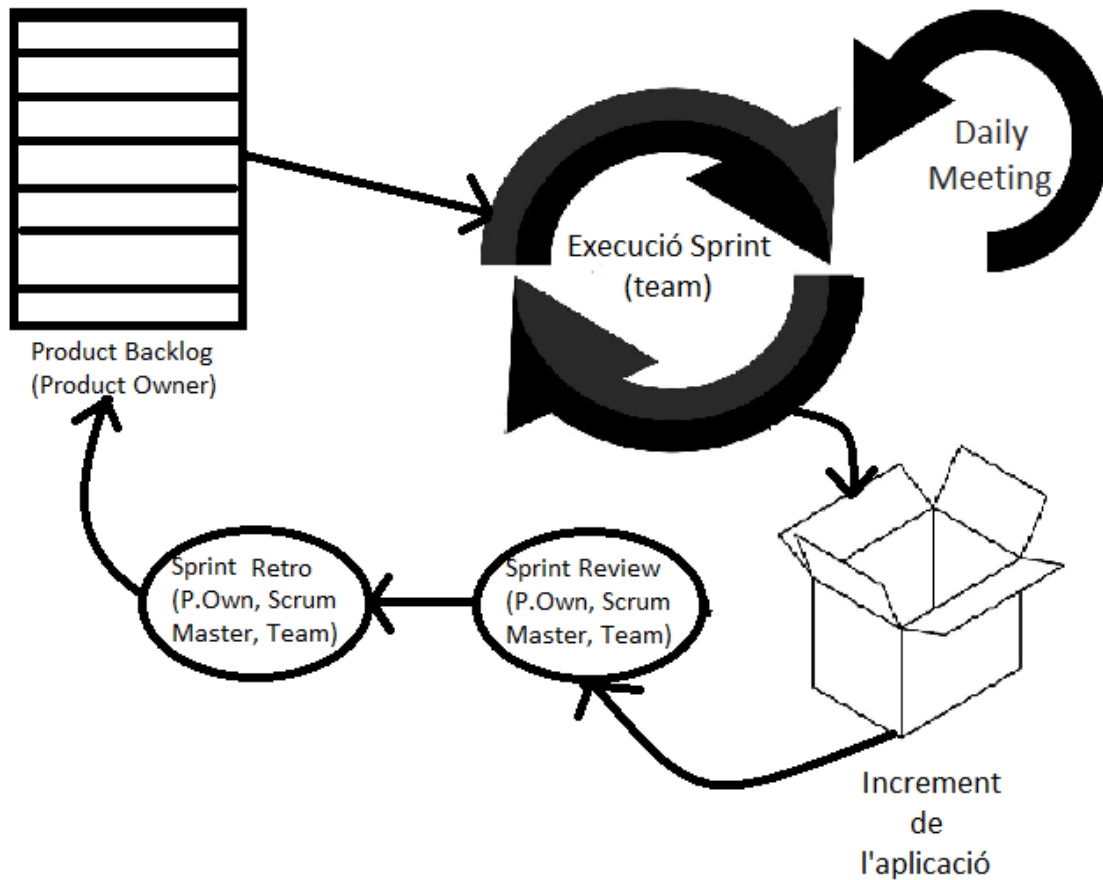
nombrosos. Alguns exemples son SCRUM, XP, DSDM, AUP, etc. Per les necessitats del projecte i pel coneixement que teníem el meu cap i jo ens vàrem decantar per seguir la metodologia SCRUM, adaptant-la a les necessitats del projecte al qual ens enfrontàvem.

El mètode de gestió àgil consta de 3 rols principals: *Scrum Master*, *Product Owner* i el propi Equip.

El *Product Owner* és similar al client. En el nostre cas, es tractava d'un projecte propi, així que el meu cap es va responsabilitzar d'aquest rol. Ell mateix, també va dur a terme el rol de *Scrum Master*, ja que m'ajudaria en tots els problemes que em sorgissin durant el desenvolupament del projecte, eliminant tots els obstacles.

Finalment, jo mateix era l'equip, realitzant totes les tasques d'anàlisi, disseny i programació de la WebApp. Cal dir que l'ideal de l'equip per SCRUM és que aquest mateix sigui auto organitzat, una pràctica que vàrem dur a terme, ja que jo mateix realitzava les tasques com creia necessàries (sempre seguint el *Product Backlog* que s'explicarà a continuació).

## 5. Metodologia de desenvolupament



Imatge 5: Cicle de vida de l'aplicació

L'esquema superior ens mostra el procés que vàrem seguir a l'hora de realitzar l'aplicació. Al principi del projecte, el meu cap em va donar una llista de totes les funcionalitats que havia de tenir l'aplicació, prioritzades segons el seu propi criteri.

Llavors, realitzàvem SPRINTS d'una setmana, en el qual jo mateix m'organitzava per enllestir les funcionalitats que m'havia proposat en el llistat, descomponent-les amb problemes més petits, però sense arribar a fer una documentació amb forma *d'User Stories*.

Cada matí, el meu cap i jo entràvem a treballar alhora, i mentre fèiem les tasques rutinàries, parlàvem de com duia el projecte en una reunió coneguda

## 5. Metodologia de desenvolupament

com a Daily Meeting, on es comentaven els possibles problemes que m'haguessin pogut sorgir el dia anterior. En aquest cas, el meu cap actuava com a *Scrum Master*, i m'intentava donar algunes idees de com podia sortir del problema. A més a més, durant aquest temps li feia una estimació de les tasques que duria a terme en aquell dia, per la qual cosa ell es feia una idea de com anava el projecte.

Quan arribava el divendres a la tarda, una hora abans d'acabar la jornada laboral, compilava l'aplicació i fèiem una *demo* de l'increment aconseguit fins aleshores en un dispositiu real. És important entendre el concepte d'increment, ja que significa que al final de cada SPRINT es presentava una aplicació completament acabada i operativa, millorant cada cop la base obtinguda fins a aquell moment.

Durant aquesta demostració (que es pot fer un símil amb l'Sprint Review), el meu cap (que en aquest cop actuava com a *Product Owner*) valorava l'increment realitzat durant aquesta setmana, i em proposava algunes millores. Aquestes millores es posaven a l'inici del Product Backlog, ja que serien les primeres tasques que es durien a terme durant la següent setmana. Generalment, aquests canvis eren mínims, pel que pràcticament no tenien un impacte en la productivitat de la setmana següent. A més a més, com a metodologia àgil que seguïem, no ens feia por fer cap canvi damunt les tasques plasmades damunt el Product Backlog, pel que si durant aquesta *demo* li sorgien noves idees d'implementació, es valoraven i es col·locaven també dintre del Product Backlog segons la seva prioritat.

Alhora, durant aquest temps, extreia les lliçons que havia après durant aquella setmana, un símil que es té amb l'Sprint Retrospective. En aquest cas, jo tenia un full i anava apuntant tot el que em deia durant aquesta demostració, com per exemple patrons de disseny que no funcionaven (per no fer-los a altes planes).

## 5. Metodologia de desenvolupament

Una vegada es realitzava tot aquest procés, el dilluns al matí es tornava a començar amb aquest cicle, que es va repetir fins al dia de finalització del meu contracte de pràctiques.

Com s'ha pogut veure en aquesta explicació, no es seguia al peu de la lletra totes les accions que es proposen dintre del model àgil SCRUM, encara que si es pot fer un símil amb aquest model.

Des del meu punt de vista, això és un encert, ja que al formar un equip tan petit, vàrem agafar totes les bones pràctiques que es proposaven i es va adaptar el model a les necessitats del projecte, una forma d'optimitzar al màxim el procés de desenvolupament de l'aplicació i de millora de la productivitat.

Seguir aquesta metodologia ens aportava alguns avantatges que no s'haguessin pogut aconseguir si s'hagués fet una gestió tradicional del projecte o si directament no s'hagués gestionat de cap forma.

Gràcies a aquesta, ens centràvem amb les característiques que havia de tenir l'aplicació i tots els canvis eren benvinguts per tal de millorar el projecte propi. A més a més, s'anava aportant valor setmana a setmana, gràcies als increments realitzats i la seva validació amb el cap, cosa que no s'hagués pogut aconseguir si tan sols hagués presentat el producte final un cop acabada l'aplicació, fent que el risc de fracàs del projecte hagués pujat de forma dramàtica.

# 6. Demostració i resultats

Per veure la implementació de totes les funcionalitats que s'han presentat al TFG, seguirem el flux lògic de l'aplicació i les diferents captures de pantalla que fan que l'usuari final pugui gaudir d'elles.

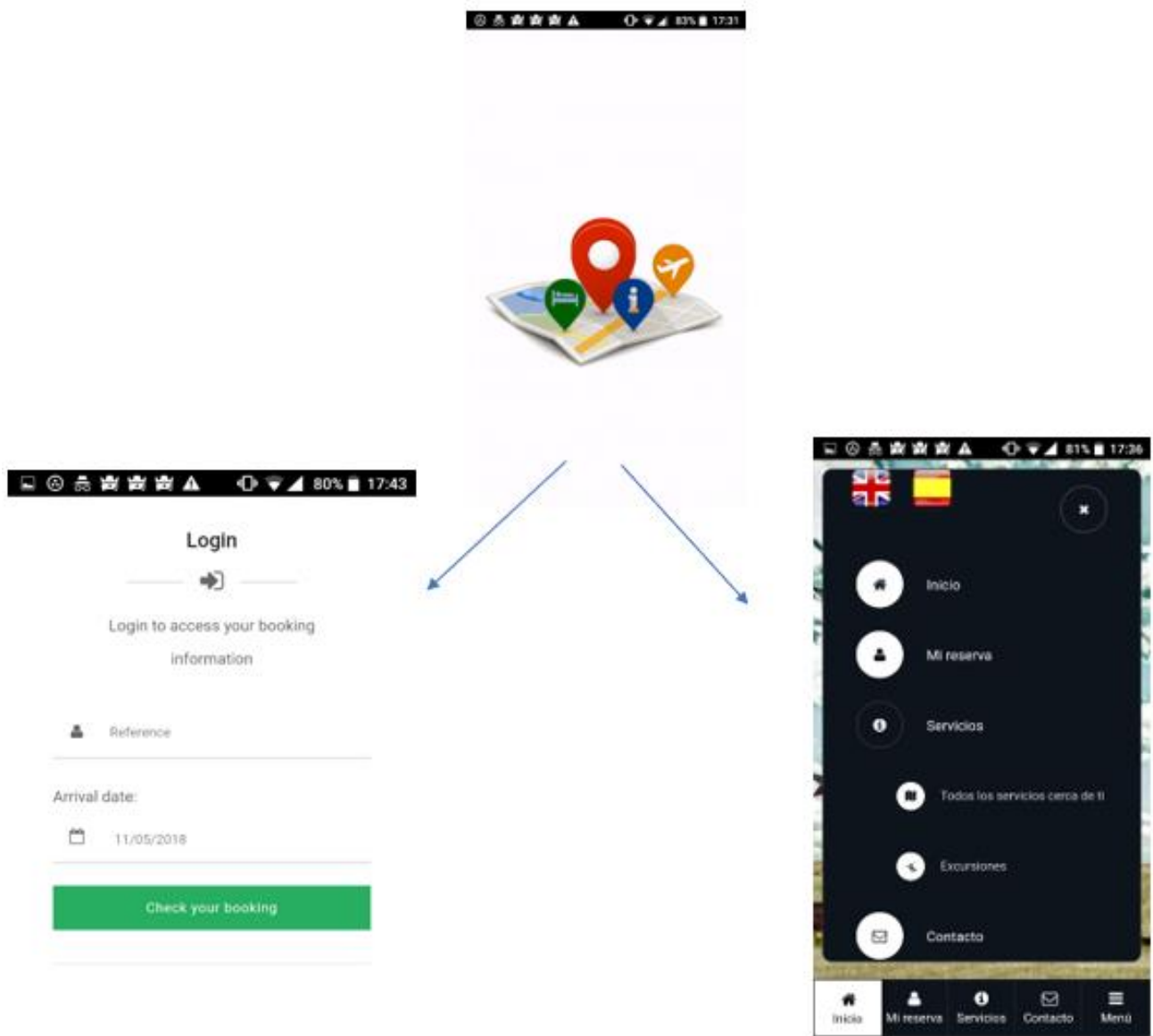
## 6.1. Inici de l'aplicació

Cada vegada que s'inicia l'aplicació, surt una animació d'uns 3 segons, que òbviament no es pot reflectir damunt el paper. Durant aquests segons d'animació, si ja es té la informació d'entrada de l'usuari, el que es fa de forma asíncrona és fer una consulta al Web Service propi de l'organització per agafar tota la informació necessària per a les altres pàgines i, posteriorment, redirigir-se cap a la pàgina principal. A més a més, en tenir la pàgina principal buida, es desplega directament el menú per oferir totes les possibilitats a l'usuari.

Per contra, si encara no ha iniciat sessió, el que es fa és redirigir directament a la pàgina de Login.



## 6. Demostració i resultats



*Imatge 6: Plana de càrrega amb decisió de redirecció al login o a la plana principal*

## 6.2. Plana ‘Mi Reserva’

La pàgina ‘Mi Reserva’ es divideix en dues subseccions per tal d’oferir tota la informació de la reserva de forma ordenada. Com es pot observar, l’accés a la plana és quasi immediat, ja que les dades ja s’han guardat durant l’animació (o després d’haver fet el login) i no fa falta consultar-les cada vegada que s’accedeix a aquesta.

La primera és anomenada Resum, i conté tot la informació referent a l’allotjament en el qual s’ha ubicat. En aquesta secció apareix la referència, les dates que es quedarà a l’hotel, l’hotel en qüestió (juntament amb un enllaç a la seva pàgina de fotografies que s’obrirà amb el plugin inAppBrowser esmentat anteriorment), la direcció i el número a telèfon. Acte seguit, sortiran tantes habitacions com s’hagi concretat en la reserva, definint el tipus i les persones que les ocuparan. A més a més, per a cada una de les persones també es pot veure el tipus d’allotjament (mitja pensió, tot inclòs, etc).

(veure imatge 7)

La segona part d’aquesta plana és la informació referent als vols que ha d’agafar els integrants del grup de la reserva.

En aquesta secció, amb l’ajuda de les icones de ‘Font Awesome’, apareix la informació del vol d’anada i de tornada (dia, hora i número de vol). A més a més, si tenen contractats serveis de transports, apareixerà l’hora que es

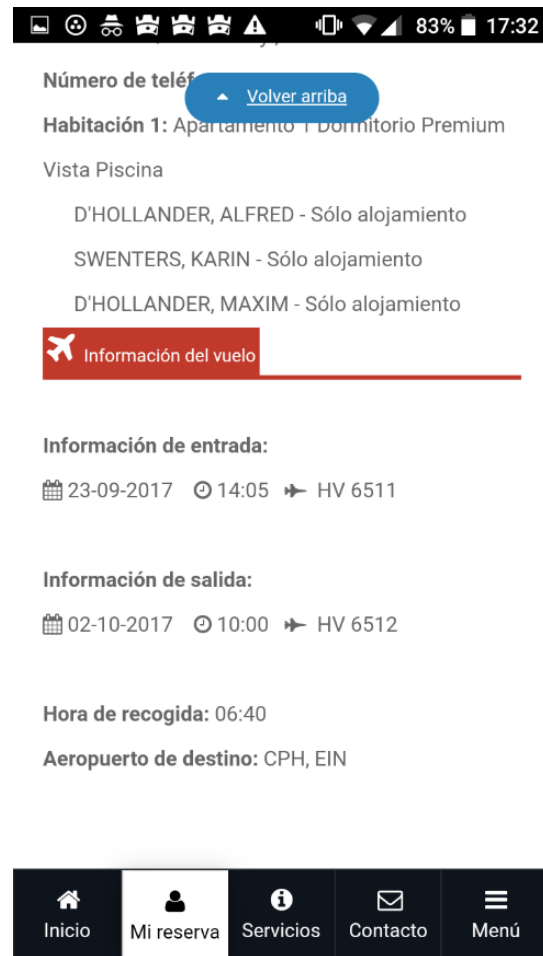
## 6. Demostració i resultats

passarà a buscar tots els membres del grup, juntament amb l'aeroport que es visitarà. (veure imatge 8)



Imatge 7: Plana d'informació de la reserva de l'usuari

## 6. Demostració i resultats



Imatge 8: Plana d'informació de vols

### 6.3. Plana 'Servicios'

La secció de serveis es subdivideix en dues grans planes, la primera (que s'obra per defecte si es fa click al menú inferior) és la dels llocs d'interès prop de la ubicació del turista, mentre que la segons plana és la de la visualització de les excursions disponibles a la zona del client.

#### 6.3.1. Llocs d'interès prop de la ubicació del turista

Una vegada s'entra a aquesta pàgina, es fa una petició a l'API de Google Maps, i s'obté el mapa del lloc on està ubicat el turista. Tot seguit,

## 6. Demostració i resultats

s'incorporen tots els punts d'interès que es troben a aquella zona en concret i s'afegeix amb un marcador més gran l'hotel al qual s'ha allotjat el client.



Imatge 9: Plana de serveis prop de la ubicació del client

A la imatge superior es pot observar tot el que s'ha esmentat a la [secció 4.4.1](#): El disseny gràfic varia segons el tipus de servei que s'ofereix, podent diferenciar diferents categories (tals com restaurants, miradors, lloguer de cotxes, supermercats, etc). A més a més, si és pitja el botó de filtratge, apareixen totes aquestes opcions juntament amb un 'checkbox', el que permet eliminar o afegir els punts que l'usuari cregui oportú. A més, si es pitja damunt qualsevol d'aquests apareixerà la informació pertinent al lloc. El botó que es troba a la part superior dreta serveix per ubicar el mapa al centre de l'hotel, per tal de millorar la seva experiència d'usuari.

### 6.3.2. Excursions

Per tal de dur a terme la contractació de les excursions, se li aporta a l'usuari un llistat de totes les que es troben per la zona a la qual s'ha allotjat, una informació que s'ha agafat durant l'inici de la sessió a l'aplicació. En aquesta llista apareix el títol de l'excursió, la seva imatge principal, una petita descripció, un enllaç per veure més informació de l'excursió i un enllaç directe a la pàgina de compra. Si es pitja sobre el títol, la imatge o l'enllaç de més informació, es redirigirà a l'usuari a una pàgina amb una descripció més extensa per poder tenir molta més informació sobre l'excursió seleccionada.



*Imatge 10: Llistat de les excursions pròximes al client*

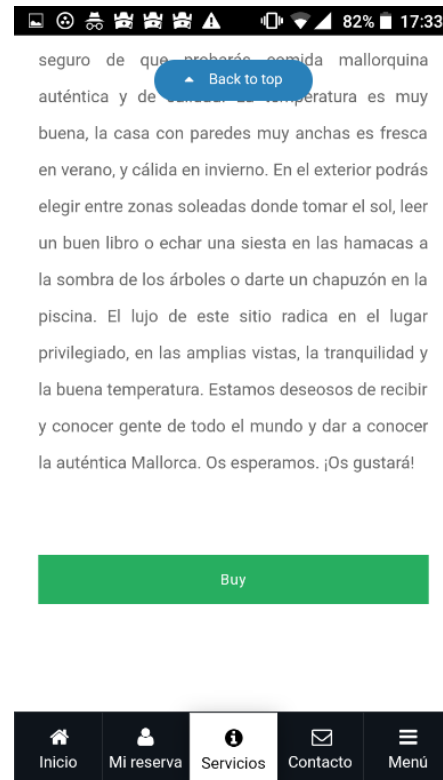
Una vegada s'entri a la pàgina de l'excursió en concret, apareixerà un descripció molt més extensa d'aquesta, amb tot el que es podrà fer a aquest

## 6. Demostració i resultats

event. A més a més, a la part superior de la pàgina sortiran totes les fotos que es tinguin disponibles sobre l'excursió en forma de *slider*. A la part inferior de la pàgina es trobarà un botó per tal de comprar l'excursió, el que desembocarà a una nova pàgina de compra.



*Imatge 12: Informació detallada de l'excursió elegida*



*Imatge 11: Botó de compra de l'excursió*

Acte seguit, si es decideix comprar mitjançant el click en aquest darrer botó de compra o des de la llista, es redirigirà a una pàgina que conté un formulari per tal de fer efectiva la contractació de l'excursió.

En aquest formulari, s'omple automàticament el nom de la persona qui la fa, sent aquest el nom del reservant del paquet de vacances. També apareix un calendari per introduir les dates del dia desitjat per realitzar l'excursió. Òbviament, no tots els dies de l'any estan disponibles totes les excursions, pel que baix del formulari es pot observar un avís de si l'excursió està

## 6. Demostració i resultats

disponible o no el dia seleccionat. En cas de que no estigui disponible, el botó de compra es desactiva automàticament per evitar pagaments erronis.

El formulari també inclou el correu electrònic, ja que després d'efectuar el pagament, s'enviarà un correu amb tota la informació de l'excursió i l'entrada corresponent.

Finalment, apareixen 2 *dropdowns* per triar el número d'integrants que assistiran a l'excursió, dividint-los entre adults i nins.

Amb tota aquesta informació recol·lectada, el preu es va variant segons les dades que introdueix l'usuari, perquè sàpiga en tot moment el que li costarà la contractació del servei.



The screenshot shows a mobile application interface for purchasing an excursion. At the top, the user's name is 'D'HOLLANDER, ALFRED'. Below this, there are input fields for 'Fecha:' (03/07/2017) and 'Mail:' (Exemple@mail.com). A progress indicator is visible. Underneath, there are two dropdown menus for 'Adultos' (set to 3) and 'Niños' (set to 0). The price is displayed as 'Precio: 204.00'. A green confirmation box states: 'Puedes reservar la excursión con al información facilitada'. At the bottom, there is a navigation bar with icons for 'Inicio', 'Mi reserva', 'Servicios', 'Contacto', and 'Menú'.

Imatge 13: Formulari de compra d'una excursió



The screenshot shows the payment screen of the application. At the top, the amount '204,00 €' is displayed. Below this, there are two payment options: 'Pagar con Tarjeta' (with logos for VISA, MasterCard, and American Express) and 'Pagar con iupay!'. The 'Pagar con Tarjeta' option is selected, showing input fields for 'Nº Tarjeta:', 'Caducidad:' (mm and aa), and 'Cód. Seguridad:'. There are 'Cancelar' and 'Pagar' buttons. Below the payment options, there is a section titled 'Datos de la operación' with the following details: 'Importe: 204,00 €' and 'Comercio: LEOMAR VIAJES, S.A. (ESPAÑA)'.

Imatge 14: Plana de pagament d'una excursió



## 6. Demostració i resultats

Finalment, per acabar amb la secció de l'excursió, entra en joc la par de la consulta a l'API externa de Redsys.

Una vegada l'usuari ha introduït totes les dades al formulari de la plana anterior i pitja comprar, s'obra una pàgina mitjançant el plugin InAppBrowser, passant la quantitat calculada com a paràmetre. En aquesta pàgina, l'usuari ha d'introduir les dades bancàries per tal d'efectuar la compra i poder assistir a l'excursió.

Una vegada s'hagi validat la compra, es tancarà el navegador extern notificant de l'èxit de la compra, juntament amb un avís de revisió del seu correu electrònic per rebre tota la informació. En tot cas, també es pot cancel·lar la compra, el que farà que es redirigeixi a la pàgina de formulari novament.

# 7. Conclusions

La principal de les conclusions és que s'ha assolit l'objectiu fonamental que ens havíem marcat. Cal tenir en compte que el desenvolupament d'una aplicació en el sector de viatges és una tasca complicada i laboriosa, a més que s'ha d'aconseguit un resultat final destacable, per tal de despuntar i destacar-se sobre la competència, donat l'alt nivell competitiu que existeix en aquest sector dins el món de la Internet.

Una bona forma de fer-ho és amb la realització d'un producte propi i únic, en el que tan sols les persones que contracten amb l'organització la poden utilitzar. D'aquesta forma, puja la satisfacció d'aquests i és més fàcil que aquest es fidelitzi amb la nostra empresa. Si el producte és molt bo, li aporta una utilitat i el client es sent còmode amb el paquet de vacances que ha contractat, és molt més probable que en la seva següent reserva pensi amb la nostra organització.

Totes les funcionalitats desenvolupades s'han realitzat sempre de cara a la utilitat dels clients, intentant treure totes les funcions òptimes que li agradaria tenir per tal d'emprendre un nou viatge.

Per aquest motiu, s'aporta tota la informació sobre la reserva de l'usuari, amb els horaris i direccions de cada una de les activitats que ha contractat.

## 7. Conclusions

A més a més, aportar un mapa on es centra automàticament a la ubicació de la reserva de l'hotel i es mostren tots els llocs d'interès que tenen al voltant, pot ser una eina molt útil pels clients de la nostra organització. Dintre de l'aplicació també s'ofereix la possibilitat d'ampliar les activitats contractades, prioritzant també per la proximitat totes les excursions disponibles. Finalment, una forma de contactar amb l'organització amb tot moment o amb els guies de les excursions més propers fa que es tanqui una aplicació molt útil de cara als clients.

Per l'altra banda, s'han de fer algunes modificacions al projecte per tal de fer el desplegament d'aquesta, encara que a la part del codi no s'ha de fer res més. Les qüestions que queden pendents són insercions a la base de dades per tal d'afegir les descripcions i fotografies de les excursions (per mostrar el producte presentable s'ha posat una fotografia per defecte i una descripció amb un text aleatori *Lorem Ipsum* en els casos que no es tinguin les reals). Un altra informació que s'ha d'afegir a la base de dades són més localitzacions dels llocs d'interès, per tal de cobrir tots els hotels que es poden contractar amb la companyia. L'avantatge de tot això és que una cada una d'aquestes modificacions que es faixin a la base de dades es veurà reflectida directament damunt l'aplicació, sense necessitat de compilar de nou el codi i extreure un nou APK.

Cal dir que les pràctiques realitzades en aquesta companyia m'han estat de gran utilitat per tal de millorar les meves capacitats de programació. Una d'elles ha estat la soltesa de programació amb HTML, JavaScript i CSS, uns llenguatges de programació molt importants en l'actualitat, ja que les pàgines web estan sent el producte estrella del present i possiblement ho seguirà sent durant molts d'anys.

A més a més, a títol personal, he descobert la programació de les aplicacions híbrides. Una tecnologia desconeguda fins aleshores. Actualment, aquestes tecnologies són molt importants i molt utilitzades en l'àmbit personal i

## 7. Conclusions

professional, pel que se m'ha presentat una molt bona oportunitat de creixement personal.

De fet, el coneixement d'aquesta tecnologia m'ha servit de molta ajuda per treure les meves primeres aplicacions en el mercat, convertint-se amb els meus primers projectes personals. Aquestes aplicacions s'han desenvolupat amb l'eina que la qual s'ha desenvolupat l'aplicació del present TFG: Cordova. De totes maneres, m'ha fet veure un món més enllà per començar a aprendre noves tecnologies d'aplicacions híbrides, ja que m'ha permès explorar altres alternatives i començar uns nous projectes amb Ionic i React Native [\[29\]](#).

Adicionalment, la realització d'aquest projecte també m'ha servit per consolidar tots els coneixements de PHP, a més a més d'altres conceptes com la creació de Web Service amb aquest mateix llenguatge de programació. A més a més, també s'ha millorat el coneixement sobre els llenguatges de transmissió de dades entre el programa client i el servidor, en aquest cas JSON. Gràcies a aquesta programació, s'ha après a crear aquests tipus d'objecte i a recuperar les dades d'un objecte JSON donat, tant dels creats en el propi Web Service com de les Apis externes.

Finalment, una altra experiència molt positiva de la realització d'aquest projecte ha estat posar en pràctica per primera vegada a un projecte real una metodologia àgil, una cosa que tant sols s'havia vist de forma teòrica i amb una simulació a una assignatura de la Universitat.

Aquest fet m'ha permès ser més crític enfront de la gestió tradicional dels projectes, confirmant el que ja pensava a títol personal: Una gestió àgil ben realitzada pot ser molt més efectiva per realitzar un projecte, pujant la satisfacció del client amb el producte final gràcies a la revisió continua del producte final.

# Referències

- [1] Aeropuertos Españoles y Navegación Aérea “Aena”, <http://www.aena.es/es/pasajeros/pasajeros.html>
- [2] Minube, <https://www.minube.com/>
- [3] Tripadvisor, <https://www.tripadvisor.es/>
- [4] AroundMe, <http://www.aroundmeapp.com/>
- [5] Prava, <http://pravaapp.com/>
- [6] Mtrip, <https://www.mtrip.com/es/>
- [7] Google Trips, <https://get.google.com/trips/>
- [8] Google Maps, <https://www.google.es/maps/>
- [9] LocalScope, <https://cynapse.com/localscope/>
- [10] izi.TRAVEL, <https://izi.travel/es>
- [11] Plana oficial de Cordova, <https://cordova.apache.org/>
- [12] Plana oficial de Ionic, <https://ionicframework.com/>
- [13] Plana oficial de Unity, <https://unity3d.com/es>
- [14] Plana oficial de Xamarin, <https://www.xamarin.com/>
- [15] ThemeForest, <https://themeforest.net/>
- [16] Theme MyMobile utilitzat per l’aplicació, [https://themeforest.net/item/mymobile-mobile-tablet-responsive-template/17110664?s\\_rank=1](https://themeforest.net/item/mymobile-mobile-tablet-responsive-template/17110664?s_rank=1)
- [17] JQuery-localize, <https://github.com/coderifous/jquery-localize>

- [18] Font de Google Roboto, <https://fonts.google.com/specimen/Roboto>
- [19] Font de Google Satisfy, <https://fonts.google.com/specimen/Satisfy>
- [20] Font Awesome, <https://fontawesome.com/>
- [21] Plugin inAppBrowser de Cordova,  
<https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-inappbrowser/>
- [22] Redsys, <http://www.redsys.es/>
- [23] Entorn de Desenvolupament Netbeans,  
<https://netbeans.org/projects/www/>
- [24] Entorn de Desenvolupament Android Studio,  
<https://developer.android.com/studio/?hl=es>
- [25] Entorn de Desenvolupament XCode,  
<https://developer.apple.com/xcode/>
- [26] Plana oficial de JSP, <http://www.oracle.com/technetwork/java/jsp-138432.html>
- [27] Plana oficial de Python, <https://www.python.org/>
- [28] Plana oficial de Ruby, <https://www.ruby-lang.org/es/>
- [29] Plana oficial de React Native, <https://facebook.github.io/react-native/>