

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria Informàtica

WHERE ARE YOU GOING?

APLICACIÓ PER A DISPOSITIUS MÒBILS AMB PLATAFORMA ANDROID
QUE PERMET PUBLICAR I CERCAR TRAJECTES

Alumne/a: Josep Torras Font

Dirigit per: Anna Muñoz Bolas

CURS 2013/14 (Febrer)

Resum

Aquest projecte, té com a objectiu principal desenvolupar les capacitacions necessàries per a poder construir aplicacions mòbils fent ús de bases de dades en plataformes de *cloud computing* (computació en el núvol), i sobre un sistema operatiu Android. Per això, els objectius que es persegueixen són els d'entendre la metodologia per a poder implementar una base de dades en el *cloud*, capacitat per a construir una base de dades que s'adeqüi al projecte i comprendre de quina manera els dispositius mòbils poden interactuar amb les dades i serveis que ofereix la computació en el núvol (*cloud computing*). Els objectius descrits, exigiran un coneixement a fons del sistema operatiu Android, les seves característiques i també les llibreries i recursos que fan falta per a poder desenvolupar aplicacions mòbils amb computació en el *cloud*. Particularment, es farà necessari un coneixement addicional que permeti treballar amb OpenStreetMap per plataforma Android. A partir d'aquí s'aniran exposant totes les tasques que es tindran que realitzar, des de la fase de recollida de requeriments, l'especificació, el disseny, la instal·lació del programari de desenvolupament necessari fins la construcció del producte. Tot plegat ha de permetre desenvolupar una aplicació que les seves funcionalitats principals, a grans trets, seran les de poder publicar un trajecte amb un origen, un destinació i dades addicionals, cercar trajectes per origen, destinació i opcionalment data, i poder visualitzar els trajectes trobats junt amb les seves dades, amb les opcions de visualitzar-los en un mapa i fer una reserva.

Índex

1.- Descripció	1
1.1.- Introducció	1
1.2.- Justificació	1
1.3.- Objectius	2
1.4.- Abast	2
1.5.- Metodologia de desenvolupament del projecte	3
1.6. - Productes finals	3
2.- Anàlisi de recursos	4
2.1. – OpenStreetMap	4
2.1.1. – Com exportar dades d’OSM	5
2.1.2. – Format del fitxer OSM XML	6
2.1.3. – Com exportar les dades a diferents formats	8
2.2. –CloudMade	9
2.2.1. – Què és CloudMade?	9
2.2.2. – Routing HTTP API docs	9
2.2.3. – GeoCoding v3 beta Engine	15
2.2.4. – GeoCoding v3 beta Server API	17
2.2.5. – Tiles HTTP API	19
2.2.6. – Resum i conclusions	21
2.3. –Nominatim	21
2.3.1. –Paraules clau especials	22
2.3.2. –Paràmetres	22
2.3.3 –Conclusions	24
2.4. –MapsForge	25
2.4.1 –Disseny i característiques	25
2.4.2 –Presentació de mapa a Android	26
2.4.3 –Conclusions	28
3.-Model i disseny	29
3.1. – Arquitectura del programari	29
3.1.1.– Tipus d’arquitectura de l’aplicació	29
3.1.2.– Identificació dels mòduls o components	29
3.2. – Funcionalitats	30
3.3. – Anàlisi	31
3.3.1. – Diagrama de casos d’ús	31
3.3.2. – Especificació dels casos d’ús	32
3.3.3. – Model conceptual de dades	36
3.3.4. – Restriccions d’integritat	37
3.3.5. – Diagrama de seqüència	37
3.3.6. – Anàlisi de pantalles	38

3.3.6.1. – Pantalla d’inici (usuari)	38
3.3.6.2. – Pantalla login	38
3.3.6.3. – Pantalla registre	39
3.3.6.4. – Pantalla opcions per usuari registrat	39
3.3.6.5. – Pantalla publicar trajecte	40
3.3.6.6. – Pantalla llistar trajectes	40
3.3.6.7. – Pantalla esborrar trajecte	41
3.3.6.8. – Pantalla modificar trajecte	41
3.3.6.9. – Pantalla cercar trajecte	42
3.3.6.10. – Pantalla veure trajecte seleccionat	42
3.3.6.11. – Pantalla fer reserva	43
3.3.6.12. – Pantalla veure mapa	43
3.3.6.13. – Pantalla comentari conductor	44
3.3.6.14. – Pantalla veure comentaris de conductor	44
4.-Instal·lació del programari	45
4.1. – Recursos de maquinari	45
4.2.- Recursos de programari	45
4.3.- Instal·lació del programari	46
4.3.1. – Instal·lació de la màquina virtual i sistema operatiu	46
4.3.2. – Instal·lació de la IDE Eclipse	49
4.3.3. – Instal·lació del SDK Android	50
3.3.4. – Instal·lació del JDK de Java	51
3.3.5. – Instal·lació de MySQL i MySQL-Query - Browser	51
3.3.6. – Instal·lació de Tomcat	56
5.- Desenvolupament del programari	58
5.1. – Descripció. Integració de l’SDK Android i Tomcat a Eclipse	58
5.1.1. – Integració de l’SDK Android	58
5.1.2. – Integració de Apache Tomcat	60
5.2. – Decisions finals sobre el disseny i eines	61
5.3. – Construcció de la base de dades	61
5.4. – Construcció del Web Service	65
5.4.1 – Algorisme de cerca de trajectes	68
5.5 – Desenvolupament aplicació per plataforma Android	69
6.- Aplicació i funcions	71
6.1 – Funcionament de l’aplicació	71
6.2– Futures millores	78
7.- Resultat final i conclusions	80
7.1 – Resultat final	80
7.2 – Conclusions	81
8.- Referències digitals	82

Índex de figures

Figura 1. Exemple d'OpenStreetMap	4
Figura 2. Exportar dades des del mapa directament.	6
Figura 3. Exemple arxiu OSM XML.	7
Figura 4. Exemple tile 1.	20
Figura 5. Exemple tile 2.	20
Figura 6. Esquema components aplicació en una arquitectura client – servidor orientada a serveis.	30
Figura 7. Diagrama de casos d'ús.	32
Figura 8. Model conceptual de dades.	36
Figura 9. Diagrama de seqüència.	37
Figura 10. Inici.	38
Figura 11. Login	38
Figura 12. Registre.	39
Figura 13. Opcions usuari	39
Figura 14. Publicar trajecte.	40
Figura 15. Llistar trajectes.	40
Figura 16. Esborrar.	41
Figura 17. Modificar trajecte.	41
Figura 18. Cercar trajecte.	42
Figura 19. Trajecte seleccionat.	42
Figura 20. Fer reserva.	43
Figura 21. Veure mapa.	43
Figura 22. Comentari a conductor.	44
Figura 23. Llistat comentaris	44
Figura 24. Selecció crear nova màquina virtual	46
Figura 25. Selecció ubicació sistema operatiu	47
Figura 26. Selecció tipus S.O. i versió	47
Figura 27. Dades per a l'instal·lació del S.O.	48
Figura 28. Selecció espai de disc.	48
Figura 29. Selecció recursos hardware per la màquina virtual	49
Figura 30. Detall carpeta descomprimida amb Eclipse	50
Figura 31. Administrador de paquets d'Android	50
Figura 32. Instal·lació del JDK	51
Figura 33. Inici instal·lació MySQL	52
Figura 34. Selecció tipus d'instal·lació	52
Figura 35. Selecció detall instal·lació	53
Figura 36. Selecció tipus server	53
Figura 37. Tipus d'ús de la base de dades	53
Figura 38. Configuració de port.	54
Figura 39. Configuració notació	54
Figura 40. Configuració contrasenya	54
Figura 41. Base de dades configurada	55
Figura 42. Instal·lació MySQL Query Browser	55
Figura 43. Selecció ruta instal·lació	55
Figura 44. Selecció tipus instal·lació	56
Figura 45. Selecció components Tomcat	56
Figura 46. Selecció de ports del Tomcat	57
Figura 47. Afegim ruta del repositori	58

<i>Figura 48. Elements per a descarregar</i>	59
<i>Figura 49. Software a instal·lar</i>	59
<i>Figura 50. SDK d'Android</i>	60
<i>Figura 51. Integració de Tomcat</i>	60
<i>Figura 52. Relació operacions – dades.</i>	64
<i>Figura 53. Esquema conceptual MVC</i>	65
<i>Figura 54. Inici</i>	71
<i>Figura 55. Registre</i>	72
<i>Figura 56. Login</i>	72
<i>Figura 57. Opcions usuari</i>	72
<i>Figura 58. Publicar Trajecte</i>	73
<i>Figura 59. Llistat trajectes</i>	73
<i>Figura 60. Opcions trajecte</i>	73
<i>Figura 61. Modificar trajecte</i>	74
<i>Figura 62. Esborrar trajecte</i>	74
<i>Figura 63. Cercar trajecte</i>	74
<i>Figura 64. Trajecte cercat</i>	75
<i>Figura 65. Trajecte seleccionat</i>	75
<i>Figura 66. Comentaris a conductor</i>	76
<i>Figura 67. Reserva</i>	76
<i>Figura 68. Vista mapa</i>	77
<i>Figura 69. Comentari a conductor</i>	77
<i>Figura 70. Comentari a conductor</i>	78

1.- Descripció

1.1.- Introducció

A l'hora de posar en marxa aquest projecte, un dels factors més importants que influiran en el seu èxit o fracàs, serà el de tenir clar quin és el seu abast amb els termes de que ha de solucionar i per tant que s'ha de fer i que no s'ha de fer. Llavors un cop establerta aquesta base, cal marcar uns objectius a assolir, i que són els que ens portaran a l'obtenció del producte final, en aquest cas, el programari de publicació de trajectes per part d'usuaris productors i cercadors d'aquests per part d'usuaris consumidors. Un altre punt fonamental és el de la planificació: aquest projecte, com tots els projectes, necessitarà un recurs molt valuós que és el temps. També com a tots els projectes té una temporització, és a dir, una data d'inici i una de finalització que serà quan ha d'estar finalitzat i es té que lliurar el producte. Per a poder complir amb el termini cal una bona planificació. Per això serà necessari en el seu disseny identificar tant els punts forts com les febleses, per a tenir-ho en compte.

El nivell de qualitat exigible no afectarà solsament a la construcció del programari sinó també a tota la documentació que l'ha d'acompanyar, la comunicació, els controls de canvis, etc. Per tant podem pensar en aquest projecte com alguna cosa més complexa que la simple codificació del programari amb els llenguatges de programació Android i Java que seran els utilitzats. En aquest apartat es tracten tots aquests elements i punts que s'han considerat, que són la base d'aquest projecte (i de qualsevol altre), i que ho emmarquem amb el títol de planificació.

1.2.- Justificació

Actualment, i potser accelerat per aquesta crisi econòmica que patim, la societat es comença a organitzar per tal de treure profit dels recursos disponibles amb el benefici del estalvi. La tecnologia existent ha permès un important canvi en la comunicació entre les persones i una interconnexió global i permanent. Així, una de les principals conseqüències importants que això ha aportat la podem definir en una paraula: compartir. Entre les innumerables coses que es comparteixen avui en dia per la xarxa, la informació té una especial rellevància. Aquesta, està repartida arreu del món, a l'abast de tothom i disponible en tot moment. Enllaçant els termes compartir i informació podem trobar infinitat de portals que ofereixen serveis fonamentats en aquests dos pilars. Entre aquests, ni ha un que va sorgir no fa massa temps i que en pot ser un bon representant: compartir trajectes en cotxe quant aquest té places disponibles. La informació, llavors, ja flueix per i des dels propis interessats. Podem veure alguns portals que ofereixen aquests tipus de servei en el qual les persones disposades a compartir places de cotxe les ofereixen a possibles consumidors, tenint com a resultat que tots en surten beneficiats. Aquest tipus de servei està disponible en format web però avui en dia tothom té molt present que els dispositius mòbils han guanyat molt de terreny i són d'ús quotidià per un nombre cada cop més gran de persones. Llavors el repte ha estat en intentar adaptar aquest servei com aplicació per a dispositius mòbils amb plataforma Android (que és la més utilitzada actualment).

1.3.- Objectius

Els objectius que es persegueixen i que es tenen d'assolir en finalitzar el projecte són:

- Haver adquirit coneixements sobre la tecnologia SIG.
- Entendre la metodologia utilitzada per implementar una base de dades en plataforma cloud computing.
- Capacitat per a construir una base de dades que sigui apropiada al projecte desenvolupat.
- Conèixer i comprendre els mecanismes que tenen a disposició els dispositius mòbils per a poder accedir a les dades i serveis presents en el cloud.
- Assolir uns coneixements sòlids del sistema operatiu Android.
- Adquirir bons coneixements de les llibreries i recursos necessaris per a poder desenvolupar aplicacions mòbils amb computació en el cloud.
- Treballar amb programari de codi obert en plataforma Android.

1.4.- Abast

Podem considerar els següents punts com línies base que engloben el projecte:

- Fer un repàs general del glossari de nomenclatura i conceptes SIG .
- Estudi, comprensió i valoració de punts forts i dèbils en l'ús de cartoDB pel desenvolupament de l'aplicació.
- Cerca d'informació i visió global de l'estructura d'OpenStreetMap per utilitzar-ho com a font de dades geogràfiques.
- Estudi de la llibreria Leaflet per veure què ofereix per a poder representar cartografia.
- Estudi i recerca d'altres recursos que poden esdevenir una oportunitat per al desenvolupament de l'aplicació
- Cerca i estudi de l'entorn de desenvolupament d'aplicacions per la plataforma Android i també de les seves API.
- Desenvolupament del producte programari.
- Redacció de la memòria.
- Elaboració de la presentació multimèdia.
- Debat final virtual.

1.5.- Metodologia de desenvolupament del projecte

En aquest projecte les tasques a desenvolupar les podem emmarcar amb dues categories inicials: una primera que és l'adquisició dels coneixements necessaris per a la construcció del producte final, i l'altre és la pròpia tasca de construcció. Aquestes tasques però no es portaran a terme d'una manera seqüencial sinó que s'aniran alternant segons l'estat en el qual es trobi el projecte. Aquest fet ve condicionat pel tipus de mètode que decidim seguir pel desenvolupament i que el podem definir com de metodologia àgil. Això significa que es desenvoluparà en primer terme unes funcionalitats concretes del disseny, es provaran, s'analitzaran i es modificaran si és el cas, repetint aquestes iteracions fins a obtenir el producte final.

1.6. - Productes finals

Un cop finalitzat el projecte, disposarem del següent material:

- Aquest document que és la memòria del projecte.
- Una presentació virtual d'aquest.
- Codi del programa desenvolupat. Documentació relacionada amb el programari.

2.- Anàlisi de recursos

En aquesta secció i en els pròxims apartats, es farà una descripció d'uns quants recursos que tenim a l'abast i que han de permetre de desenvolupar el producte. Faig notar i remarco l'expressió uns quants perquè ni ha una extensa varietat i per tant, el seu anàlisi seria una tasca molt extensa, quedant fora de l'abast d'aquest treball. Per tant, s'avaluaran les eines que a criteri propi, podent ser més adients i s'ajusten als objectius per finalment, seleccionar les que poden ser òptimes. Llavors, les eines que s'analitzaran seran les següents: OpenStreetMap, CloudMade, OSMDroid i MapsForge (entre d'altres que finalment no s'utilitzaran i que s'han analitzat com Leaflet, Postgres (amb mòdul PostGIS) i osm2pgsql).

2.1. - OpenStreetMap

OpenStreetMap és un projecte col·laboratiu que té per objectiu elaborar, oferir i compartir dades de tipus geogràfic. Qualsevol persona interessada en el projecte es pot donar d'alta com a usuari i pot contribuir a ampliar la base de dades aportant nova informació geogràfica. Aquest fet proporciona la gran avantatge de que la informació està sempre actualitzada ja que a més de incorporar-ne de nova, es posa al dia o es corregeix (en cas necessari) la existent. També cal dir que no es necessari estar registrat per poder obtenir la informació que hi ha emmagatzemada i que, mitjançant aquesta, es poden elaborar diversitat de mapes. La distribució de les dades que hi han emmagatzemades es fa sota llicència oberta Open Database License. Aquesta iniciativa neix davant la necessitat de poder disposar de mapes i d'informació geogràfica de lliure ús per a tothom, degut a que la major part de material existent no ho és. A l'actualitat, hi han diverses empreses que fan ús d'aquesta base de dades i que fins i tot alguna, realitza aportacions econòmiques per a finançar aquesta fundació (Aeroterra, Cloud Made, Apple, Nestoria, Geofebic, Mapnik Services, etc.). A la web www.openstreetmap.org podem veure un exemple del que es pot fer amb OSM (OpenStreetMap) permetent-nos veure unes quantes de les diferents capes de mapa. En la següent figura podem veure un exemple:

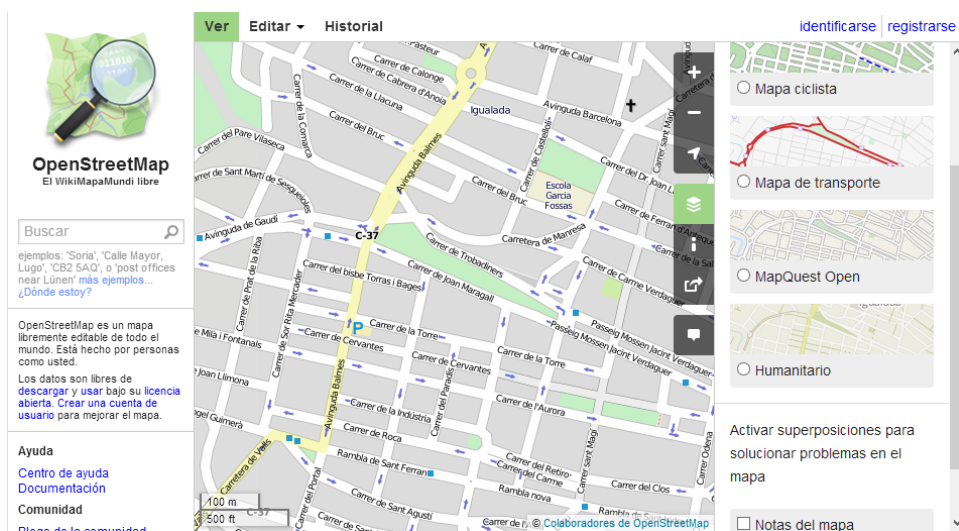


Figura 1. Exemple d'OpenStreetMap

Així, una petita enumeració dels usos de OSM seria la següent:

- Base d'informació per a la construcció de mapes per a diversos usos.
- Solucions digitals per a serveis de localització
- Conversió de dades a formats SIG.
- Traçat de rutes turístiques, de muntanya, etc.
- Càlcul de rutes òptimes.

Per últim dir que tota la informació emmagatzemada en la base de dades de OSM no només prové de voluntaris que han anat pujant les dades que han recopilat, sinó que també s'ha incorporat informació de dades públiques alliberades per institucions governamentals i altres.

2.1.1. – Com exportar dades d'OSM

OpenStreetMap ofereix diferents formats i maneres de descarregar la informació que té emmagatzemada a la base de dades (anteriorment era MySQL però actualment és PostgreSQL amb el complement PostGIS per donar suport a dades espacials). Hi ha l'opció Share a l'abast en el mapa de OSM que permet exportar la imatge als formats ràster més comuns (JPEG i PING) i també a formats vectorials com PDF, SVG, i PostScript. Però una altre de les possibilitats que ofereix la descarrega és fer-ho amb algun format SIG, per la qual cosa es tenen que descarregar les dades amb el format natiu d'OpenStreetMap, en un fitxer OpenStreetMap XML Data (.osm).

Això es pot fer de dues maneres:

- Descarregant directament l'arxiu Planet.osm que conté la informació geogràfica a nivell planetari i que ocupa uns quants GB (lloc de descàrrega: <http://planet.openstreetmap.org/planet/>).
- Seleccionat una àrea (que no pot ser molt gran) del mapa i mitjançant l'opció "Exportar" s'obté l'arxiu amb les dades (lloc de selecció i descàrrega: <http://www.openstreetmap.org/export#map=11/41.5834/1.6243&layers=T>). Podem veure el mapa en la següent imatge:

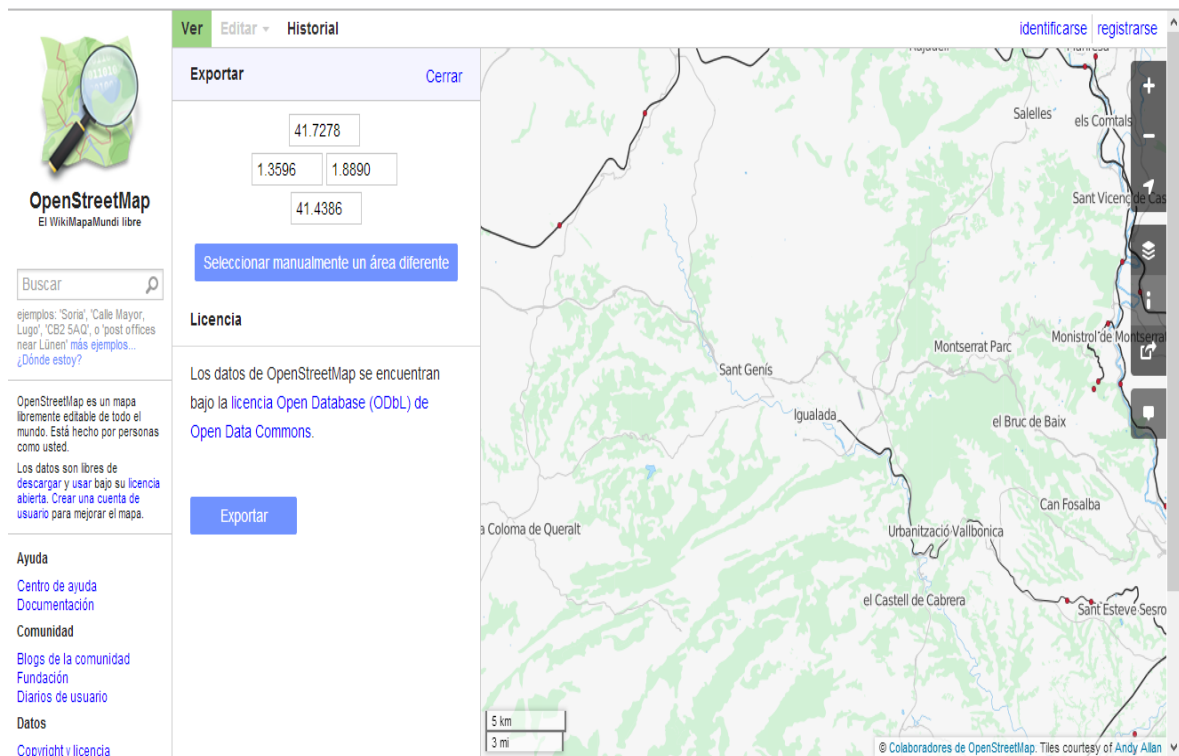


Figura 2. Exportar dades des del mapa directament.

Existeixen altres alternatives per a poder descarregar la dades directament amb arxius .osm i dividits per regions. És el cas de Cloud Made, una web que utilitza OpenStreetMap i entre els serveis que dóna hi ha el de poder descarregar totes les dades de OSM però dividides en arxius classificats per continents, regions, països, i fins i tot comunitats dintre d'un país (com és el cas d'Espanya). D'aquesta manera podem descarregar la informació que necessitem sens d'haver de manejar un excés de dades innecessàries, cosa que pot passar per exemple, amb Planet.osm. L'únic inconvenient és que no estan tan actualitzats com les de la web d'OSM però per moltes aplicacions, això pot no ser cap inconvenient. Per a més informació i descàrrega es pot visitar la web¹.

Una altre alternativa és Geofabrik. En aquest lloc també es poden descarregar els arxius de dades de OSM però no tant subdividits com en el cas de CloudMade. Es pot trobar més informació i descarregar els arxius a la web².

2.1.2. – Format del fitxer OSM XML

Les principals eines en l'univers OSM utilitzen un format XML seguint aquesta definició d'esquema XML que es va utilitzar per primera vegada només per l'API. Bàsicament es tracta d'una llista d'instàncies de les dades primitives (nodes, camins, i les relacions) que són l'arquitectura del model de l'OSM.

¹ <http://downloads.cloudmade.com/>

² <http://download.geofabrik.de/>

En la següent imatge es mostra un exemple d'arxiu OSM XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">
  <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900" maxlon="12.2524800"/>
  <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636" timestamp="2008-09-21T21:37:45z"/>
  <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter" uid="36744" visible="true" version="1" changeset="323878" timestamp="2008-05-03T13:39:23z"/>
  <node id="1831881213" version="1" changeset="12370172" lat="54.0900666" lon="12.2539381" user="lafkor" uid="75625" visible="true" timestamp="2012-07-20T09:43:19z">
    <tag k="name" v="Neu Broderstorf"/>
    <tag k="traffic_sign" v="city_limit"/>
  </node>
  ...
  <node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636" timestamp="2008-09-21T21:37:45z"/>
  <way id="26659127" user="Masch" uid="55988" visible="true" version="5" changeset="4142606" timestamp="2010-03-16T11:47:08z">
    <nd ref="292403538"/>
    <nd ref="298884289"/>
    ...
    <nd ref="261728686"/>
    <tag k="highway" v="unclassified"/>
    <tag k="name" v="Pastower Straße"/>
  </way>
  <relation id="56688" user="kmvar" uid="56190" visible="true" version="28" changeset="6947637" timestamp="2011-01-12T14:23:49z">
    <member type="node" ref="294942404" role=""/>
    ...
    <member type="node" ref="364933006" role=""/>
    <member type="way" ref="4579143" role=""/>
    ...
    <member type="node" ref="249673494" role=""/>
    <tag k="name" v="Küstenbus Linie 123"/>
    <tag k="network" v="VfV"/>
    <tag k="operator" v="Regionalverkehr Küste"/>
    <tag k="ref" v="123"/>
    <tag k="route" v="bus"/>
    <tag k="type" v="route"/>
  </relation>
  ...
</osm>
```

Figura 3. Exemple arxiu OSM XML.

L'estructura de dades del arxiu d'exemple de la figura anterior es descriu a continuació:

- Es declara el tipus de codificació i que és UTF-8.
- Un element OSM que conté la versió de l'API (i per tant les seves característiques) i el generador amb el qual s'ha confeccionat el fitxer (normalment una eina d'edició).
 - o Un bloc de nodes que contenen la ubicació espacial en el sistema de referència WGS84.
 - Les etiquetes de cada node.
 - o Un bloc de trams.
 - Les referències als nodes per a cada tram.
 - Les etiquetes de cada tram.
 - o Un bloc de relacions.
 - Les referències als membres per cada relació.
 - Les etiquetes de cada relació.

Per a més informació sobre el tema es pot visitar la següent pàgina d'on s'ha extret aquesta informació com a resum³

³ <http://wiki.openstreetmap.org/wiki/osm>

2.1.3. – Com exportar les dades a diferents formats

Un cop que s’han descarregat les dades OSM sigui del lloc que sigui, pot ser necessari convertir-les en altres tipus de formats perquè així ho requereix l’aplicació que les té de tractar. Per portar a terme aquesta tasca existeixen una sèrie de convertidors que se’n ocupen. A continuació es presenta una taula resum extreta de la wiki d’OpenStreetMap i que per a més informació la podeu veure a la web⁴

Format	Descarregar dades	Convertors
SVG (amb estil osmarender)	-	http://wiki.openstreetmap.org/wiki/Osmarender/Howto
Adobe Illustrator	-	http://wiki.openstreetmap.org/wiki/Exporting_to_Adobe_Illustrator
Polish format	-	http://wiki.openstreetmap.org/wiki/Osm2mp
Garmis GPS	http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/Download	http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin
GML	-	http://wiki.openstreetmap.org/wiki/Converting_OSM_to_GML http://www.wasser-muenchen.de/rguwiki/index.php?title=Osm2shp&php://wiki.openstreetmap.org/wiki/GeoConverterrintable=yes&printable=yes
KML (i altres formats GPS)	-	http://www.gpsbabel.org/download.html
Manifold GIS	-	http://forum.manifold.net/forum/t53330.11
Shapefile	http://download.geofabrik.de/ http://downloads.cloudmade.com/ http://mapserver.flightgear.org/shpdl/	http://www.polygongis.com/index.php/tools/ https://code.google.com/p/osm2shp/ http://wiki.openstreetmap.org/wiki/GeoConverter
Mapinfo (tab, mid/mif)	-	http://www.polygongis.com/index.php/tools/ http://wiki.openstreetmap.org/wiki/GeoConverter
PostgreSQL	Descarregar l’arxiu .osm, crear un nou esquema PostGIS a PostgreSQL, obrir la consola i escriure: osm2pgsql –slim –H localhost –U postgres –d nomEsquema ./arxiu.osm (-W si té contrasenya).	http://wiki.openstreetmap.org/wiki/Osm2pgsql
PostgreSQL	Descarregar un arxiu .osm	http://pgrouting.org/docs/tools/osm2pgrouting.html
SQLite	-	https://www.gaia-gis.it/fossil/spatialite-tools/index

⁴ <http://wiki.openstreetmap.org/wiki/ES:Export>

2.2. –CloudMade

En aquest apartat analitzem una de les eines de les que s'utilitzaran per el desenvolupament de l'aplicació. Es tracta d'un software i serveis, que aquesta empresa posa a disposició tant de clients com de desenvolupadors per a utilitzar-les en les seves aplicacions. Els productes que desenvolupen estan tots ells relacionats amb el món de les dades geogràfiques i geolocalització. El software i serveis que desenvolupen no està solsament orientat al món comercial sinó que també ni ha una part orientada al software lliure i serveis gratuïts.

2.2.1. – Què és CloudMade?

CloudMade és una empresa de desenvolupament de software i serveis, que ofereix tant a clients com a desenvolupadors d'aplicacions tot un seguit d'innovadores eines i una API, que permet de crear aplicacions basades en geolocalització en plataformes majoritàriament mòbils i web. Totes aquestes solucions de software mantenen una connexió amb el cloud que s'encarrega de mantenir els mapes i les dades actualitzades, i sincronitzant-los d'aquesta manera amb els del usuari a través de pantalles i enllaços. En els seus inicis, CloudMade va començar comptant com a única font de dades OpenStreetMap. En la actualitat no es limita a aquesta font de dades sinó que ha afegit moltes altres fonts disponibles per tot el món. També cal dir que van engegar el desenvolupament de la llibreria escrita en JavaScript, de codi obert i orientada a l'elaboració de mapes interactius per a incrustar tant a pàgines web com a dispositius mòbils, Leaflet. Per a ampliar aquesta informació es pot visitar la seva web⁵.

2.2.2. – Routing HTTP API docs

En aquest apartat es presenta un abstracte sobre les característiques de l'API de routing i que mostra com es poden fer consultes a través de la web⁶. Es veuran els diferents formats en que poden retornar les consultes i amb quins paràmetres es poden fer.

L'estructura d'una consulta per URL és la següent:

```
http://routes.cloudmade.com/<APIKEY>/api/0.3/start_point,[transit_point1,...,transit_pointN],end_point/route_type[/route_type_modifier].output_format[?lang=(Two letter ISO 3166-1 code)][&units=(km|miles)]
```

Si la consulta exigeix habilitació del servei de seguretat SSL⁷, es té que substituir la part inicial de l'adreça per⁸.

⁵ <http://cloudmade.com/>

⁶ <http://cloudmade.com/documentation/routing>

⁷ http://ca.wikipedia.org/wiki/Transport_Layer_Security

⁸ https://ssl_routes.cloudmade.com

Paràmetres comuns de la URL:

start_point	Latitud i longitud del punt d'inici de la ruta separats per coma.
end_point	Latitud i longitud del punt final de la ruta separats per coma.
route_type	Tipus de ruta ("car", "foot", "bicycle").
output format	Format del resultat de retorn ("js", "gpx").

Paràmetres opcionals de la URL:

transit_points	Llista de punts per on passa la ruta per ordre de pas i entre claudàtors ([]).
route_type_modifier	Opcions de la ruta: més curta (shortest), més ràpida (fastest), per ara només disponible per a tipus "car".
lang	Llenguatge en format ISO. Si no s'especifica es pren per defecte és Anglès. Els possible valors són: de, en, es, fr, hu, it, nl, ro, ru, si, vi, zh. És possible agregar una nova traducció a l'adreça: http://cloudmade.com/documentation/routing#routingTranslation
units	Mesura de la distància en unitats Km o Milles (Milles per defecte).
callback	Paràmetre opcional per a la resposta JS. Si per exemple callback = routeLoaded la resposta serà routeLoaded(JSON_object).

Significat dels missatges retornats

Missatge	Significat
m	Metre.
km	Kilòmetre.
ft	Peus.
mile	Milla.
miles	Milles.
head_on_pattern	Capçalera %s a %s.
head_on_pattern_unnamed	Capçalera %s.
earth_direction_N	Nord.
earth_direction_NE	Nord-est.
earth_direction_E	Est.
earth_direction_SE	Sud-est.
earth_direction_S	Sud.
earth_direction_SW	Sud-oest.
earth_direction_W	Oest.
earth_direction_NW	Nord-oest.
turn_C	Continuar a %s.
turn_TSLR	Girar a la dreta a %s.
turn_TSHR	Girar a la dreta a %s.
turn_TR	Girar a la dreta a %s.
turn_TL	Gireu a l'esquerra a %s.
turn_TSLL	Girar a l'esquerra a %s.
turn_TSHL	Girar a l'esquerra a %s.
turn_TU	Fer un canvi de sentit.

turn_C_unnamed	Continuar
turn_TSLR_unnamed	Girar a la dreta.
turn_TSHR_unnamed	Girar a la dreta.
turn_TR_unnamed	Girar a la dreta.
turn_TL_unnamed	Gireu a l'esquerra.
turn_TSLL_unnamed	Girar a l'esquerra.
turn_TSHL_unnamed	Girar a l'esquerra.
turn_TU_unnamed	Fer un canvi de sentit.
unknown_road	Carretera desconeguda.
unknown_road_near	Carretera desconeguda pro de %s
roundabout_take_the_exit	A la rotonda, agafar la sortida, %s a %s
roundabout_take_the_exit_unnamed	A la rotonda, agafar la sortida %s
exit_1	Primer
exit_2	Segon
exit_3	Tercer
exit_4	Quart
exit_5	Cinquè
exit_6	Sisè
exit_7	Setè
exit_8	Vuitè
exit_9	Novè
total_length	Longitud total
total_duration	Temps total
total_route_distance	Distancia total de la ruta
total_route_duration	Temps total de la ruta
sec	Segon
secs	Segons
hr	Hora
hrs	Hores
min	Minut
mins	Minuts

Exemples:

<http://routes.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/api/0.3/47.25976,9.58423,47.26117,9.59882/car/shortest.js>

<http://routes.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/api/0.3/47.25976,9.58423,47.26117,9.59882/bicycle.gpx>

[http://routes.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/api/0.3/51.22545,4.40730,\[51.22,4.41,51.2,4.41\],51.23,4.42/car.js?lang=de&units=miles](http://routes.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/api/0.3/51.22545,4.40730,[51.22,4.41,51.2,4.41],51.23,4.42/car.js?lang=de&units=miles)

JS respostes (JSON)

```
{
  version:0.3,
  status: 0 - OK, 1 - Error,
  status_message: Error message string,
  route_summary: {
    total_distance: Distance in meters,
    total_time: Estimated time in seconds,
    start_point: Name of the start point of the route,
    end_point: Name of the end point of the route,
    transit_points: [[point_name,lat,lon],...,[point_nameN,latN,lonN]] - Transit
    points if they are present in request.
  },
  route_geometry: [Array of nodes from start to end in 4326 projection
  ([lat1,lon1],...,[latM,lonM])]
  route_instructions: [[instruction, length, position, time, length_caption,
  earth_direction, azimuth, turn_type, turn_angle ],...,[instructionN, lengthN,
  ...]]
}
```

JSON paràmetres

Nom	Tipus	Descripció
instruction	String	Instrucció de text. Exemple: Gira a l'esquerre a ...
length	double	Longitud de segment en metres.
position	int	Índex del primer punt del segment en una route_geometry.
time	int	Temps estimat en segons del trajecte.
length_caption	String	Longitud del trajecte.
earth_direction	String	Codi d'inici del segment (N, NE, E, SE, S, SW, W, NW).
azimuth	double	http://en.wikipedia.org/wiki/Azimuth
turn_type	String	Codi del tipus de gir (opcionalment absent pel primer segment).
turn_angle	degree	Angle en graus del gir entre dos segments. 90 per dreta, 270 per esquerra, 180 per canvi de sentit, 0 seguir recta.

Tipus possibles de paràmetre de gir	
C	Continuar recte.
TL	Gir a l'esquerra.
TSLL	Gir lleu a l'esquerra.
TSHL	Gir pronunciat a l'esquerra.
TR	Gir a la dreta.
TSLR	Gir lleu a la dreta.
TSHR	Gir pronunciat a la dreta.
TU	Canvi de sentit.

Exemple de resposta GPX

```
<?xml version="1.0" encoding="UTF-8"?>
  <gpx creator="" version="1.1" xmlns="http://www.topografix.com/GPX/1/1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.topografix.com/GPX/1/1 gpx.xsd ">
    <extensions>
      <distance>293</distance>
      <time>34</time>
      <start>Perckhoevelaan</start>
      <end>Goudenregenlaan</end>
    </extensions>
    <wpt lat="51.17702" lon="4.39630" />
    <wpt lat="51.17656" lon="4.39655" />
    <wpt lat="51.17639" lon="4.39670" />
    <wpt lat="51.17612" lon="4.39696" />
    <wpt lat="51.17640" lon="4.39767" />
    <wpt lat="51.17668" lon="4.39828" />
    <wpt lat="51.17628" lon="4.39874" />
    <wpt lat="51.17618" lon="4.39888" />
    <rte>
      <rtept lat="51.17702" lon="4.39630">
        <desc>Head south on Perckhoevelaan, 0.1 km</desc>
        <extensions>
          <distance>111</distance>
          <time>13</time>
          <offset>0</offset>
          <distance-text>0.1 km</distance-text>
        </extensions>
      </rtept>
    </rte>
  </gpx>
```

```

        <direction>S</direction>
        <azimuth>160.6</azimuth>
    </extensions>
</rtept>
<rtept lat="51.17612" lon="4.39696">
    <desc>Turn left at Laarstraat, 0.1 km</desc>
    <extensions>
        <distance>112</distance>
        <time>13</time>
        <offset>3</offset>
        <distance-text>0.1 km</distance-text>
        <direction>NE</direction>
        <azimuth>58.1</azimuth>
        <turn>TL</turn>
        <turn-angle>269.0</turn-angle>
    </extensions>
</rtept>
<rtept lat="51.17668" lon="4.39828">
    <desc>Turn right at Goudenregenlaan, 70 m</desc>
    <extensions>
        <distance>70</distance>
        <time>8</time>
        <offset>5</offset>
        <distance-text>70 m</distance-text>
        <direction>SE</direction>
        <azimuth>143.4</azimuth>
        <turn>TR</turn>
        <turn-angle>89.8</turn-angle>
    </extensions>
</rtept>
</rte>
</gpx>

```

Per a més informació es pot consultar la documentació sobre aquest tema a la web⁹

2.2.3. – GeoCoding v3 beta Engine

La geocodificació és el procés de cerca de coordenades geogràfiques (longitud i latitud) amb informació d'entrada textual com adreces (ciutat, carrer, número, etc.), punts d'interès, etc. GeoCoding v3 és basa en un mètode especialitzat en aquest tipus de recerca d'alt rendiment, en el qual la consulta es fa mitjançant URL que incorpora tots els termes de cerca en format text. La resposta arriba en format XML. Quan es fa una cerca d'una adreça, aquesta ha d'incloure tots o alguns dels següents components: país, ciutat, comunitat, districte, codi postal, carrer, número de la casa, punt d'interès. El procés de consulta el podem descriure com: es processa la consulta d'entrada de cerca amb les restriccions imposades per alguns paràmetres, es calculen els resultats de la cerca i finalment, es preparen i prioritzen les dades de geocodificació que es retornen. Quan arriba la consulta en primer lloc, aquesta es prepara i codifica per adaptar-la a la semàntica pròpia. Això es fa en tres subprocessos: precompilació de la cadena de cerca, es detalla i finalment es normalitza.

- Precompilació: per raons d'eficiència i rendiment amb qualsevol cadena de cerca que envia l'usuari pot ser precompilada amb dos passos:
 - o Pre – processament independent local i regional.
 - o Una segona etapa de processament dependent de la regió i lloc.

Si cal, la compilació independent local / regional afegeix uns separadors addicionals a la cadena de recerca original, com per exemple caràcters d'espai abans de les lletres majúscules i nombres.

- Detall de la cadena de cerca: la cadena de cerca pot incloure separadors febles com espais i separadors forts com punts i comes o comes. Un separador feble significa que els dos termes en realitat, pertanyen a una mateixa part mentre que un separador fort significa que ambdós termes pertanyent a parts diferents. El "itemitzador" calcula una llista possible d'elements de direcció de la cadena de recerca. Per exemple, amb la consulta = "Stuttgart Alemanya; leuschner Strasse 45" l'itemitzador calcula els següents ítems: Stuttgart Alemanya, Stuttgart, Alemanya, Leuschner Strasse 45, Leuschner Strasse, Leuschner, Strasse 45, Strasse, 45.
- Normalització de la cerca: cada cadena que surt del pas anterior es normalitza segons les regles següents:
 - o Eliminar tots els espais i caràcters especials (per exemple parèntesis).
 - o Passar totes les lletres majúscules a minúscules.
 - o Eliminar tots els accents, dièresi, etc. de totes les lletres.
 - o Normalitzar caràcters Unicode amb NFKD¹(Normalization from Compatibility Decomposition).

⁹ <http://cloudmade.com/documentation/routing>

Com exemple: un element anomenat originalment Varieté Düsseldorf esdevé “varietedusseldorf”.

Una recerca directa de carrers o punts d'interès pot ser molt costosa degut a la gran quantitat d'informació que hi ha a la base de dades. No obstant, una consulta té com a paràmetres un codi postal o una regió. Per tant, primer es fa una reducció de l'àrea mitjançant l'extracció d'aquesta informació prèviament formatada.

- Cerca per codi postal: la base de dades conté de forma ordenada tots els codis postals disponibles en un determinat mapa. Un codi postal pot ser un atribut d'un element geogràfic com una regió, carrer, punt d'interès, etc. El motor de geocodificació cerca coincidències dintre de la base de dades per cada codi postal passat. Normalment només hi pot haver un resultat. Amb el codi postal trobat, el motor de cerca té accés directe a totes les característiques geogràfiques que té aquest atribut.

Hi ha dues maneres d'arribar a les regions amb el codi postal determinat. En algunes bases de dades ja hi ha regions emmarcades per polígons per a cada codi postal. En aquest cas, la regió resultat de la construcció per codi postal és simplement la unió de tots els polígons que tenen per atribut dit codi postal. Podria ser que hi hagués més d'una regió a la unió fruit de codis postals iguals però de diferents països. En OpenStreetMap no hi ha polígons associats a un codi postal¹⁰. No obstant això, hi ha algunes característiques de tipus carrer o punts d'interès els quals tenen aquest atribut. En aquest cas, el motor de geocodificació calcula polígons aproximats mitjançant l'agrupació de núvols de punts.

- Determinar els noms del lloc: en general això funciona de manera similar a com ho fa la cerca per codis postals però amb algunes diferències. A causa de la gran quantitat de llocs de tot el món, el motor de cerca comença a fer una recerca binària amb les tres primeres lletres o caràcters de la paraula. Si no troba una coincidència exacta, llavors s'incrementa el nombre d'errors permesos en la coincidència en funció de la longitud de la cadena a cercar.

Quan una cerca pot donar resultats molt diversos, si no hi ha cap regió exacta perquè es trobi una solució, es fa una aproximació traçant un cercle al voltant del centre de la ciutat amb un radi en funció de la població i la importància de la ciutat o districte.

Finalment dir que si tant la cerca per codi postal com la cerca per regió són vàlides, la regió final de la solució és la intersecció de les dues.

Càlcul de carrer resultant: La determinació dels noms de carrer dins de la regió / solució de codi postal és la que coincideix amb almenys un dels elements de la cerca. La tolerància d'error és la mateixa que per la cerca de llocs que s'ha explicat anteriorment.

En primer lloc es recullen totes les característiques del carrer a la base de dades amb el nom del carrer com atribut. S'agrupen els resultats segons aquestes característiques ja que hi poden haver carrers amb el mateix nom. Es calcula la posició que representa aquest grup. Es tracta de detectar un nombre de casa dintre dels elements de cerca i es

¹⁰ http://en.wikipedia.org/wiki/Unicode_equivalence

veu si un membre d'un grup de carrer conté aquest número. Si és així, aquesta posició és la que es dona per a tot el grup, sinó es pren el punt més central.

Després dels passos anteriors, tenim un número de carrer amb una posició representativa. Mitjançant l'aplicació d'un tipus de procés de geocodificació inversa el motor afegeix tota la informació disponible de la zona i que encara falta (país, estat, província, codi postal, districte, etc.) a cada entrada d'un resultat. Si no s'hagués trobat un resultat concret i per tant s'hagués pres una petita regió, s'enriqueix amb tota la informació màxima disponible d'aquest resultat.

Cada resultat representa una adreça i conté les següents parts de la direcció: número de carrer, número de la casa, etc. Cadascuna d'aquestes parts de la direcció té un pes determinat, que descriu la seva prioritat. En general no totes les parts són conegudes per una sola direcció. Per a cada part de la direcció es calcula una prioritat individual (entre 0.0 i 1.0). S'identifica la millor combinació entre un element de recerca i la cadena de part de la direcció. Com millor sigui la coincidència millor serà la prioritat. Per algunes parts de la direcció hi ha paràmetres addicionals que influeixen en la prioritat. Per acord, es pren que la importància i població compte més perquè per exemple París de França pondera més que París de Canadà. Si no hi ha regions de llocs exactes, es considera la distància entre el centre de la ciutat i la posició de la direcció. Com més gran sigui la distància menor és la prioritat. Després, per determinar la prioritat general per una adreça, el motor recull les prioritats de cada part d'aquesta de manera individual i les pondera. Finalment, tots els resultats s'ordenen segons la seva prioritat. Els que tenen una prioritat molt baixa es desestimen.

2.2.4. – GeoCoding v3 beta Server API

GeoCoding v3 beta compta amb un potent motor de cerca de punts d'interès mitjançant mapes vectorials MicroMap. L'API HTTP té un format d'URL com es mostra a continuació:

```
http://beta.geocoding.cloudmade.com/v3/<APIKEY>/api/geo.location.search.2?format=xml&source=OSM&enc=UTF-8&limit=10&locale=de&q=Leinfelden-Echterdingen
```

El significat de cada camp és el següent:

Camp	Significat
http://beta.geocoding.cloudmade.com/v3/	Link a Geocoding v3.
APIKEY	La key de desenvolupador que permet accedir en els serveis de CloudMade.
geo.location.search.2?	Sol·licitud de tipus de cerca.
format=xml	Format de la resposta.
source=OSM	Font de les dades (OpenStreetMap).
enc=UTF-8	Codificació del codi.
limit=10	Màxim nombre de resultats.
locale=de	Localitat de recerca actual.
q=Leinfelden-Echterdingen	Cadena de consulta d'entrada.

Les consultes tenen una sintaxis i uns paràmetres determinats que s'han de respectar. A continuació es presenten els paràmetres de les variables d'entrada més importants.

Paràmetre	Comentari
bbox=lonMin, latMin;lonMax, latMax	La seva finalitat és delimitar l'àrea de cerca dintre d'aquestes coordenades.
limit=number	Limita el número de resultats que es retornen.
format=xml	Format dels resultats que es retornen. Per defecte és xml però poden ser: json, sdf (Structure Data Format), i html.
source=name od the Data sources	S'utilitza per seleccionar l'origen de les dades per a la consulta (per exemple OSM).
enc=Encoding format	Especifica el format en que es codifiquen les dades d'entrada (per exemple UTF-8).
locale=locale	Especifica la configuració regional de cerca actual considerant el país i el idioma. Quan es defineix com a local, fa referència a l'Alemanya.
q="abc..."	S'especifica la cadena de consulta que es vol fer.

Per tal de resoldre una adreça postal, a la consulta es passen un o més dels següents camps: nom del país, nom de l'estat, nom de la ciutat, codi postal, nom del carrer, punts d'interès (POI). Normalment una entrada basada en text, s'estructura amb format de lliure estil, formulari d'adreça o una barreja d'ambdues. En totes aquestes recerques, el caràcter "q" s'utilitza com abreviatura de consulta i se li assignaran les adreces seguint la sintaxis d'acord a cada tipus.

- Estil lliure: la recerca d'estil lliure (recerca d'una sola línia o cerca difusa) es caracteritza per la dificultat en discernir que és el que es busca exactament. Per exemple, la consulta: q="Leinfelden-Echterdingen Alemanya Ludwigstr 4", és difícil pel motor de cerca decidir el que és "Leinfelden-Echterdingen" i el que és "Alemanya" perquè el primer no està especificat com una ciutat i el segon com un país. Per tant, els resultats d'aquesta cerca possiblement seran força deficients.
- Formulari d'adreça: aquest estil que també s'anomena de recerca estructurada, normalment respon a una sintaxis concreta: [clau=valor]. Aquesta parella clau – valor ha d'estar entre claudàtors. Els valors de les claus suportats són: país, estat, comarca, codi postal, ciutat, districte, carrer, número de la casa i vista. Els valors són els noms específics per aquestes claus (per exemple, Alemanya, 25, Avinguda Diagonal, etc.). Amb aquest tipus de cadenes de consulta, el motor de cerca és capaç de reconèixer per exemple "Alemanya" com un país, "Leinfelden – Echterdingen" com una ciutat, i així amb tots els valors de les diferents claus. Com a conseqüència, aquest tipus de cerca és molt més ràpida i efectiva.
- Les etiquetes de clau més importants són: country, state, zip, city, district, street, housenumber, sight, bbox, circle.
- Tipus híbrid: finalment hi ha el tipus híbrid que és una barreja dels dos estils anteriors. Un exemple de consulta híbrida seria la següent: [pais=Alemanya] Karlstr. Leinfelden-Echterdingen 70771[housenumber=12].

Resultats d'una consulta

El tipus explícit de consulta determina l'exactitud dels resultats de la cerca. Es pot fer un petit resum dels resultats que s'obtidran depenent del tipus de consulta que es faci:

- Si no s'especifica el número de casa en la consulta o no es pot trobar en la font de les dades, s'obtidrà una resposta que retornarà el mig del carrer com a resultat.
- Si en la consulta només s'especifica el codi postal, la resposta retornarà un punt de referència de la zona.
- Si a la consulta només s'especifica el nom de la ciutat, a la resposta es retornarà el punt central d'aquesta.
- Si només s'especifica el nom del carrer en la consulta i hi ha més d'un carrer amb aquest nom, els carrers trobats s'ordenen per ciutat d'on pertanyen.

Els paràmetres de les variables de sortida més comuns són els que es mostren a continuació:

Etiqueta	Significat
<city>City name</city>	Nom de la ciutat.
<country>Country name</country>	Nom del país.
<district>District name</district>	Nom del districte.
<position>Coordinates</position>	Coordenades del resultat.
<state>State name</state>	Nom de l'estat.
<zip>Zip-Code</zip>	Codi zip.
<status>Status of result</status>	Informa de si la cerca ha tingut èxit o no.
<duration>Time</duration>	Temps total de la cerca en mil·lisegons.

Finalment dir que també és possible fer una cerca per coordenades i el resultat en cas d'èxit serà el nom del lloc al qual pertanyen les coordenades que s'han passat.

Per a completar tota aquesta informació és molt recomanable visitar la pàgina on es detalla tota aquesta API i que és d'on s'ha estret aquest resum¹¹.

2.2.5. – Tiles HTTP API

CloudMade permet accedir el seu servidor de tiles mitjançant la seva API HTTP de tiles. Això pot ser de gran utilitat quan altres APIs de tercers no encaixen sigui per les raons que siguin en el projecte que s'està desenvolupant. L'estructura d'accés a aquesta API és simple i es pot reconèixer de manera immediata si s'està familiaritzat amb els tiles d'OpenStreetMap. En els següents exemples es mostren les URL per aconseguir dos tiles d'alta resolució de l'àrea de Berlín:

<http://b.tile.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/1/256/15/17599/10746.png>

Podem veure el tile aconseguit en la següent figura:

¹¹ <http://cloudmade.com/documentation/geocoding>



Figura 4. Exemple tile 1.

[http://b.tile.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/1\[b\]@2x\[/b\]/256/15/17599/10746.png](http://b.tile.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/1[b]@2x[/b]/256/15/17599/10746.png)

Podem veure el tile aconseguit en la següent figura:



Figura 5. Exemple tile 2.

Convenció de numeració pels tile: l'API HTML pels tiles utilitza projecció Mercator esfèrica, en la qual, la vista del món és circular amb diversos nivells de zoom fix disponibles. En l'exemple anterior bé definit per 15/17599/10746.png i es pot traduir o considerar com: zoom/x/y.png amb origen 0/0 al nord-est en cada tile.

Conversió de longitud i latitud de tiles: per saber quin és el tile que es necessita per una determinada longitud i latitud, en primer lloc es té de triar el nivell de zoom desitjat i després fer els següents càlculs:

- $n = 2 * \text{zoom}$
- $\text{xtile} = ((\text{lon_deg} + 180) / 360) * n$
- $\text{ytile} = (1 - (\ln(\tan(\text{lat_rad}) + \sec(\text{lat_rad})) / \text{Pi})) / 2 * n$

Per arrodonir xtile i ytile per a obtenir el número enter que correspon al tile, s'han de fer les següents operacions:

- $n = 2 * \text{zoom}$
- $\text{lon_deg} = \text{xtile} / n * 360.0 - 180.0$
- $\text{lat_rad} = \arctan(\sinh(\text{Pi} * (1 - 2 * \text{ytile} / n)))$
- $\text{lat_deg} = \text{lat_rad} * 180.0 / \text{Pi}$

Això és tot el que s'ha de fer per a obtenir els tile que es necessitin.

2.2.6. – Resum i conclusions

En els anteriors apartats s'ha fet un petit resum de que és CloudMade i de quines funcionalitats i eines posa a l'abast dels desenvolupadors. A part de les que s'han vist n'hi ha d'altres com NAV SDK¹² i MAP SDK¹³ que són unes classes amb els seus mètodes i interfícies que permeten llegir dades de diferents fonts de bases de dades per tal de poder confeccionar després tan mapes, com presentar informació geogràfica i relacionada diversa. Aquesta empresa té una part comercial i una altre que permet fer ús del seus serveis d'una manera però moderada. És una bona opció a tenir en compte degut a la potència del seu motor de cerca i la seva flexibilitat a l'hora de recuperar les dades ja que aquestes s'ofereixen en diferents formats. Una de les seves fonts de dades és OpenStreetMap que com ja sabem és una base de dades open source i per tant de lliure ús.

2.3. –Nominatim

Nominatim és una eina que permet cercar dades d'OSM a partir d'un nom i una adreça. També però permet geocodificació inversa, és a dir, a partir d'unes coordenades geogràfiques retorna el nom del lloc. Els índex de Nominatim, fan referència tant als de les dades d'OpenStreetMap com a d'altres amb característiques no identificades en aquest com poden ser bars, hotels, esglésies, etc. En la cerca, certes paraules clau es tradueixen amb una cerca d'etiquetes OSM específiques. Per exemple, al introduir la paraula "pub" Nominatim cerca la informació etiquetada com a "oci=pub". Aquest índex de Nominatim tenen l'estructura bàsica de clau=valor. Les dades de Nominatim s'estan indexant constantment degut a que té un rati d'actualització alt (aproximadament cada 20 minuts).

¹² <http://cloudmade.com/products/nav-sdk>

¹³ <http://cloudmade.com/products/map-sdk>

2.3.1. -Paraules clau especials

A Nominatim, com ja hem comentat en l'anterior apartat, és fa ús de paraules clau en forma clau = valor per a fer cerques. Actualment, s'estan recopilant tota una sèrie de frases especials ja existents per a fer cerques. Aquestes frases tenen un sentit especial per el motor de cerca. Existeix tota una nomenclatura que el motor entén i reacciona apropiadament davant certa simbologia. Així per exemple, si una mateixa frase pot ser expressada en singular i en plural si la posem en plural posarem una etiqueta de "plural=Y", en cas en que estigui en singular l'etiqueta serà "plural=N". També per exemple, posarem els operadors "in", "near" o "named" quant la localitat que cerquem estigui situada "a", "a prop de" o "anomenada". Per exemple, per "aeroport Barcelona" amb aquests prefixes podrem distingir aeroport de Barcelona, aeroport prop de Barcelona o aeroport que s'anomena Barcelona. Podem trobar una extensa taula de frases especials i per diferents idiomes en la web¹⁴.

2.3.2. -Paràmetres

Les consultes a Nominatim tenen el següent format:

```
http://nominatim.openstreetmap.org/search?<params>  
http://nominatim.openstreetmap.org/search/<query>?<params>
```

El format de retorn pot ser de tres tipus:

```
format=[html|xml|json|jsonv2]
```

Per a la resposta es pot seleccionar idioma preferit o fins i tot més d'un enumerant-los separats per comes. El format és el següent:

```
accept-language=<browser language string>
```

Cadena de consulta a cercar:

```
q=<query>
```

¹⁴ http://wiki.openstreetmap.org/wiki/Nominatim/Special_Phrases

Una alternativa a la cadena pot ser les consultes estructurades. Aquestes tenen el següent format:

```
street=<houenumber> <streetname>  
city=<city>  
county=<county>  
state=<state>  
country=<country>  
postalcode=<postalcode>
```

Es pot limitar la cerca a un país específic o una llista de països:

```
countrycodes=<countrycode>[,<countrycode>][,<countrycode>]...
```

També es pot especificar una zona preferida per a trobar el que s'està cercant:

```
viewbox=<left>,<top>,<right>,<bottom>  
or viewboxlbrt=<left>,<bottom>,<right>,<top>
```

Es poden restringir els resultats només als elements que figuren en el límit del quadre. Per exemple una cerca de [pub] normalment es rebutja. Però amb el paràmetre bounded = 1 es traduirà en una llista d'elements que coincideixen dintre del límit de quadre.

```
bounded=[0|1]
```

Si volem un desglossament de la direcció en els seus diferents elements:

```
addressdetails=[0|1]
```

Si es desitja que certs llocs que conté OpenStreetMap no apareguin en el resultat de la cerca, es pot posar una llista dels llocs (separats per comes) que no es vol que apareguin.

```
exclude_place_ids=<place_id>,[place_id],[place_id]>
```

També es pot limitar el nombre de resultats que retorna una cerca:

```
limit=<integer>
```

Si es vol que el resultat de la cerca d'una geometria estigui encapsulada en format JSON:

```
polygon_geojson=1
```

Si la resposta d'una geometria es vol en kml:

```
polygon_kml=1
```

Si la resposta d'una geometria es vol en format WKT:

```
polygon_text=1
```

A continuació podem veure tres exemples de consultes:

```
http://nominatim.openstreetmap.org/search?q=135+pilkington+avenue,+birmingham&format=xml&polygon=1&addressdetails=1
```

```
http://nominatim.openstreetmap.org/search/135%20pilkington%20avenue,%20birmingham?format=xml&polygon=1&addressdetails=1
```

```
http://nominatim.openstreetmap.org/search/gb/birmingham/pilkington%20avenue/135?format=xml&polygon=1&addressdetails=1
```

Per a més informació i per veure els paràmetres de la geocodificació inversa és convenient visitar la wiki¹⁵.

2.3.3 –Conclusions

Nominatim és una bona eina per obtenir dades de localitats puntuals. Disposa d'un potent motor de cerca i d'una gran flexibilitat a l'hora de fer consultes. Podem obtenir molts detalls de punts d'interès tant de ciutats i pobles com de llocs fora d'aquests. El seu ús es senzill en fer-se les consultes via URL i també té flexibilitat de formats a l'hora de tornar les respostes. En definitiva, serà una bona eina per a poder obtenir les coordenades de les localitats.

¹⁵ <http://wiki.openstreetmap.org/wiki/Nominatim>

2.4. –MapsForge

MapsForge és un projecte de codi obert amb llicència LGPL3 i està dissenyat per aplicacions basades amb OpenStreetMap. Actualment ofereix una biblioteca per a poder renderitzar mapes en aplicacions escrites per plataforma Android. La vista dels mapes és relativament senzilla d'utilitzar i molt semblant a l'API de Google. Es poden personalitzar els estils dels mapes mitjançant la seva configuració en arxius .xml.

2.4.1 –Disseny i característiques

El mapa binari de format d'arxiu MapsForge està dissenyat per a la representació de mapes en els dispositius mòbils que tenen recursos limitats. Es permet l'emmagatzematge de la informació geogràfica (dades d'OpenStreetMap per exemple) de forma molt eficient, accés als tiles de manera ràpida i filtratge dels objectes del mapa pel nivell de zoom.

L'arxiu del mapa està compost per diferents capes i cadascuna d'aquestes emmagatzema els objectes a representar segons el nivell de zoom. Els diferents intervals de zoom formen cada capa.

Coses a tenir en compte

- Totes les coordenades expressades amb altitud i longitud s'emmagatzemen en micrograus (graus * 10⁶).
- En els camps, el byte més significatiu s'emmagatzema primer (Big Endian).
- Els camps numèrics de les variables de byte sense signe, estan marcats amb VBE-U INT i s'emmagatzemen de la següent manera:
 - o el primer bit de cada byte s'utilitza per informació de continuïtat dels següents set bits de la dada.
 - o El valor del primer bit és 1 si el byte següent pertany al camp i 0 en cas contrari.
 - o Cada byte té 7 bits de valor per la dada i que coincideixen amb els de menys pes.
- Els camps numèrics signats amb una codificació de bytes variables estan marcats amb VBE-S INT i s'emmagatzemen de la següent manera:
 - o el primer bit de cada byte s'utilitza per informació de continuïtat dels següents sis (els sis primers) o set (tota la resta) bits de la dada.
 - o El valor del primer bit és 1 si el byte següent pertany al camp i 0 en cas contrari.
 - o Cada byte té sis (els sis primers) o set (tota la resta) bits de valor numèric començant amb els de menys pes.
 - o El segon bit en el primer byte indica el signe del número: 0 significa número positiu i 1 negatiu.
- Totes les cadenes s'emmagatzemen en UTF-8 de la següent manera:
 - o La longitud L de la cadena codificada amb bytes com VBE-U INT.
 - o L bytes de la cadena codificats en UTF-8.

Estructura dels arxius

Per a cada nivell de zoom es genera un sub-arxiu. Aquest consta d'un segment d'índex de tile que emmagatzema un punter per a cada peça creada en el segment de dades de tiles. La manera d'emmagatzemar els tiles en el segment de dades de tile i els seus corresponents punters en l'índex de tile és per fila i com una columna. Les files i les columnes venen donades internament per la disposició del enreixat que forma el tile i que està definit pel requadre rectangular que el conté. Per a cada tile hi ha metainformació disponible a la seva capçalera juntament amb les seves dades de càrrega útils (PDI i formes).

Per a llegir les dades d'un tile especificades en el sub-arxiu, el punter de grandària fixa en l'índex es calcula a partir de les coordenades del tiles. Els punts d'entrada d'índex pel desplaçament, en el sub-arxiu on s'emmagatzemen les dades.

Esquemàticament, aquesta estructura la podem representar de la següent manera:

```
# meta data
file header

# sub-files
for each sub-file
  # tile index segment
  tile index header
  tile index entries

  # tile data segment
  for each tile
    tile header
    for each POI
      POI data
    for each way
      way properties
      way data
```

Per a més informació sobre l'estructura de l'arxiu, capçaleres de l'arxiu, índex del tile, entrada d'índex, POIs, propietats dels way, etc. es pot consultar a la web¹⁶ que s'especifica amb detall.

2.4.2 -Presentació de mapa a Android

En aquesta secció es mostra com escriure una aplicació simple per Android que presenti un mapa. Aquest mapa que es presentarà en pantalla es pot moure i aplicar-hi zoom i estarà guardat a la memòria sd¹⁷ del dispositiu. El codi de la lògica és el que es presenta a continuació:

¹⁶ <https://code.google.com/p/mapsforge/wiki/SpecificationBinaryMapFile>

¹⁷ http://es.wikipedia.org/wiki/Secure_Digital

```

import android.os.Bundle;
import java.io.File;
import org.mapsforge.android.maps.MapActivity;
import org.mapsforge.android.maps.MapView;

public class HelloMapView extends MapActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MapView mapView = new MapView(this);
        mapView.setClickable(true);
        mapView.setBuiltInZoomControls(true);
        mapView.setMapFile(new File("/sdcard/path/to/mapfile.map"));
        setContentView(mapView);
    }
}

```

Podem notar que aquí no es té que importar cap llibreria de mapes de Google. En lloc d'aquesta, s'importa la de MapsForge. Tampoc no es necessita, a diferència de Google Maps, cap API key. El que si es té que fer és afegir a l'Android manifest el permís per escriure a la memòria externa sd. Es mostra a continuació:

```

<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Un altre exemple però ara fent ús d'un mapa on line, és a dir, sens utilitzar cap mapa emmagatzemat en memòria és el següent:

```

private MapView mapView;

public void onCreate(Bundle savedInstanceState) {
    Super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_osmaps);
    mapView = (MapView) findViewById(R.id.mapView);

    mapView.setCenter(new GeoPoint(41.38, 2.15));
    mapView.getController().setZoom(13);

    mapView.setClickable(true);
    mapView.setBuiltInZoomControls(true);
}

```

En el manifest ara es té que donar permís per connexió a Internet.

```

<uses-permission android:name="android.permission.INTERNET"/>

```

Finalment, a l'arxiu xml tindrem el següent:

```
<org.mapsforge.android.maps.mapView  
    android:id="@+id/mapa"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent" />
```

Amb això ja estaria tot complert per poder visualitzar un mapa centrat en el punt (41.38, 2.15) en una aplicació Android i es tindria control de zoom.

Per tal de visualitzar elements diferents sobre el mapa MapsForge compte amb tres classes que són:

- `ArrayItemizedOverlay`: per a poder visualitzar punts.
- `ArrayCircleOverlay`: per a poder visualitzar àrees.
- `ArrayWayOverlay`: per a la visualització de rutes.

Per a més informació cal consultar l'API de Mapsforge¹⁸.

2.4.3 –Conclusions

Com s'ha dit, MapsForge és una llibreria de codi obert i llicència LGPL3 que té un bon rendiment i amb possibilitats de presentar mapes vectorials. Aquesta última característica és important si es volen emmagatzemar en memòria per presentar-los sense connexió a Internet, ja que aquests, ocupen poc espai. No és una llibreria excessivament evolucionada però té moltes funcionalitats que permeten presentar mapes interactius amb presentació de una gran varietat d'elements i també una gran qualitat de presentació. També hem de tenir en compte que utilitza OpenStreetMap com a subministrador de mapes i informació geoespacial. Amb tot això cal dir que és també una bona alternativa a tenir en compte per utilitzar-lo en aplicacions que necessitin presentar mapes en plataforma Android.

¹⁸ <https://code.google.com/p/mapsforge/wiki/OverlayAPI>

3.-Model i disseny

Aquest projecte té com objectiu la construcció d'un producte programari que té que complir una sèrie de requisits o funcionalitats i pensat per dispositius mòbils que utilitzin plataforma Android. Més concretament, es tractarà d'una aplicació que permetrà posar en contacte usuaris de transport privat, que realitzin rutes similars, amb la finalitat de compartir recursos. Aquest tipus d'aplicacions existeixen en format web¹⁹, a les quals es pot accedir des d'un navegador, però ara el repte és fer l'aplicació especialment per dispositius mòbils.

En els següents apartats es detalla pas a pas tot el disseny de l'aplicació que està dividit seguint com a criteri les diferents funcionalitats que aquesta ofereix.

3.1. – Arquitectura del programari

L'arquitectura del programari la podríem definir com el disseny de més alt nivell de l'estructura d'un programa o aplicació i en la qual, s'han identificat els mòduls principals del sistema, les funcionalitats i responsabilitats de cadascun d'ells i finalment, les seves interaccions possibles utilitzant un control i flux de dades, protocols d'interacció i comunicació, etc. Això permet la comprensió de sistemes complexos, i el raonament per part del dissenyador sobre l'estructura pròpia dels sistemes i les seves propietats. La representació d'una arquitectura de programari es fa en termes d'una col·lecció de components i de totes les relacions possibles que hi poden haver entre aquests i la manera com interaccionen.

3.1.1.- Tipus d'arquitectura de l'aplicació

Els diferents patrons arquitectònics defineixen una varietat d'estils. Ni han que no descriuen un tipus concret sinó que són una barreja d'altres, formant part llavors d'un tipus heterogeni. Podem enumerar els principals estils com: sistemes de flux de dades, sistemes organitzats en capes o nivells, arquitectura client – servidor, arquitectures heterogènies basades en client – servidor i organitzat en capes, sistemes orientats a objectes distribuïts, arquitectures basades en esdeveniments i arquitectures orientades a serveis. El tipus d'arquitectura utilitzat per aquest treball serà de tipus heterogeni on es barreja una arquitectura basada en client – servidor (amb servidor pesant i client lleuger) i també orientada a serveis.

3.1.2.- Identificació dels mòduls o components

Passem a identificar els mòduls bàsics d'aquesta arquitectura:

- El component client consta d'una sèrie d'interfícies gràfiques en les quals interaccionarà l'usuari. Aquestes interfícies es limiten a recollir i a mostrar dades. Responen així a cada petició de l'usuari en el seu ús de les diferents funcionalitats que ofereix l'aplicació.

¹⁹ <http://www.blablacar.com>

- Un altre component client, internament recull les dades que l'usuari ha introduït i interacciona amb la part servidor fent ús dels diferents serveis que aquest ofereix.
- Un tercer component client internament retornarà dades cap el component client de presentació per retornar informació dels mòduls servidor i que l'usuari necessita.
- Un component servidor de dades estàndard que ofereix serveis per guardar i subministrar informació cap i des de la base de dades relacional.
- Un component servidor de dades espacials en el núvol que ofereix un servei de recuperació de dades geogràfiques a partir de coordenades.
- Un component servidor en el núvol que ofereix un servei de traducció d'un nom de localitat a coordenades geogràfiques.

En la següent imatge es mostra esquemàticament els components i com s'interrelacionen:

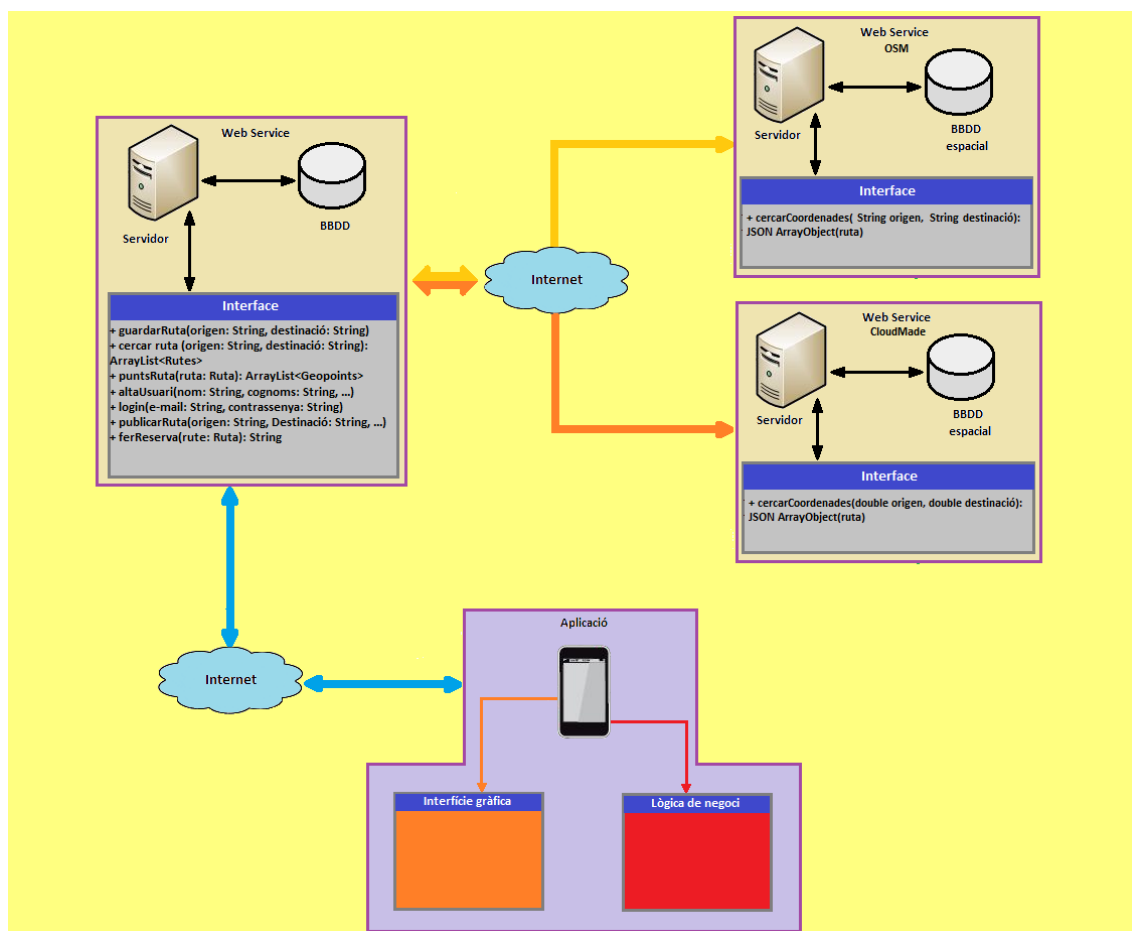


Figura 6. Esquema components aplicació en una arquitectura client – servidor orientada a serveis.

3.2. – Funcionalitats

En aquest apartat es presenten les funcionalitats de l'aplicació des del punt de vista de la navegació entre activitats. Es fa una descripció introductòria que s'anirà ampliant en cada un dels apartats posteriors.

- Pantalla d'inici: presentarà les opcions “donar-se d'alta”, “login”, “sortir”.
- Pantalla “donar-se d'alta”: apareix un formulari en el qual hi ha els diferents camps de dades personals requerides per a poder passar a ser un usuari registrat i fer ús del servei.
- Pantalla de login: apareix un formulari que demana el nom d'usuari i la contrasenya per a poder entrar en el sistema com a usuari registrat.
- Pantalla d'usuari registrat: apareixen les opcions “publicar trajecte”, “llistar trajectes propis”, “cercar trajecte”, “afegir comentari d'un conductor” i “ajuda”.
- Pantalla publicar trajecte: apareix un formulari amb les dades necessàries per a publicar un nou trajecte.
- Pantalla llistar trajecte: apareixen els trajectes publicats per l'usuari amb l'opció de esborrar o modificar el trajecte seleccionat.
- Pantalla cercar trajecte: apareix un formulari amb les dades necessàries per cercar un trajecte. Permet seleccionar-ne un.
- Pantalla trajecte seleccionat (per l'usuari que cerca un trajecte): es presenten les dades de la ruta seleccionada amb les opcions de veure el trajecte en un mapa, la de fer la reserva i la de veure els comentaris i puntuació atorgats en el conductor.
- Pantalla confirmació de reserva: si l'usuari ha fet la reserva se li comunica el resultat (si s'ha fet satisfactòriament o si no s'ha pogut fer).
- Pantalla mapa: es presenta el mapa de la ruta seleccionada i l'opció de tornar a la pantalla anterior.
- Pantalla fer comentari: l'usuari registrat podrà fer un comentari del conductor i puntuar-lo.
- Pantalla ajuda: es mostrarà una breu descripció de les funcionalitats que ofereix l'aplicació.

3.3. – Anàlisi

En aquest apartat es farà l'anàlisi de l'aplicació a alt nivell. Es representarà mitjançant diferents diagrames (casos d'ús i seqüència) la manera en que un usuari interactuarà amb el sistema. També es descriuran les regles d'integritat i les restriccions en aquesta relació aplicació - usuari.

3.3.1. – Diagrama de casos d'ús

A continuació es mostra el diagrama de casos d'ús. Podem veure que hi ha dos actors que són l'usuari registrat i l'usuari no registrat. Podem apreciar les diferents funcions a les quals tenen accés cadascun. Aquest diagrama dóna una visió global del que ofereix el sistema a l'usuari sense entrar en detalls ni tenir en compte la tecnologia emprada a l'hora de desenvolupar l'aplicació.

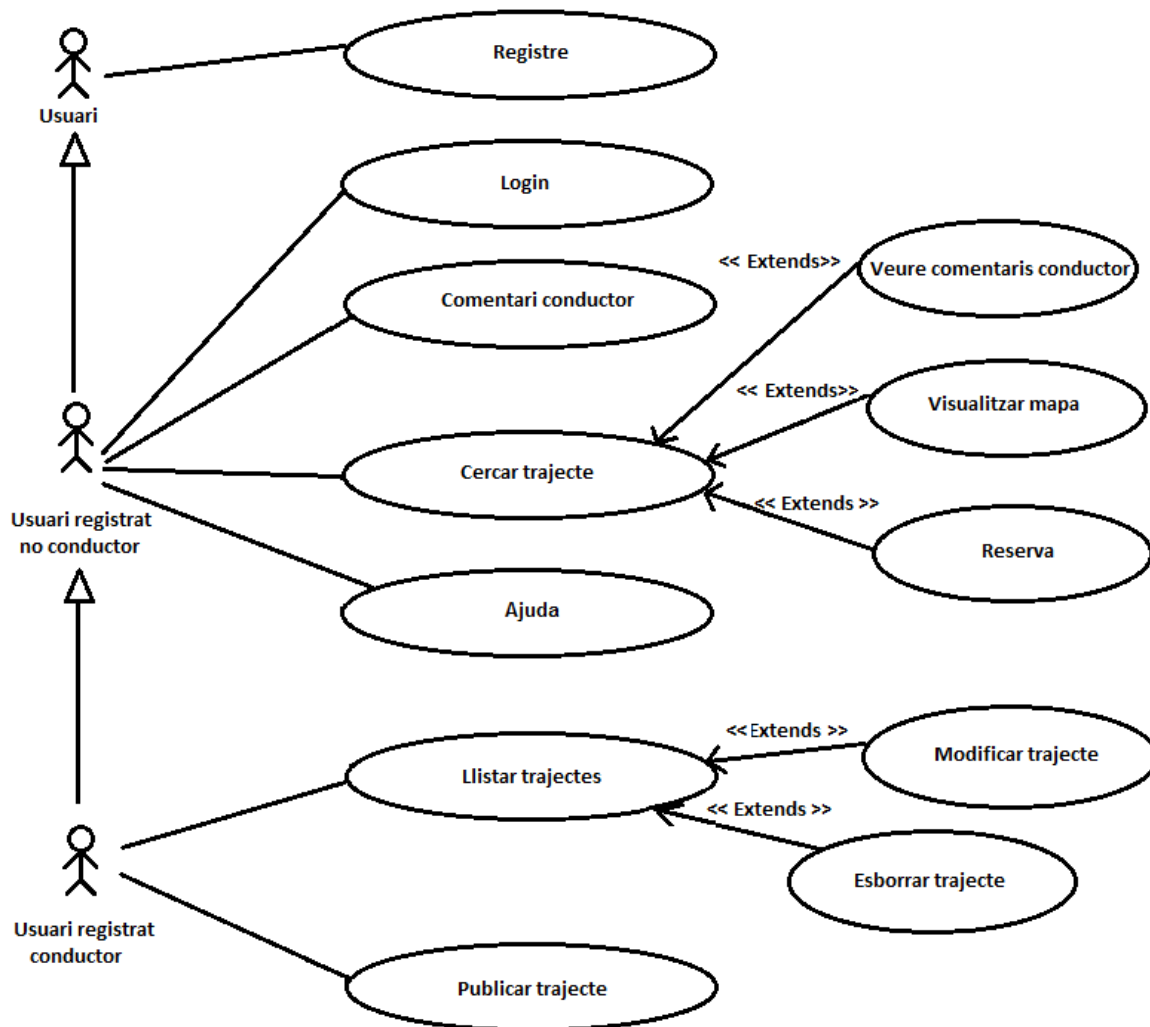


Figura 7. Diagrama de casos d'ús.

3.3.2. – Especificació dels casos d'ús

A continuació es fa una descripció detallada de cada cas d'ús representat a l'apartat anterior.

Nom: Registre

Actors: Usuari

Resum de la funcionalitat: donar-se d'alta en el sistema.

Flux d'esdeveniments principal:

- 1) L'usuari es vol donar d'alta en el sistema en un moment determinat.
- 2) S'executa el cas d'ús registre.
- 3) L'usuari omple el formulari amb les dades que es demanen.
- 4) L'usuari envia les dades en confirmar el registre.
- 5) El sistema dona d'alta en el nou usuari.

Flux d'esdeveniments alternatiu:

- 1) El sistema no pot donar d'alta a l'usuari i retorna un error en el registre.
- 2) L'usuari decideix en l'últim moment no donar-se d'alta i avorta l'operació.
- 3) L'usuari no omple totes les dades obligatòries i el sistema retorna un error.

Nom: Login

Actors: Usuari registrat

Resum de la funcionalitat: l'usuari entra en el sistema amb el seu nom d'usuari i contrasenya.

Flux d'esdeveniments principal:

- 1) L'usuari que ja està registrat en el sistema hi vol accedir.
- 2) S'executa el cas d'ús login.
- 3) L'usuari omple el formulari amb les dades que es demanen (nom i contrasenya).
- 4) L'usuari envia les dades un cop ha omplert el formulari.
- 5) El sistema dona accés a l'usuari registrat.

Flux d'esdeveniments alternatiu:

- 1) L'usuari no està registrat i el sistema retorna un error en el login.
- 2) La informació que introdueix l'usuari registrat no és vàlida i el sistema retorna un error en el login .
- 3) El sistema retorna un error per problemes tècnics.

Nom: Publicar trajecte

Actors: Usuari conductor registrat

Resum de la funcionalitat: l'usuari registrat conductor ha entrat en el sistema i publica un trajecte.

Flux d'esdeveniments principal:

- 1) L'usuari registrat conductor ja ha accedit al sistema.
- 2) S'executa el cas d'ús publicar trajecte.
- 3) L'usuari conductor omple el formulari amb les dades que es demanen.
- 4) L'usuari conductor envia les dades un cop ha omplert el formulari.
- 5) El sistema guarda les dades del nou trajecte i n'informa a l'usuari conductor registrat.

Flux d'esdeveniments alternatiu:

- 1) L'usuari conductor registrat no omple totes les dades obligatòries i el sistema informa del error.
- 2) El sistema retorna un error per problemes tècnics.
- 3) L'usuari conductor registrat avorta la publicació del trajecte.

Nom: Cercar trajecte

Actors: Usuari conductor registrat, usuari no conductor registrat

Resum de la funcionalitat: L'usuari (conductor o no)registrat ha entrat en el sistema i cerca un trajecte.

Flux d'esdeveniments principal:

- 1) L'usuari (conductor o no) registrat ja ha accedit al sistema.
- 2) S'executa el cas d'ús cercar trajecte.
- 3) L'usuari omple el formulari amb les dades que es demanen .
- 4) L'usuari envia les dades un cop omplert el formulari.
- 5) El sistema presenta els resultats de la consulta.

Flux d'esdeveniments alternatiu:

- 1) L'usuari (conductor o no) registrat no entre totes les dades obligatòries i el sistema retorna un error.
- 2) El sistema retorna un error per problemes tècnics.
- 3) L'usuari (conductor o no) registrat avorta la cerca.

Nom: Visualitzar mapa

Actors: Usuari conductor registrat, usuari no conductor registrat

Resum de la funcionalitat: L'usuari (conductor o no) registrat ha entrat en el sistema, ha cercat un trajecte i el visualitza en el mapa.

Flux d'esdeveniments principal:

- 1) L'usuari (conductor o no) registrat ja ha accedit al sistema.
- 2) S'executa el cas d'ús visualitzar mapa.

Flux d'esdeveniments alternatiu:

- 1) El sistema retorna un error per problemes tècnics.
- 2) No hi ha informació del mapa per aquell trajecte.

Nom: Reserva

Actors: Usuari (conductor o no) registrat

Resum de la funcionalitat: l'usuari (conductor o no) registrat ha entrat en el sistema, ha cercat un trajecte, n'ha seleccionat un i fa una reserva de plaça.

Flux d'esdeveniments principal:

- 1) L'usuari (conductor o no) registrat ja ha accedit al sistema.
- 2) S'executa el cas d'ús reserva.
- 3) L'usuari (conductor o no) registrat decideix fer una reserva del trajecte.
- 4) El sistema presenta a l'usuari registrat el resultat de la reserva.

Flux d'esdeveniments alternatiu:

- 1) El sistema retorna un error per problemes tècnics.

Nom: Veure comentaris conductor

Actors: Usuari (conductor o no) registrat

Resum de la funcionalitat: l'usuari (conductor o no) registrat ha entrat en el sistema, ha cercat un trajecte, n'ha seleccionat un i vol veure els comentaris que han fet altres usuaris en el conductor del trajecte seleccionat.

Flux d'esdeveniments principal:

- 5) L'usuari (conductor o no) registrat ja ha accedit al sistema.
- 6) S'executa el cas d'ús veure comentaris conductor.
- 7) El sistema presenta a l'usuari registrat els comentaris fets al conductor del trajecte seleccionat i la puntuació.

Flux d'esdeveniments alternatiu:

- 2) El sistema retorna un error per problemes tècnics.

Nom: Llistar trajectes

Actors: Usuari conductor registrat

Resum de la funcionalitat: l'usuari conductor registrat selecciona veure el llistat dels trajectes que ha publicat.

Flux d'esdeveniments principal:

- 1) L'usuari conductor registrat ja ha accedit al sistema.
- 2) S'executa el cas d'ús llistar trajectes.
- 3) L'usuari conductor registrat accedeix als trajectes que té publicats
- 4) El sistema presenta a l'usuari conductor registrat la llista de trajectes que té publicats.

Flux d'esdeveniments alternatiu:

- 1) El sistema retorna un error per problemes tècnics.

Nom: Modificar trajecte

Actors: Usuari conductor registrat

Resum de la funcionalitat: L'usuari conductor registrat ha entrat en el sistema, ha llistat els trajectes que ha publicat, n'ha seleccionat un i modifica les dades d'aquest.

Flux d'esdeveniments principal:

- 1) L'usuari conductor registrat ja ha accedit al sistema.
- 2) S'executa el cas d'ús modificar trajecte.
- 3) L'usuari conductor registrat decideix modificar les dades d'un trajecte.
- 4) El sistema presenta a l'usuari conductor registrat el trajecte amb les dades actuals.
- 5) L'usuari conductor registrat modifica les dades del trajecte.

Flux d'esdeveniments alternatiu:

- 1) L'usuari conductor registrat deixa algun camp obligatori en blanc i el sistema mostra l'error.
- 2) El sistema retorna un error per problemes tècnics.

Nom: Esborrar trajecte

Actors: Usuari conductor registrat

Resum de la funcionalitat: l'usuari conductor registrat ha entrat en el sistema, ha seleccionat llistar trajectes, n'ha seleccionat un i l'esborra.

Flux d'esdeveniments principal:

- 8) L'usuari conductor registrat ja ha accedit al sistema.
- 9) S'executa el cas d'ús esborrar trajecte.
- 10) L'usuari conductor registrat ha triat un trajecte que va publicar i l'esborra.
- 11) El sistema presenta a l'usuari conductor registrat el resultat de l'operació.

Flux d'esdeveniments alternatiu:

- 1) El sistema retorna un error per problemes tècnics.

Nom: Comentari conductor

Actors: Usuari no conductor registrat

Resum de la funcionalitat: l'usuari no conductor registrat ha entrat en el sistema i fa un comentari sobre un conductor a més de donar-li una puntuació.

Flux d'esdeveniments principal:

- 2) L'usuari no conductor registrat ja ha accedit al sistema.
- 3) S'executa el cas d'ús comentari conductor.

- 4) L'usuari no conductor registrat escriu un comentari sobre un conductor i el puntua.
- 5) El sistema presenta a l'usuari no conductor registrat el resultat de l'operació.

Flux d'esdeveniments alternatiu:

- 3) L'usuari no conductor no ha puntuat en el conductor i el sistema mostra l'error.
- 4) El sistema retorna un error per problemes tècnics.

3.3.3. – Model conceptual de dades

En aquest apartat es presenta el diagrama estàtic d'anàlisi (model conceptual de dades) UML de l'estructura lògica de dades de l'aplicació. Aquest diagrama conté les classes, atributs, associacions que són més rellevants pel sistema.

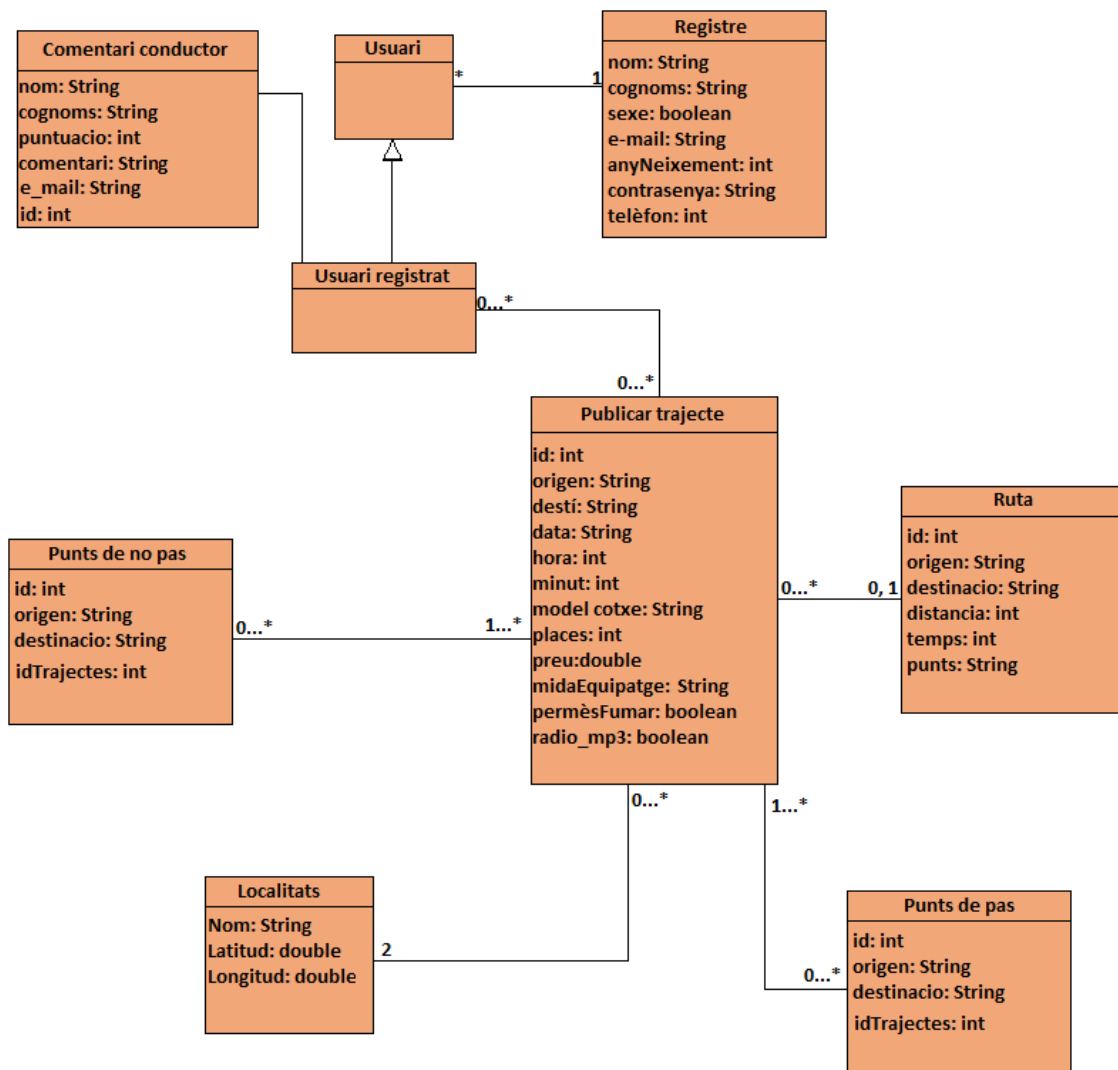


Figura 8. Model conceptual de dades.

3.3.4. – Restriccions d'integritat

Les restriccions d'integritat les podem definir com les condicions que han de satisfer les instàncies del model. En farem una descripció en forma textual per veure de manera clara quines hi ha.

1. L'usuari registrat s'identifica per la seva direcció de correu electrònic i la seva contrasenya.
2. A l'hora de modificar o esborrar algun trajecte es farà segons el trajecte seleccionat.
3. Un trajecte pot ser esborrat per l'usuari que l'ha publicat.
4. Tant els enters d'hora com de minut seran més grans o iguals a zero.
5. El preu del trajecte serà per una reserva i el seu valor serà positiu.

3.3.5. – Diagrama de seqüència

En aquest apartat veurem el diagrama de seqüència de la interfície gràfica d'usuari amb tots els seus casos d'ús relacionats. D'aquesta manera podem veure de manera esquemàtica com es portarà a terme la navegació per a la interfície gràfica d'usuari per a cada funcionalitat descrita.

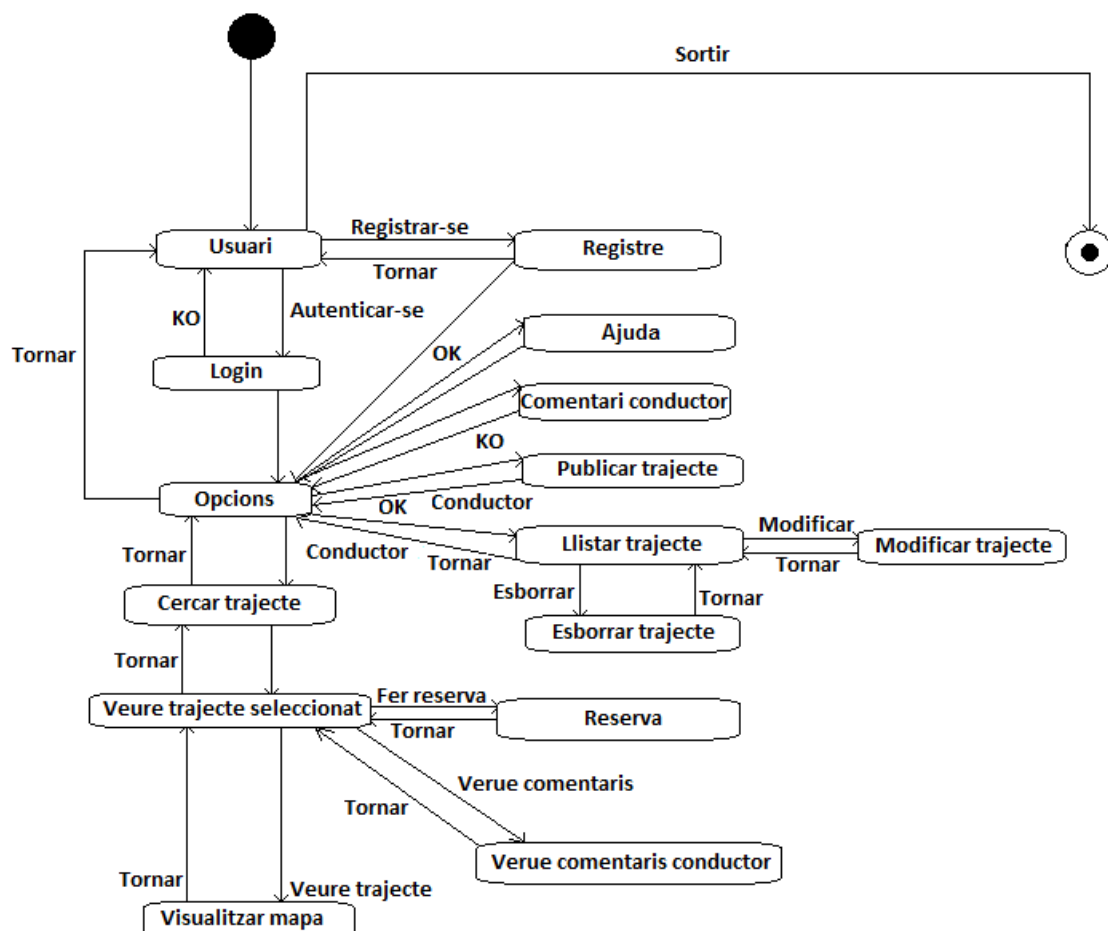


Figura 9. Diagrama de seqüència.

3.3.6. – Anàlisi de pantalles

En aquest apartat anirem veient un esbós de cada pantalla per tenir una aproximació de la seva aparença. Continuem amb un model i disseny que ara com ara és independent de la tecnologia seleccionada pel desenvolupament de l'aplicació i per tant, és completament vàlid per a qualsevol.

3.3.6.1. – Pantalla d'inici (usuari)

Aquesta pantalla serà la primera que l'usuari, registrat o no es trobarà en posar en marxa l'aplicació. Se li presentaran tres opcions: registre , login i sortir. D'aquesta manera l'usuari nou es podrà donar d'alta en el sistema omplint un formulari; l'usuari que ja està registrat podrà entrar en el sistema mitjançant el seu correu electrònic i la contrasenya que va triar; finalment es dóna l'opció de sortir de l'aplicació. L'esbós gràfic de la pantalla la veiem a continuació:

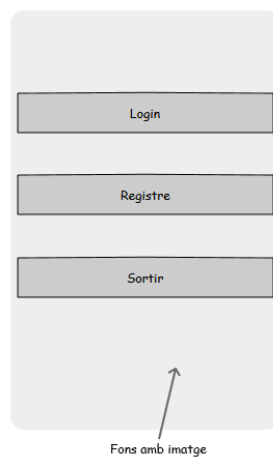


Figura 10. Inici.

3.3.6.2. – Pantalla login

L'usuari que ja està donat d'alta en el sistema es podrà autenticar en aquesta pantalla. Se li demanarà el nom d'usuari i la contrasenya. Tindrà l'opció de sortir si finalment no vol continuar. A continuació es mostra l'esbós gràfic de la pantalla que es mostrarà:

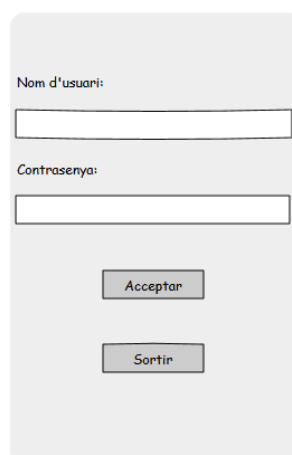


Figura 11. Login

3.3.6.3. – Pantalla registre

L'usuari nou que no està donat d'alta en el sistema es podrà registrar en aquesta pàgina. Se li presenta un formulari en el qual tindrà d'omplir tots els camps. Un cop fet, podrà enviar el formulari per completar l'operació de registre. També es dóna l'opció de sortir si es repensa i no es vol registrar. A continuació es mostra l'esbós gràfic de la pantalla que es mostrarà:

Formulari de registre amb els següents camps i opcions:

- Nom:
- Cognoms:
- e-mail:
- Any de neixament (yyyy):
- N° de telèfon:
- Sexe: Femení
- Contrasenya:
- Tlf. públic: si
- e-mail públic: Si
- Botons: Enviar, Sortir

Figura 12. Registre.

3.3.6.4. – Pantalla opcions per usuari registrat

Quan l'usuari s'ha registrat o ha entrat mitjançant l'opció de login, la pantalla que li apareix és la de les opcions disponibles, que li permetrà seleccionar la funcionalitat que més li convingui. L'esbós de la pantalla és el que es mostra a continuació:

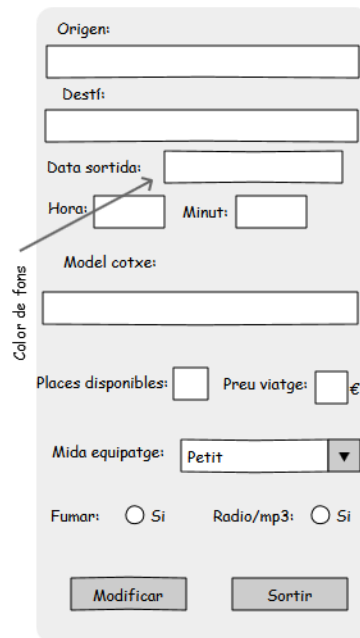
Pantalla d'opcions per usuari registrat amb els següents botons:

- Publicar trajecte
- Llistar trajectes
- Cercar trajecte
- Comentari conductor
- Sortir

Figura 13. Opcions usuari

3.3.6.5. – Pantalla publicar trajecte

L'usuari registrat podrà publicar un trajecte. Aquesta pantalla és la que veurà i en la qual apareix un formulari amb les dades a omplir de la ruta. Un cop ple el formulari, podrà enviar les dades per a completar la publicació. També en aquesta pantalla tindrà l'opció de sortir. L'esbós de la pantalla es mostra a continuació:



Origen:

Destí:

Data sortida:

Hora: Minut:

Model cotxe:

Places disponibles: Preu viatge: €

Mida equipatge:

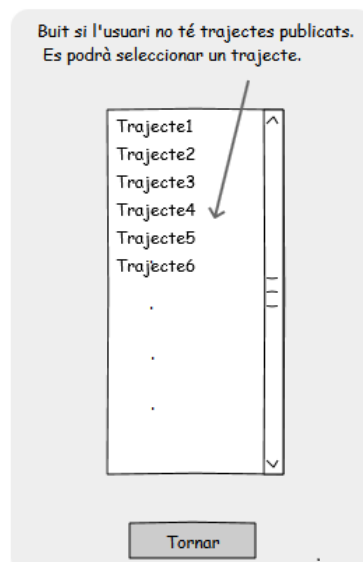
Fumar: Sí No Radio/mp3: Sí No

Color de fons

Figura 14. Publicar trajecte.

3.3.6.6. – Pantalla llistar trajectes

La pantalla següent és la que es mostra a l'usuari si ha seleccionat l'opció llistar trajectes. Si no té cap trajecte seleccionat, aquesta estarà buida. Si té trajectes, en podrà seleccionar un que se li obrirà amb més detall en una altre finestra junt amb tota la informació.



Buit si l'usuari no té trajectes publicats.
Es podrà seleccionar un trajecte.

- Trajecte1
- Trajecte2
- Trajecte3
- Trajecte4
- Trajecte5
- Trajecte6
- .
- .
- .

Figura 15. Llistar trajectes.

3.3.6.7. – Pantalla esborrar trajecte

La pantalla d'esborrar trajecte mostra el trajecte que ha seleccionat l'usuari propietari (el qui el va publicar), i li dóna l'opció d'esborrar-lo. Això només ho podrà fer si es compleixen les regles d'integritat. En la següent imatge es veu l'esbós de la pantalla que es presentarà.

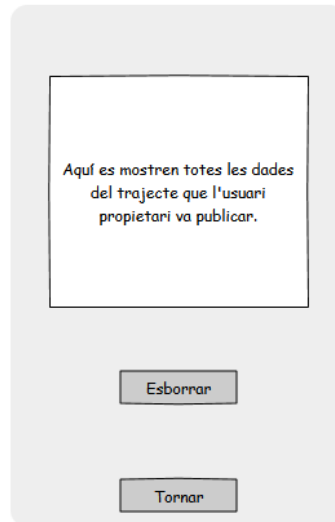


Figura 16. Esborrar.

3.3.6.8. – Pantalla modificar trajecte

Quan l'usuari accedeix a aquesta pantalla és perquè té publicat com a mínim un trajecte i té la intenció de modificar alguna dada (si les regles d'integritat li permeten fer-ho). També tindrà l'opció de tornar a la pàgina anterior si es repensa. A la següent imatge podem veure la pantalla que se li presenta.

A wireframe of a mobile application screen for editing a route. The form contains the following fields and controls:

- Origen: Text input with "Lleida".
- Destí: Text input with "Mannesa".
- Data sortida: Text input with "25/11/2013".
- Hora: Text input with "06".
- Minut: Text input with "15".
- Model cotxe: Text input with "SEAT Panda".
- Places disponibles: Text input with "3".
- Preu viatge: Text input with "2" followed by a "€" symbol.
- Mida equipatge: Dropdown menu with "Petit" selected.
- Fumar: Radio button with "Si".
- Radio/mp3: Radio button with "Si".
- Buttons: "Modificar" and "Sortir".

Figura 17. Modificar trajecte.

3.3.6.9. – Pantalla cercar trajecte

Quan l'usuari arriba a aquesta pantalla té l'opció de cercar un trajecte per la qual cosa, tindrà d'introduir la població origen, la població destí i la data. D'aquesta manera se li presentaran els trajectes existents a la base de dades amb aquestes característiques. També tindrà l'opció de tornat enrere. La pantalla que veurà serà la que es mostra a continuació.



Origen:

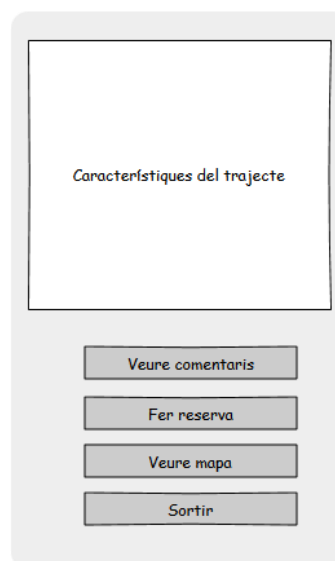
Destí:

Data sortida (dd/mm/yyyy):

Figura 18. Cercar trajecte.

3.3.6.10. – Pantalla veure trajecte seleccionat

L'usuari que ha cercat un trajecte, i n'ha triat un entre els que s'han trobat que s'ajustaven a les característiques requerides, se li presenta la següent pantalla en la qual veu el detall del trajecte i se li presenten tres opcions: fer la reserva, veure el trajecte en el mapa o veure comentaris. La següent imatge ens mostra la pantalla que veurà l'usuari.



Característiques del trajecte

Figura 19. Trajecte seleccionat.

3.3.6.11. – Pantalla fer reserva

En aquesta pantalla l'usuari entra per a fer una reserva del trajecte que ha triat. Llavors el que podrà veure serà el telèfon i /o adreça de correu electrònic del conductor si és que aquest permet publicar aquestes dades. En cas negatiu veurà l'adreça de correu electrònic del web màster i ja serà aquest el que farà el tràmit amb el conductor. D'aquesta manera, la pantalla que es visualitzarà serà la de la imatge següent.

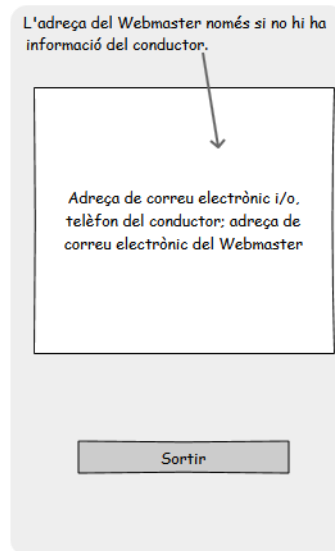


Figura 20. Fer reserva.

3.3.6.12. – Pantalla veure mapa

En aquesta pantalla l'usuari podrà veure el trajecte sobre el mapa. La vista és pot veure a la següent imatge.

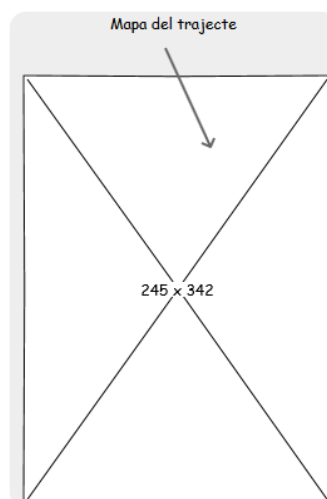
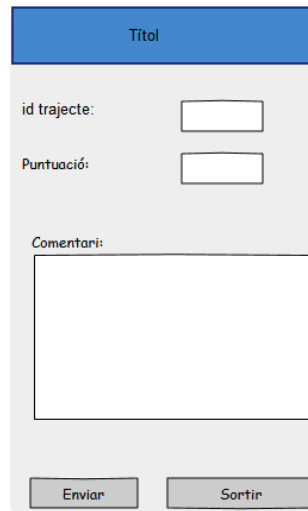


Figura 21. Veure mapa.

3.3.6.13. – Pantalla comentari conductor

En aquesta pantalla l'usuari podrà deixar un comentari sobre el conductor i també el podrà avaluar. La següent pantalla mostra el formulari que es presenta per omplir.



Titol

id trajecte:

Puntuació:

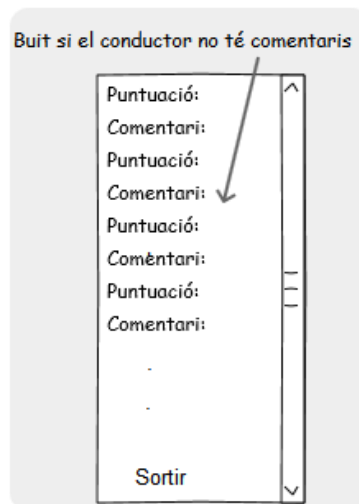
Comentari:

Enviar Sortir

Figura 22. Comentari a conductor.

3.3.6.14. – Pantalla veure comentaris de conductor

En aquesta pantalla l'usuari podrà veure els comentaris i la puntuació fets a un conductor abans de llogar una plaça. La següent pantalla mostra el que veurà l'usuari.



Buit si el conductor no té comentaris

Puntuació: ^

Comentari: ↓

Puntuació:

Comentari:

Puntuació:

Comentari:

Puntuació:

Comentari:

Sortir

∨

Figura 23. Llistat comentaris

4.-Instal·lació del programari

En els següents subapartats, es farà una descripció de l'entorn en el qual es desenvoluparà el producte objectiu del projecte, els recursos de software necessaris, les eines que finalment s'utilitzaran d'entre el ventall d'aquestes, i es donaran les pautes seguides per a la seva instal·lació i configuració. També veurem les diferents opcions en quant a versions i el perquè se'n trien unes i no unes altres.

4.1. - Recursos de maquinari

- PC amb microprocessador Intel® Core(TM) i5-2450M CPU @ 2.50 GHz 2.50 GHz de 64 bits .
- Memòria física instal·lada a l'equip de 8,00 GB.
- Targeta gràfica nVidia GeForce 410M amb 1024 MB de memòria gràfica.
- Monitor de 15.5 polsades 1280 x 800 16:9.
- Disc dur SATA de 640 GB.

4.2.- Recursos de programari

Els recursos de programari amb els quals es desenvolupa el software són els següents:

- Sistema operatiu Windows XP SP3 instal·lat en màquina virtual VMWare Player.
- Eclipse com a IDE de desenvolupament del programari (qualsevol versió amb suport JEE).
- SDK Android que l'integro dintre d' Eclipse.
- jdk 1.7.25 de Oracle.
- Sistema de gestió de base de dades relacional MySQL de Oracle.
- Contenedor de pàgines web dinàmiques Apache Tomcat versió 6.0.37 (també es pot utilitzar la versió 7 o 8).

A partir d'aquí utilitzarem tot un conjunt de llibreries i recursos que els anirem presentant a mesura que els necessitem en el desenvolupament del software.

4.3.- Instal·lació del programari

En aquest subapartat veurem el programari que es té d'instal·lar per el desenvolupament de l'aplicació i de suport a aquesta. També veurem els passos necessaris que s'han seguit per a instal·lar-lo.

4.3.1. – Instal·lació de la màquina virtual i sistema operatiu

En primer lloc instal·larem la màquina virtual VMware²⁰ – player que serà el suport pel sistema operatiu i la resta de programes. Un cop descarregat, premem sobre l'executable i seguim els passos d'instal·lació (ubicació, etc.). Un cop s'ha instal·lat ja podem executar-la per a crear una màquina virtual i posar-hi el sistema operatiu que en aquest cas serà, com ja s'ha dit, Windows XP SP3. En primer lloc executem l'aplicació i veurem la següent interfície:

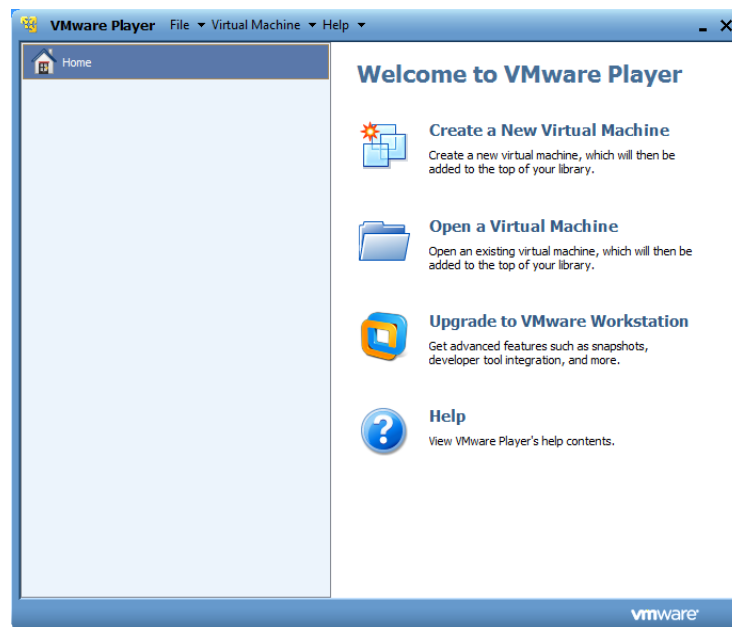


Figura 24. Selecció crear nova màquina virtual

Ara premem sobre la primera opció que és la de crear un nova màquina virtual. La interfície que ens apareix és la següent:

²⁰ https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/4_0

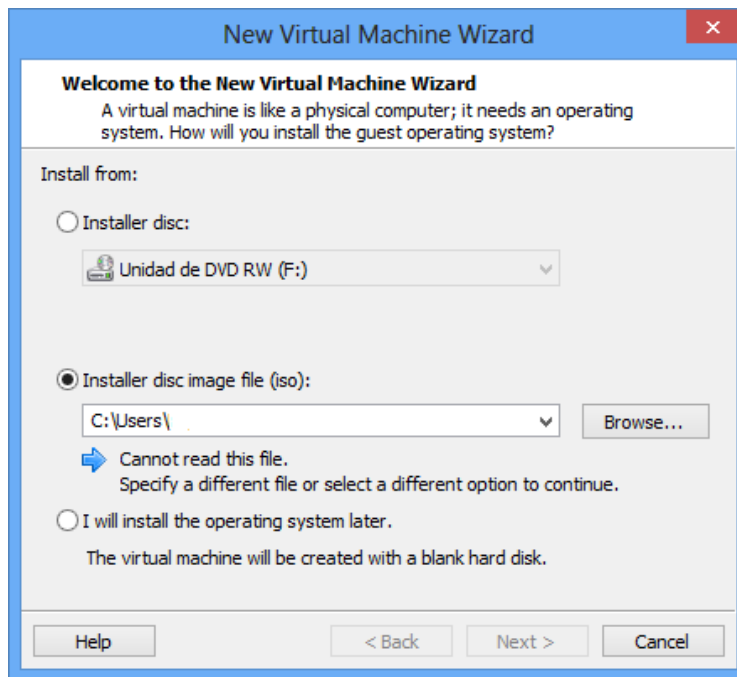


Figura 25. Selecció ubicació sistema operatiu

Aquí ens demana el sistema operatiu que volem instal·lar i que cerquem per tal de donar-li la ruta. També podem posposar la instal·lació per a més tard seleccionant l'opció que tenim a sota. Un cop escollida l'opció premem sobre següent (Next) i ara se'ns obre una interfície que ens demana el tipus de sistema operatiu i la versió. Ho podem veure en la següent imatge:

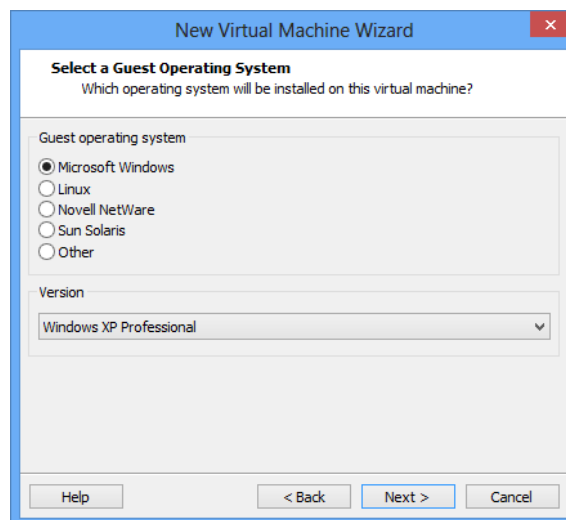


Figura 26. Selecció tipus S.O. i versió

Escollim Microsoft Windows i la versió Windows XP Professional. Continuem i ara ens demana les dades d'instal·lació: la key del producte, el nom d'usuari, contrasenya, etc. Ho podem veure a la imatge següent:

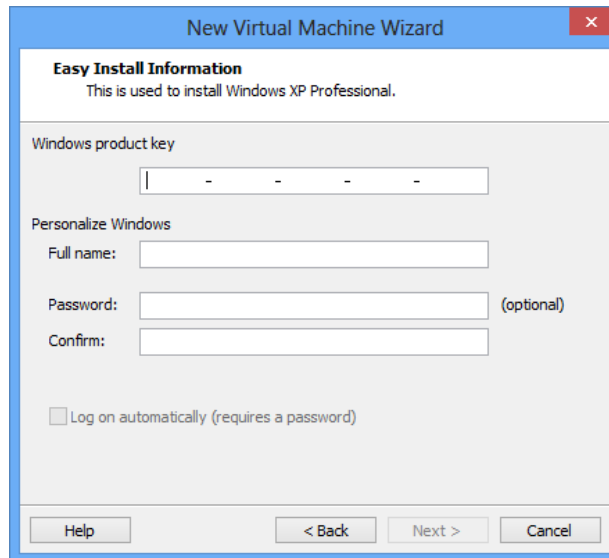


Figura 27. Dades per a l'instal·lació del S.O.

Un cop ple, ens demanarà el nom de la màquina virtual, el màxim d'espai de disc dur que volem que utilitzi i si volem que la guardi en un sol arxiu o en varis. Els valors que presenta per defecta són suficients i ho deixem d'aquesta manera. Ho veiem a la següent imatge:



Figura 28. Selecció espai de disc.

Finalment ens fa un resum de les característiques de la màquina que ens crearà i que podem modificar si hi ha alguna cosa que trobem insuficient. El que farem serà atorgar 2 GB de memòria ja que per defecte ho deixa a 512MB. També traurem la unitat de disquet ja que no en tenim. La següent imatge mostra la interfície de les característiques de la màquina:

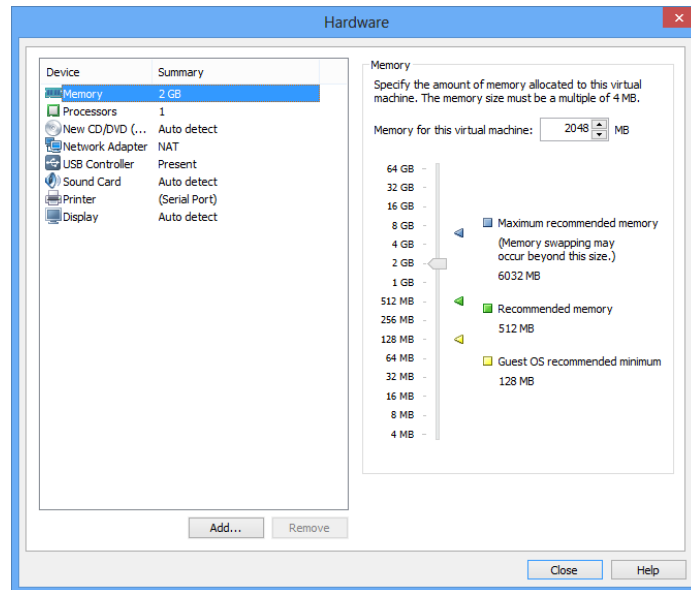


Figura 29. Selecció recursos hardware per la màquina virtual

Ara ja finalitzem i ens sortirà la interfície que ens insta a arrancar la màquina virtual. En fer-ho ens instal·larà el sistema operatiu, i ens l'arrencarà. Quant sortim d'aquest, la propera vegada que arranquem la màquina virtual ja ens engegarà el Windows automàticament i evidentment, el procés serà més curt degut a que bàsicament es comportarà com una arrancada del PC.

4.3.2. – Instal·lació de la IDE Eclipse

Eclipse és una IDE que no requereix cap instal·lació, simplement el descarreguem de la pàgina web propietària²¹, el descomprimim i guardem la carpeta resultant en el lloc que considerem més adient. Dintre de la carpeta disposem d'un executable que serà el que ens posa en marxa l'aplicació. La versió pot ser qualsevol (Indigo, June, Kepler, etc.) amb suport JEE (Java Enterprise Edition). En la següent imatge podem veure l'interior de la carpeta amb l'executable, el qual podem crear un accés directe i col·locar-lo en un lloc fàcilment accessible, per tal de poder executar el programa en comoditat:

²¹ <http://www.eclipse.org/downloads/>

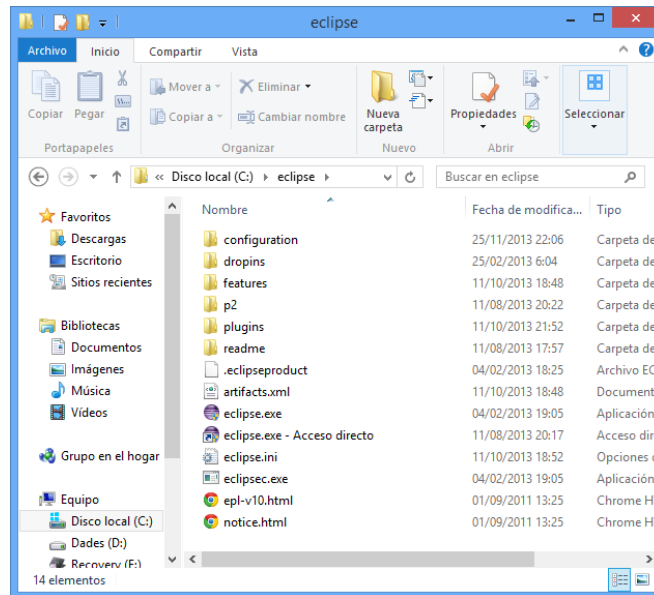


Figura 30. Detall carpeta descomprimida amb Eclipse

4.3.3. – Instal·lació del SDK Android

El SDK d'Android²² (Software Development Kit) serà necessari per a desenvolupar l'aplicació client.

Un cop descarregat el descomprimim i dintre de la carpeta i trobem l'arxiu executable SDK Manager.exe que ens permet instal·lar els paquets que necessitem i actualitzacions. És necessari esperar a que el gestor descarregui la llista d'actualitzacions disponible. La interfície que es mostra, la podem veure a la següent imatge:

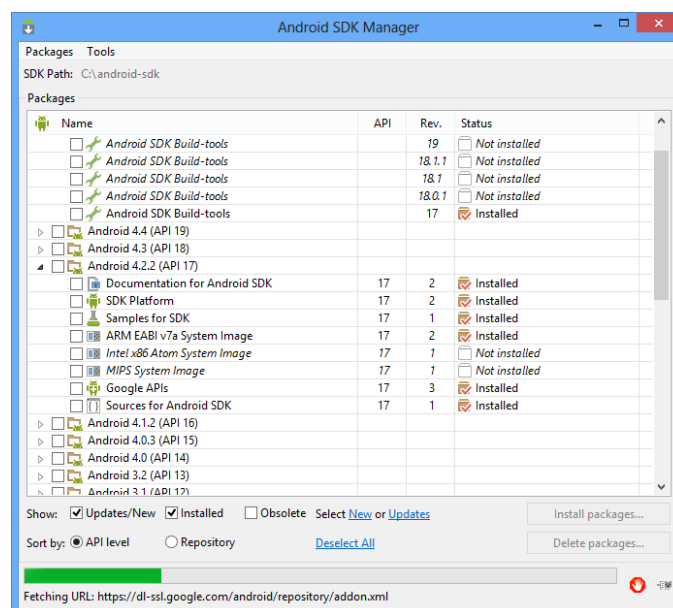


Figura 31. Administrador de paquets d'Android

²² <http://developer.android.com/sdk/index.html#download>

S'opta per instal·lar tots els paquets (els marcarem a les caselles de la part esquerra que hi ha a tal efecte) però en realitat, instal·lant els de la versió per la qual volem desenvolupar el programa ja ni ha prou. Ara es guarda la carpeta en el lloc que es cregui més adient (en el meu cas directori arrel) i recordem on ja que en integrar L'SDK a Eclipse necessitarem dir-li on està ubicada. D'aquesta manera ja tindrem a punt el SDK d'Android.

3.3.4. – Instal·lació del JDK de Java

El JDK de Java²³ (Java Development Kit) són un conjunt d'eines i llibreries que ens permetran desenvolupar el nostre Web Service en llenguatge de programació Java. Existeix per a diferents plataformes i també disposa del JRE (Java Runtime Environment) per a poder executar i provar el codi que anem escrivint. Un cop descarregat, descomprimim l'arxiu i l'executem. Ens demanarà el que volem instal·lar i ho deixem tot per defecte. També tenim l'opció de canviar el directori on s'instal·larà. Ho podem veure en la següent imatge:

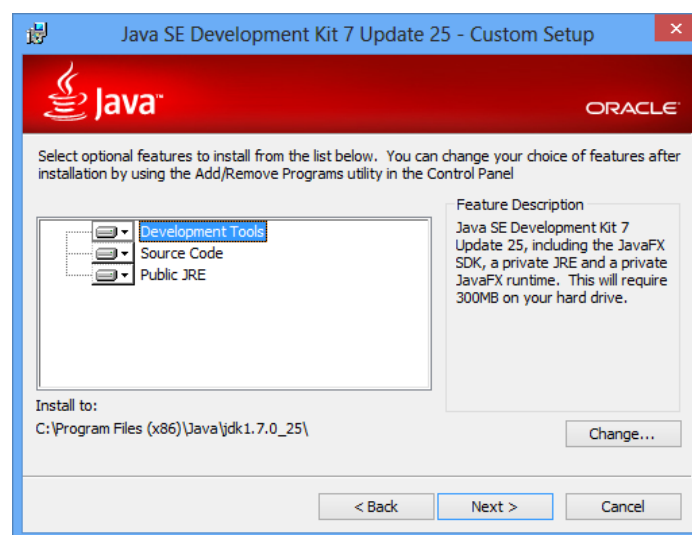


Figura 32. Instal·lació del JDK

A partir d'aquí es porta a terme la instal·lació fins que en arribar al final ens anuncia que el JDK ja està correctament instal·lat en el nostre ordinador, arribant d'aquesta manera al final.

3.3.5. – Instal·lació de MySQL i MySQL-Query - Browser

La base de dades que s'utilitza finalment és MySQL. Ens serà útil i suficient per les dades que volem guardar i instal·larem una versió que ja té cert temps però que és molt estable i lleugera cobrint totes les nostres necessitats. També descarregarem un editor per a poder crear de manera fàcil les taules i editar-les: es tracta de mysql_query_browser.

²³ <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>

Tant el sistema de gestió de base de dades en la versió 5.5.29²⁴ com l'editor²⁵ els podem descarregar de les seves web.

Un cop tot descarregat instal·larem primer el sistema de gestió executant l'arxiu amb extensió .msi que ens hem baixat.

La instal·lació de MySQL la iniciem executant l'arxiu obtingut i ens presenta la següent interfície:

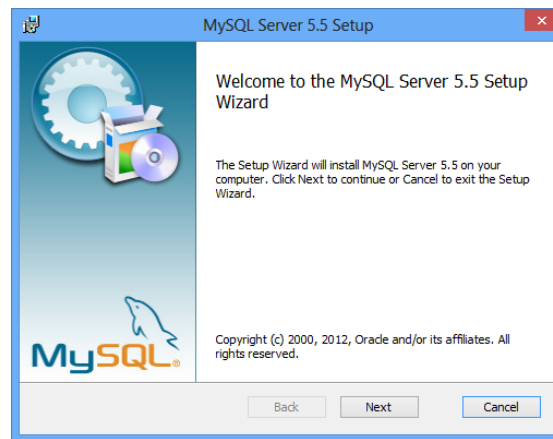


Figura 33. Inici instal·lació MySQL

Continuant amb la instal·lació ens apareix la següent interfície:

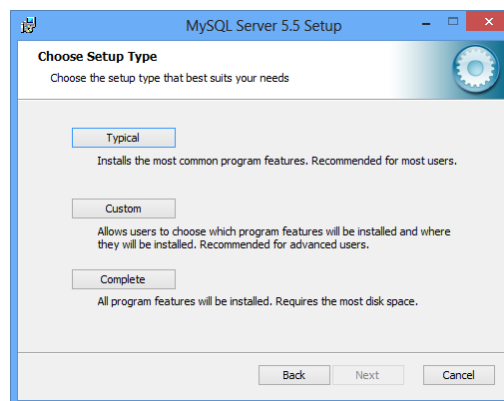


Figura 34. Selecció tipus d'instal·lació

²⁴ http://www.filehippo.com/es/download_mysql/download/435aa7afc7eef27bd731a3b2eba3ec72/

²⁵ <http://downloads.mysql.com/archives/query/>

Seleccionem la instal·lació típica, seguim fins a obtenir la següent interfície:

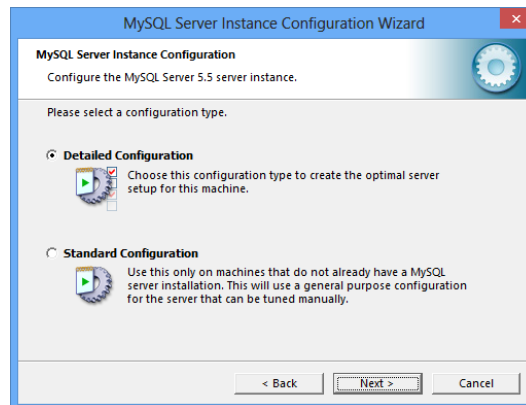


Figura 35. Selecció detall instal·lació

Seleccionem la configuració detallada per a poder seleccionar preferències de configuració i seguim fins a obtenir la següent interfície:

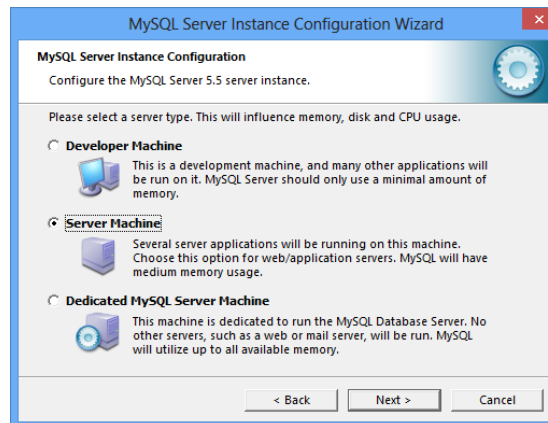


Figura 36. Selecció tipus server

Ara seleccionem que s'instal·li com a servidor i continuem fins a la següent interfície:

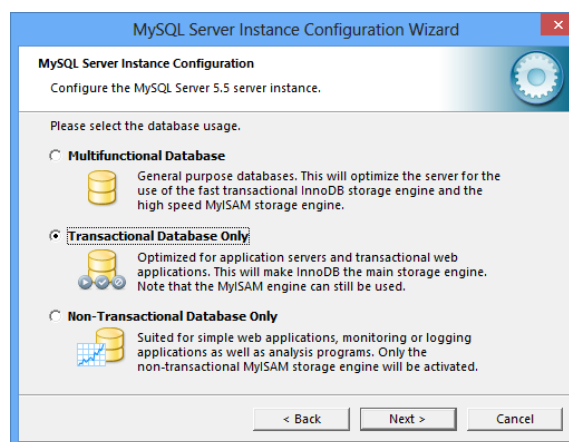


Figura 37. Tipus d'ús de la base de dades

Aquí seleccionem que utilitzarem la base de dades com a base de dades transaccional solsament. Continuem fins obtenir la següent interfície:

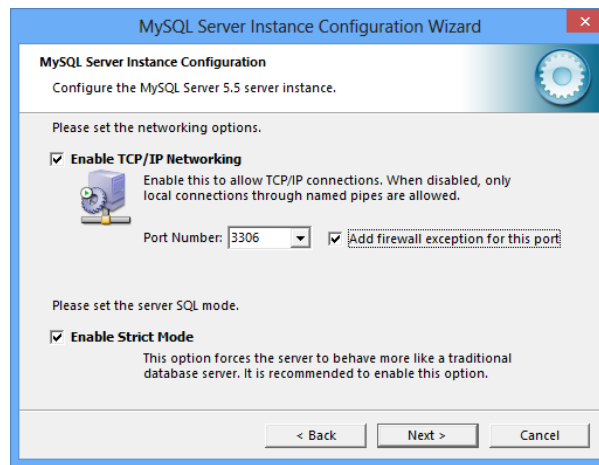


Figura 38. Configuració de port.

Deixem el port 3306 que és el que ens suggereix per defecte i marquem la casella que ens generi una excepció a les regles del tallafocs (per si decidim en fer proves, accedir des de l'exterior). Continuem fins a la següent interfície:

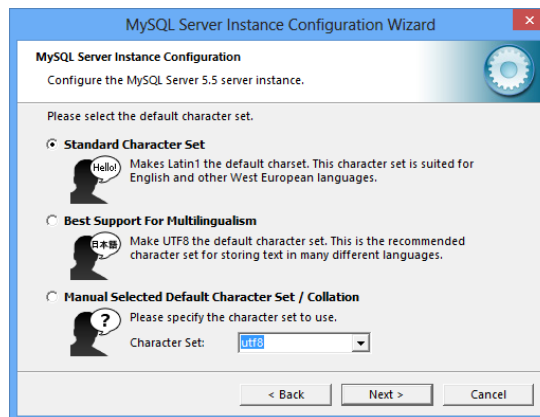


Figura 39. Configuració notació

Aquí seleccionem que utilitzem notació UTF-8 per compatibilitat de llenguatge. Continuem fins a la propra interfície:

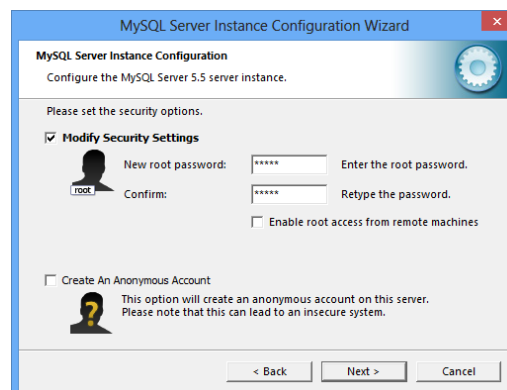


Figura 40. Configuració contrasenya

Posem una contrasenya que en el meu cas és “admin”. Seguir fins la propera interfície:

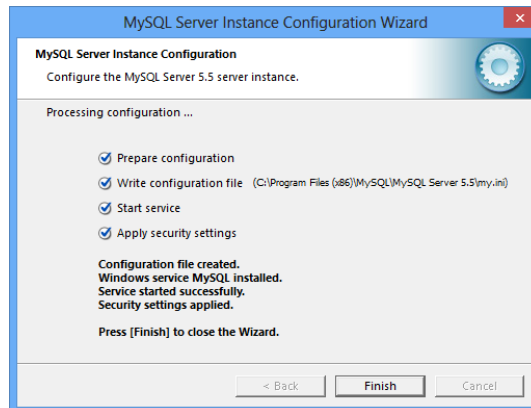


Figura 41. Base de dades configurada

Aquí ja hem arribat al final i podem veure com arrenca el servei, aplica la seguretat i les configuracions. En aquest punt, ja tenim MySQL instal·lat i en funcionament. Ara instal·larem MySQL-Query-Browser. Per començar, executem l'arxiu que hem descarregat. Ens presenta la següent interfície:



Figura 42. Instal·lació MySQL Query Browser

Continuem i la següent interfície que se'ns mostra ens convida a acceptar la llicència. Ho fem (si és que el volem instal·lar) i arribem a la següent interfície:

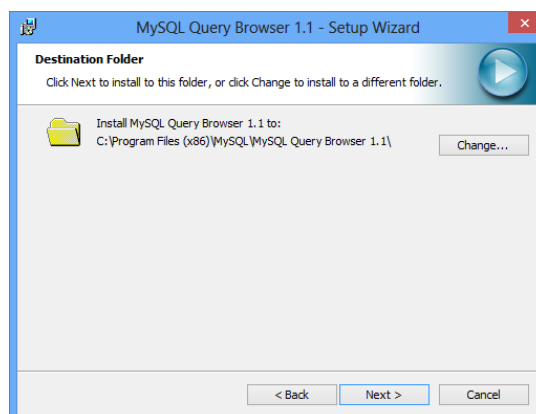


Figura 43. Selecció ruta instal·lació

Aquí se'ns dóna la possibilitat de canviar el directori d'instal·lació. Continuem i arribem a la següent interfície:

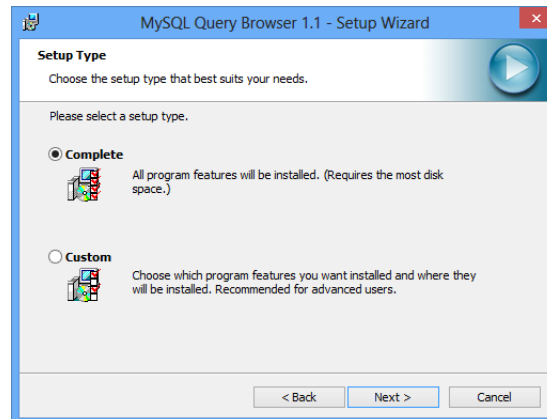


Figura 44. Selecció tipus instal·lació

Ens deixa seleccionar el tipus d'instal·lació. Ho deixem en complerta, continuem i arribem al final de la instal·lació. En aquest punt ja tenim el programa instal·lat.

3.3.6. – Instal·lació de Tomcat

En aquest apartat veurem com instal·lar el servidor de servlets Apache Tomcat. Aquí s'ubicarà el Web Service que oferirà els seus serveis i que estarà gestionat per aquest servidor. El primer que farem serà descarregar la versió que instal·larem de Apache Tomcat²⁶ i que he triat la 6.0.37 per a la seva estabilitat. De totes maneres també es podria treballar amb la 7.0 o fins i tot amb la 8.0 que no fa massa que ha sortit (les últimes versions però, sempre poden presentar algun problema d'estabilitat). Un cop descarregat l'arxiu l'executem per a instal·lar-lo. Ens surt la pàgina d'inici, acceptem els termes de llicència i seleccionem el que volem instal·lar de la interfície que se'ns presenta i que és la següent:

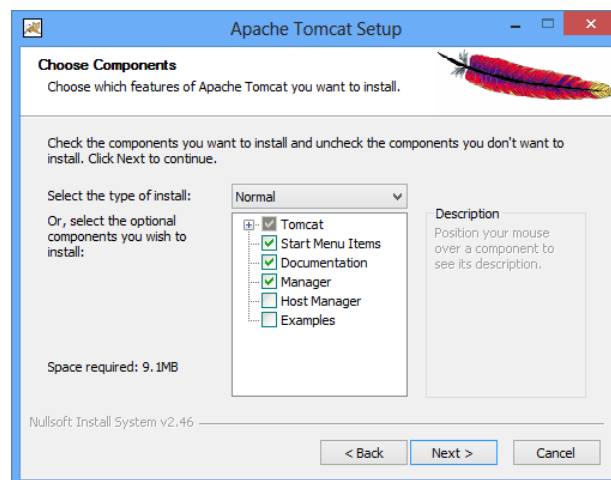


Figura 45. Selecció components Tomcat

²⁶ <http://tomcat.apache.org/download-60.cgi>

A la següent interfície que se'ns presenta podem escollir els ports en els quals escoltarà el servidor. També podem posar una contrasenya per entrar a la part administrativa i triar el nom en que veurem el servei dintre de Windows. Es poden deixar els per defecta o triar-ne d'altres. Aquí triarem el 8887, 8888 i 8889, per la simple raó de que hi han altres màquines corrent amb el port per defecta ocupat. També posem una contrasenya per l'entrada a la part administrativa, doncs de no fer-ho, no podríem entrar i tindríem que editar l'arxiu de configuració per posar-ni una. La interfície és la següent:

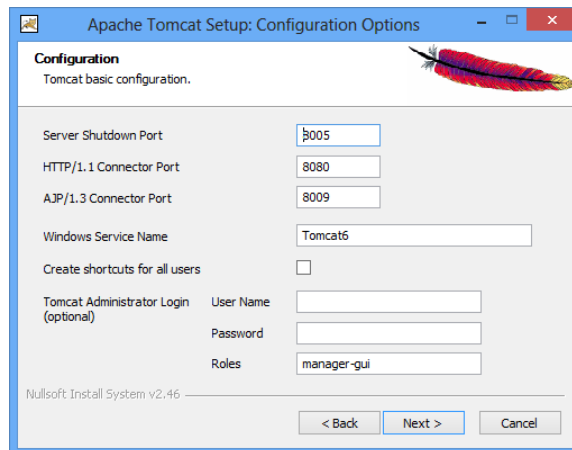


Figura 46. Selecció de ports del Tomcat

Seguidament se'ns dóna l'opció de canviar la ruta d'instal·lació i s'instal·la. Un cop instal·lat el podem configurar perquè arrenqui en inicialitzar el sistema o fer-ho nosaltres de manera manual. D'aquesta manera ja tenim instal·lat i en funcionament el servidor.

5.- Desenvolupament del programari

En aquest apartat es podrà veure la manera en que s'ha desenvolupat el programari, el que ha fet falta per a començar la tasca, les decisions finals que s'han pres abans de començar i tot el que ha fet falta i s'ha anat definint.

5.1. – Descripció. Integració de l'SDK Android i Tomcat a Eclipse

En aquest apartat es descriuen les línies base del desenvolupament del producte final, les decisions que es van prenent, i com va evolucionant. També es mostra de quina manera s'integren les eines amb l'objectiu de fer possible un desenvolupament complet i compacte. Es detallaran les proves realitzades en cada funcionalitat implementada per a poder assegurar en un elevat percentatge que les coses responen de la manera esperada i que el programari no queda en estats inestables, o presentant resultats incongruents.

L'IDE Eclipse serà l'eina de desenvolupament tant de l'aplicació client del producte com del Web Service o part del servidor que proveirà de serveis a la primera. La part client, es desenvolupa per a plataforma Android, per tant necessitarem integrar el SDK per tal de poder desenvolupar el software i també ens permetrà fer les proves pertinents en un espai de simulació, per assegurar que les funcionalitats implementades responen de manera coherent a les especificacions. Pel desenvolupament de la part servidor, s'utilitza el llenguatge de programació Java que el tenim integrat a Eclipse i amb el suport del JDK, aquí la única cosa a integrar és el servidor de pàgines que en el nostre cas i com ja hem mencionat anteriorment serà Apache Tomcat.

5.1.1. – Integració de l'SDK Android

Per integrar l'SDK d'Android a Eclipse seguirem els següents passos:

Dintre de Eclipse seleccionem el menú "Help" i dintre el desplegable seleccionem "Install New Software". Seleccionem "Add..." a la part superior i escrivim el següent: a "Name" i podem posar Android, a "Location": <http://dl-ssl.google.com/android/eclipse/> i seleccionem a "ok". Amb això ens baixarà les eines del repositori d'Android que instal·larem en el nostre Eclipse. Ho podem veure a la següent imatge:

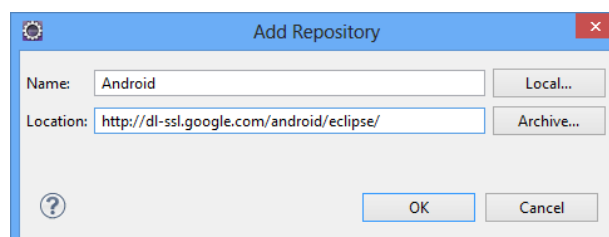


Figura 47. Afegim ruta del repositori

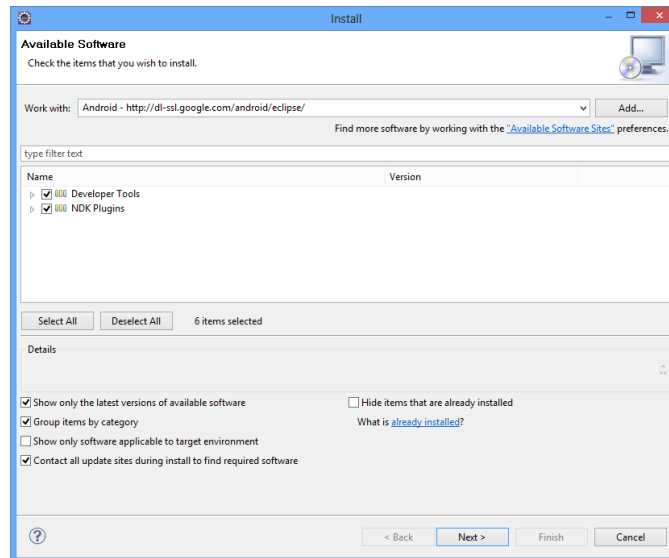


Figura 48. Elements per a descarregar

Seleccionem el dos ítems que ens arriben del repositori i premem el botó “Next”. Ara arribem a la següent pantalla que ens mostra els elements que s’instal·laran:

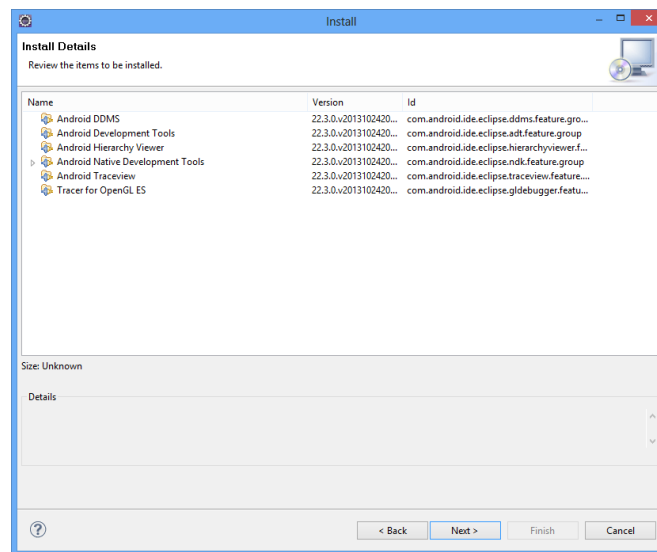


Figura 49. Software a instal·lar

Premem un altre cop sobre “Next”, a la següent pantalla acceptem els termes de llicència i premem sobre “Finish”. Amb això es descarregarà tot el software d’Android que s’integra a Eclipse. Finalment ens demanarà de reinicialitzar Eclipse, acceptem i un cop reinicialitzat el configurem perquè pugui accedir al SDK que ja havíem instal·lat. Anem al menú “Windows”, seleccionem “Preferences” i a l’apartat “Android” seleccionarem la ruta on havíem guardat el SDK que ens havíem descarregat. Un cop fet això premem sobre “Apply” i ens mostrarà el SDK instal·lat. Acceptem i ja tenim el SDK d’Android integrat a Eclipse. Podem veure en la següent imatge aquesta última interfície:

5.2. – Decisions finals sobre el disseny i eines

En els apartats anteriors hem descrit les eines que s'utilitzaran pel desenvolupament del programari. Aquest el podem dividir en dos blocs: un és el de l'aplicació en si pels dispositius mòbils amb sistema operatiu Android. L'altre és un Web service que donarà suport a l'aplicació gestionant i operant sobre tot el conjunt de dades tant produïdes com requerides per l'aplicació. Per a desenvolupar l'aplicació tant d'un bloc com de l'altre, utilitzem una IDE que és, com ja s'ha mencionat, Eclipse. El contenidor del Web service serà Apache Tomcat. Per la persistència de les dades utilitzem el sistema de gestió de base de dades MySQL. Les dades geogràfiques, tant les coordenades de les poblacions com els punts de les rutes, els cercarem via Web mitjançant dos serveis externs: Nominatim de OSM i l'API de Routing de CloudMade. La representació dels mapes amb la ruta traçada ho farem mitjançant la llibreria de codi obert MapsForge que també s'ha vist i comentat en l'apartat 2.4.1. En els següents capítols s'expliquen els aspectes més rellevants del codi que es desenvolupa.

5.3. – Construcció de la base de dades

La construcció de la base de dades la farem d'acord amb el model de dades descrit en el capítol de disseny 3.3.3. Començarem creant un nou esquema que li direm "rutejar" i construïm 6 taules que les anomenem de la següent manera: ruta, trajectepublicat, registre puntsdepas, puntsdenopas, localitats, i comentariskonductor.

La taula "ruta", ens emmagatzemarà les dades que utilitzarà l'aplicació tant per mostrar el trajecte sobre un mapa com per donar informació sobre el temps i distància estimats de la ruta. Aquests atributs, el tipus que són i la seva descripció la podem veure en la següent taula:

Nom taula: ruta		
Nom atribut	Tipus	Descripció
id	int	Identificador de trajecte autogenerat.
origen	varchar	Nom localitat origen de la ruta.
destinacio	varchar	Nom localitat destinació de la ruta.
distancia	int	Distància de la ruta.
temps	int	Temps estimat en realitzar-la.
punts	mediumtext	Punts geogràfics per on passa la ruta.

La taula "trajectepublicat" emmagatzema les dades referents a un trajecte que un usuari conductor publica. Són les dades bàsiques que caracteritzen el trajecte i que ja hem descrit en l'apartat de model i disseny. Les podem veure en La següent taula:

Nom taula: trajectepublicat		
Nom atribut	Tipus	Descripció
id	int	Identificador de trajecte autogenerat.
origen	varchar	Nom localitat origen de la ruta.
destinacio	varchar	Nom localitat destinació de la ruta.
data	varchar	Dia, mes i any en que es realitzarà el trajecte.
hora	int	Hora de sortida.

minut	int	Minut de sortida.
model_cotxe	varchar	Model del cotxe en que es fa el trajecte.
places	int	Número de places disponibles.
preu	double	Preu del trajecte per passatger.
e_mail	varchar	Adreça del correu electrònic del conductor.
permes_fumar	boolean	A 1 indica que està permès de fumar dintre del cotxe.
radio_mp3	boolean	A 1 indica que el cotxe disposa de radio i mp3.
distancia	int	Distància de la ruta.
temps	int	Temps estimat en realitzar-la.

La taula “registre” emmagatzemarà totes les dades personals d’un usuari que es registri. Es poden veure a la següent taula juntament amb el seu tipus i la descripció:

Nom taula: registre		
Nom atribut	Tipus	Descripció
id	int	Identificador del registre autogenerat.
nom	varchar	Nom de l’usuari.
cognoms	varchar	Cognoms de l’usuari.
e_mail	varchar	Adreça del correu electrònic de l’usuari.
any_naixement	int	Any de naixement de l’usuari.
telefon	varchar	Número de telèfon de l’usuari.
contrasenya	int	Contrasenya elegida per l’usuari.
sexe	boolean	A 1 → sexe femení.
telefon_public	boolean	A 1 → es pot facilitar el número de telèfon a un altre usuari.
E_mail_public	boolean	A 1 → es pot facilitar l’adreça de correu a un altre usuari.

Quant un usuari cerca un trajecte, es porta a terme la cerca entre tots els que estan publicats. Les coincidències entre origen i destinació en la major part del casos no es donaran, iniciant-se llavors una cerca per punts, és a dir, compararem les coordenades de origen i destinació amb els punts per on passa la ruta per veure si hi estan inclosos. Aquesta taula emmagatzema poblacions que pertanyen a una ruta i que no coincideixen amb el seu origen ni destinació. Això estalviarà posteriorment tornar a fer-ne la cerca per punts. El detall de les dades que emmagatzema aquesta taula els podem veure a continuació:

Nom taula: puntsdepas		
Nom atribut	Tipus	Descripció
id	int	Identificador de punt de pas autogenerat.
origen	varchar	Nom localitat origen de la ruta.
destinacio	varchar	Nom localitat destinació de la ruta.
idtrajecte	int	Id del trajecte al qual pertany.

La idea d’aquesta taula (puntsdenopas) està basat en el concepte de la taula anterior però per les poblacions que no pertanyent a una ruta.

En podem veure el detall i descripció dels seus atributs a continuació:

Nom taula: puntsdenopas		
Nom atribut	Tipus	Descripció
id	int	Identificador de punt de no pas autogenerat.

origen	varchar	Nom localitat origen de la ruta.
destinacio	varchar	Nom localitat destinació de la ruta.
idtrajecte	int	Id del trajecte al qual pertany.

La finalitat de la taula "localitats" és la de guardar les coordenades d'una població per tal de no haver de tornar a cercar-les fora quan es tornin a necessitar. Les dades i descripció es detallen a la següent taula:

Nom taula: localitats		
Nom atribut	Tipus	Descripció
nom	varchar	Nom de la localitat.
Longitud	double	Coordenada longitud de la localitat.
latitud	double	Coordenada latitud de la localitat.

La taula "comentarisconductor" guarda les dades referents a un comentari que se li fa a un conductor. Les podem veure detallades a la següent taula junt amb la seva descripció:

Nom taula: comentarisconductor		
Nom atribut	Tipus	Descripció
id	int	Identificador del comentari autogenerat.
nom	varchar	Nom del conductor.
cognoms	varchar	Cognoms del conductor.
e_mail	varchar	Correu electrònic del conductor.
puntuacio	int	Puntuació atorgada en el conductor.
comentari	varchar	Comentari fet al conductor

Aquesta estructura respon a la necessitat d'emmagatzemar els diferents tipus de dades. Cada taula guarda i subministra informació relacionada amb cadascuna de les funcionalitats que l'aplicació ofereix. En la següent figura, es mostra de manera gràfica aquestes relacions i interaccions entre les funcionalitats o operacions amb les diferents taules que conformen la base de dades:

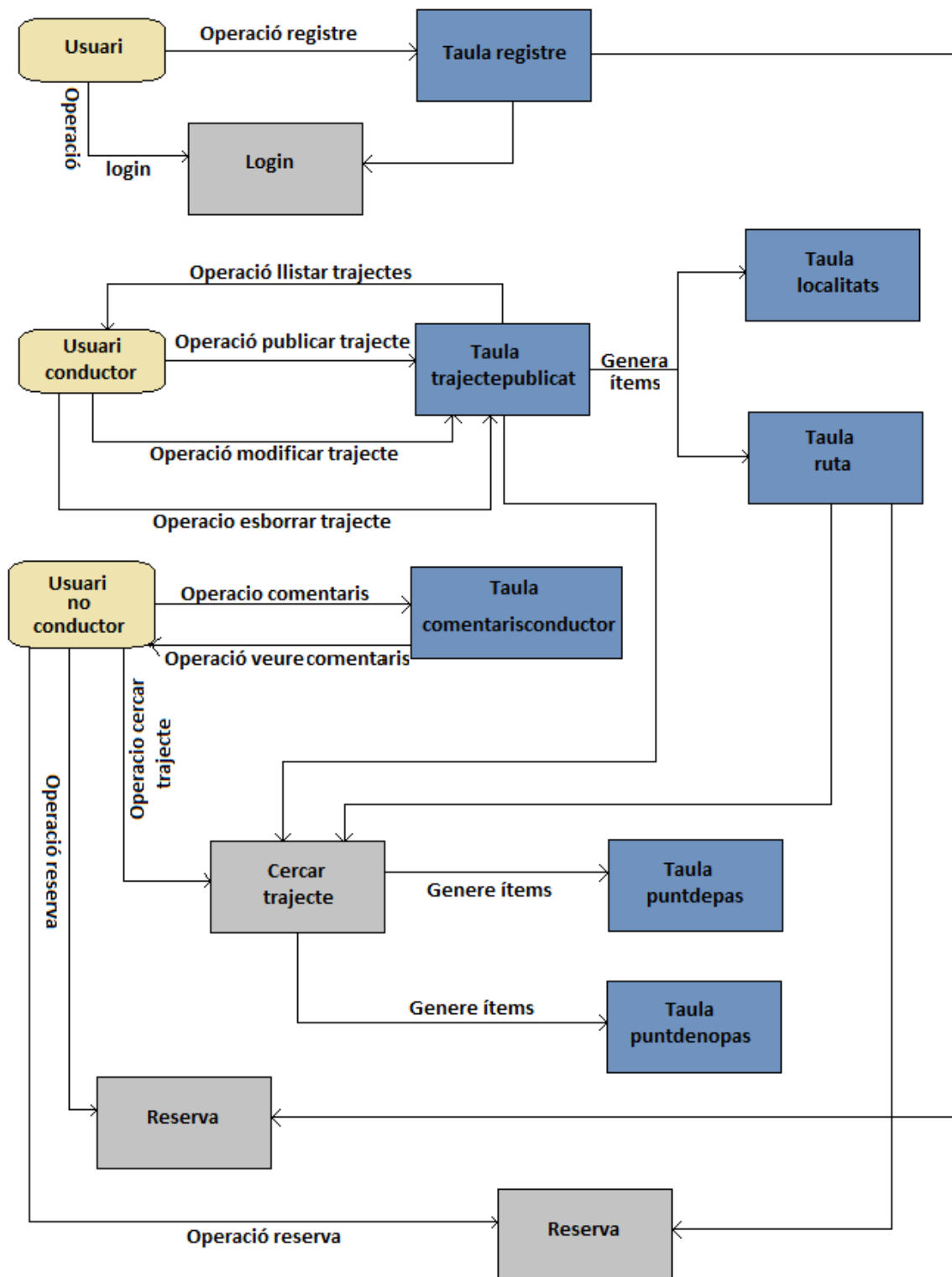


Figura 52. Relació operacions – dades.

5.4. – Construcció del Web Service

D'una manera molt simple podríem definir un Web service com un màquina que espera peticions de clients i en resposta, els subministra els recursos que aquests sol·liciten dintre dels disponibles. Hi ha uns estàndards reglamentats en el món dels Web Service²⁷, nosaltres però ens el farem a la nostra mida degut a que aquest, no tindrà que publicar serveis ni serà de caràcter públic, el contrari serà particular i concret per a l'aplicació i per tant, no té que ser visible més enllà d'aquesta.

Per a muntar el nostre Web service, farem ús del patró d'arquitectura de software MVC (Model – Vista – Controlador). Aquest model, separa la lògica de negoci de la interfície d'usuari i encara que normalment relacionem la vista amb pàgines html, jsp, php, etc. en aquest cas no serà així i simplement es limitarà a enviar la informació d'usuari en format de text pla i com objectes JSON²⁸. La següent imatge mostra gràficament el concepte d'aquest patró:

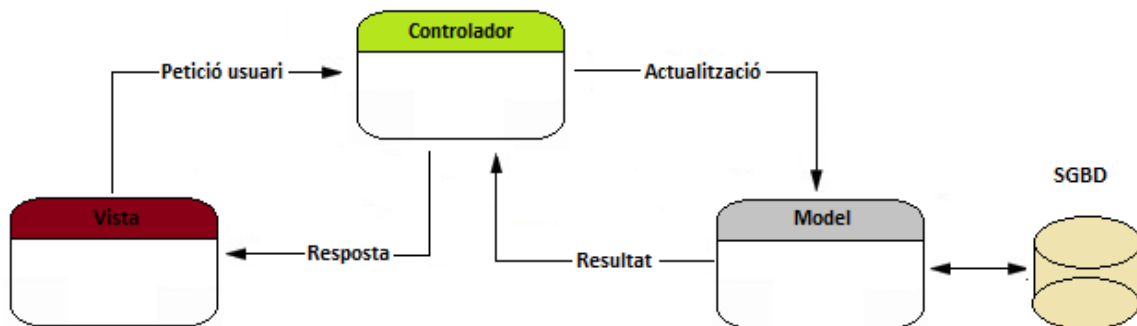


Figura 53. Esquema conceptual MVC

Model

Com ja s'ha comentat, el model conté la lògica de negoci i les dades que són objecta de l'aplicació i que li donen sentit. En aquesta part hi trobem tots els mètodes de tractament i manipulació de les dades i la interacció amb la base de dades que és un element fonamental per a donar persistència en aquestes. En aquest bloc s'han programat els mètodes que gestionen les dades i que fan possible l'accés a la base de dades, així com les diferents entitats. S'han dividit en 5 paquets: `cat.pepweb.pep.beans`, `cat.pepweb.pep.jdbc`, `cat.pepweb.pep.negoci`, `cat.pepweb.pep.utilitats` i `org.json`. Passem a fer una enumeració de les classes de cada paquet junt amb una descripció del seu contingut i funcionalitat.

²⁷ http://es.wikipedia.org/wiki/Servicio_web

²⁸ <http://es.wikipedia.org/wiki/JSON>

Nom del paquet: cat.pepweb.pep.beans	
Classe	Descripció
Comentari.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquesta que defineixen un objecte d'aquesta classe.
Localitats.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.
Ruta.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.
Trajectes.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.
Usuari.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.
Puntdepas.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.
Puntdenopas.java	Conté els atributs, constructors i mètodes set i get de cadascun d'aquests que defineixen un objecte d'aquesta classe.

Nom del paquet: cat.pepweb.pep.jdbc	
Classe	Descripció
Connexio.java	Aquesta classe gestiona la connexió amb la base de dades.
OperacioAfegirUnComentari.java	Aquesta classe conté els mètodes CRUD i d'específics per a poder fer consultes molt concretes a la taula de la base de dades comentaris.
OperacioLogin.java	Aquesta classe conté l'operació de consulta a la taula registre per verificar posteriorment el nom d'usuari (e_mail en aquest cas) i la contrasenya.
OperacionsLocalitats.java	Aquesta classe conté els mètodes per a crear un nou ítem a la taula localitats de la base de dades i per llegir-ne un de determinat.
OperacionRegistreUsuari.java	Aquesta classe conté els mètodes que permeten crear un nou ítem a la taula registre i recuperar informació de ella tal com les dades d'un usuari .
OperacionsRuta.java	Aquesta classe disposa dels mètodes adients per a crear un nou ítem al la taula ruta i per llegir-ne de determinats o extreure'n un llistat.
OperacionsTrajectesUsuaris.java	En aquesta classe hi podem trobar els mètodes necessaris per a crear nous ítems a la taula trajectepublicat, per a modificar-ne, per esborrar-ne i per extreure'n una llista.
OperacionsPuntsDePas.java	En aquesta classe hi trobarem tots els mètodes necessaris per a crear nous ítems a la taula puntdepas, i també per a fer consultes amb uns determinats criteris.
OperacionsPuntsDeNoPas.java	En aquesta classe hi trobarem tots els mètodes necessaris per a crear nous ítems a la taula puntdepas, i també per a fer consultes amb uns determinats criteris.

Nom del paquet: cat.pepweb.pep.negoci	
Classe	Descripció
AfegirComentari.java	Aquesta classe gestiona les operacions necessàries i el flux de dades per a portar a terme la creació d'un nou comentari a la taula comentaris de la base de dades, amb les dades rebudes.
AfegirTrajecte.java	Aquesta classe gestiona les dades i operacions necessàries per donar d'alta un nou trajecte a la taula trajectepublicat de la base de dades.
EsborrarTrajecte.java	Aquesta classe gestiona la funcionalitat d'esborrar un trajecte de la taula trajectepublicat de la base de dades.
Login.java	Aquesta classe Gestiona l'autenticació de l'usuari cercant les dades a la taula registre de la base de dades.
ModificarTrajecte.java	Aquesta classe gestiona mitjançant les operacions necessàries el canvi d'alguns atributs d'una tupla de la taula trajectepublicat de la base de dades.
Registre.java	Aquesta classe gestiona la creació d'una nova tupla a la taula registre de la base de dades.
CercarComentaris.java	Aquesta classe gestiona totes les operacions necessàries per retornar una llista amb els comentaris que s'han fet d'un determinat conductor, i les seves puntuacions. S'extreu la informació de la taula comentarisconductor.
CercarPunts.java	Aquesta classe gestiona les operacions necessàries per retornar tots els punts pels quals passa una ruta. Els extreu de la taula ruta.
CercarTrajectes.java	Aquesta classe efectua totes les operacions necessàries per retornar una llista amb els trajectes que coincideixen amb els publicats segons un origen o destinació concrets. Interactua amb les taules localitats, trajectepublicat, puntdepas i puntdenopas.
LlistarTrajectesUsuari.java	Aquesta classe retorna, mitjançant les operacions necessàries, una llista amb els trajectes que un usuari té publicats. Interactua amb la taula trajectepublicat.
Reserva.java	Aquesta classe retorna mitjançant les operacions necessàries, un número de telèfon i adreça de correu electrònic (si s'escau), d'un usuari conductor. Interactua amb la taula registre.

Nom del paquet: cat.pepweb.pep.utilitats	
Classe	Descripció
CercarCoordenades.java	Aquesta classe conté els mètodes necessaris per a fer una gestió de dades encaminades a retornar les coordenades d'una localitat guardades en una llista. No interactua amb cap taula de la base de dades i es tracta d'una classe auxiliar utilitzada per altres classes.
CercarRuta.java	Aquesta classe conté els mètodes necessaris per a fer una gestió de dades encaminades a retornar els punts de pas d'una ruta, distància i temps estimat encapsulat dintre d'una llista. No interactua amb cap taula de la base de dades i es tracta d'una classe auxiliar utilitzada per altres classes.
GestioDates.java	Aquesta classe proporciona funcionalitats de format de dates i transformació de d'String a date i a l'inrevés. És utilitzada per altres classes que necessiten d'aquest serveis.

- En el paquet `org.json`²⁹ i podem trobar 17 classes que donen suport (almenys algunes d'elles) a codificar i descodificar informació en format JSON. Aquest paquet, de codi obert, no és propi i s'ha incorporat per aquest propòsit. S'ha deixat d'aquesta manera com a classes de codi font però en realitat es podria importar com una llibreria.

Controlador

El controlador és l'element intermediari entre el client (o aplicació client) i la capa del model. Quant l'aplicació (en aquest cas) requereix d'un servei, el controlador rep la petició, mira de quina petició es tracta, recull les dades i passa tota la informació en el model que serà l'encarregat de realitzar les operacions necessàries per a subministrar una resposta adient al servei que el client ha requerit. Llavors, quan el model té la resposta, l'envia a la capa controlador i aquesta pren les dades, i les subministra a l'usuari (en aquest cas aplicació). Si es tractés d'un servidor web la resposta amb tota probabilitat seria en codi html, php, etc. per tal que l'usuari, en el navegador (programari amb el qual interactuaria amb el servidor), se li obrís una pàgina ja que aquest és capaç d'entendre i executar dit codi. No és el nostre cas ja que aquí l'aplicació no interactua amb el Web service des d'un navegador sinó que ho fa invocant un servei directament, per tant les dades que rebrà seran diferents i en el nostre cas serà de text pla i objectes empaquetats en format JSON (si invoquéssim el servei des d'un navegador com a resposta veuríem les dades amb text pla). Llavors el nostre controlador està format per un servlet que és l'encarregat d'atendre les peticions mitjançant els seus mètodes GET i POST. Amb el primer mètode escolta les peticions en les quals les dades arriben en la pròpia adreça URL, i amb el segon, les dades que arriben empaquetades en format JSON. El primer mètode l'usem quant el número de dades a enviar és molt petit (màxim 3) i l'altre quan la mida d'aquestes és considerable (per exemple les dades d'un registre, trajecte, etc.). També des del controlador s'envien dades en text pla o empaquetades depenent de la seva longitud (per exemple amb text pla quan s'envia un codi per a saber si l'operació ha tingut èxit o empaquetades quan s'envia per exemple un llistat de trajectes). El paquet `cat.pepweb.pep.controlador` disposa de la classe `Controlador.java` que és la que proporciona tota aquesta funcionalitat.

Vista

Tal com ja hem dit, aquí no podem parlar de vistes pròpiament perquè no s'interactua amb un navegador que ens presenta el resultat de les peticions en format de pàgina web sinó que directament, ens subministra dades en text pla i empaquetades en format JSON. El format de les respostes les proporciona el propi controlador.

5.4.1 – Algorisme de cerca de trajectes

Una part important en les funcionalitats que proporciona el Web service és la cerca de trajectes. Els trajectes que publiquen els usuaris conductors, tenen com a dades importants l'origen i la destinació. Quan un usuari cerca un trajecte, ho fa segons un origen i una destinació que coincideix amb les del seu interès. Moltes vegades però, pot passar que el trajecte cercat no coincideixi amb els que estan publicats però que aquest quedi inclòs dintre d'aquests (o algun

²⁹ <http://json.org/java/>

altre) perquè en realitat són punts de pas. Amb aquesta premissa, el mecanisme a l'hora de cercar un trajecte és el següent:

- Cerquem els trajectes que coincideixin amb els paràmetres origen i destinació del trajecte que s'està cercant a la taula "trajectepublicat" i posem en una llista els que coincideixen.
- Dels trajectes que no han coincidit en el pas anterior, mirem a les taules "puntdepas" i "puntdenopas" si l'origen i destinació del trajecte cercat hi apareixen. Si apareixen a la taula "puntdepas", afegirem el trajecte a la llista del punt anterior. Si apareixen a la taula "puntdenopas", els descartem.
- Finalment, els trajectes restants, compararem per cadascun si les coordenades de l'origen i la destinació (més una tolerància) del trajecte cercat, coincideix amb algun punt del trajecte que es compara (publicat) amb l'ordre origen primer destinació segon. Si i ha coincidència, el trajecte es posa a la llista de trajectes trobats del primer pas, i les poblacions origen, destinació s'afegeixen a la taula "puntdepas" per la ruta en qüestió (d'aquesta manera ampliem els punts de pas de la ruta i això ens estalviarà iteracions sobre els punts d'aquesta en cerques posteriors). Si els punts origen i destinació no coincideixen amb la ruta que es comparen, llavors s'afegeixen a la taula "puntdenopas" per, en cerques posteriors, determinar directament sense necessitat d'iterar sobre els punts de la ruta, que aquell origen i destinació no són punts de pas de la ruta en qüestió.
- Finalment es retorna la llista amb els trajectes que són candidats a la cerca que es fa.

Amb aquest sistema optimitzem, a mesura que aquestes taules que s'han mencionat creixen, les cerques de trajectes degut a que cada cop més (com més cerques diferents es fan) es té millor informació sobre les poblacions per les que passa una ruta determinada.

5.5 – Desenvolupament aplicació per plataforma Android

Pel desenvolupament de l'aplicació crearem un nou projecte Android i a partir d'aquí, la nostra tasca consistirà en anar implementant les vistes (interfícies gràfiques en les quals interactuarà l'usuari) i la lògica de negoci. Les vistes les implementarem d'acord amb el disseny fet i comentat en l'apartat de disseny 3.3.6 i la lògica de negoci respondrà a les funcionalitats també esmentades en l'apartat 3.2 i amb les restriccions imposades. En les diferents vistes per les quals podrà navegar l'usuari, aquest introduirà dades i en veurà. Les dades introduïdes s'enviaran en el web service per a portar a terme la funció escollida i les dades presentades vindran d'aquest com a resposta per tal de satisfer les expectatives de l'usuari. La lògica de negoci s'encarrega de presentar la vista sol·licitada per l'usuari en cada moment, recollir les dades entrades en la vista, verificar que les dades són correctes i complertes abans d'enviar-les en el Web service, establir el diàleg amb aquest codificant cada petició, codificar les dades correctament per a enviar-les, descodificar les que s'han rebut, interpretar-les i preparar-les per a poder ser visualitzades, activar la vista que es correspongui en cada moment. Com podem veure, la construcció de l'aplicació passa per dues etapes: part gràfica que es correspon a la construcció de les vistes i lògica de negoci que és el veritable nucli de l'aplicació.

A l'hora del disseny de la part gràfica, s'intenta utilitzar tipus de Layouts, botons etc. que es puguin adaptar a diferents mides de pantalla, però moltes vegades existeixen diferències força

pronunciades en fer corre l'aplicació en un o altre tipus de dispositiu. En no ser dissenyada per un tipus de dispositiu específic, finalment es pren la determinació que el disseny gràfic s'ajusti perquè es vegin bé totes les interfícies en un emulador de terminal (proporcionat pel mateix SDK d'Android) de les següents característiques: terminal de 4.0'' WVGA (480 x 800 hdpi), amb el target 2.2 – Api level 8 (d'aquesta manera es pot executar amb gairebé totes les versions), amb una RAM de 512 MB i amb memòria interna de 1GB. En aquest terminal es fan també totes les proves de test de l'aplicació per veure que tot funciona de manera correcta.

6.- Aplicació i funcions

En aquest apartat veurem com funciona l'aplicació. Farem un recorregut per totes les seves funcionalitats a través de les interfícies gràfiques que la componen. De fet, podrem veure la materialització del disseny que en el seu moment es va portar a terme per a donar una solució a les expectatives del projecte.

6.1 – Funcionament de l'aplicació

Ara posem en marxa l'aplicació i ho fem en l'emulador que hem configurat dintre les possibilitats que ens ofereix l'SDK Android i les seves característiques ja les hem descrit al final de l'apartat anterior (es poden veure al final de la pàgina anterior).

La interfície gràfica d'usuari que ens presenta en posar-la en marxa és la següent:



Figura 54. Inici

Ara, si un usuari vol accedir en el sistema es tindrà de registrar o si ja ho està s'autenticarà. Llavors se li presentarà la primera o segona interfície segons sigui el cas:



Figura 55. Registre

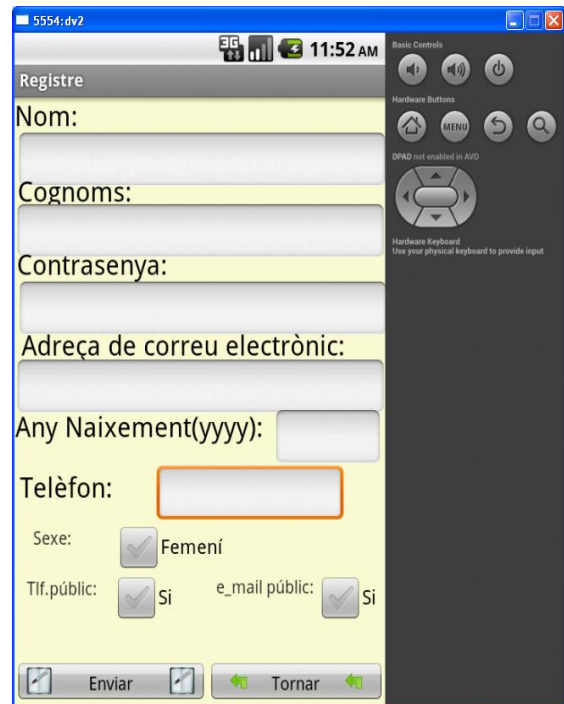


Figura 56. Login

Un cop l'usuari ha accedit en el sistema sigui mitjançant el registre o el Login se li presentarà la següent interfície gràfica que li mostra les opcions que té a disposició:



Figura 57. Opcions usuari

Si es selecciona l'opció publicar trajecte o la de llistar trajectes es mostren les següents interfícies (formulari o llistat de trajectes) segons sigui l'opció escollida:

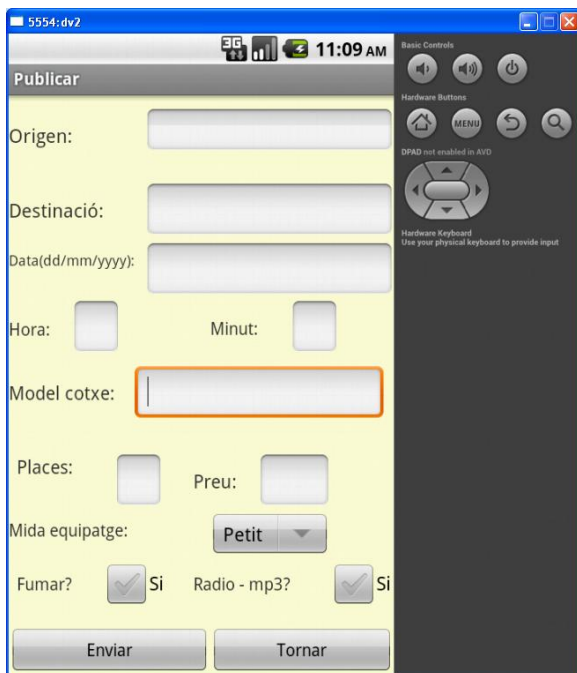


Figura 58. Publicar Trajecte



Figura 59. Llistat trajectes

L'usuari es pot desplaçar per la llista i ara té l'opció de seleccionar un trajecte dels que se li presenten. En fer-ho se li mostrarà la següent interfície amb les dades del trajecte:



Figura 60. Opcions trajecte

Com es pot apreciar, en aquest punt hi ha la possibilitat d'esborrar el trajecte o de fer-hi modificacions. Les modificacions es permetent en tots els camps excepte l'origen i destinació degut a que modificar aquests camps és sinònim de trajecte nou. L'opció esborrar ens presenta una interfície en la qual es demana confirmació abans de fer-ho:



Figura 61. Modificar trajecte

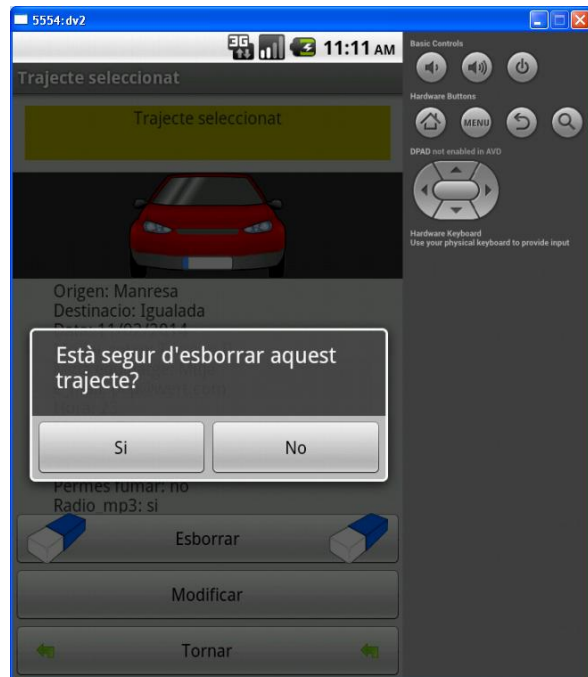


Figura 62. Esborrar trajecte

Si es confirma s'esborra el trajecte i si es confirma modificar es modifica. Després es torna a la interfície que presenta les opcions. En el menú opcions, una altre opció és la de cercar trajecte. En seleccionar-la se'ns presenta la següent interfície gràfica:



Figura 63. Cercar trajecte

Aquí l'usuari entrarà com a dades l'origen i destinació del trajecte que cerca i opcionalment la data. Com a resultat se li presentarà la següent interfície gràfica amb els trajectes que s'han trobat:

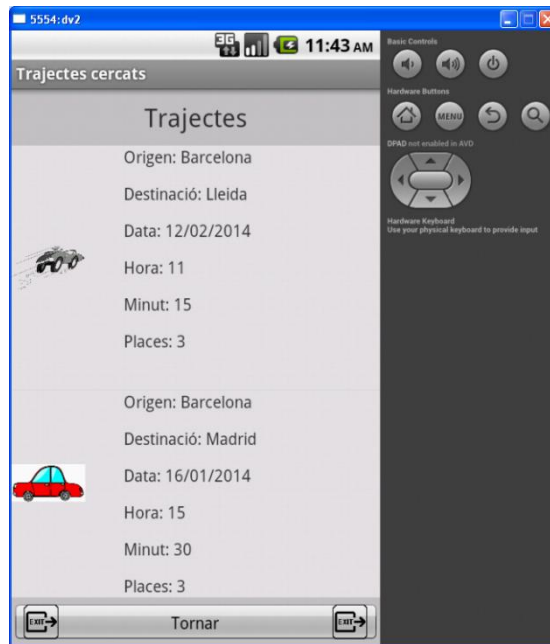


Figura 64. Trajecte cercat

Un cop mostrats els trajectes que s'han trobat en la cerca (en l'imatge que es mostra només s'ha n'ha trobat un), l'usuari en pot seleccionar un i se li mostra la següent interfície gràfica:



Figura 65. Trajecte seleccionat

En aquest punt l'usuari pot veure els comentaris que han fet altres usuaris en el conductor del trajecte seleccionat, fer una reserva i veure el trajecte pintat en un mapa. Si es seleccionen la primer i segona opció se'ns presenten les següents interfícies:



Figura 66. Comentaris a conductor



Figura 67. Reserva

En fer la reserva, es mostra el número de telèfon del conductor i la seva adreça de correu electrònic si en registrar-se ho ha marcat com a públic. En cas contrari, mostra l'adreça de correu del Webmàster ja que llavors la reserva es farà a través d'ell. També es mostra un anunci recordant a l'usuari que guardi l'id del trajecte si posteriorment vol deixar un comentari sobre el conductor ja que el necessitarà.

Si en les opcions anteriors es selecciona l'opció veure mapa, obtindrem la ruta marcada en un mapa. Ho podem veure en la següent imatge:

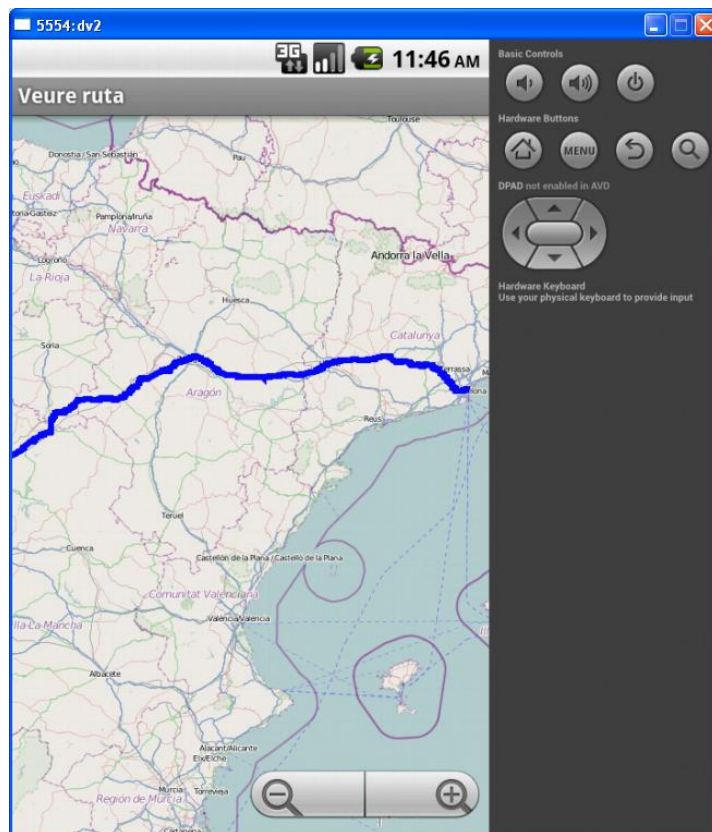


Figura 68. Vista mapa

En el menú opcions també podem triar la funcionalitat de puntuar i posar un comentari per al conductor amb el qual s'hagi fet un determinat trajecte.



Figura 69. Comentari a conductor

Finalment en el menú opcions tenim l'opció ajuda que ens fa una breu però suficient descripció de les diferents funcionalitats:

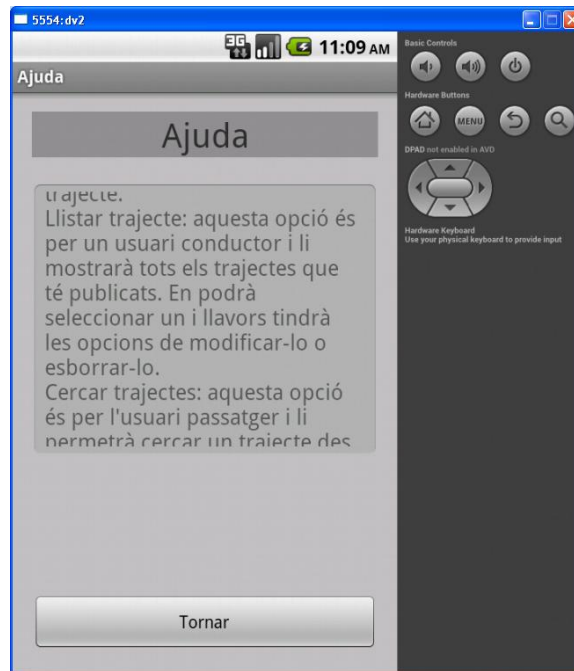


Figura 70. Ajuda

Aquestes són totes les funcionalitats que ofereix l'aplicació i donen resposta als requeriments inicials del projecte. Podem veure que el seu funcionament és compacte i la seva navegabilitat simple i intuïtiva. En totes les diferents interfícies gràfiques que ens ofereix, tenim la possibilitat de tornar enrere o sortir, per facilitar els salts entre les diferents opcions. També s'ha intentat etiquetar els botons al màxim de descriptius per a no crear confusió i fer el màxim d'agradable l'experiència de l'usuari. En les tasques que presumiblement poden requerir més temps, es llença un quadre informatiu avisant d'aquest fet perquè l'usuari no tingui dubtes en quant a un mal funcionament de l'aplicació. Si en alguna operació es produeix un error se n'informa a l'usuari i s'intenta concretar sempre que és possible, el tipus d'error que s'ha produït.

6.2- Futures millores

Aquest projecte, es va pensar per a donar resposta a una necessitat plantejada concreta: construir una aplicació que fes possible el posar en contacte persones que realitzessin trajectes semblants per tal de poder aprofitar el transport compartint-lo. Però també per a complir uns objectius més genèrics com conèixer més a fons el sistema operatiu Android per tal de poder-hi desenvolupar aplicacions, conèixer les eines de codi obert existents per a poder satisfer les necessitats de recursos de l'aplicació, etc (els objectius estan descrits en l'apartat 1.3). El resultat final cobreix les

expectatives marcades en aquests objectius, donant una solució pràctica i efectiva a les necessitats inicials. La solució proposada no és la única. D'entre el ventall que ni han s'analitzen les que poden resultar més efectives, eficients i simples tenint en compte diferents criteris. Finalment, en el transcurs de la pròpia construcció del producte, o en etapes finals, s'aprecien detalls que sense aportar un gran canvi en el resultat, poden aportar alguna millora tant interna (i que probablement no serà apreciada per l'usuari però pot dotar de major robustesa l'aplicació) com externa en forma de nova funcionalitat. A continuació en detallaré tres, la primera com exemple de millora interna i les altres dues com millores externes (apreciables per l'usuari).

- En la part servidor de l'aplicació a l'hora de cercar les coordenades de les localitats i els punts de pas d'una ruta ho fem mitjançant consultes externes. Una altre manera de fer-ho seria importar les dades d'OSM i emmagatzemar-les en una base de dades local amb suport per dades espacials (per exemple Postgres, amb el mòdul PostGIS). Això no tindria repercussió per l'usuari però dotaria de més autonomia a l'aplicació degut a que no tindria d'anar a cercar aquestes dades a l'exterior.
- En la part aplicació es podria dotar aquesta de la possibilitat de triar entre més d'un idioma. Actualment només està disponible en Català i això la restringeix a un ús més local. Les modificacions a fer en aquest sentit tenen un cost baix. També es podria dotar alguna funcionalitat amb alguna dada més com per exemple a l'hora de publicar un trajecte donar la possibilitat de posar algun punt intermedi. Actualment en publicar un trajecte sempre es cerca la ruta més ràpida però això pot resultar ser no exacte, dons hi pot haver algun trajecte que no utilitzi aquest tipus de ruta.
- Per últim també considero que seria mol útil la revisió de la part gràfica (interfícies d'usuari) per part d'un dissenyador gràfic, doncs la visualització del conjunt de l'aplicació seria més atractiva. Això penso que resulta crucial perquè una aplicació tingui èxit, no solsament té que funcionar bé i oferir les funcionalitats necessàries sinó que una presentació visual atractiva i agradable atrau l'atenció de l'usuari.

7.- Resultat final i conclusions

En els següents apartats es farà un balanç del que ha estat el projecte, fins on s'ha arribat en l'assoliment dels objectius i quines són les impressions personals i experiència adquirida en el transcurs del seu desenvolupament.

7.1 – Resultat final

El contingut d'aquesta memòria, detalla com s'ha anat desenvolupant el projecte i mostra d'una manera sintètica les eines que s'han utilitzat per cobrir les necessitats requerides. De totes maneres, no hi queda reflectida la tasca total realitzada degut a que també s'ha fet un estudi d'altres eines que finalment no s'han utilitzat però que han requerit del recurs temps. En aquest apartat es fa una anàlisi dels objectius plantejats en l'inici del projecte i del seu grau d'acompliment.

- **Comprendre la metodologia per implementar una base de dades en una plataforma de cloud computing:** inicialment, es va fer una valoració per utilitzar un servei d'aquestes característiques (cartoDB en la seva versió gratuïta) però es va desestimar per raons de capacitat i també perquè en la meua opinió, estava més orientat a facilitar la presentació d'un altre tipus de mapes amb dades de densitat, estadístiques, etc. Llavors es va assumir un repte de majors proporcions: elaborar una plataforma al núvol pròpia. D'aquesta manera es va implementar un Web service que oferia les funcionalitats requerides.
- **Construir una base de dades geogràfica apropiada al projecte:** es va plantejar d'utilitzar una base de dades amb suport per a dades espacials. Finalment les dades geogràfiques d'una localitat i ruta (que eren les que es necessitaven) es va decidir cercar-les a l'exterior i emmagatzemar-les en la base de dades local sense suport espacial perquè realment no feia falta. El sistema gestor de base de dades implementat va ser MySQL i compleix amb els requisits donant cobertura suficient a l'aplicació.
- **Comprendre els mecanismes de que disposen els dispositius mòbils per accedir a dades i serveis en el núvol:** l'aplicació, bàsicament recull dades de l'usuari i n'hi presenta. A partir d'aquí té necessitat d'emmagatzemar dades (dotar-les de persistència), recuperar-ne i presentar-ne de tractades. Tot això ho fa a través de serveis en el núvol i que en aquest cas els hi ofereix el Web service que s'ha elaborat. El mecanisme del qual es serveix és el de l'accés per URL i les dades s'envien i reben tant directament per la pròpia adreça (utilitzant llavors el mètode "get" del servlet) com empaquetats en format de dades JSON (utilitzant llavors el mètode "put" del servlet).
- **Conèixer a fons el SO Android:** degut a que l'aplicació client s'ha escrit per aquest sistema operatiu, això ha obligat a conèixer molts aspectes d'aquest. S'ha tingut d'explotar la seva part de funcionalitats que ofereix per a les interfícies gràfiques, la part referent a enviar dades, rebre'n i donar-les-hi format, creació de diferents fils d'execució, presentació de dades, importar llibreries noves per a poder presentar mapes dibuixant-hi rutes, etc.
- **Conèixer les diferents llibreries i recursos necessaris per a desenvolupar aplicacions mòbils amb computació en el núvol:** com he comentat en l'apartat anterior bàsicament, les llibreries a importar en estat les necessàries per a la presentació dels mapes. Android

ja porta integrades una gran quantitat de llibreries que donen resposta a diverses funcionalitats.

- **Treballar amb OpenStreerMap per Android:** per la presentació dels mapes es fa ús d'OpenStreetMaps. La part servidor també en fa ús per l'adquisició de coordenades geogràfiques.

7.2 - Conclusions

Aquest projecte, tal com descriuen els objectius, a servit per a conèixer i integrar diferents eines i disciplines de coneixement amb l'objectiu de generar un producte que cobreix unes necessitats, en un àmbit actualitzat. El producte final obtingut és el resultat d'aplicar una sèrie d'eines i recursos però que no són tots els que s'han analitzat. Personalment, abans de desenvolupar aquest projecte tenia uns coneixements nuls en uns casos i bàsics en d'altres per tant, s'ha tingut que realitzar una tasca d'investigació i aprenentatge considerable. En segon terme, com ja s'ha comentat, existia la possibilitat d'utilitzar diferents eines per arribar a la solució final, aquest fet ha exigut tenir-les en consideració per a veure quines podien ser més adients. Al final, penso que s'han assolit els objectius plantejats però també admeto que per falta de temps (recurs molt valuós i limitat) m'ha condicionat o potser millor dit, m'ha fet decidir per un tipus d'implementació en detriment d'algun altre. Com a conseqüència d'aquest fet però, també cal dir que he conegut uns recursos que d'altre manera m'haurien passat desapercebuts. Amb tot això destaco que per primer cop he afrontat un projecte que encara que de dimensions reduïdes, m'ha permès experimentar totes les fases en situació real, i viure en primera persona les dificultats que sorgeixen, tant les analitzades en la fase de planificació (a l'anàlisi de riscos) com d'altres no previstes, veure per altre banda també com una bona planificació és del tot necessària i útil, aprendre a fer reajustaments en certs moments, i finalment, trobar solucions als problemes que van sorgint i que és indispensable per a tirar endavant el projecte i arribar al final. Per tant no és solsament tots els coneixements adquirits en quan a programació, dades geogràfiques i conceptes afins, llibreries que ofereixen solucions funcionals que ajuden de manera determinant a la programació de diferents parts de l'aplicació, etc. sinó que un aspecte també molt important ha estat el d'afrontar l'experiència de emprendre un projecte informàtic.

8.- Referències digitals

- Informació OSM [www.openstreetmap.org]
- Exportar dades selecció OSM [<http://www.openstreetmap.org/export#map=11/41.5834/1.6243&layers=T>]
- Descarregar arxiu Planet de OSM [<http://planet.openstreetmap.org/planet/>]
- Descarregar dades OSM per regions a CloudMade [<http://downloads.cloudmade.com/>]
- Descarregar dades OSM subdividides a Geofabrik [<http://download.geofabrik.de/>]
- Estructura de les dades OSM en XML [<http://wiki.openstreetmap.org/wiki/.osm>]
- Exportar dades OSM en diferents formats [<http://wiki.openstreetmap.org/wiki/ES:Export>]
- Informació general CloudMade [<http://cloudmade.com/>]
- CloudMade Routing API [<http://cloudmade.com/documentation/routing>]
- CloudMade Geocoding API [<http://cloudmade.com/documentation/geocoding>]
- CloudMade navegació [<http://cloudmade.com/products/nav-sdk>]
- CloudMade mapes [<http://cloudmade.com/products/map-sdk>]
- OSMDroid informació [<http://www.slf4j.org/android/>]
- MapsForge informació [<https://code.google.com/p/mapsforge/wiki/SpecificationBinaryMapFile>]
- MapsForge API [<https://code.google.com/p/mapsforge/wiki/OverlayAPI>]
- VMware[https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/4_0]
- Eclipse [<http://www.eclipse.org/downloads/>]
- SDK Android [<http://developer.android.com/sdk/index.html#download>]
- JDK Java [<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>]
- MySQL [<http://downloads.mysql.com/archives/installer/>]
- MySQL Browser [<http://downloads.mysql.com/archives/query/>]
- Apache Tomcat [<http://tomcat.apache.org/download-60.cgi>]
- Web Service conceptes [http://es.wikipedia.org/wiki/Servicio_web]
- JSON conceptes [<http://es.wikipedia.org/wiki/JSON>]
- Projecte json [: <http://json.org/java/>]