

'MyOwnTrip': Obtenció i localització de punts d'interès I  
Memòria de projecte

**UAB**  
Universitat Autònoma de Barcelona

**UAB**

**Universitat Autònoma  
de Barcelona**

# '*MyOwnTrip*': Obtenció i localització de punts d'interès I

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de *Sistemes*  
realitzat per  
*Victor Casado Sached*  
i dirigit per  
*Jordi Duran i Cals*

**Escola d'Enginyeria**

Sabadell, *Juny de 2012*

El sotasignat, **Jordi Duran i Cals**,  
professor de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present memòria  
ha estat realitzat sota la seva direcció  
per en **Victor Casado Sached**

I per a que consti firma la present.

Sabadell, **Juny** de **2012**

-----  
Signat: **Jordi Duran i Cals**

## ÍNDEX

1. Introducció	6
1.1 Motivacions	6
1.2 Marc de treball	6
1.3 Objectius generals	7
1.4 Estructura de la memòria del projecte	8
2. Estudi de viabilitat	9
2.1 Introducció	9
2.2 Tipologia i paraules clau	9
2.3 Revisió i catalogació dels objectius del projecte	9
2.4 Estudi de la situació actual	10
2.5 Context	11
2.6 Lògica del sistema	11
2.7 Descripció física del sistema	12
2.8 Requisits del projecte	13
2.8.1 Llistat de requisits funcionals	13
2.8.2 Llistat de requisits no funcionals	13
2.8.3 Catalogació dels requisits	14
2.9 Alternatives	15
2.9.1 Alternativa 1: Java	15
2.9.2 Alternativa 2: Java + API de Google	15
2.10 Solució proposada	15
2.11 Planificació del projecte	16
2.12 Recursos del projecte	16
2.12.1 Recursos humans	16
2.12.2 Recursos materials	17
2.13 Tasques del projecte	18
2.14 Planificació temporal	20
2.15 Avaluació de riscos	21
2.16 Pla de contingència	22

2.17 Pressupost	23
2.17.1 Estimació del cost de personal	23
2.17.2 Estimació del cost dels recursos materials	23
2.17.3 Resum i anàlisi cost / benefici	24
2.18 Conclusions	24
3. Disseny tècnic i implementació del sistema	25
3.1 Model de desenvolupament	25
3.2 Entorn de desenvolupament	25
3.3 Disseny del sistema	27
3.3.1 Disseny i implementació de la base de dades	29
3.3.2 Disseny i implementació del mòdul d'extracció automàtica de punts d'interès	32
3.3.3 Disseny i implementació de la interfície de l'aplicació Android	40
3.3.4 Disseny i implementació de la funcionalitat d'extracció de definicions de la Wikipèdia	50
3.4 Diagrames del sistema	52
3.4.1 Diagrama de casos d'ús	52
3.4.2 Diagrama de seqüència	53
3.4.3 Diagrama d'activitats	55
4. Test i proves	57
4.1 Proves d'unitat de la base de dades	57
4.2 Proves d'unitat del mòdul d'extracció de punts d'interès	58
4.3 Proves d'unitat del mòdul d'extracció de definicions de la Wikipedia	60
4.4 Proves d'unitat de la interfície Android	60
4.5 Proves d'usuari de la interfície Android	61
Com es pot observar en base als resultats, la navegació dels usuaris per l'aplicació és, en general, molt satisfactòria. Podem afirmar, per tant, que la interfície de l'aplicació Android aconsegueix els objectius de disseny centrats en l'usuari que hem utilitzat com a base de creació.	63
4.6 Proves d'integració	63
5. Línies futures	65
5.1 Millora del mòdul d'extracció de punts d'interès	65
5.1.1 Millora 1: Freeling i API de Google Maps	66

5.1.2 Justificació de la no implementació	67
5.2 Millora de la interfície de l'aplicació Android	68
5.2.1 Millora 1: Temes d'usuari	68
5.2.2 Millora 2: Adaptació a dispositius tablet	68
5.2.3 Justificació de la no implementació	69
5.3 Millora del mòdul d'extracció de definicions de la Wikipedia	69
5.3.1 Millora 1: Informació en diferents idiomes	70
5.3.2 Millora 2: Imatges d'un punt d'interès	70
5.3.3 Justificació de la no implementació	71
6. <i>Conclusions</i>	72
6.1 Objectius aconseguits i no aconseguits	72
6.2 Seguiment dels errors de desenvolupament	73
6.3 Valoració personal	75
7. <i>Glossari</i>	77
7.1 Relació de taules	77
7.2 Relació d'imatges	78
8. <i>Bibliografia</i>	79
8.1 Apunts	79
8.2 Llibres	79
8.3 Direccions web	79

## **1. Introducció**

### **1.1 Motivacions**

El present treball vol desenvolupar un sistema capaç de treballar amb conceptes tan actuals i innovadors com poden ser el Processament del Llenguatge Natural (PLN), la Realitat Augmentada i o la geo-localització. Per fer-ho, es pensa en un prototip que ofereixi a l'usuari, de manera dinàmica, la possibilitat d'obtenir, a partir d'una col·lecció de documents i enllaços web i de forma totalment automàtica, un conjunt de punts d'interès amb la seva corresponent geo-localització. A més, permet manipular aquests punts d'interès per tal de visualitzar-los en un mapa o traçar rutes.

La idea original sorgeix al 2010 durant les classes de l'assignatura Teoria d'Autòmats i Llenguatges Formals i, més concretament, quan es parla de Màquines d'Estats i Autòmats.

Més endavant, durant la reunió de presa de contacte amb el director del projecte, aquest proposa el sistema anteriorment exposat, marcant molt clarament que hauria de ser capaç d'extreure de forma automàtica els punts d'interès i que s'inclouria una aplicació Android capaç de manipular-los.

Així doncs, i amb la complementació del treball d'un altre company, s'ha creat un prototip que satisfà els dos objectius principals que inicialment es van proposar.

### **1.2 Marc de treball**

Abans de començar a parlar plenament del projecte que defensa aquesta memòria, cal aclarir que es tracta d'un treball de desenvolupament de programari conjunt.

Així doncs, un cop definides les especificacions generals del projecte a desenvolupar cal diferenciar dues línies de treball molt marcades, de forma que cada component del grup tingui clar què és el que ha de fer. Es fa aquesta diferenciació amb l'objectiu, a més, de donar total independència al treball de cada alumne, de forma que en el cas que un dels components del grup no pogués arribar a desenvolupar la seva part, l'altre pogués presentar quelcom funcional.

És llavors quan es fa el que vam definir com un tall horitzontal en el desenvolupament del projecte i se'ns assignen unes tasques a cadascun dels membres. La separació horitzontal de les funcions a desenvolupar permet a tots els components del grup prendre part en totes les fases del projecte, motiu pel qual vam decidir que es fes així.

Per tant, un cop separades i diferenciades les parts del projecte vam poder començar a treballar cadascun per la seva part, tenint sempre en compte, al menys des d'un punt de vista a molt alt nivell, la màxima compatibilitat amb el treball del company.

Per fer-ho, s'acorda treballar amb un llenguatge de programació actual com és el Java i enfocar l'aplicació mòbil de cara a dispositius *Android*. A més, per a cada mòdul, es defineixen clarament les entrades i sortides amb els quals es treballaran.

És llavors quan ens podem posar a treballar i desenvolupar el projecte.

### **1.3 Objectius generals**

Els objectius generals que pretén satisfer el projecte són:

1. Proporcionar un servei capaç de detectar i extreure localitzacions a partir de text pla.
2. Proporcionar una interfície web que permeti als usuaris registrar-se i accedir al sistema, així com generar peticions d'extracció.
3. Proporcionar un conjunt d'eines que permetin calcular o assignar geo-localitzacions a un conjunt de punts d'interès.
4. Proporcionar un mòdul de cerca de definicions de punts d'interès a la *Wikipèdia*.
5. Dissenyar la interfície d'usuari de l'aplicació *Android*.
6. Crear una aplicació *Android* que permeti a l'usuari la visualització dels resultats de l'extracció.
7. Aplicar els coneixements adquirits a l'assignatura Teoria d'Autòmats i Llenguatges Formals.
8. Aprofundir en el coneixement del llenguatge Java.
9. Aprofundir en el coneixement de l'entorn de desenvolupament Eclipse i la integració d'aquests amb altres mòduls.
10. Aprofundir en el coneixement del desenvolupament d'aplicacions mòbils.

De tots aquests objectius, els que pretén satisfer la nostra part del projecte són l'1, el 3, el 4, el 5, el 7, el 8, el 9 i el 10. Més endavant, en l'apartat 2.3 *Revisió i catalogació dels objectius del sistema* s'analitzen més a fons.

## **1.4 Estructura de la memòria del projecte**

La memòria del projecte s'estructura en set capítols:

1. Introducció: al primer apartat es descriuen les motivacions, marc de treball, objectius i estructura que envolten el projecte que es desenvolupa.
2. Estudi de viabilitat: a la segona part, descriurem breument el context que envolta el projecte, així com la lògica, requisits i tasques de que es compona. A més, s'analitzaran els riscos, alternatives i la solució final escollida així com tota la planificació del projecte.
3. Disseny Tècnic i Implementació del Sistema: el tercer apartat és on es troba la part més important de la memòria on s'expliquen quins són els models que s'han seguit a l'hora de treballar i com s'han dut a terme el disseny i implementació de cadascun dels mòduls que conformen el sistema.
4. Fase de proves: un cop explicat el procés que s'ha seguit per a desenvolupar el sistema, al quart apartat s'especifiquen també les proves d'unitat i integració que s'han realitzat per tal de validar el producte final.
5. Conclusions: en aquest apartat es valoren els objectius finals del treball i s'exposen breument les línies de treball futures i possibles millores del sistema. A més, es fa una valoració personal del projecte.
6. Bibliografia: per últim, llistem el conjunt de documents i direccions web usades com a reforç.



## **2. Estudi de viabilitat**

### **2.1 Introducció**

La intenció d'aquest apartat és la de dur a terme un breu però acurat anàlisi dels aspectes principals del projecte que es vol desenvolupar.

Això inclou una primera part de definició del context a partir del qual es parteix i dels requisits esperats o desitjats que la solució ha de satisfer.

Més endavant cal analitzar quines són les diferents alternatives possibles i escollir-ne justificadament la més adient, així com fer una planificació aproximada de les tasques que cal dur a terme.

Abans de tirar endavant el projecte cal, a més, avaluar i prevenir riscos i calcular el pressupost i beneficis esperats per calibrar si el projecte té expectatives d'arribar a ser quelcom vàlid.

El punt clau és que amb el nostre programari resultant volem oferir una eina innovadora i, a la vegada, de qualitat en quant a desenvolupament.

### **2.2 Tipologia i paraules clau**

La tipologia d'aquest projecte té dues vessants:

- a. Desenvolupament de programari, ja que es tracta d'una aplicació que combina parts de desenvolupament Client/Servidor, desenvolupament web i desenvolupament Android.
- b. De recerca, ja que moltes de les eines de desenvolupament a les que ens enfrontem són desconegudes i és necessari tenir clars els seus fonaments i funcionament.

Les paraules clau associades al projecte són:

Java, Android, PLN (Processament del Llenguatge Natural), Geo-localització, Realitat augmentada.

### **2.3 Revisió i catalogació dels objectius del projecte**

Els objectius principals que el projecte pretén satisfer són els que s'han especificat en el punt *1.3 Objectius*.

Podem classificar aquests objectius segons la seva naturalesa en:

Objectius	Crític	Prioritari	Secundari
1	X		
2	X		
3		X	
4		X	
5		X	
6			X
7			X
8			X

*Taula 1. Classificació dels objectius*

## 2.4 Estudi de la situació actual

El darrer any s'ha produït el que es coneix com la revolució de les xarxes socials. Potser és difícil veure com aquest fenomen pot afectar al context del projecte que s'està desenvolupant però, en realitat, aquesta revolució i el caràcter de l'aplicació que es vol dur a terme estan íntimament relacionats.

El perfeccionament de l'accés a Internet sense cables (*wireless*) ha proliferat tota una col·lecció de dispositius portàtils capaços d'oferir a l'usuari accés a la xarxa des de pràcticament qualsevol punt.

A més, la continua connexió a les xarxes socials, provoca que la majoria de persones i, fins i tot sectors de la població que en un principi s'havien quedat fora de l'era tecnològica, disposin de dispositius mòbils tals com telèfons intel·ligents (*smartphone*) o tabletas amb connexió a Internet permanent.

Això provoca l'aparició de la que podríem considerar la vessant tecnològica en més auge de l'actualitat: el desenvolupament d'aplicacions mòbils. A partir d'aquí, tot un conjunt d'empreses i persones s'esforcen per:

- a. Desenvolupar i donar a conèixer llenguatges específics per al desenvolupament d'aplicacions mòbils com és l'Android Development Kit en el cas de Google Android.
- b. Aprendre la sintaxi i l'ús de les eines que aquests llenguatges proporcionen al desenvolupador d'aplicacions. És important remarcar que aquest aprenentatge és pràcticament autodidacta i a través de la xarxa ja que es tracta de quelcom tan nou que no s'ensenya oficialment i els llibres que tracten el tema són poc específics.

Així doncs, ens trobem amb un mercat d'aplicacions mòbils en que, per una banda, qualsevol desenvolupador pot arribar a crear una aplicació i posar-la a la venda i, per l'altra, qualsevol persona pot obtenir i usar per un mòdic preu aquesta aplicació en un dels seus dispositius mòbils.

És, per tant, un repte doble ja que ens haurem de submergir un entorn de desenvolupament que pràcticament acaba d'originar-se i haurem de ser capaços de crear un producte que ofereixi quelcom atractiu en un mercat de milers i milers d'aplicacions.

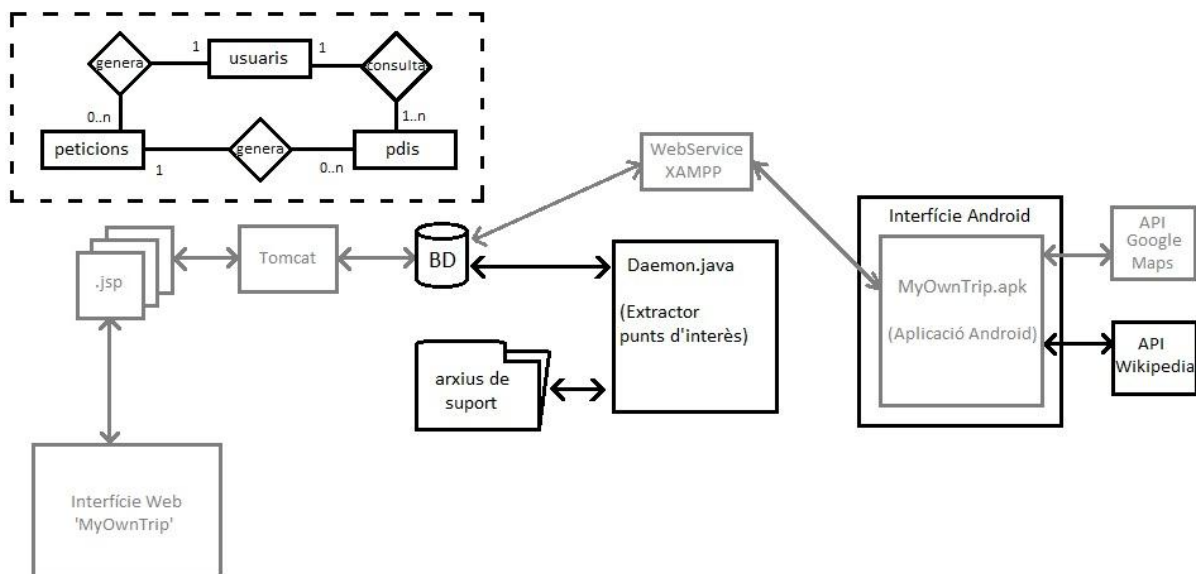
## 2.5 Context

Aspecte	Solució proposada
Entorn de desenvolupament	Eclipse
Llenguatge de programació	Java
Bases de dades	MySQL a través d'un servidor Apache
Plataforma mòbil	Google Android (Android Development Kit)

Taula 2. Context

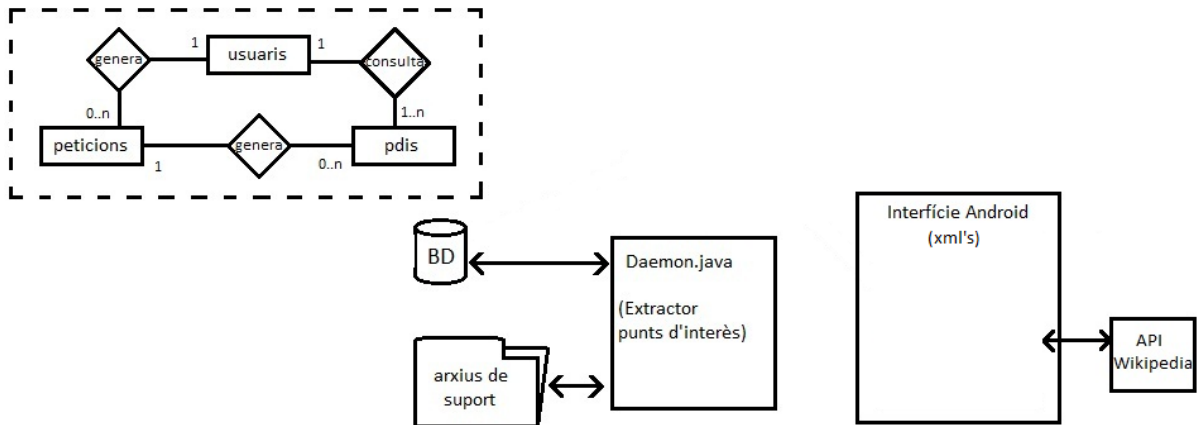
## 2.6 Lògica del sistema

La lògica del sistema a desenvolupar és la següent:



Imatge 1. Lògica del sistema complet

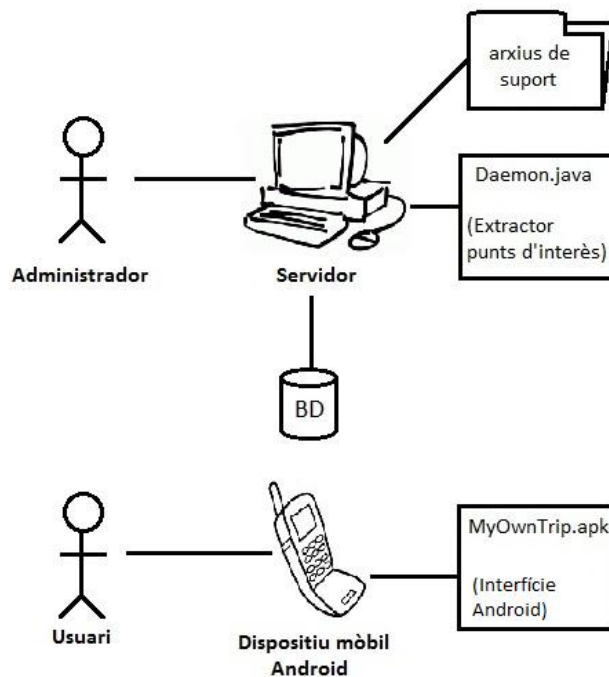
I dins d'aquest esquema general, la part que desenvolupa i documenta aquesta memòria és la que es marca en negre, que es correspon amb la següent il·lustració:



*Imatge 2. Lògica del sistema parcial*

## 2.7 Descripció física del sistema

La descripció física del sistema a desenvolupar és la següent:



*Imatge 3. Descripció física del sistema*

Concretament, el maquinari amb el que es treballa és:

	Descripció
Equip local	Dell Inspiron 1120 amb Sistema Operatiu Windows 7
Dispositiu mòbil Android	Mòbil ZTE Blade amb Android 2.2

*Taula 3. Maquinari*

## 2.8 Requisits del projecte

### 2.8.1 Llistat de requisits funcionals

Els requisits funcionals del sistema a desenvolupar són:

1. Manteniment i gestió (altes, baixes, modificacions) de les dades dels usuaris del sistema.
2. Manteniment, gestió (pujades, baixades, accessos i modificacions) i actualitzacions dels arxius de suport al servidor.
3. Manteniment i gestió (pujades, accessos i modificacions) dels arxius d'usuaris al servidor.
4. Manteniment, gestió i actualització de les peticions dels usuaris a la base de dades.
5. Manteniment, gestió i actualització de la informació dels punts d'interès d'usuaris a la base de dades.
6. Administració de la base de dades i dels documents d'usuaris per part de l'administrador del sistema.
7. Comunicació als usuaris dels drets de rectificació o modificació de les dades.
8. Sistema d'ajuda en línia.
9. Sistema de còpies de seguretat i recuperació de dades.

### 2.8.2 Llistat de requisits no funcionals

Els requisits no funcionals del sistema a desenvolupar són:

1. La capacitat del servidor s'ha d'ajustar al volum d'ús que els usuaris en faran d'ell.
2. El sistema ha de poder suportar diferents sessions d'usuari simultànies.
3. El codi font del programari ha de ser el més modulats possible per tal de permetre la fàcil integració amb la part complementària del projecte i facilitar futures possibles ampliacions.

4. Normalització de la base de dades i accés segons l'estàndard MySQL.
5. Tolerància a dades i accions incorrectes.
6. Control de totes les entrades i sortides de l'usuari.
7. Interfície d'usuari Android amigable i senzilla.
8. Documentació de tot el producte.
9. Compliment de la normativa dels projectes de final de carrera de l'Escola d'Enginyeria.

### 2.8.3 Catalogació dels requisits

Podem classificar els requisits segons la seva naturalesa en:

Requisit	Essencial	Condicional	Opcional
RF1	X		
RF2	X		
RF3	X		
RF4	X		
RF5	X		
RF6		X	
RF7		X	
RF8			X
RF9			X
RNF1	X		
RNF2	X		
RNF3	X		
RNF4		X	
RNF5		X	
RNF6			X
RNF7		X	
RNF8			X
RNF9		X	

*Taula 4. Requisits funcionals i no funcionals del projecte*

## 2.9 Alternatives

A continuació s'analitzen les diferents alternatives.

### 2.9.1 Alternativa 1: Java

La primera alternativa és desenvolupar completament tot el sistema amb l'entorn de desenvolupament Eclipse i el llenguatge Java. Les característiques d'aquesta solució són:

Avantatges	Inconvenients
Totalment ajustable als requisits del sistema	Es parteix de zero i, per tant, s'ha de codificar la totalitat del sistema
Totalment adaptable als recursos del sistema	La capacitat d'integració i la optimitat del sistema depèn total i directament del desenvolupador
Totalment modificable	

Taula 5. Anàlisi de l'alternativa 1

### 2.9.2 Alternativa 2: Java + API de Google

La segona alternativa és desenvolupar part de l'aplicació fent ús de l'Eclipse (extracció automàtica dels punts d'interès) i altra part utilitzant les eines de desenvolupament de l'API de Google (aplicació Android per a dispositius mòbils). Les característiques d'aquesta solució són:

Avantatges	Inconvenients
Sistema parcialment ajustable als requisits	Comunicació entre dos sistemes diferents
Implementació senzilla de determinades funcionalitats gràcies a l'API de Google	Necessitat d'adaptació del format de les dades d'un sistema a l'altre

Taula 6. Anàlisi de l'alternativa 2

## 2.10 Solució proposada

Si fem una comparació de les alternatives:

Alternativa	Complexitat	Nivell d'integració	Documentació	Costos
Alternativa 1	Alta	Alt	Online	0 €
Alternativa 2	Mitja	Mitjà	Online	0 €

Taula 7. Comparació de les alternatives

A la vista de les característiques de les diferents alternatives, sembla que l'alternativa 2 és la més equilibrada i, per tant, la que és preferible escollir.

Així doncs, es desenvolupa un sistema que combina per una part un conjunt de funcionalitats que es programen partint de zero en llenguatge Java i, per l'altra, un conjunt d'eines que ens proveeixen les API's de Google.

Cal, per tant, tenir en compte que s'ha de mirar en tot moment la compatibilitat entre ambdues eines per tal de fer possible la presentació d'un tot funcional final.

## 2.11 Planificació del projecte

L'eina utilitzada per a fer la planificació del projecte és Microsoft Project 2010.

La data de començament del projecte va quedar fixada al 3 d'octubre de 2011 i la data aproximada de finalització del mateix es planteja cap a finals de juny de 2012, un cop feta la presentació davant del tribunal, que és quan podem considerar el cicle del projecte com a tancat.

Fent la planificació amb Microsoft Project 2010, i tenint en compte que dediquem 10 hores setmanals al desenvolupament del projecte, la durada aproximada del projecte seria d'unes 30 setmanes. Partint d'aquesta estimació comptem amb dos calendaris de treball diferents, un per a cada component de l'equip:

Component	Rol	Dies	Hores
Jordi Duran Cals	Director de projecte	Dilluns (cada 2 setmanes)	18.00h - 19.00h

*Taula 8. Component 1 del projecte*

Això representen 15 setmanes \* 1,5 hores/setmana = 22,5 hores

Component	Rol	Dies	Hores
Victor Casado Schez	Analista / Programador	Dilluns, Dimecres i Divendres	15.00h - 17.00h
		Diumenges	17.00h - 18.00h

*Taula 9. Component 2 del projecte*

Això representen 30 setmanes \* 7 hores/setmana = 210 hores

## 2.12 Recursos del projecte

### 2.12.1 Recursos humans

El conjunt de recursos humans està format per l'equip de projecte:



Nom	Correu electrònic	Càrrec	Responsabilitats	Calendari
Jordi Duran Cals	<a href="mailto:jdurant@deic.uab.cat">jdurant@deic.uab.cat</a>	Director de projecte	-Supervisar el correcte desenvolupament del projecte així com la seva viabilitat i finalització.	1
Victor Casado Sachez	<a href="mailto:vcasado90@gmail.com">vcasado90@gmail.com</a>	Analista / Programador	-Anàlisi de requeriments. -Desenvolupament. -Proves. -Documentació. -Implementació. -Manteniment.	2

Taula 10. Recursos humans del projecte

## 2.12.2 Recursos materials

El maquinari disponible és el següent:

Dispositiu	Tipus	Característiques	Propietari	Disponibilitat	Càrrec
Dell Inspiron 1120	Ordinador portàtil	- <u>Processador</u> : Intel Core 2 Duo T5800 a 2.00 GHz - <u>RAM</u> : 4,00 GB - <u>Disc dur</u> : 200 GB - <u>Tarja gràfica</u> : ATI Mobily Radeon HD 3430 - <u>Connectivitat</u> : Ethernet i WiFi - <u>SO</u> : Windows 7 Ultimate (64 bits)	Victor Casado Sachez	Total	500 €
Mòbil ZTE Blade	Dispositiu mòbil tipus	- <u>RAM</u> : 512,00 MB	Victor Casado	Total	150 €

	smartphon e	- <u>Connectivitat</u> : Ethernet i WiFi  - <u>SO</u> : Android 2.2	Sachez		
--	----------------	---------------------------------------------------------------------------------	--------	--	--

*Taula 11. Recursos materials*

El programari disponible és el següent:

Programa	Llicència	Cost pel projecte
Windows 7 Ultimate	Pagament	320 €
Microsoft Office 2010	Pagament	139 €
Microsoft Project 2010	Pagament	1067 €
Eclipse	Lliure	0 €
XAMPP	Lliure	0 €
Tomcat v5.5	Lliure	0 €
Android Development Kit (ADK)	Lliure	0 €

*Taula 12. Programari disponible*

## 2.13 Tasques del projecte

Les tasques del projecte són:

Id	Nom de la tasca	Durada	Precedència	Recursos
1	Inici del projecte: assignació i matriculació del projecte	2 hores		DP
2	<b>Planificació</b>	<b>22 hores</b>		
3	Estudi de viabilitat	10 hores	1	A
4	Aprovació de l'estudi de viabilitat	1 hora	3	DP
5	Pla de projecte	10 hores	4	A
6	Aprovació del pla de projecte	1 hora	5	DP
7	<b>Anàlisi de l'aplicació</b>	<b>24 hores</b>		

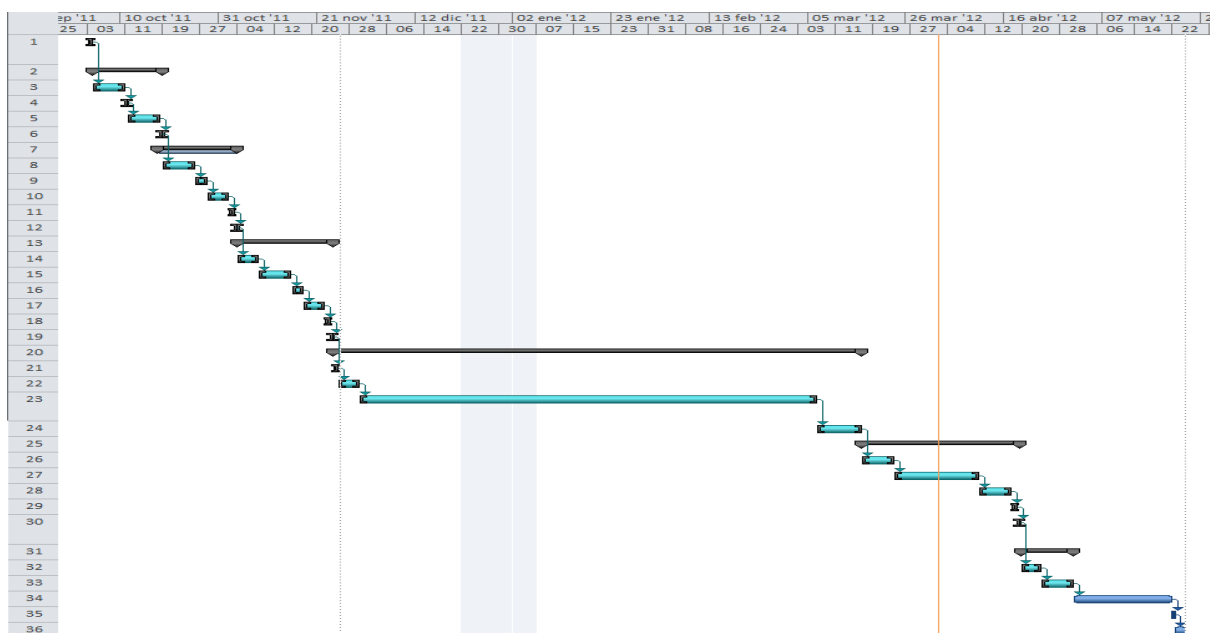
<b>8</b>	Anàlisi de requisits (cassos d'ús)	10 hores	6	A
<b>9</b>	Anàlisi de dades (base de dades)	5 hores	8	A
<b>10</b>	Anàlisi de seguretat i legalitat	5 hores	9	A
<b>11</b>	Documentació de l'anàlisi	3 hores	10	A
<b>12</b>	Aprovació de l'anàlisi	1 hora	11	DP, A
<b>13</b>	<b>Disseny de l'aplicació</b>	<b>29 hores</b>		
<b>14</b>	Disseny de la base de dades	5 hores	12	A, P
<b>15</b>	Disseny modular de l'aplicació	10 hores	14	A, P
<b>16</b>	Disseny de la interfície i ajuda en línia	5 hores	15	A, P
<b>17</b>	Disseny de proves	5 hores	16	A, P
<b>18</b>	Documentació del disseny	3 hores	17	A
<b>19</b>	Aprovació del disseny (PC)	1 hora	18	DP, A
<b>20</b>	<b>Desenvolupament de l'aplicació</b>	<b>132 hores</b>		
<b>21</b>	Preparació entorn de desenvolupament	2 hores	19	P
<b>22</b>	Configuració de la base de dades	5 hores	21	P
<b>23</b>	Mòdul d'adquisició de dades i funcionalitat de l'aplicació	110 hores	22	P
<b>24</b>	Desenvolupament interfície d'usuari	15 hores	23	P
<b>25</b>	<b>Test i proves</b>	<b>34 hores</b>		
<b>26</b>	Proves unitàries	10 hores	24	P

<b>27</b>	Proves d'integració	10 hores	26	P
<b>28</b>	Proves d'estrés (incidències i riscos)	10 hores	27	P
<b>29</b>	Documents de desenvolupament i test	3 hores	28	P
<b>30</b>	Aprovació del desenvolupament i test	1 hora	29	DP, A, P
<b>31</b>	<b>Implantació</b>	<b>15 hores</b>		
<b>32</b>	Instal·lació	5 hores	30	A, P
<b>33</b>	Proves reals	10 hores	32	A, P
<b>34</b>	Generació de documents (memòria)	30 hores	33	CP
<b>35</b>	Tancament del projecte (PC)	1 hora	34	DP
<b>36</b>	Defensa del projecte (PC)	5 hores	35	DP

*Taula 13. Tasques del projecte*

## 2.14 Planificació temporal

La planificació temporal generada amb Microsoft Project 2010 és la següent:



*Imatge 4. Planificació del projecte*

## 2.15 Avaluació de riscos

Risc	Origen	Descripció	Efecte
R1. Planificació temporal optimista	Planificació del projecte	No s'ha planificat de forma correcta la durada de les tasques del projecte	El projecte no s'acaba en la data prevista
R2. Canvi de requisits	Anàlisi	Els requisits del projecte canvien durant el seu desenvolupament	Endarreriment del projecte, augment dels recursos
R3. Eines de desenvolupament inadequades	Desenvolupament	Les eines escollides per al desenvolupament del sistema no resulten útils, provoquen molts problemes o són incompatibles	Endarreriment del projecte, disminució de la qualitat del sistema
R4. No es fa correctament la fase de test	Test i prova	Les proves que es duen a terme sobre el sistema no cobreixen tots els possibles casos	Manca de qualitat, deficiència en el funcionament, insatisfacció dels usuaris
R5. Manca d'adopció de mesures de seguretat	Durant tot el projecte	No s'implementen tots els mecanismes necessaris per a la recuperació de dades davant fallades	Pèrdua d'informació

R6. Accident fatal de l'equip informàtic	Durant tot el projecte	L'equip informàtic amb el que comptem per a desenvolupar el projecte s'espalla de forma fatal	Pèrdua de tota la informació del projecte
R7. Abandonament del projecte abans de la finalització	Durant tot el projecte	S'abandona la realització del projecte per causes alienes a la voluntat dels participants	No finalització del projecte

*Taula 14. Riscos del projecte*

Podem classificar aquests riscos segons el seu efecte sobre el projecte en:

Risc	Probabilitat [0..10]	Impacte [0..10]
R1	3	5
R2	6	3
R3	4	7
R4	3	5
R5	3	5
R6	4	9
R7	2	9

*Taula 15. Anàlisi de riscos*

## 2.16 Pla de contingència

El pla de contingència dels diferents riscos estimats és:

Risc	Solució
R1	Ajornar algunes funcionalitats, dedicar més temps del previst.
R2	Renegociar funcionalitats, re-planificar el projecte.
R3	Preveure altres alternatives, millorar la formació de l'equip.
R4	Dissenyar els tests amb antelació, fer tests automàtics.
R5	Revisar la seguretat en cada fase, dur a terme polítiques de seguretat actives.

<b>R6</b>	Fer còpies de seguretat del sistema periòdicament, utilitzar un sistema de control de versions.
<b>R7</b>	No té solució

*Taula 16. Pla de contingència*

## 2.17 Pressupost

### 2.17.1 Estimació del cost de personal

Tal com s'ha calculat als apartats 2.11 i 2.12 quan s'ha parlat de la planificació del projecte i els recursos humans disponibles:

Recurs	Hores	Preu/Hora	Cost total
<b>Cap de projecte</b>	22,5h	40 €	900 €
<b>Analista/Programador</b>	210h	20 €	4200 €

*Taula 17. Cost de personal*

La suma total derivada dels recursos és de 5100 €.

### 2.17.2 Estimació del cost dels recursos materials

Tal com s'ha calculat als apartats 2.11 i 2.12 quan s'ha parlat de la planificació del projecte i els recursos materials disponibles:

Recurs	Cost d'amortització	Cos unitari	Període d'amortització	Període d'ús
<b>Dell Inspiron 1120</b>	250 €	500 €	18 mesos	9 mesos
<b>Mòbil ZTE Blade</b>	33,5 €	150 €	18 mesos	4 mesos
<b>Windows 7 Ultimate</b>	160 €	320 €	18 mesos	9 mesos
<b>Microsoft Office 2010</b>	69,5	139 €	18 mesos	9 mesos
<b>Microsoft Project 2010</b>	178 €	1067 €	18 mesos	3 mesos

*Taula 18. Cost de recursos*

Això representa un total de 691 € derivats dels recursos materials.

### 2.17.3 Resum i anàlisi cost / benefici

El cost del projecte tenint en compte tots els conceptes suma un total de:

- Costos de recursos humans: 5100 €.
- Costos de recursos materials: 691 €.

Un total d'aproximadament 5800 €.

Encara que el cost total del producte és alt, aquest es pot amortitzar si, per exemple, posem a la venda l'aplicació al Market d'Android amb un preu assequible (per exemple 0,70 €).

Cal tenir en compte, a més, que les funcionalitats que el sistema ofereix a través de l'aplicació són quelcom en auge en el paradigma actual, de manera que, en un període aproximat d'un any i mig s'haurà amortitzat el cost total i s'haurà obtingut un benefici.

### 2.18 Conclusions

A continuació es pot valorar la viabilitat del projecte amb una taula de beneficis i inconvenients lligats al projecte:

Avantatges	Inconvenients
Millorar l'experiència en programació Java i disseny d'aplicacions mòbils	Inversió important amb amortització esperada a un any i mig.
Proporcionar a qualsevol usuari un servei capaç d'extreure de forma automàtica punts d'interès i la seva corresponent geo-localització en base als seus documents / direccions web visitades.	
Realitzar un projecte que més tard altres desenvolupadors poden fer servir com a base per a altres plataformes o futures ampliacions.	Necessitat d'integració de diferents eines.
Treballar en un projecte que, un cop acabada la carrera, pugui servir com a carta de presentació i mostra de l'estil de treball.	

*Taula 19. Avantatges i inconvenients del projecte*

Per tant, Beneficis + Inconvenients = Projecte viable.



### **3. Disseny tècnic i implementació del sistema**

#### **3.1 Model de desenvolupament**

Per al desenvolupament del projecte, s'han fet servir de forma combinada dos dels més coneguts i utilitzats models de desenvolupament de programari: el model de prototipus i el model en cascada.

Per una banda, en un principi partim d'uns objectius ideals a un alt nivell conceptual. A partir d'aquests objectius, hem anat obrint camí en paral·lel al desenvolupament del prototipus sempre amb una clara línia marcada. És en aquest procés en el qual s'aplica el model de prototipus, que permet un modelat, disseny i construcció de quelcom vàlid de forma més o menys ràpida.

En el disseny individual de cada mòdul, però, és a la vegada present el model en cascada. Així, una vegada definits els requisits principals d'una part del sistema, i tenint sempre a la vista els objectius generals del projecte, procedim al disseny, codificació, prova i implantació del programari corresponent. A més, a mesura que el projecte va creixent, es duu a terme un manteniment i, en alguns casos, actualització dels mòduls anteriorment desenvolupats.

Cal afegir que es tracta d'un projecte ambiciós que consta de diferents parts desenvolupades per diferents participants i en el que, per tant, l'èxit és aconseguir la correcta comunicació entre elles. A més, és important que el resultat sigui capaç de resoldre en gran part els objectius principals.

#### **3.2 Entorn de desenvolupament**

El llenguatge de programació base escollit per a la implementació del projecte és el Java i el seu framework per al desenvolupament l'Eclipse. Aquest llenguatge de programació és un dels més senzills, ensenyats i utilitzats internacionalment. A més, l'Eclipse és un entorn de desenvolupament multi-plataforma lliure i que, per tant, està a disposició pública. Cal dir, però, que tot i que està pensat per a ser utilitzat principalment a entorns Linux, per qüestions de equipament disponible ens hem decantat per fer servir un sistema operatiu de tipus Windows, entorn en el qual hem pogut fer servir l'Eclipse sense cap problema.

Respecte al Java, cal dir que es tracta d'un llenguatge de programació orientat a objecte, que agafa gran part de la seva sintaxi dels llenguatges C i C++. La seva característica principal és la proporció d'un model d'objectes molt simple i d'una eina de gestió de memòria anomenada recollidor de brossa, que permet a l'usuari oblidar-se de tasques relacionades amb la manipulació directa de punters i memòria.

Es tracta, a més, d'un llenguatge multi-plataforma capaç d'integrar-se en una llarga llista d'entorns com poden ser dispositius mòbils, sistemes encastats, navegadors web, sistemes de tipus servidor o aplicacions d'escriptori, entre altres. Molts fabricants i desenvolupadors de programari ofereixen, a més, un conjunt de funcionalitats en forma d'API (Application Programming Interface) a les quals es pot accedir de forma quasi natural amb el llenguatge Java.

Així doncs, podem emmarcar el Java dins d'un dels paradigmes de programació actuals més importants com és el de la programació orientada a objecte. Els llenguatges d'aquest tipus proporcionen tota una sèrie d'eines per tal de facilitar la tasca del disseny de codi, com són:

-Classe: definició de les propietats i comportament d'un tipus concret d'objecte. La instanciació és la lectura d'aquestes definicions i la creació d'un objecte a partir d'aquestes.

-Atribut: propietat d'una determinada classe. Pot ser dinàmic o estàtic, segons si el seu valor és variable o no, i públic, protegit o privat, segons la seva visibilitat.

-Mètode: algoritme associat a un objecte o a un classe d'objectes l'execució del qual es desencadena com a conseqüència de la recepció d'un missatge.

-Herència: és la facilitat mitjançant la qual una classe A hereta en ella cadascun dels atributs i mètodes d'una classe B, com si aquests atributs i operacions haguessin estat definits a la mateixa A.

-Objecte: entitat proveïda d'una sèrie de propietats o característiques (atributs) i de comportaments o funcionalitats (mètodes), que conseqüentment reaccionaran davant de determinants esdeveniments.

A més, el llenguatges orientats a objecte tenen una sèrie de característiques que els distingeixen de la resta:

-Abstracció: és la capacitat que té un objecte de treballar, informar, canviar el seu estat i comunicar-se amb altres objectes sense revelar com s'implementen aquestes accions.

-Encapsulament: és la propietat per la qual tots els elements que poden considerar-se pertanyents a una mateixa entitat, estan al mateix nivell d'abstracció de manera que s'augmenti la cohesió dels components del sistema.

-Modularitat: és la propietat que permet subdividir una aplicació en parts més petites (mòduls) cadascuna de les quals ha de ser el més independent possible de l'aplicació en sí i de les altres parts.

-Principi d'ocultació: cada objecte està aïllat de l'exterior, és un mòdul natural, i cada tipus d'objecte exposa una interfície a altres objectes que especifica com poden interactuar amb ell. Aquest aïllament protegeix l'objecte de modificacions internes per part d'altres objectes.

-Polimorfisme: és la propietat per la qual comportaments diferents, associats a objectes diferents, poden compartir el mateix nom, de manera que en realitzar la crida amb aquest nom, s'executarà un comportament o un altre, segons l'objecte que la fa.

-Herència: és la capacitat de les classes de relacionar-se entre sí, formant una jerarquia de classificació.

-Recol·lecció de la brossa: és la tècnica per la qual l'entorn d'objectes s'encarrega de destruir automàticament i, per tant, d'alliberar la memòria associada als objectes que hagin quedat sense cap referència.

Així doncs, es tracta d'un tipus de paradigma que utilitza objectes com a elements fonamentals en la construcció de la solució.

L'Eclipse per la seva part, potencia aquesta gran capacitat d'integració del llenguatge Java, fent possible l'addició i fàcil utilització de milers de funcionalitats extra o complementàries a allò que s'està programant. Per fer-ho, es val de llibreries de tipus .jar (Java ARchive) o fins i tot de classes externes i internes que l'usuari pot importar de forma molt ràpida i fàcil. Cal dir, que el fet de que es tracti d'un programari de codi lliure fa que siguin els propis usuaris els qui l'enriqueixen de forma que es crea tot un sistema que s'alimenta per sí mateix i des d'on és possible accedir a múltiples funcionalitats i implementacions.

Proveeix també una sèrie d'eines molts útils per a la codificació, com el debugger en temps real, la prova i verificació del codi desenvolupat, com el JUnit, o de sistemes de control de versions, com el CVS.

Es tracta, per tant, d'una de les millors, si no la millor opció que podem escollir per al desenvolupament del projecte.

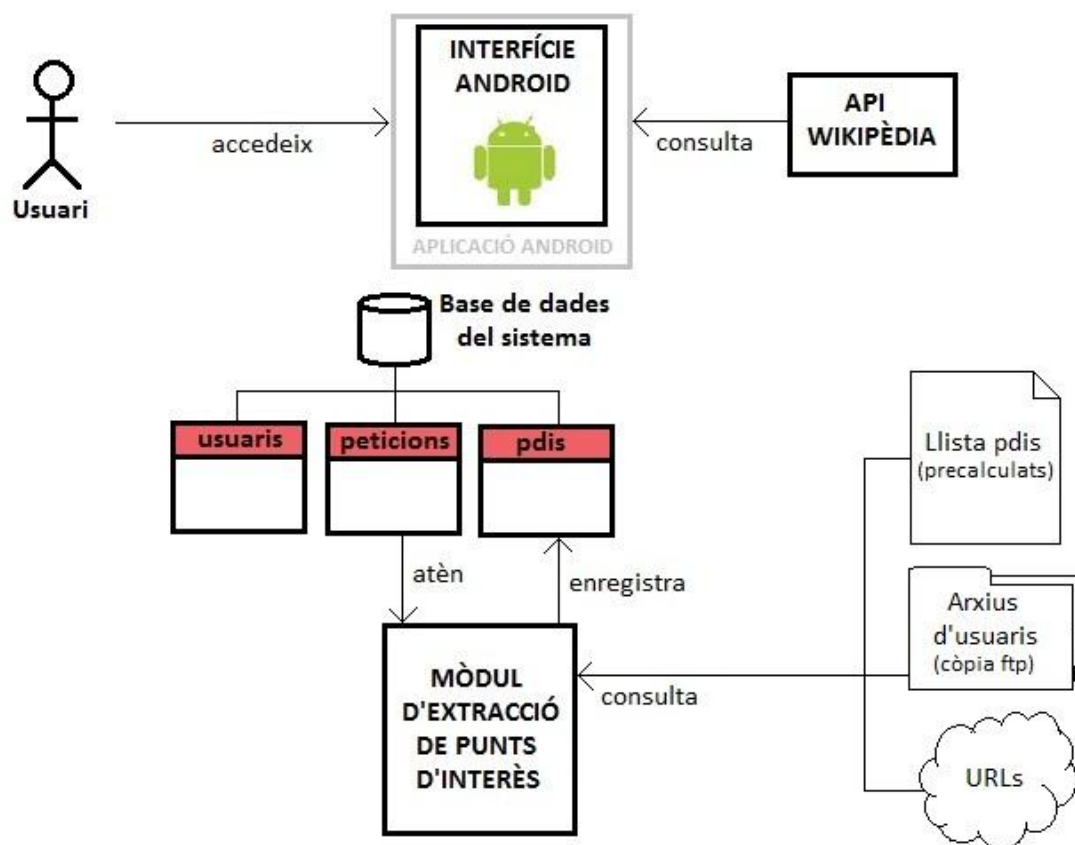
### **3.3 Disseny del sistema**

Per al plantejament del disseny del sistema s'ha fet servir una estratègia de *divideix i venceràs*. Així doncs, a partir d'una idea global del projecte, i posteriorment al repartiment de les tasques entre els membres de l'equip de treball, s'han identificat els principals objectius a satisfer per part de la nostra parcel·la del desenvolupament total. Aquest objectius, que definiran la segmentació del treball a realitzar, són:

1. Proporcionar un mòdul capaç d'extreure de forma automàtica un conjunt de punts d'interès a partir d'una col·lecció de documents i direccions web.
2. Proporcionar una interfície d'usuari per a l'aplicació Android.
3. Proporcionar un sistema d'extracció de definicions de la Wikipèdia per a cadascun dels punts d'interès.

Tot aquest sistema requereix una base de dades pròpia per tal de contenir la informació relativa als usuaris, a les peticions d'extracció i als punts d'interès extrets. Aquesta base de dades, forma part del sistema però d'alguna forma n'és externa. Estaria allotjada en un servidor i des del nostre producte hi tindriem una connexió d'escriptura, lectura o escriptura/lectura, per tal de llançar-hi de forma automàtica sentències MySQL generades pels diferents mòduls del sistema total.

A la següent imatge podem veure el mapa complet del flux de dades entre tots els mòduls del sistema:



*Imatge 5. Flux de dades del sistema*

Mirant el graf, podem afirmar que el sistema realitza les següents accions:

1. El mòdul d'extracció de punts d'interès atén peticions dels usuaris.

2. Per a cada petició, consulta l'arxiu o direcció web (url) assignada i, a partir d'una llista precalculada, n'extreu els punts d'interès.
3. Enregistra cada punt d'interès que troba a la base de dades.
4. Per una altra banda, l'usuari accedeix a la interfície d'usuari Android a través d'un *smartphone*.
5. L'aplicació Android consulta l'API de la Wikipèdia per tal d'extreure una definició per a cada punt d'interès.

És vital remarcar la importància de l'establiment de l'estructura de la base de dades i de les entrades i sortides de cada mòdul entre els diferents participants en el desenvolupament del sistema general. Aquests acords són la base del correcte funcionament final del sistema una vegada posades en comú les dues parts.

### 3.3.1 Disseny i implementació de la base de dades

El subministrador de bases de dades triat és el MySQL i el seu gestor principal via web *phpMyAdmin*.

Això és perquè hem treballat amb el servidor XAMPP i resulta molt més còmode, ràpid, lleuger i adaptable treballar amb MySQL. Entre d'altres funcionalitats aquest permet exportar bases de dades completes en fitxers senzills de la mateixa manera que importar-ne una de guardada. A més, es tracta de software lliure i s'integra total i completament amb el sistema operatiu Windows i amb el llenguatge de programació escollit: Java. D'altra banda, cal comentar que l'Eclipse permet la connexió directa amb qualsevol dels subministrador de dades més comuns, cosa que fa que el nostre sistema summament portable a qualsevol altre entorn.

Pel nostre sistema es requereix una base de dades molt senzilla i òptima de només tres taules. El nostre sistema llença les consultes sobre aquesta usant una connexió de lectura / escriptura per tal d'analitzar, afegir i actualitzar la informació.

La base de dades del sistema consta de tres taules: *users*, *requests* i *pdis*. D'aquestes tres, la part del sistema que desenvolupem només interactua amb dues d'elles (*requests* i *pdis*). Tot i així, a continuació s'explica detalladament l'estructura de totes tres taules.

La taula *users* ens serveix per emmagatzemar la informació dels usuaris del sistema.

L'estructura d'aquesta taula, extreta directament del gestor *phpMyAdmin* és la següent:

Columna	Tipo	Nulo	Predeterminado	Extra
<b>id</b>	int(11)	No	Ninguna	AUTO_INCREMENT
<b>username</b>	varchar(5)	No	Ninguna	- - -
<b>password</b>	varchar(50)	No	Ninguna	- - -

Taula 20. Estructura de la taula 'users' de la BD

L'ús de cada camp és el següent:

- id*: és un índex auto-incremental i és la clau primària de la taula.
- username*: és la columna on es guarda el nom d'usuari.
- password*: és la columna on es guarda la contrasenya de l'usuari.

La taula *requests* és la més important del sistema: per una banda permet als usuaris inserir noves peticions a la base de dades; per una altra, proveeix al mòdul d'extracció de punts d'interès de la informació necessària per a que pugui realitzar la seva funció.

Un dels problemes que es van trobar a l'hora de dissenyar el sistema és el cas que un mateix usuari o dos usuaris independents poguessin pujar arxius amb noms idèntics. En un cas com l'esmentat, per defecte, l'arxiu nou substituiria l'antic, sense tenir en compte si el contingut n'és el mateix. Per solucionar aquest problema, es pren la decisió d'assignar a cada petició un identificador aleatori (*ticket*) que el fa diferenciable de la resta, per tal d'evitar aquesta problemàtica situació.

L'estructura d'aquesta taula, extreta directament del *phpMyAdmin* és la següent:

Columna	Tipo	Nulo	Predeterminado	Extra
<b>id</b>	int(11)	No	Ninguno	AUTO_INCREMENT
<b>source</b>	Longtext	No	Ninguno	- - -
<b>type</b>	varchar(100)	No	Ninguno	- - -
<b>state</b>	varchar(50)	No	Ninguno	- - -
<b>user</b>	varchar(50)	No	Ninguno	- - -
<b>topic</b>	varchar(100)	No	Ninguno	- - -
<b>ticket</b>	varchar(50)	No	Ninguno	- - -
<b>visible</b>	Varchar(50)	No	Private	- - -

Taula 21. Estructura de la taula 'requests' de la BD

L'ús de cada camp és el següent:

-*id*: és un índex auto-incremental i és la clau primària de la taula.

-*source*: és la columna on es guarda el nom de l'arxiu o la direcció web a analitzar.

-*type*: és la columna on es guarda el tipus de petició (arxiu [*file*] o direcció web [*url*]).

-*state*: és la columna on es guarda l'estat de la petició (pendent [*pending*], processant [*processing*], done [*processada*] o error [*error*]).

-*user*: és la columna on es guarda el nom de l'usuari que ha generat la petició.

-*topic*: és la columna on es guarda el tema al que fa referència la petició (p.e.: *Costa Catalana, Gaudí, Barcelona...*).

-*ticket*: identificador únic assignat a la petició. Per solucionar el problema de que dos usuaris puguin arxius amb noms idèntics, els arxius que es pugen al servidor FTP s'enregistren com '*ticket + nom\_de\_l'arxiu*'.

-*visible*: és la columna on s'especifica si es tracta d'un tòpic privat (només el pot veure l'usuari que ha generat la petició) o públic (el poden veure tots els usuaris del sistema).

La taula *pdis* ens serveix per emmagatzemar la informació relativa als punts d'interès que el sistema extreu dels documents d'usuari.

L'estructura d'aquesta taula, extreta directament del *phpMyAdmin* és la següent:

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Extra</b>
<b>id</b>	<i>int(11)</i>	<i>No</i>	<i>Ninguno</i>	<i>AUTO_INCREMENT</i>
<b>name</b>	<i>varchar(50)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>latitude</b>	<i>varchar(50)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>longitude</b>	<i>varchar(50)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>source</b>	<i>longtext</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>topic</b>	<i>varchar(100)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>user</b>	<i>varchar(50)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>
<b>ticket</b>	<i>varchar(50)</i>	<i>No</i>	<i>Ninguno</i>	<i>- - -</i>

Taula 22. Estructura de la taula '*pois*' de la BD

L'ús de cada camp és el següent:

-*id*: és un índex auto-incremental i és la clau primària de la taula.

-*name*: és la columna on es guarda el nom del punt d'interès.

-*latitude*: és la columna on es guarda el valor de latitud del punt d'interès.

-*longitude*: és la columna on es guarda el valor de longitud del punt d'interès.

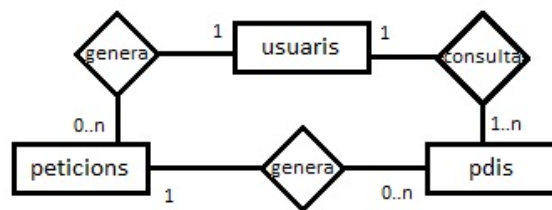
-*source*: és la columna on es guarda el nom del document o direcció web del qual s'ha extret el punt d'interès.

-*topic*: és la columna on es guarda el tema que té assignada la petició de la qual s'ha extret el punt d'interès.

-*user*: és la columna on es guarda l'usuari que té assignada la petició de la qual s'ha extret el punt d'interès.

-*ticket*: és la columna on es guarda el *ticket* que té assignada la petició de la qual s'ha extret el punts d'interès.

A continuació podem veure el model relacional de les taules exposades anteriorment:



*Imatge 6. Model relacional de la BD*

Com es pot veure, un usuari pot generar un número indeterminat de peticions, cadascuna de les quals, a la seva vegada, poden generar un número indeterminat de punts d'interès que, més tard, l'usuari pot consultar.

### **3.3.2 Disseny i implementació del mòdul d'extracció automàtica de punts d'interès**

El mòdul d'extracció automàtica de punts d'interès és un dels grans reptes del projecte. Es tracta d'un subsistema capaç de processar llenguatge natural i identificar dins del text els punts d'interès que hi puguin aparèixer. Un cop identificat un punt d'interès cal, també, que sigui capaç d'assignar-li una geo-localització en forma d'un parell de valors de latitud i longitud.



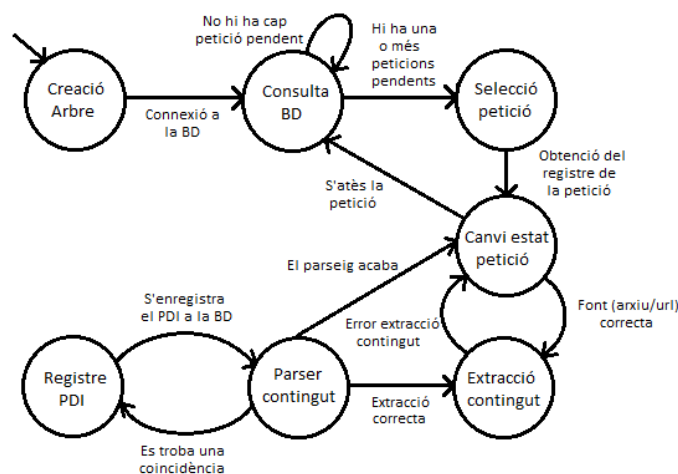
Des d'un primer moment es planteja el funcionament d'aquest mòdul com el d'un procés de tipus dimoni, és a dir, s'executa de forma contínua (infinita) i sempre en segon pla, sense ser controlat directament per l'usuari.

El disseny inicial d'aquest mòdul s'ha fet de forma bàsica però molt eficient, tot utilitzant una col·lecció finita de punts d'interès precalculats. A partir d'aquest conjunt de punts, el mòdul crea una estructura en forma d'arbre per tal de realitzar la cerca de forma més còmoda més endavant. Un cop creat l'arbre, es consulten a la base de dades les peticions pendents i, per a cadascuna, es llença un *thread* de tipus processador. Aquest processador obre el fitxer o direcció web al qual fa referència la petició, n'extreu el contingut i el parseja fent servir l'arbre que s'ha creat anteriorment. Per a cada coincidència, el processador enregistra un punt d'interès a la base de dades. Cal afegir que, en paral·lel a l'anàlisi i parseig del document o direcció web, el processador va mantenint actualitzat l'estat de la petició.

Així doncs, a grans trets els passos que duu a terme aquest mòdul són:

1. Creació de l'arbre a partir de la col·lecció preestablerta de punts d'interès.
2. Consulta de peticions pendents a la base de dades.
3. Per a cada petició pendent:
  - a. Canvia l'estat de la petició a processant.
  - b. Extreu el contingut.
  - c. Parseja el contingut.
    - i. Enregistra les coincidències a la base de dades.
  - d. Canvia l'estat de la petició a processada.

Així doncs, el diagrama d'estats associats a aquest mòdul és el següent:



**Imatge 7. Diagrama d'estats del mòdul d'extracció de punts d'interès**

Per a dur a terme aquest procés s'utilitzen tot un conjunt de classes que a continuació es detallen.

### -Classe *Node.java*

La classe *Node.java* permet la creació d'objectes de tipus *node* que s'utilitzaran per a crear l'arbre de punts d'interès a partir del qual es farà l'extracció.

Els atributs de la classe *Node.java* són:

-*key*: caràcter que identifica l'objecte *node*.

-*children*: vector d'elements de tipus *char* (caràcter) que conté tots els possibles caràcters (*keys*) als que es pot accedir a partir d'aquest *node*.

-*nodes*: vector d'elements de tipus *node* que conté tots els nodes fill del *node* actual.

-*finalState*: bolean que indica si es tracta d'un *node* fulla (*true*) o no (*false*).

-*poi*: cadena de caràcters que en el cas de tractar-se d'un *node* fulla conté el nom del punt d'interès; a la resta de casos (*nodes* no terminals), és una cadena buida.

-*geolocation*: cadena de caràcters que en el cas de tractar-se d'un *node* fulla conté la geo-localització del punt d'interès amb el següent format: "*latitude,longitude*"; a la resta de casos (*nodes* no terminals), és una cadena buida.

Els mètodes associats a la classe *Node.java* són:

-*Node(char key)*: constructor de la classe, que assigna a l'atribut *key* el caràcter que rep com a paràmetre.

-*addChild(char child)*: funció que (a) afegeix el caràcter que rep com a paràmetre al vector *children* i (b) crea un nou objecte *Node* al que li passa com a *key* el caràcter que rep com a paràmetre i que afegeix al vector *nodes*.

Es tracta, per tant, de la classe més senzilla dins el mòdul d'extracció de punts d'interès, però d'importància vital per al seu correcte funcionament.

### Classe *Tree.java*

La classe *Tree.java* permet la creació d'objectes de tipus arbre, essencials per a emmagatzemar de forma eficient la informació relativa als punts d'interès precalculats. Aquesta classe facilita, a més, la posterior tasca de cerca de coincidències en els documents i direccions web.

Els atributs de la classe *Tree.java* són:

-*root*: objecte de tipus *Node* que apunta a l'arrel de l'arbre, que per defecte conté el caràcter '/'.

-*source*: cadena de caràcters que conté la ruta a l'arxiu de text pla amb la informació precalculada dels punts d'interès.

Els mètodes associats a la classe *Tree.java* són:

-*Tree(String source)*: constructor de la classe, que assigna a l'atribut *source* la cadena de caràcters que rep com a paràmetre.

-*createTree()*: mètode principal de la classe i pilar de l'estructura en arbre que crea el mòdul. El pseudocodi associat a aquest mètode és el següent:

```
Creació d'un objecte File a partir de l'atribut source
Creació del node arrel de l'arbre

node_actual = node_arrel

Si existeix l'arxiu {
    Per a cada línia {
        Per a cada caràcter de la línia {
            key = caràcter

            Si el vector children del node no conté el
            caràcter {
                Afegeix un nou node_fill al node_actual
                node_actual = node_fill
            } si no {
                node_actual = node_fill
            }
        }
    }
}
```

```
        node_actual.finalState = true

        node_actual.poi = nom_poi

        node_actual.geolocation = geo-localització_poi

        node_actual = node_arrel

    }

}
```

Es tracta, per tant, de la classe principal per a dur a terme una de les tasques més importants del mòdul: la creació d'una estructura on emmagatzemar la informació relativa als punts d'interès precalculats.

#### Classe *Processor.java*

La classe *Processor.java* és el motor del mòdul i el seu correcte funcionament i eficiència són els que determinen l'èxit o el fracàs de la tasca que ha de dur a terme el mòdul d'extracció. Cal dir que es tracta d'una classe que hereta de la classe *Thread* de Java i que, per tant, té accés a tots els seus mètodes.

Els atributs de la classe *Processor.java* són:

- id*: número d'identificació de la petició que s'està processant.
- source*: cadena de caràcters que conté el nom del document o direcció web que s'està processant i d'on s'extreu el contingut a analitzar.
- type*: cadena de caràcters que indica si la petició fa referència a un arxiu o a una direcció web.
- user*: cadena de caràcters que conté el nom de l'usuari que ha generat la petició.
- topic*: cadena de caràcters que conté el nom del tema que l'usuari ha assignat a la petició.
- ticket*: cadena de caràcters que conté l'identificador únic de la petició.

-*tree*: objecte de tipus *Tree* que el processador utilitza per a fer el parseig del contingut del *source*. Aquest objecte, com s'ha vist anteriorment, conté la informació relativa als punts d'interès precalculats.

Els mètodes associats a la classe *Processor.java* són:

-*processor(int id, String source, String type, String user, String topic, String ticket, Tree tree)*: constructor de la classe, que assigna a cada atribut de l'objecte el valor que rep com a paràmetre.

-*run()*: sobrecàrrega del mètode *run()* propi de la classe *Thread* de la qual hereta la classe *Processor*. Controla el flux principal de procés del mòdul i el seu pseudocodi associat és el següent:

```
Connexió a la base de dades del sistema

Actualització de l'estat de la petició a 'processant'

Si es tracta d'un arxiu {

    Creació d'un objecte File a partir de l'atribut
    source

    Si es pot obrir l'arxiu {

        contingut = contingut_arxiu

    } si no {

        Actualització de l'estat de la petició a 'error'

    }

}

Si es tracta d'una direcció web {

    Si es pot obrir la direcció web {

        contingut = contingut_direcció_web

    } si no {
```

```
        Actualització de l'estat de la petició a 'error'
    }
}

Crida al mètode extractor(contingut, tree)

Actualització de l'estat de la petició a 'processada'

Desconnexió de la base de dades
```

-*extractor*(String contentString, Tree tree): és el mètode encarregat de fer el parseig del contingut de l'arxiu o direcció web i d'extreure'n els punts d'interès que hi apareixen tot fent servir l'arbre *tree*. El pseudocodi associat a aquest mètode és el següent:

```
Creació d'un objecte Node node_actual que apunta a l'arrel
de l'arbre

Per a cada caràcter del contingut {

    Si node_actual té com a fill al caràcter {

        node_actual = node_fill

        Si node_actual és un node fulla {

            Connexió a la base de dades

            Enregistrament del punts d'interès

            Desconnexió de la base de dades

        }

    } si no {

        node_actual = node_arrel

    }

}
```

Cal assenyalar que el darrer mètode *extractor* és el que implementa l'algoritme de cerca de punts d'interès. En un principi, aquest algoritme el que feia era, per a cada punt d'interès precalculat comprovar si es trobava una coincidència a la cadena del contingut de l'arxiu o direcció web. Aquesta solució era, però poc eficient (ordre  $n^2$ ); amb la implementació de l'estructura en arbre i l'algoritme de cerca final, hem aconseguit una cerca molt més ràpida i eficient (ordre  $n$ ).

Es tracta, per tant de la classe que suporta el major pes del mòdul d'extracció de punts d'interès i que determinarà, per tant, el seu funcionament i rendiment.

### Classe *Main.java*

En darrer lloc es crea la classe *Main.java* com a conductora del flux principal del mòdul. El pseudocodi associat a aquesta classe és el següent:

```
Creació de l'arbre a partir de l'arxiu font
while (true) {          --> Per sempre
    Connexió a la base de dades
    Obtenció de les peticions pendents
    Per a cada petició (5 simultànies com a màxim) {
        Llançament d'un thread de tipus Processor
    }
}
```

Davant la possibilitat que el número d'usuaris del sistema sigui molt gran, hem hagut de pensar en un mecanisme que permeti que el mòdul compleixi la seva funció sense saturar-se. És en aquest punt quan es pren la decisió de gestionar les peticions amb cues, que permeten, de forma ràpida i senzilla, fixar un número màxim d'atenció de peticions simultànies.

Per fer-ho, fem servir una estratègia de consumidor i productor. Per una banda, els usuaris generen peticions a la base de dades i per l'altra el dimoni va atenent-les. En principi, el dimoni llençava tants fils de processament com peticions pendents existissin. Un cop madurat el codi, i analitzant les complicacions que aquesta forma de treballar pot

portar, es crea un sistema de producció-consum on, encara que existeixin més de  $n$  peticions, el dimoni només pot atendre'n  $n$  simultàniament. Per fer-ho assigna a cada petició un identificador únic entre 1 i  $n$ , que no podrà assignar a cap altre petició nova fins que la inicial estigui processada.

Aquesta és, per tant, de la classe central del mòdul d'extracció de punts d'interès.

L'execució d'aquesta classe i del mòdul en conjunt es fa exportant el projecte com a un arxiu de tipus .jar (Java ARchive). Aquest tipus d'arxius són fàcilment executables des de la consola de Windows. En el nostre cas, el llacem mitjançant l'execució d'aquest senzill script:

```
cd C:\Users\Victor\Desktop\  
java -jar daemon.jar  
pause
```

, que permet fer córrer l'arxiu *daemon.jar* com un procés de tipus dimoni. Aquest tipus de processos s'executen de forma infinita i sempre en segon pla, és a dir, sense ser controlat per cap usuari. De totes formes, a la finestra de la consola el dimoni va informant de totes aquelles accions que realitza, tals com connectar-se a la base de dades o enregistrar un punt d'interès, entre altres. A més, per seguretat, es genera automàticament un document de log fent servir la classe *Logger* que Java proporciona.

### 3.3.3 Disseny i implementació de la interfície de l'aplicació Android

A l'hora de dissenyar la interfície d'usuari de l'aplicació Android, hem volgut tenir en compte les característiques principals que diferencien una bona interfície d'usuari d'una altre que no ho és. Aquestes característiques són:

1. Claredat: una interfície neta no necessita manuals, ni dona lloc a confusions. Amb això assegurem que els nostres usuaris cometin menys fallades mentre l'utilitzen.
2. Concisió: està relacionada amb la qualitat anterior i sosté que com més es sobrecarregui la nostra interfície, més difícil serà per l'usuari trobar allò que busca. Hem de ser concisos i especificar tot breument.



3. Familiaritat: s'han d'aprofitar i reutilitzar elements preestablerts a la societat ja que ajuden bastant a comprendre fàcilment la nostra interfície les primeres vegades que l'usuari l'utilitza.
4. Sensibilitat: es refereix a dos conceptes:
  - a. Rapidesa de la interfície.
  - b. Bona realimentació: l'usuari ha de saber en tot moment què està fent, tant si ho aconsegueix, com si no.
5. Consistència: una vegada que el usuari aprenguin certes parts del maneig de la interfície, es pot extrapolar aquest maneig a noves àrees i funcionalitat de la interfície.
6. Estètica: una interfície visual atractiva fa que l'ús de la interfície sigui més agradable i augmenti la satisfacció dels usuaris.
7. Eficiència: el temps és or i una bona interfície d'usuari hauria d'estalviar-nos molt de temps i esforç.
8. Errors: la forma amb la que la interfície controla els errors serà una prova de qualitat global. Una bona interfície no ha de castigar els usuaris pels seus errors sinó que ha de proporcionar la intenció de posar-los remei.

Dissenyar una interfície d'usuari que contingui totes aquestes qualitats és difícil perquè la modificació d'alguna qualitat en concret afecta a la resta. D'un bon equilibri entre totes aquestes qualitats dependrà l'èxit i la qualitat de la nostra interfície d'usuari.

A més, cal dir que s'ha aplicat un disseny centrat en l'usuari. Aquesta és una filosofia de disseny que té per objectiu la creació de productes que resolguin necessitats concretes dels seus usuaris finals, aconseguint la major satisfacció i millor experiència d'ús possible amb el mínim esforç per parts d'aquests. Pren forma com un procés en el qual s'utilitzen una sèrie de tècniques multidisciplinàries i on cada decisió que es pren ha d'estar basada en les necessitat, objectius, expectatives, motivacions i capacitats dels usuaris.

La majoria de processos que fan disseny centrat en l'usuari segueixen el següent esquelet:

1. Identificar els usuaris finals de l'aplicació: en el nostre cas, qualsevol usuari d'smartphone amb sistema operatiu Android. En general, es tracta de persones joves i amb terminals amb una sèrie de característiques específiques com són l'accés a Internet, entrada tàctil o l'ús del GPS.
2. Dissenyar un producte que resolgui les seves necessitats i s'ajusti a les seves capacitats, expectatives i motivacions: en el nostre cas, una interfície que permeti a l'usuari loguejar-se en el sistema, accedir a una llista dels seus tòpics i punts d'interès i, per a cada punt d'interès, obtenir la seva definició de la Wikipèdia, la seva posició en el mapa i una ruta de com arribar-hi des de la seva posició actual.

3. Posar a prova el disseny: utilitzant test unitari, com es veurà en el següent apartat de la memòria.

Per fer el disseny de la interfície d'usuari s'ha creat un nou projecte de tipus Android Application a l'entorn de desenvolupament Eclipse i s'ha especificat l'estructura dels diferents *layouts* o vistes que l'aplicació utilitza.

Així doncs, i tenint en compte tots aquests aspectes, s'han dissenyat les vistes que a continuació s'especifiquen i amb les quals definim per complet el que seria la interfície amb la que l'usuari de l'aplicació mòbil Android interactuarà. Tot plegat, un disseny senzill, intuïtiu i fàcilment implementable.

#### Vista *main.xml*

La primera vista o vista principal *main.xml* proporciona a l'usuari un simple formulari per tal de que pugui loguejar-se al sistema. Per fer-ho, s'ha codificat la següent estructura de *layouts*:

```
<LinearLayout> //orientació vertical i contingut centrat
    <TableLayout> //taula de continguts
        <TableRow> //primera fila de la taula
            <TextView> //etiqueta 'Usuari'
            </TextView>
            <EditText> //input nom_usuari
            </EditText>
        </TableRow>
        <TableRow> //segona fila de la taula
            <TextView> //etiqueta 'Contrasenya'
            </TextView>
            <EditText> //input contrasenya_usuari
            </EditText>
        </TableRow>
    </TableLayout>
</LinearLayout>
```

```
<TableRow> //tercera fila de la taula  
    <Button> //botó de login  
    </Button>  
</TableRow>  
</TableLayout>  
</LinearLayout>
```

, a partir de la qual es genera una vista com la següent:



*Imatge 8. Vista main.xml*

Un cop introduïts el nom d'usuari i la contrasenya, es poden produir dues situacions:

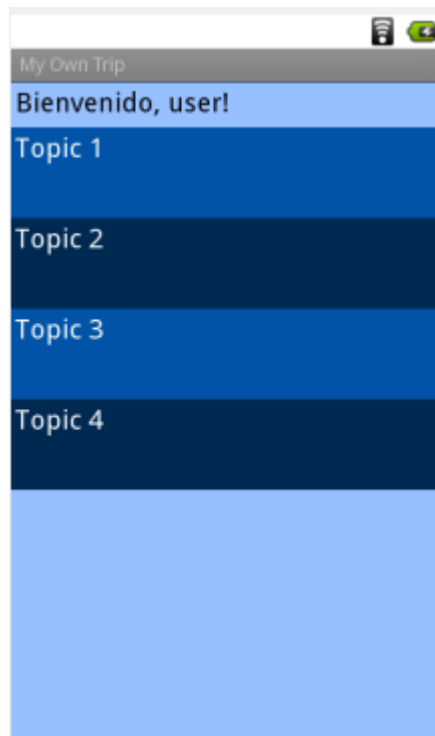
1. No es pot iniciar sessió correctament: davant d'aquesta opció, es mostra un missatge d'error.
2. Es pot iniciar sessió correctament: davant d'aquesta opció, es mostra la vista *usertopics*.

Vista *usertopics.xml*

Un cop s'ha iniciat la sessió, la segona vista *usertopics.xml* proporciona a l'usuari una llista dels seus temes o *topics*, en base als seves peticions a la base de dades. Per fer-ho s'ha codificat la següent estructura de *layouts*:

```
<LinearLayout> //orientació vertical
    <TextView> //salutació a l'usuari
</TextView>
    <ScrollView> //vista que permet scroll
        <LinearLayout> //orientació vertical
            <TextView> //nom del topic 1
            </TextView>
            <TextView> //nom del topic 2
            </TextView>
            ...
            <TextView> //nom del topic n
            </TextView>
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

, a partir de la qual es genera una vista com la següent:



**Imatge 9. Vista usertopics.xml**

Clicant a sobre d'un dels temes disponibles, es poden produir dues situacions:

1. El GPS no està disponible i no es pot realitzar cap acció: davant d'aquesta opció, es mostra un missatge d'error.
2. Es calcula la posició actual i es calculen els deu punts d'interès més propers: davant d'aquesta opció, es mostra la vista *userpois*.

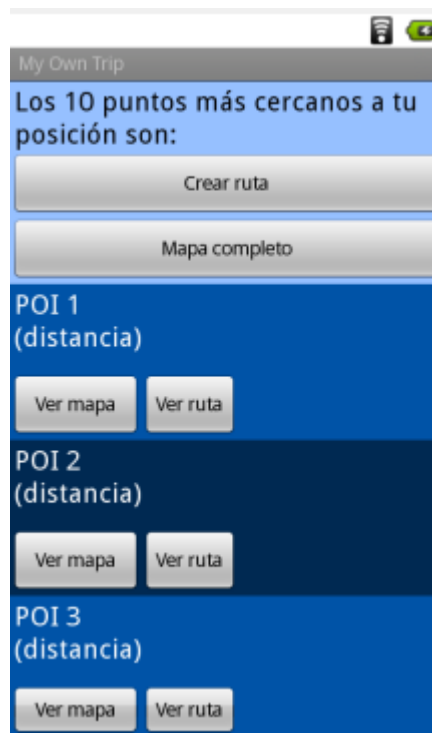
Vista *userpois.xml*

Un cop s'ha clicat sobre un dels temes d'usuari disponibles, la tercera vista *userpois.xml* proporciona a l'usuari una llista dels deu punts d'interès més propers a la seva posició. Per fer-ho, s'ha fet servir la següent estructura de *layouts*:

```
<LinearLayout> //orientació vertical
    <TextView>
    </TextView>
    <ScrollView> //vista que permet scroll
        <Button> //botó ruta personalitzada
        </Button>
        <Button> //botó mapa amb tots els punts
        </Button>
        <LinearLayout> //orientació vertical
            <TextView> //nom del poi 1 (distancia real)
            </TextView>
            <LinearLayout> //orientació horitzontal
                <Button> //botó mapa
                </Button>
                <Button> //botó com arribar-hi
                </Button>
            </LinearLayout>
            <TextView> //nom del topic 2
            </TextView>
            ...
    </ScrollView>
</LinearLayout>
```

```
<TextView> //nom del topic n
    </TextView>
</LinearLayout>
</ScrollView>
</LinearLayout>
```

, a partir de la qual es genera una vista com la següent:



*Imatge 10. Vista userpois.xml*

En aquesta interfície es poden realitzar diferents accions:

1. Clicar al botó 'Crear ruta': s'obre una nova vista que a continuació s'explica i que permet a l'usuari crear una ruta amb una subselecció del total de punts que es mostren.
2. Clicar al botó 'Mapa completo': es mostra un mapa estàtic amb tots els punts d'interès.

3. Clicar sobre el nom del punt d'interès: s'obre un pop-up amb una breu descripció del punt d'interès, extreta de la Wikipèdia.
4. Clicar al botó 'Ver mapa': es mostra el punt d'interès en un mapa.
5. Clicar al botó 'Ver ruta': es mostra un mapa amb la ruta des de la posició actual fins al punt d'interès.

#### Vista *userroute.xml*

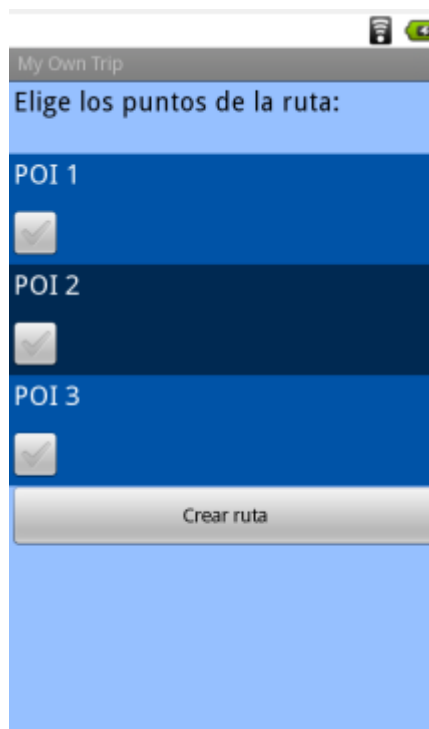
Clicant al botó 'Crear ruta', la quarta vista *userroute.xml* proporciona a l'usuari una llista de punts d'interès similar a la de la vista anterior però amb un checkbox a cada punt, que permet decidir quins punts incloure a la ruta. Per fer-ho s'ha fet servir la següent estructura de *layouts*:

```
<LinearLayout> //orientació vertical
    <TextView>
    </TextView>
    <ScrollView> //vista que permet scroll
        <LinearLayout> //orientació vertical
            <TextView> //nom del poi 1
            </TextView>
            <CheckBox>
            </CheckBox>
            <TextView> //nom del topic 2
            </TextView>
            <CheckBox>
            </CheckBox>
            ...
            <TextView> //nom del topic n
            </TextView>
            <CheckBox>
            </CheckBox>
```



```
</LinearLayout>  
  
</ScrollView>  
  
</LinearLayout>
```

, que genera una vista com la següent:



*Imatge 11. Vista userroute.xml*

Un cop escollits els punts, fent clic a 'Crear ruta', es mostra un mapa amb la ruta corresponent.

### Menu.xml

Des de totes les vistes especificades anteriorment l'usuari pot, a més, configurar una sèrie de propietats mitjançant un menú. Per fer-lo, s'ha fet servir la següent estructura:

```
<menu>
  <item>                                //Opció principal: Configuració
    <item></item>                        //Sub-opció 1: Canviar IP
    <item></item>                        //Sub-opció 2: Canviar n° pdis
  </item>
```

, que genera un menú molt intuïtiu i on l'usuari pot:

-Canviar la IP: un dels inconvenients de treballar de forma local és que segons on treballem, la IP del servidor canvia. Per agilitzar la feina i no haver d'instal·lar l'aplicació cada vegada que ens movem de xarxa, es dóna la possibilitat en aquest prototip de canviar la IP on es volen fer les consultes. Cal dir però, que un cop implantat el sistema a la web, aquesta opció desapareixeria.

-Canviar el número de punts d'interès: per defecte, quan fem clic sobre un topic, es mostra una vista amb els deu punts d'interès més propers a la posició actual. Amb aquesta opció es pot configurar el número de punts d'interès que volem que es calculin.

### **3.3.4 Disseny i implementació de la funcionalitat d'extracció de definicions de la Wikipèdia**

Una de les funcionalitats que el sistema ha de ser capaç d'oferir és la de que, donat el nom d'un punt d'interès, l'usuari pugui obtenir una breu descripció d'aquest en base a l'article que la Wikipèdia ofereix.

A l'hora de plantejar el disseny i implementació d'aquesta funcionalitat hem vist que existeixen diferents alternatives:

1. Accedir directament a l'article de la Wikipèdia i obrir la direcció web amb el navegador.
2. Utilitzar l'API que la Wikipèdia proveeix i que permet, a partir de la construcció d'una direcció web url, accedir al contingut de l'article.

3. Utilitzar l'aplicació Android que la Wikipèdia posa a disposició de forma gratuïta al Market GooglePlay.

Un cop analitzades les tres alternatives, vam posar per davant l'eficiència i facilitat d'ús de cadascuna i vam arribar a la conclusió de que la més òptima a utilitzar és la segona. Aquesta opció ens permet:

-Obtenir el contingut de l'article sencer de la Wikipèdia disposant-ne només del nom del punt d'interès.

-Crear una funció que parsegi el contingut i identifiqui el primer paràgraf de l'article, que normalment és el que conté la informació essencial.

La implementació del mètode és tan senzilla com utilitzar la següent url:

```
http://es.wikipedia.org/w/api.php?action=parse&prop=text&page=NOM_PUNT_INTE  
RÈS&format=json
```

, posant com a valor del paràmetre *page* el nom del punt d'interès que vulguem consultar. La única consideració que cal tenir en compte és que si el nom del punt d'interès conté espais en blanc, caldrà substituir-los per la seqüència '%20'.

Així doncs, un cop construïda la direcció web, es guarda en una variable de tipus *string* el contingut de l'article de la Wikipedia. Per identificar i extreure el primer paràgraf de l'article, es segueix la següent metodologia:

1. Fer un *split* del contingut utilitzant com a clau l'etiqueta que marca l'inici del primer paràgraf ('<p>'). Aquest *split* genera dues subcadena, la segona de les quals conté tot allò que hi ha immediatament després de l'etiqueta d'inici de paràgraf.

```
_____ <p> _____  
(subcadena 1)      (subcadena 2)
```

2. Fer un *split* de la segona subcadena utilitzant com a clau l'etiqueta que marca el final del primer paràgraf ('</p>'). Aquest *split* genera dues subcadena, la primera de les quals conté el que hi ha abans del final del paràgraf.

```
_____ </p> _____  
(subcadena 1)      (subcadena 2)
```

3. Agafar el contingut que quedi fora d'etiquetes html.
4. Substituir caràcters especials per evitar errors.

En alguns cassos, el resultat final conté 'REDIRECCIÓN' o 'REDIRECT'. Quan això passa, vol dir que no s'ha trobat cap article associat al punt d'interès. La bona notícia, és

però, que la Wikipedia proposa una clau alternativa, que podem utilitzar per a fer una nova cerca.

### 3.4 Diagrames del sistema

Els diagrames associats al sistema que hem desenvolupat ajuden a entendre millor i d'una forma una mica més detallada el funcionament i la interacció de cada mòdul amb la resta i del sistema en general.

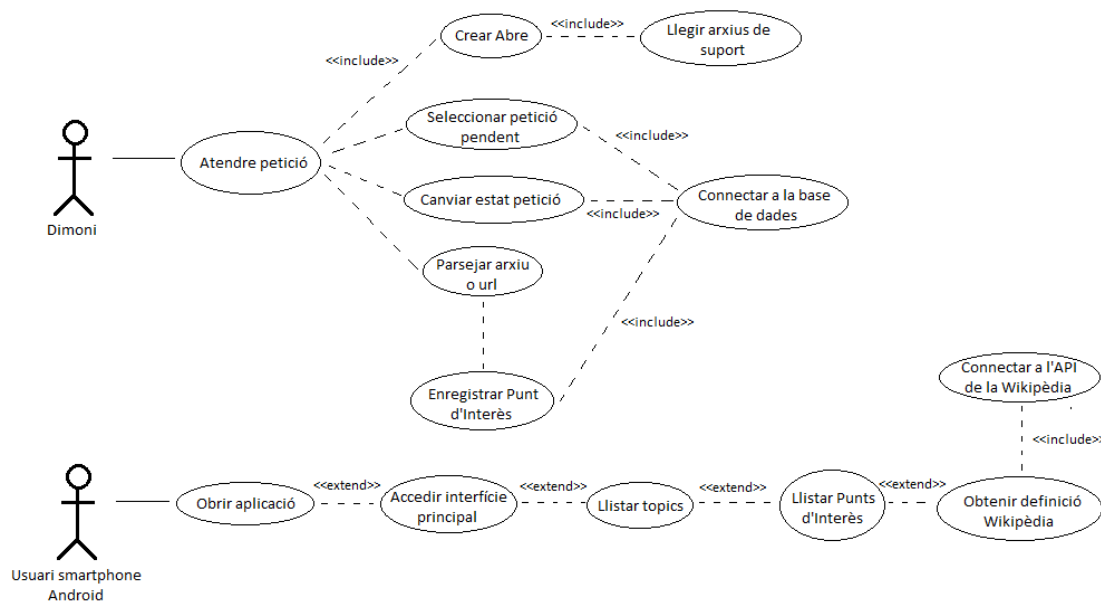
La claredat amb la que el conjunt de diagrames descriuen el sistema és també un indicatiu del correcte disseny i estructuració d'aquest i conformen una part molt important del projecte en sí.

A continuació, exposem els diagrames principals associats al sistema.

#### 3.4.1 Diagrama de casos d'ús

El diagrama de casos d'ús és el principal del sistema. Ens ajuda a identificar els diferents actors presents en el sistema i les accions que poden realitzar, de forma directa o indirecta, cadascun.

El diagrama de casos d'ús associat al sistema és el següent:



**Imatge 12. Diagrama de casos d'ús del sistema**

Al diagrama es poden identificar dos actors principals:

1. El mòdul d'extracció de punts d'interès o dimoni.

## 2. Usuari smartphone Android.

, així com les principals accions a les que poden accedir cadascun, que constitueixen els casos d'ús:

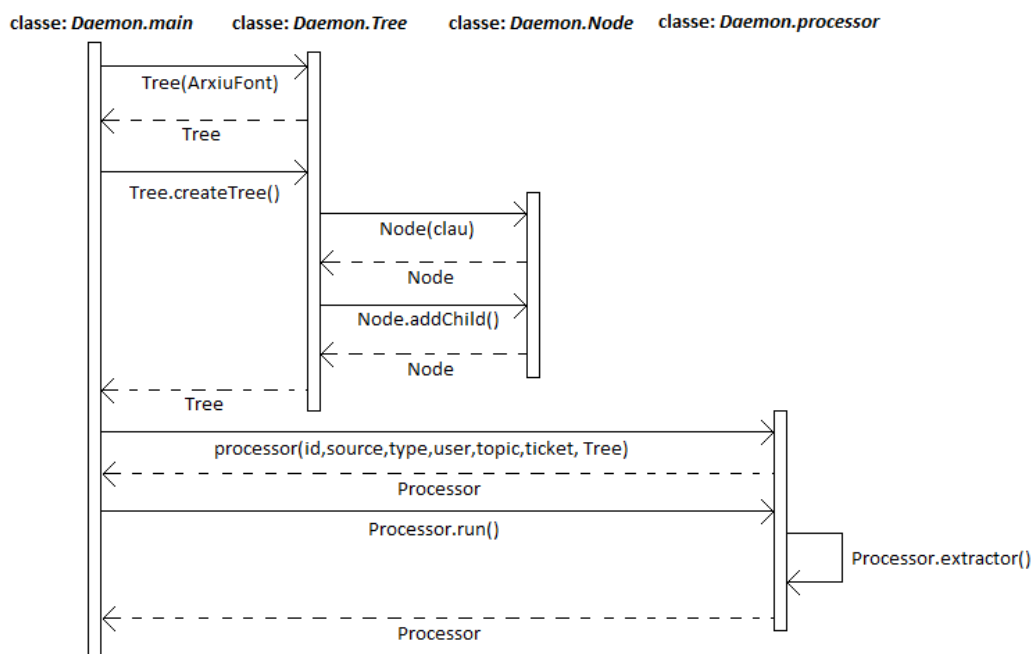
1. Atendre petició: el dimoni, com a actor, la única acció que és capaç de dur a terme de forma directa és la d'atendre peticions. Un cop que l'acciona, però, és capaç d'accedir a altres casos d'ús secundaris i derivats del principal com són crear l'arbre, parsejar arxius o enregistrar punts d'interès, entre altres.
2. Accedir a l'aplicació: l'usuari Android, com a actor, és capaç d'obrir l'aplicació i moure's d'una vista a un altra. A més, a partir d'una de les vistes, també pot obtenir l'article de la Wikipèdia relatiu a un punt d'interès.

Així doncs, gràcies al diagrama de casos d'ús hem pogut identificar els dos actors principals del nostre sistema així com els principals cassos d'ús, que farem servir per a la definició de la resta de diagrames.

### 3.4.2 Diagrama de seqüència

Els diagrames de seqüència mostren la interacció del conjunt de classes o objectes de l'aplicació en el temps i se'n modela un per a cada cas d'ús. A més, permet veure quin és l'intercanvi de missatges entre els diferents elements del sistema.

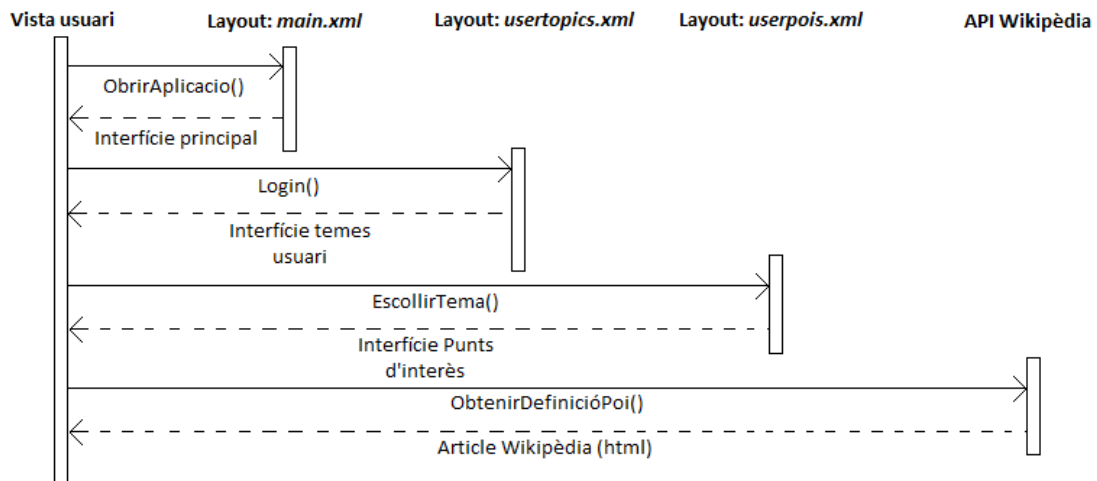
El diagrama de seqüència associat al cas d'ús *Atendre petició* és el següent:



Imatge 13. Diagrama de seqüència del cas d'ús 'Atendre petició'

En aquest cas, el diagrama de seqüència ens mostra la interacció entre les diferents classes del mòdul d'extracció de punts d'interès. Com es pot veure, aquest diagrama mostra bàsicament el procés que duu a terme la classe principal d'aquest mòdul, que s'ha descrit anteriorment mitjançant pseudocodi.

El diagrama de seqüència associat al cas d'ús *Accedir a l'aplicació* és el següent:



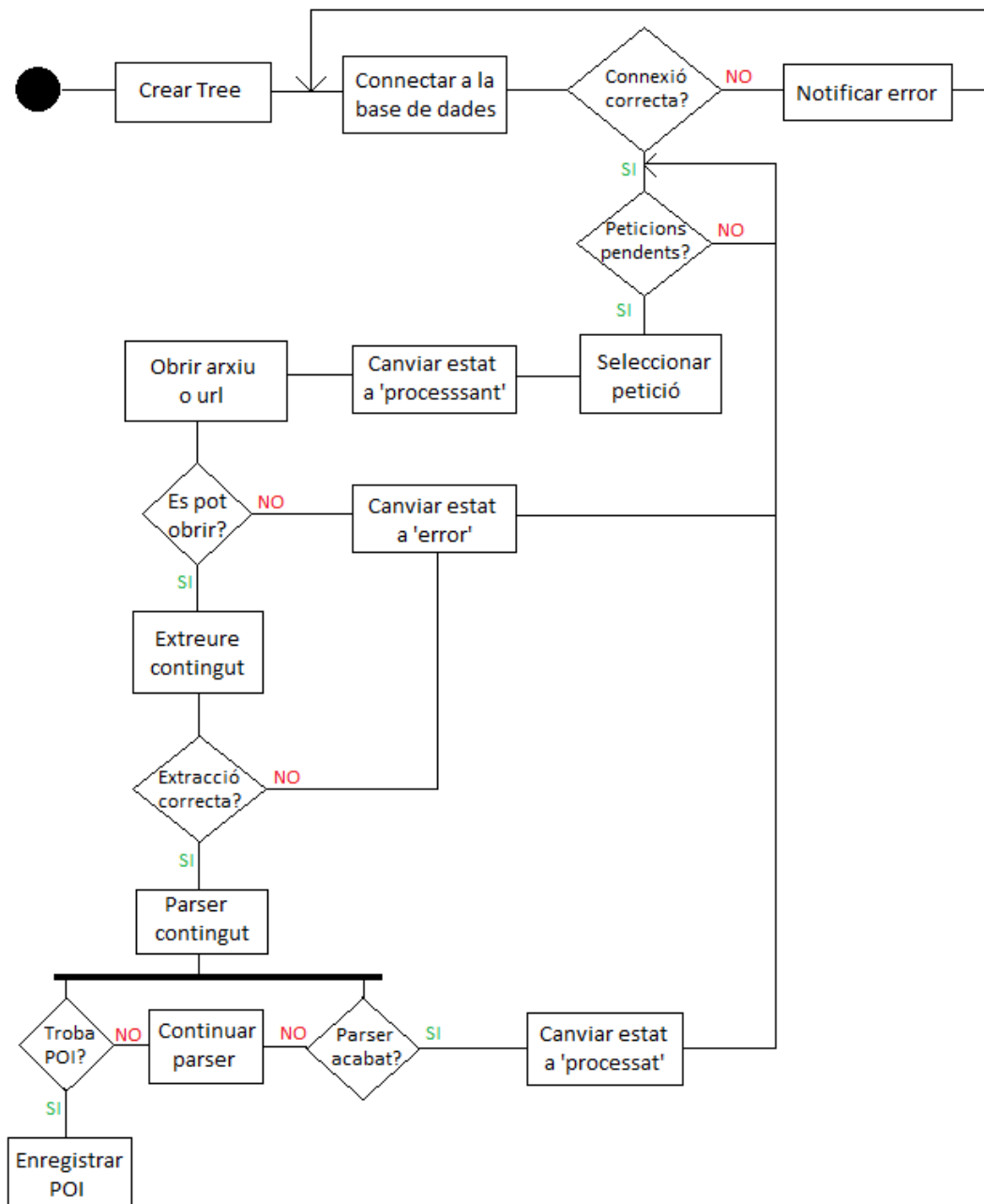
**Imatge 14. Diagrama de seqüència del cas d'ús 'Accedir a l'aplicació'**

En aquest cas, el que podem veure és com la vista de l'usuari va canviant segons amb quin *layout* es comunica l'usuari. Observem a més, que és un accés seqüencial en el que no es pot accedir a qualsevol *layout* en qualsevol moment, sinó que s'ha de saltar un d'un a un altre.

### 3.4.3 Diagrama d'activitats

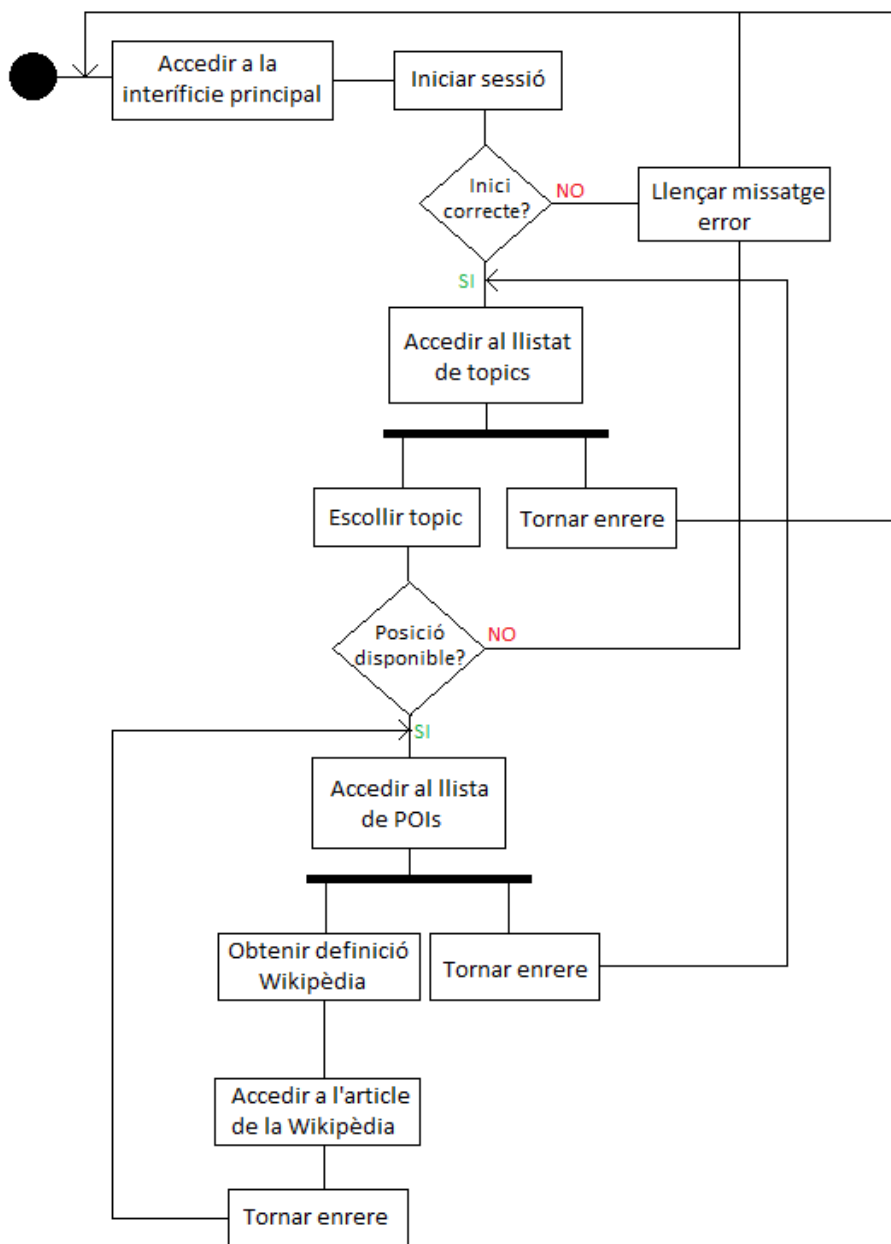
Per últim, els diagrames d'activitats ens donen una idea dels diferents fluxos de treball presents al sistema. Són molt explicatius i també se'n representa un per a cada cas d'ús.

El diagrama d'activitats associat al cas d'ús *Atendre petició* és el següent:



*Imatge 15. Diagrama d'activitats del cas d'ús 'Atendre petició'*

El diagrama d'activitats associat al cas d'ús *Accedir a l'aplicació* és el següent:



*Imatge 16. Diagrama d'activitats del cas d'ús 'Accedir a l'aplicació'*

En tots dos diagrames podem veure els diferents camins possibles dels fluxos principals i descriptius del sistema.



## 4. Test i proves

Per aquest apartat hem de tenir molt en compte que per al desenvolupament del sistema hem fet servir un disseny en forma de mòduls quasi independents. Es tracta, per tant, d'un projecte on es pot comprovar el correcte funcionament de l'aplicació aplicant proves unitàries a cadascun dels mòduls.

A més, cal recordar que aquest projecte no és sinó una part d'un sistema encara més gran, on caldrà avaluar que totes dues parts s'integrin de forma correcta.

Així doncs, a l'hora de realitzar les proves s'han aplicat uns criteris de test basats principalment en la funcionalitat del sistema. Tot i així, altres aspectes importants, com n'és la seguretat, també juguen un paper molt important en aquesta fase, en tant que es tracta d'un programari que es podria fer servir com a versió, si no final, quasi final del sistema.

S'han aplicat, per tant, proves d'unitat mòdul a mòdul per comprovar de forma independent que cadascun d'ells funciona correctament.

Paral·lelament, cada cop que un mòdul superava correctament les proves d'unitat per assegurar-ne la funcionalitat en els casos bàsics, s'han anat agrupant en el sistema per aplicar proves d'integració pas a pas.

D'altra banda, en no tenir un client que ens hagi encarregat el sistema, les proves d'acceptació han estat validades per l'equip de desenvolupadors.

### 4.1 Proves d'unitat de la base de dades

A la següent taula podem veure les proves principals a les quals hem sotmès a la Base de Dades:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Accés complet a les taules del sistema a través de phpMyAdmin.	Poder connectar a la base de dades a través del navegador a la direcció <a href="http://localhost/phpmyadmin">http://localhost/phpmyadmin</a> a Windows 7.	Com l'esperat.	Victor
2	Connexió a la base de dades des de Java.	Poder connectar a la base de dades a través de codi Java.	Com l'esperat.	Victor

<b>3</b>	Execució de sentències SQL des de l'interfície de phpMyAdmin.	Executar <i>SELECT</i> , <i>INSERT</i> , <i>UPDATE</i> , <i>DELETE</i> .	Com l'esperat.	Victor
<b>4</b>	Execució de sentències SQL des de Java.	Executar <i>SELECT</i> i <i>UPDATE</i> a través de codi Java.	Com l'esperat.	Victor
<b>5</b>	Exportació de tot el contingut de la BD per a còpies de seguretat.	Obtenir un fitxer de sortida amb tota l'estructura de la BD per poder importar-la en cas de desastre.	Com l'esperat.	Victor

**Taula 23. Proves d'unitat de la BD**

#### 4.2 Proves d'unitat del mòdul d'extracció de punts d'interès

A la següent taula podem veure les proves principals a les quals hem sotmès al mòdul d'extracció de punts d'interès:

<b>#</b>	<b>Requeriment</b>	<b>Resultat esperat</b>	<b>Resultat obtingut</b>	<b>Tester</b>
<b>1</b>	Lectura d'un fitxer de text pla.	Obtenir el contingut d'un fitxer de text pla.	Com l'esperat.	Victor
<b>2</b>	Construcció de l'arbre de cerca a partir del contingut d'un arxiu de text pla.	Crear un objecte de tipus arbre correctament estructurat.	Com l'esperat.	Victor
<b>3</b>	Connexió a la base de dades.	Poder connectar a la base de dades a través de codi Java.	Com l'esperat.	Victor
<b>4</b>	Consulta a la base de dades.	Poder obtenir registres de la taula <i>requests</i> de la BD i fer servir la informació dels diferents camps.	Com l'esperat.	Victor
<b>5</b>	Actualització a la base de dades.	Poder canviar el valor <i>state</i> dels registres de la taula <i>requests</i> de la BD.	Com l'esperat.	Victor
<b>6</b>	Integració de la llibreria <i>Tika.jar</i> dins del	Poder utilitzar les diferents classes i mètodes de la	Com l'esperat.	Victor

	sistema.	llibreria Java <i>Tika.jar</i> per a l'extracció de continguts de de fitxers i direccions web.		
<b>7</b>	Extracció del contingut de qualsevol tipus d'arxiu.	Obtenir el contingut de qualsevol tipus de fitxer com poden ser .doc, .odt o .pdf entre altres, per tal de poder analitzar-lo.	Com l'esperat.	Victor
<b>8</b>	Extracció del contingut d'una direcció web.	Obtenir el contingut sense tags html de qualsevol direcció web, per tal de poder analitzar-lo.	Com l'esperat.	Victor
<b>9</b>	Parseig del contingut d'un fitxer o direcció web fent servir l'arbre de cerca.	Obtenir els punts d'interès que apareguin al fitxer o direcció web fent un únic recorregut del contingut.	Com l'esperat.	Victor
<b>10</b>	Inserció de nous registres a la base de dades.	Poder inserir nous registres a la taula <i>pois</i> de la BD.	Com l'esperat.	Victor
<b>11</b>	Generació d'un arxiu executable JAR del mòdul.	Obtenir un arxiu executable de tipus JAR que executa la classe <i>Main</i> del projecte corresponent al mòdul d'extracció de punts d'interès.	Com l'esperat.	Victor
<b>12</b>	Execució de l'arxiu JAR a través de la consola de Windows.	Visualitzar l'execució de la funcionalitat que permet l'extracció de punts d'interès a partir de les peticions pendents a la BD.	Com l'esperat.	Victor
<b>13</b>	Generació d'un arxiu de log.	Obtenir un arxiu que registri totes les accions que duu a terme el mòdul per tal de determinar possibles errors d'execució.	Com l'esperat.	Victor

Taula 24. Proves d'unitat del mòdul d'extracció de punts d'interès

### 4.3 Proves d'unitat del mòdul d'extracció de definicions de la Wikipedia

A la següent taula podem veure les proves d'unitat principals a les quals hem sotmès al mòdul d'extracció de punts d'interès:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Extracció del contingut de la Wikipedia.	Obtenir la descripció en HTML de l'article de la Wikipedia relatiu a un determinat punt d'interès.	Com l'esperat.	Victor
2	Extracció del primer paràgraf d'un article de la Wikipedia.	Obtenir el contingut que hi ha entre les primeres etiquetes html <code>&lt;p&gt;</code> d'un article de la Wikipedia.	Com l'esperat.	Victor
3	Sanejament del contingut del primer paràgraf de la Wikipedia.	Obtenir el contingut del primer paràgraf d'un article de la Wikipedia sens les etiquetes html.	Com l'esperat.	Victor
4	Reconeixement d'error	Diferenciar un resultat vàlid d'un altre que no ho és.	Com l'esperat.	Victor
5	Parseig d'alternatives	Obtenir el contingut d'un article alternatiu si es dóna el cas de que no se'n troba cap per al punt d'interès especificat.	Com l'esperat.	Victor

*Taula 25. Proves d'unitat del mòdul d'extracció de definicions de la Wikipedia*

### 4.4 Proves d'unitat de la interfície Android

En quant a la interfície Android, cal dir que més que a proves funcionals, l'hem sotmès a proves d'estil, centrant-nos en els aspectes que faran que l'usuari la vegi com una interfície amigable o no.

A la següent taula podem veure les proves principals a les quals hem sotmès a la interfície Android:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Correcta visualització de les dades per pantalla.	Les dades es visualitzen de manera que la informació es perfectament llegible i no s'aglomera a la pantalla.	Com l'esperat.	Victor
2	Entrades de text i botons accessibles.	Les entrades de text i els botons tenen la mida adequada per al seu accés / accionament.	Com l'esperat.	Victor

*Taula 26. Proves d'unitat de la interfície Android*

#### 4.5 Proves d'usuari de la interfície Android

En el cas de la interfície Android, a més de les proves d'unitat, que verifiquen la funcionalitat d'aquesta interfície, hem cregut important realitzar una sèrie de proves d'usuari.

Aquest tipus de prova permeten, a través de l'exposició d'un conjunt d'usuaris a l'aplicació, analitzar com aquests s'enfronten, perceben i interactuen amb la interfície, per tal de determinar si el disseny ha estat correcte o no.

Per fer-ho, hem posat a cinc usuaris d'entre quinze i cinquanta anys davant la interfície i els hem anat demanant que realitzessin una sèrie d'accions que a continuació presentem en forma de formulari:

Prova 1: Iniciar sessió.

·Resultats:

- Usuari 1: Resultat satisfactori
- Usuari 2: Resultat satisfactori
- Usuari 3: Resultat satisfactori
- Usuari 4: Resultat satisfactori
- Usuari 5: Resultat satisfactori

Èxit: 100%

Prova 2: Canviar el número de punts més propers a buscar.

·Resultats:

- Usuari 1: Resultat satisfactori

- Usuari 2: Resultat no satisfactori (No troba la opció)
- Usuari 3: Resultat no satisfactori
- Usuari 4: Resultat satisfactori
- Usuari 5: Resultat satisfactori

Èxit: 80%

Prova 3: Generar un mapa amb tots els punts d'interès.

·Resultats:

- Usuari 1: Resultat satisfactori
- Usuari 2: Resultat satisfactori
- Usuari 3: Resultat satisfactori
- Usuari 4: Resultat satisfactori
- Usuari 5: Resultat satisfactori

Èxit: 100%

Prova 4: Visualitzar la informació d'un dels punts d'interès.

·Resultats:

- Usuari 1: Resultat satisfactori
- Usuari 2: Resultat satisfactori
- Usuari 3: Resultat no satisfactori (No troba la opció)
- Usuari 4: Resultat satisfactori (No troba la opció)
- Usuari 5: Resultat satisfactori

Èxit: 60%

Prova 5: Visualitzar un dels punts en un mapa.

·Resultats:

- Usuari 1: Resultat satisfactori
- Usuari 2: Resultat satisfactori
- Usuari 3: Resultat satisfactori
- Usuari 4: Resultat no satisfactori (S'ha equivocat d'opció)
- Usuari 5: Resultat satisfactori

Èxit: 80%

Prova 6: Visualitzar la ruta fins a un dels punts.

·Resultats:

- Usuari 1: Resultat no satisfactori (S'ha equivocat d'opció)
- Usuari 2: Resultat no satisfactori (S'ha equivocat d'opció)
- Usuari 3: Resultat satisfactori
- Usuari 4: Resultat satisfactori
- Usuari 5: Resultat satisfactori

Èxit: 60%

Prova 7: Crear una ruta.

·Resultats:

- Usuari 1: Resultat satisfactori
- Usuari 2: Resultat satisfactori
- Usuari 3: Resultat satisfactori
- Usuari 4: Resultat satisfactori
- Usuari 5: Resultat satisfactori

Èxit: 100%

Èxit final: 82,85%

Com es pot observar en base als resultats, la navegació dels usuaris per l'aplicació és, en general, molt satisfactòria. Podem afirmar, per tant, que la interfície de l'aplicació Android aconsegueix els objectius de disseny centrats en l'usuari que hem utilitzat com a base de creació.

#### **4.6 Proves d'integració**

Les proves d'integració serveixen per avaluar la funcionalitat del sistema quan el complementem amb la resta del projecte que, com s'ha explicat a l'apartat d'introducció, desenvolupa un altre company.

A la següent taula podem veure les proves d'integració principals a les quals hem sotmès el sistema:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Lectura dels registres generats per l'Aplicació Web.	El mòdul extractor de punts d'interès és capaç d'accedir a les peticions que els usuaris han generat a l'Aplicació Web i utilitzar-ne les dades assignades a aquests registres.	Com l'esperat.	Victor
2	Modificació dels registres generats per l'Aplicació Web.	El mòdul extractor de punts d'interès és capaç de modificar l'estat de les peticions que els usuaris han generat a l'Aplicació Web.	Com l'esperat.	Victor
3	Inserció de registres accessibles a la BD.	El mòdul extractor de punts d'interès és capaç d'inserir a la BD registres als quals l'Aplicació Web i l'Aplicació Android poden accedir correctament.	Com l'esperat.	Victor
4	Proporció del servei d'extracció de definicions de la Wikipedia.	La funció d'extracció de definicions de la Wikipedia pot ser utilitzada correctament per l'Aplicació Android.	Com l'esperat.	Victor
5	Proporció de les vistes de la interfície Android.	L'Aplicació Android reconeix com a vàlids els arxius xml corresponents als diferents layouts o vistes de la interfície Android.	Com l'esperat.	Victor

**Taula 27. Proves d'integració**



## 5. Línies futures

A l'hora de resoldre cadascun dels problemes que ens ha plantejat cada funcionalitat a desenvolupar del sistema, hem fet servir sempre el mateix sistema de treball: primer, explorar les diferents alternatives possibles per a satisfer el requisit; a continuació, valorar els avantatges i inconvenients associats a cadascuna de les possibilitats; i, per últim escollir-ne justificadament la millor.

Dit això es podria pensar, per tant, que un cop desenvolupat el sistema, aquest acompliria les seves funcions de la forma més eficient dins d'un marc de possibilitats. Això no implica, però, que, a mesura que hem anat desenvolupant mòduls, hagin pogut aparèixer altres alternatives, fruit de la cerca, replantejament o compatibilitat que donen també solució a problemes que ja s'havien resolt.

És llavors, i sobretot un cop acabat i analitzat el sistema general, quan apareixen possibles millores o línies futures de treball a implementar sobre allò que s'ha desenvolupat. El fet de no haver-les dut a terme, es deu en molt gran part a la falta de temps o a la no adaptabilitat del sistema envers determinades eines, però en cap cas a no saber com implementar o canviar la funcionalitat.

És per això que hem volgut afegir un apartat de millores i ampliacions una mica més extens del que normalment es fa. Això és degut a que analitzem cada decisió mòdul per mòdul i detallam la millora que es podria dur a terme tal i com l'hauríem implementat.

D'aquesta manera, anem més enllà d'exposar les potencials millores o ampliacions del sistema per a especificar detalladament de quina manera caldria modificar-lo per dur-la a terme.

Tot plegat, un apartat que pot perfectament servir de guia per a futurs desenvolupadors que s'animessin a treballar sobre el sistema existent.

### 5.1 Millora del mòdul d'extracció de punts d'interès

Recordem molt breument el funcionament final del mòdul d'extracció de punts d'interès:

1. Es parteix d'un fitxer de text pla que conté una llista de punts d'interès, cadascun amb la seva corresponent geo-localització.
2. Es crea un arbre que emmagatzema la informació de l'arxiu de text pla.
3. A partir d'una consulta a la base de dades s'accedeix a un document o direcció web i se n'extreu el contingut.

4. Amb l'ajuda de l'arbre, s'identifiquen els punts d'interès que apareixen al document o direcció web.
5. S'enregistren a la base de dades cadascun dels punts d'interès extrets, amb la seva geo-localització.

### 5.1.1 Millora 1: Freeling i API de Google Maps

El principal problema d'aquest mòdul és el manteniment de la llista de punts d'interès. A l'hora d'implementar el mòdul, a aquest aspecte no se li dóna massa importància perquè l'estructura en arbre i el parseig del contingut del document o direcció web ens ajuda a posar en pràctica els coneixements adquirits a l'assignatura de Teoria d'Autòmats i Llenguatges Formals. D'aquesta manera, la solució implementada aconsegueix un objectiu tant important com és el de servir com a model de resolució d'un tipus de problema après durant la carrera i, per tant, és vàlida i molt acceptable.

A la pràctica, però, el fet de que l'administrador hagi d'estar actualitzant aquesta llista o, en el pitjor dels casos, hagi de crear-la de zero, és una tasca massa dura que cal automatitzar d'alguna manera. Així, aconseguirem un sistema més autònom i millor.

Per fer-ho, es proposa fer servir una combinació de dues eines que permetria de forma més o menys senzilla que el sistema canviés radicalment.

La primera d'aquestes eines s'anomena *Freeling*. Es tracta d'un programari gratuït d'anàlisi sintàctic multilingüe el qual, donat un conjunt de frases, és capaç de reconèixer-les i analitzar-les morfo-sintàcticament. Això és, per a cada paraula, dir quines són les seves possibles categories gramaticals (cadascuna amb el seu tant per cert de probabilitat) i la seva funció sintàctica dins de l'oració.

Així doncs, *Freeling* ens permet reconèixer les Entitats amb Nom (Named Entities), moltes de les quals són candidates a referir-se a punts d'interès.

La segona eina, és una funcionalitat molt útil que ofereix l'API de Google Maps que permet, donat el nom d'un punt d'interès i mitjançant la construcció d'una senzilla url, obtenir, si és que n'hi ha, una geo-localització associada al nom proporcionat.

Així si, per exemple, *Freeling* detecta l'entitat amb nom *Palau de la Música*, la següent direcció web:

<http://maps.google.com/maps/geo?q=Palau%20de%20la%20Música&output=csv&sensor=false>

, retorna un vector amb quatre valors, els dos darrers del qual es corresponen amb les coordenades de latitud i longitud associades, en aquest cas, al *Palau de la Música*. Cal afegir que, en cas que l'API de Google Maps no trobés cap geo-localització assignada al nom que li passem com a paràmetre, els dos últims valors del vector valen ambdós zero.

Així doncs, el funcionament millorat del mòdul d'extracció automàtica de punts d'interès és el següent:

1. A partir d'una consulta a la base de dades s'accedeix a un document o direcció web i se n'extreu el contingut.
2. Es passa el contingut a l'analitzador sintàctic *Freeling* i s'agafen les paraules la categoria sintàctica de les quals sigui similar a 'NP00000'.
3. Per a cada Entitat amb Nom es crida a l'API de Google Maps i es parseja la resposta.
4. Si la geo-localització és diferent de 0,0 s'enregistra el punt d'interès a la base de dades.

Cal denotar que el *Freeling* és una eina multilingüe i que, per tant, permet l'anàlisi morfo-sintàctic en una gran varietat d'idiomes com són català, castellà, gallec, anglès, italià, portuguès o rus. Això permet, per tant, que les peticions que l'usuari fa a la base de dades es puguin fer sobre documents en diferents idiomes, i que tots ells generin resultats.

Es tracta, doncs, d'una millora substancial que fa que el sistema passi d'estar restringit a la detecció i extracció d'un conjunt de cinquanta, cent o dos cents punts d'interès, a ser capaç de detectar quasi qualsevol punt d'interès sense que ni l'usuari administrador ni els usuaris del sistema hagin de fer cap tipus de manteniment. Cal remarcar que això és possible gràcies a que el detector d'Entitats amb Nom, *Freeling*, es realimenta a sí mateix mitjançant processos d'aprenentatge automàtic.

### **5.1.2 Justificació de la no implementació**

Cal aclarir, per últim, el motiu de no haver implementat aquesta millora. En aquest cas, no es tracta d'un problema de temps ja que les eines se'ns donen preparades per a utilitzar i el problema realment es redueix a un parell o tres d'*scripts* que creïn el flux de treball del mòdul. El principal problema que trobem és que es tracta d'una alternativa que descobrim que existeix un cop acabat el sistema i que no és fàcilment integrable al programari que hem desenvolupat. Això és perquè el *Freeling* està pensat per a ser utilitzat sobre plataformes Linux i nosaltres, des d'un primer moment, fixem com a Sistema Operatiu de treball Windows 7.

Es tracta, per tant, d'aplicar una millora que implica agafar el projecte des de zero i adaptar-lo en la seva totalitat a un Sistema Operatiu nou. És per això, que després d'intentar per diferents medis que l'eina funcionés correctament a Windows 7 i davant del fracàs d'aquesta intenció, quan es decideix no dur a terme la millora.

De totes formes s'explora, tal com hem pogut veure, aquesta alternativa de funcionament tant interessant.

## **5.2 Millora de la interfície de l'aplicació Android**

Des d'un primer moment es fixa com a objectiu principal de la interfície Android fer un disseny centrat totalment en l'usuari. El resultat final d'aquesta part del sistema, resulta molt satisfactori, ja que aconseguim desenvolupar un conjunt de vistes equilibrades, intuïtives, que presenten la informació de forma clara i específica i que, en general, satisfà tant l'usuari com els participants del projecte.

### **5.2.1 Millora 1: Temes d'usuari**

En aquest cas, llavors, les possibles millores se centren més en l'aspecte visual de la interfície (colors i tipus de lletra, entre altres) que en el disseny d'estructura i continguts de la mateixa.

I és clar que és un aspecte clau a satisfer perquè l'usuari final, al cap i a la fi, es deixa emportar per les sensacions que l'aplicació li transmet visualment. El problema és que, com diu l'expressió '*per a gustos, els colors*'. Així doncs, un cop hem aconseguit que l'usuari se senti a gust manegant l'aplicació, li podem oferir la possibilitat de canviar l'aspecte visual mitjançant temes que modifiquin la combinació de colors dels diferents elements de les vistes o la possibilitat de canviar la font que s'utilitza.

Per fer-ho, es podrien utilitzar un conjunt de variables que emmagatzemin els codis de color assignats al tema que l'usuari hagi escollit. Així, a l'hora de crear els *layouts* s'accediria a aquestes variables i es crearia la interfície en conseqüència. Es disposaria, a més, d'una variable addicional per a guardar el tipus de lletra que es vol utilitzar.

### **5.2.2 Millora 2: Adaptació a dispositius tablet**

A banda d'aquestes millores en el camp estilístic de la interfície, es pot plantejar la implementació d'una de les vies d'evolució natural d'una aplicació: l'adaptació de la interfície a dispositius de tipus tablet.

En el cas del sistema desenvolupat, pel seu caràcter clarament lligat a la presentació d'una part de la informació mitjançant mapes, l'adaptació de la interfície a un dispositiu tablet permetria una millora considerable en quant a la visualització d'aquesta informació.

Implementar aquesta millora, per una altra part, es redueix a un problema de mides. Per una banda, s'hauran d'adaptar les imatges que es fan servir com a fons de l'aplicació. Per una altra, s'haurà de canviar la mida dels *layouts* i *textviews* que conformen les vistes. Per a resoldre aquest problema, es pot recórrer a dues solucions:

- a. Dissenyar una aplicació alternativa, amb els canvis de mida pertinents. D'aquesta manera, podríem trobar al *Market* d'Android la mateixa aplicació en dues versions, segons el tipus de dispositiu que l'usuari utilitzi.
- b. Redissenyar l'aplicació fent servir mides relatives (en tant per cent). Aquesta solució permetria una única aplicació al *Market d'Android* que s'adaptaria al dispositiu de l'usuari.

Òbviament la solució més eficient és la opció b, que permet una integració total de l'aplicació, a més, en dispositius mòbils i tablet sense importar la mida o resolució de la pantalla d'aquests, és a dir, totalment adaptativa.

### **5.2.3 Justificació de la no implementació**

En aquest cas, les dues millores proposades no s'han pogut dut a terme per falta de temps, sobretot en el cas de la creació de temes. Tot i així, que podem veure que es tracta de canvis que, un cop llençada a la venda l'aplicació, serien necessaris i quasi immediats degut a l'impacte estètic que poden tenir front els clients potencials.

### **5.3 Millora del mòdul d'extracció de definicions de la Wikipedia**

Recordem molt breument quin és el funcionament final del mòdul d'extracció de definicions de la Wikipedia:

1. A partir d'una paraula clau (que fa referència a un punt d'interès) es crea una direcció web que crida a l'API de la Wikipedia.
2. Aquesta direcció web pot retornar dos valors:
  - a. El contingut de l'article de la Wikipedia referent al punt d'interès.
  - b. Un missatge amb la paraula REDIRECT i, a continuació, una suggerència correctiva.
3. En el primer cas, s'agafa el contingut del primer paràgraf.
4. En el segon cas es torna al punt 1 fent servir la nova paraula clau suggerida.

Tot i que es tracta d'un disseny prou bo en el que cal remarcar la capacitat d'autocorrecció del mòdul davant del problema de no trobar cap article relatiu a la cerca, l'API de la Wikipedia és una eina prou potent com per permetre'ns veure alguns usos extrems per als quals podem utilitzar-la.

Totes les millores, però, es basen en oferir informació relativa al punt d'interès tenint en compte que partim de tant sols un mot o un conjunt de mots.

### **5.3.1 Millora 1: Informació en diferents idiomes**

Per a dur a terme la primera millora, ens aprofitarem de la internacionalitat de la Wikipèdia. Aquesta, com a eina utilitzada mundialment, disposa de gran quantitat d'articles en diferents idiomes. Això implica, que moltes de les entrades, mostren exactament la mateixa informació però en diferents idiomes. Un cop identificat l'article relatiu a un cert punt d'interès, podem, directament, verificar si existeix el seu corresponent en l'idioma que prèviament l'usuari hagi seleccionat.

Fer aquest canvi és tan fàcil com canviar les dues primeres lletres de la direcció web que es construeix (en comptes de *es*, utilitzar *fr*, *it* o *de*, entre altres). El principal problema, però, és que moltes vegades no existeix un mateix article en diferents idiomes, a no ser que es tracti de punts d'interès molt famosos o amb una certa rellevància. En altres casos, a més, si existeix un equivalent, però cal fer la traducció del nom del punt d'interès i això implicaria, per exemple, fer servir complementàriament l'API del traductor de Google.

Una altra possible solució, ja que hem esmentat l'eina de traducció de Google, és la d'extreure el contingut seguint la metodologia exposada inicialment, i fer servir aquest servei per traduir el resultat extret a l'idioma que l'usuari hagi escollit.

### **5.3.2 Millora 2: Imatges d'un punt d'interès**

La segona millora, aprofita que hi ha un punt de l'extracció en el que disposem de l'article sencer de la Wikipedia per donar, a més, un conjunt d'imatges relacionades amb el punt d'interès. Més concretament, les imatges que l'article pugui proporcionar, que, en general, es tracta d'una imatge del monument o edifici i, en alguns casos de mapes o imatges històriques associades a aquest. Aquesta és una opció molt interessant que permet obtenir tota una col·lecció d'imatges relacionades amb un punt d'interès amb una seguretat quasi del cent per cent de que fan referència a allò que s'està buscant.

Per a dur a terme aquesta millora, en la fase de selecció del primer paràgraf de l'article, es faria a més una cerca de totes les etiquetes html relatives a imatges. Cadascuna d'aquestes referència a l'usuari, a través d'un accés directe, a la imatge desitjada.

Una altra possibilitat d'implementació d'aquesta millora seria la de realitzar una cerca a Google imatges fent servir com a clau el nom del punt d'interès. Aquest mètode, però, té l'inconvenient de no tenir una certesa massa gran de si l'usuari trobarà imatges relatives al punt d'interès o no.

### **5.3.3 Justificació de la no implementació**

En quant a la implementació d'aquestes millores, la primera es va intentar posar en pràctica, però no es van aconseguir el resultats desitjats degut a que calia ajustar alguns paràmetres i la manca de temps no va poder fer-ho possible. En quant a la segona, es tracta exclusivament de falta de temps i potser del fet que no s'observés com a funcionalitat en un principi, que no s'hagi dut a terme.

Tot i així, totes dues funcionalitats donarien un punt molt positiu i personal al projecte i seria una molt bona línia futura per a continuar amb el desenvolupament d'aquest.

## 6. Conclusions

En aquest capítol finalitzarem amb l'avaluació dels objectius un cop finalitzat el projecte, repassarem les desviacions de la planificació i exposarem una valoració personal de l'experiència.

### 6.1 Objectius aconseguits i no aconseguits

El nostre objectiu principal és el d'oferir a l'usuari la possibilitat d'extreure de forma automàtica un conjunt de punts d'interès i informació relativa a aquests, a partir d'un conjunt de documents i direccions web. Mitjançant l'anàlisi del contingut d'aquests documents o direccions web i comparant-los amb tota una col·lecció pre-calculada de punts d'interès, el sistema és capaç de detectar els punts d'interès sobre els que l'usuari ha llegit o ha consultat informació. Un cop detectats, a més, se'ls assigna una geo-localització i es permet a l'usuari treballar amb ells a fi de, entre altres funcions, obtenir una breu ressenya informativa sobre el punt d'interès o veure'l en un mapa.

Un cop dissenyat i implementat el sistema final que ofereix la funcionalitat esperada i aconsegueix els objectius inicials, podem afirmar que és una eina viable a desenvolupar.

Si anem al detall i repassem els objectius propis marcats a la introducció (apartat 1.3), podem analitzar vuit objectius:

1. Proporcionar un servei capaç de detectar i extreure localitzacions a partir de text pla. **(Complert)**
2. Proporcionar un conjunt d'eines que permetin calcular o assignar geo-localitzacions a un conjunt de punts d'interès. **(No complert)**
3. Proporcionar un mòdul de cerca de definicions de punts d'interès a la *Wikipedia*. **(Complert)**
4. Dissenyar la interfície d'usuari de l'aplicació *Android*. **(Complert)**
5. Aplicar els coneixements adquirits a l'assignatura Teoria d'Autòmats i Llenguatges Formals. **(Complert)**
6. Aprofundir en el coneixement del llenguatge Java. **(Complert)**



7. Aprofundir en el coneixement de l'entorn de desenvolupament Eclipse i la integració d'aquests amb altres mòduls. **(Complert)**
  
8. Aprofundir en el coneixement del desenvolupament d'aplicacions mòbils. **(Complert)**

Tal com podem veure, l'objectiu número dos no s'ha pogut dur a terme. Encara que es tracta d'un objectiu prioritari (tal com s'indica a l'apartat 2.3) cal aclarir per què no es satisfà. El motiu és que la implementació escollida del mòdul d'extracció de punts d'interès no requereix d'una eina autònoma de càlcul de coordenades ja que en el mateix fitxer de text pla on s'emmagatzemen els punts d'interès, es guarda també la geo-localització assignada a cadascun d'ells.

Tot i així, com hem vist en una de les millores d'aquest mòdul, l'API de Google Maps ofereix un servei que donada una clau, permet obtenir un parell de coordenades associades.

Podem dir, per tant, que, en aquest cas, el que al principi semblava un objectiu per sí sol, ha acabat solucionant-lo la implementació d'un dels altres objectius i, per tant, no s'ha desenvolupat una eina específica per a complir-lo.

Així doncs, es tracta d'un projecte factible que aconsegueix tots els objectius proposats.

## 6.2 Seguiment dels errors de desenvolupament

Quan es duu a terme la planificació d'un projecte es tendeix sempre a ser optimista. Això es deu en part a que, a priori, hom té una idea general d'allò que ha de fer però sense pensar massa en la forma detallada de com dur a terme cada tasca. Es normal, llavors, que quan comencen a desenvolupar el projecte, ens trobem amb que les eines que utilitzem per a treballar ens donen problemes.

En el nostre cas, no podria ser d'una altra manera i, a continuació, podem veure una relació dels problemes de desenvolupament que ens hem anat trobant al llarg del projecte i que hem hagut de solucionar fent un esforç extra.

Problema	Solució	Efecte
No existeix una col·lecció de punts d'interès amb la seva geo-localització.	Creació de zero d'una llista de punts d'interès, cadascun amb la seva corresponent geo-localització	Endarreriment de la primera fase del projecte.
La llibreria d'extracció de	Consulta de la documentació	Pràcticament sense

continguts Tika no s'integra correctament a l'Eclipse.	de Tika.	efecte.
No sabem com generar un arxiu jar i executar-lo a través de consola.	Consulta de diferents webs i articles relatius al tema.	Pràcticament sense efecte.
Obrir els ports de l'ordinador en el que treballàvem per poder donar accés al localhost al company de projecte	Consulta de la documentació de Windows sobre com permetre l'accés local a ordinador dins de la mateixa xarxa.	Breu endarreriment del projecte.
Problema físic a la font d'alimentació del portàtil principal.	Migració de tot el projecte a un altre portàtil, amb la corresponent posada a punt dels paràmetres del sistema.	Endarreriment del projecte.
Extracció de les dades de la Wikipedia en un format correcte. Érem capaços d'extreure el contingut de tot un article de la Wikipedia però en un format que no ens permetia treballar correctament amb ell.	Després d'analitzar els diferents formats disponibles vam arribar a la conclusió de que el que a nosaltres ens interessava és JSON.	Breu endarreriment del projecte.
Jerarquia de continguts a Android dins de cada vista	Consulta de la documentació de desenvolupadors Android	Breu endarreriment del projecte.

**Taula 28. Problemes de desenvolupament**

Tot i els endarreriment que han provocat alguns d'aquests problemes, cal assenyalar que vam fer una planificació temporal bastant pessimista, de manera que, encara que algunes tasques han trigat més del normal, altres que vam planificar amb més temps hem aconseguit resoldre-les abans de l'esperat.

Així doncs, i tenint en compte que vam establir unes 200 hores de treball en total, podem dir que la planificació ha estat bastant acurada, doncs tenim el projecte enllestit fent servir més o menys les hores especificades.

### **6.3 Valoració personal**

A nivell personal, cal remarcar que des de l'inici em vaig plantejar el desenvolupament d'aquest projecte no com a exercici acadèmic de final de carrera sinó com a una prova conclusiva dels meus coneixements, capacitats i habilitats adquirides durant els meus anys d'estudi sobre un escenari molt pròxim al món laboral i el marc tecnològic actual.

A nivell tècnic considero que he adquirit moltes habilitats. En primer lloc, en quant al llenguatge de programació Java, dir que ja havíem treballat amb ell en algunes assignatures de la carrera però ha estat realment durant el desenvolupament d'aquest projecte quan he après a utilitzar-lo. Això m'ha donat molta fluïdesa, punt que considero important en tant que es tracta d'una de les eines de programació més presents actualment.

En segon lloc, en quant a les API's que tant la Wikipedia com Google Maps proporcionen, dir que són eines de les que havia sentit parlar però amb les que no havia treballat. Aquest projecte m'ha donat la possibilitat de descobrir les potents funcionalitats que ofereixen aquests serveis web concrets i de descobrir-ne d'altres.

Per últim, arribem a la part del projecte que des d'un principi es va atreure més i em va semblar més interessant: la interfície Android. Durant tot l'estiu previ al desenvolupament del projecte i després d'haver adquirit un terminal amb sistema operatiu Android vaig estar documentant-me sobre com crear aplicacions per a aquesta innovadora plataforma. Així doncs, quan el meu tutor ens va proposar un projecte en el qual hi havia una part de desenvolupament Android no vaig haver-me de pensar ni un instant si prenia o no l'oportunitat de desenvolupar quelcom real, factible i funcional. Aquesta és, a nivell tècnic, l'àrea en la que més he après i segurament la que m'impulsarà a no quedar-me en el que sé per anar a aprendre més sobre plataformes mòbils en general.

A nivell personal, a part del coneixements tècnics que com he exposat anteriorment he adquirit, m'agradaria remarcar l'aprenentatge del treball en equip que he experimentat. En tractar-se d'un projecte conjunt, he après que la comunicació en totes les fases d'un projecte és vital per a l'acompliment dels objectius i per a la final satisfacció del client que necessita el software.

D'altra banda, també he millorat en les habilitats transversals ja que he après a enfrontar-me a una planificació, formalitzar tots els tràmits previs al projecte necessaris per a dur-lo a terme i ser-ne responsable justificant cada pas. També he après a autogestionar-me el temps i les activitats, sent jo desenvolupador, analista i assumint les tasques de cap de projecte en alguns casos.

Com a conclusió final dir que tant jo, com el meu company de projecte com el cap de projecte em treballat fort i em obtingut uns resultats molt positius.

## 7. Glossari

### 7.1 Relació de taules

<i>Taula 1. Classificació dels objectius</i>	10
<i>Taula 2. Context</i>	11
<i>Taula 3. Maquinari</i>	13
<i>Taula 4. Requisits funcionals i no funcionals del projecte</i>	14
<i>Taula 5. Anàlisi de l'alternativa 1</i>	15
<i>Taula 6. Anàlisi de d'alternativa 2</i>	15
<i>Taula 7. Comparació de les alternatives</i>	15
<i>Taula 8. Component 1 del projecte</i>	16
<i>Taula 9. Component 2 del projecte</i>	16
<i>Taula 10. Recursos humans del projecte</i>	17
<i>Taula 11. Recursos materials</i>	18
<i>Taula 12. Programari disponible</i>	18
<i>Taula 13. Tasques del projecte</i>	20
<i>Taula 14. Riscos del projecte</i>	22
<i>Taula 15. Anàlisi de riscos</i>	22
<i>Taula 16. Pla de contingència</i>	23
<i>Taula 17. Cost de personal</i>	23
<i>Taula 18. Cost de recursos</i>	23
<i>Taula 19. Avantatges i inconvenients del projecte</i>	24
<i>Taula 20. Estructura de la taula 'users' de la BD</i>	30
<i>Taula 21. Estructura de la taula 'requests' de la BD</i>	30
<i>Taula 22. Estructura de la taula 'pois' de la BD</i>	31
<i>Taula 23. Proves d'unitat de la BD</i>	58
<i>Taula 24. Proves d'unitat del mòdul d'extracció de punts d'interès</i>	59
<i>Taula 25. Proves d'unitat del mòdul d'extracció de definicions de la Wikipedia</i>	60

<i>Taula 26. Proves d'unitat de la interfície Android</i>	61
<i>Taula 27. Proves d'integració</i>	64
<i>Taula 28. Problemes de desenvolupament</i>	74

## **7.2 Relació d'imatges**

<i>Imatge 1. Lògica del sistema complet</i>	11
<i>Imatge 2. Lògica del sistema parcial</i>	12
<i>Imatge 3. Descripció física del sistema</i>	12
<i>Imatge 4. Planificació del projecte</i>	20
<i>Imatge 5. Flux de dades del sistema</i>	28
<i>Imatge 6. Model relacional de la BD</i>	32
<i>Imatge 7. Diagrama d'estats del mòdul d'extracció de punts d'interès</i>	33
<i>Imatge 8. Vista main.xml</i>	43
<i>Imatge 9. Vista usertopics.xml</i>	45
<i>Imatge 10. Vista userpois.xml</i>	47
<i>Imatge 11. Vista userroute.xml</i>	49
<i>Imatge 12. Diagrama de casos d'ús del sistema</i>	52
<i>Imatge 13. Diagrama de seqüència del cas d'ús 'Atendre petició'</i>	53
<i>Imatge 14. Diagrama de seqüència del cas d'ús 'Accedir a l'aplicació'</i>	54
<i>Imatge 15. Diagrama d'activitats del cas d'ús 'Atendre petició'</i>	55
<i>Imatge 16. Diagrama d'activitats del cas d'ús 'Accedir a l'aplicació'</i>	56

## 8. Bibliografia

### 8.1 Apunts

- Apunts i materials de l'assignatura *Enginyeria del Software I* cursada al primer semestre del curs 2010-2011.
- Apunts i material de l'assignatura *Enginyeria del Software II* cursada al segon semestre del curs 2010-2011.
- Apunts i materials de l'assignatura *Metodologia i Gestió de Projectes* cursada al primer semestre del curs 2011-2012.
- Apunts i materials de l'assignatura *Teoria d'Autòmats i Llenguatges Formals* cursada al primer i segon semestre del curs 2010-2011.

### 8.2 Llibres

- *Cómo programar Java*, Harvey M. Deitel, Prentice Hall Mexico, 2008
- *Android: Guia para desarrolladores*, Frank Ableson, Robi Sens, Chris King, Anaya Multimedia, 2011
- *UML y patrones*, Graig Larman, Pearson Educación, 2002

### 8.3 Direccions web

- Eclipse – The Eclipse Foundation opensource community website, <http://www.eclipse.org/>, 2012
- MySQL :: The world's most popular opensource database, <http://www.mysql.com/>, 2012
- Apache friends – XAMPP, <http://www.apachefriends.org/es/xampp.html>, 2012
- phpMyAdmin, [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php), 2012
- Wikipedia – La enciclopedia libre, <http://es.wikipedia.org/>, 2012
- Google Maps API – Google Developers, <https://developers.google.com/maps/?hl=es>, 2012
- MediaWiki, <http://www.mediawiki.org/wiki/API>, 2012
- Apache Tika, <http://tika.apache.org/>, 2012
- Logger (Java 2 Platform SE v1.4.2), <http://docs.oracle.com/javase/1.4.2/docs/api/java/util/logging/Logger.html>, 2012

·Freeling home page, <http://nlp.lsi.upc.edu/freeling/>, 2012

·Android Developers, <http://developer.android.com/index.html>, 2012

·Normativa de Projectes d'Enginyeries Tècniques en Informàtica,  
[http://www.uab.cat/Document/541/595/Normativa\\_PFCNovembre2010.pdf](http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf), 2012