

VisitBCN

Aplicació iOS de turisme per Barcelona

Lliurament Final: Fita inicial

ÍNDEX

1 INTRODUCCIÓ	4
1.1 FORMULACIÓ DEL PROBLEMA.....	4
1.2 OBJECTIUS DEL PROJECTE.....	5
1.3 CONTEXTUALITZACIÓ.....	6
1.4 ACTORS IMPLICATS	6
1.4.1 DESENVOLUPADOR, DISSENYADOR I BETA TESTER	6
1.4.2 TUTOR DEL PROJECTE	6
1.4.3 DESENVOLUPADORS DE MAPES	6
1.4.4 USUARIS	6
2 ESTAT DE L'ART	7
2.1 EL PROBLEMA DEL VIATGER CANADENC	7
2.2 ALGORITMOS DE CERCA	7
2.2.1 DIJKSTRA.....	8
2.2.2 A*	10
2.2.3 OPTIMITZACIONS DE DIJKSTRA I A*	11
2.3 DISSENY D'APLICACIONS MÒBILS.....	12
2.4 CONCLUSIONS.....	12
3 DEFINICIÓ DE L'ABAST	13
3.1 ABAST	13
3.2 POSSIBLES OBSTACLES.....	14
3.2.1 ERRORS EN EL CODI	14
3.2.2 CALENDARI	14
3.2.3 CANVI EN EL FORMAT DE LES DADES.....	14
3.3 METODOLOGIA I RIGOR.....	14
3.3.1 CICLES DE DESENVOLUPAMENT CURTS.....	15
3.3.2 FEEDBACK DEL CLIENT	15
3.3.3 DESENVOLUPAMENT GUIAT PER PROVES.....	15
3.4 EINES PER EL DESENVOLUPAMENT	16
3.5 SEGUIMENT DEL PROJECTE	16

4 PLANIFICACIÓ TEMPORAL.....	17
4.1 PROGRAMA	17
4.2 PLA DE PROJECTE	17
4.2.1 PLANIFICACIÓ DEL PROJECTE (FITA INICIAL)	17
4.2.2 ANÀLISI I DISSENY DEL PROJECTE	17
4.2.3 DESCRIPCIÓ DE LES TASQUES	18
4.2.4 FITA FINAL	18
4.3 DURACIÓ APROXIMADA	19
4.4 VALORACIÓ D'ALTERNATIVES.....	19
4.5 DIAGRAMA DE GANTT	20
4.6 RECURSOS.....	21
4.6.1 HARDWARE.....	21
4.6.2 SOFTWARE.....	21
5 GESTIÓ ECONÒMICA I SOSTENIBILITAT	22
5.1 GESTIÓ ECONÒMICA.....	22
5.1.1 CONSIDERACIONS I COMENTARIS.....	22
5.1.2 PRESSUPOST DE RECURSOS HUMANS.....	22
5.1.3 PRESSUPOST DE HARDWARE	23
5.1.4 PRESSUPOST DE SOFTWARE	23
5.1.5 PRESSUPOST DE LLICÈNCIES	23
5.1.6 DESPESES INDIRECTES.....	24
5.1.7 PRESSUPOST TOTAL.....	24
5.1.8 CONTROL DE DESVIACIONS.....	24
5.2 ÀREA SOCIAL	25
5.3 ÀREA AMBIENTAL.....	25
5.4 PUNTUACIÓ SOSTENIBILITAT	25
6 BIBLIOGRAFIA	26

1 Introducció

1.1 Formulació del problema

El problema del viatger canadenc¹ és un dels problemes més comuns en l'àmbit de la informàtica i ha estat un cas d'estudi freqüent durant molts anys. Tanmateix, aquest problema té moltes variants que, alhora, són diverses, ja que, només modificant un paràmetre de l'heurística que volem per tal de traçar el millor camí, el problema pot canviar. Totes aquestes variants del problema les podem trobar sovint al nostre dia a dia.

En aquest cas ens centrem en una situació bastant típica entre els viatgers i és la de trobar el camí més ràpid per anar des d'on ens trobem fins algun punt d'interès de la ciutat, amb el transport públic.

Durant molts anys s'han implementat moltes eines (“Google Maps” o “Maps” de Apple) que proporcionen aquesta informació mitjançant la utilització de grafs i variacions millorades d'algoritmes com el A*² o el Dijkstra. Però cal tindre en compte una cosa i és que a més dels problemes que poden sorgir al no torbar-nos a la nostra ciutat, podem afegir-hi la manca de connexió a internet que ens dificulta obtenir l'itinerari.

Aquest projecte vol trobar la millor variació i implementació d'aquests algoritmes per tal de connectar l'usuari amb certs punts d'interès de Barcelona mitjançant el transport públic, el bus o el metro. A més a més, es vol trobar una solució al problema sorgit de la connexió a internet.

¹ “Canadian Traveller Problem” o “CTP”.

² A estrella.

1.2 Objectius del projecte

Ara que s'ha explicat el problema que pretén solucionar aquest projecte, podem passar a analitzar els diferents objectius a assolir durant la realització d'aquest.

Per tal de oferir als nostres futurs usuaris el millor servei, caldrà fer un estudi de quines són les necessitats dels viatgers quan es troben a una ciutat desconeguda, per tant no només caldrà fer un bon producte informàtic sinó que també caldrà posar èmfasi en altres objectius primaris.

Podríem organitzar els objectius de la següent manera:

1. Estudiar les necessitats dels nostres clients potencials mitjançant una enquesta realitzada a població de Barcelona i a on se'ls hi pregunta per les seves necessitats al viatjar.
2. Estudiar i implementar la millor solució informàtica per el problema de software i computacional que se'ns presenta (CTP).
3. Realitzar una interfície gràfica senzilla i intuïtiva amb la que la corba d'aprenentatge sigui curta.
4. Tractar les dades si cal perquè siguin més precises i ofereixin millors resultats i així donar més confiança a l'usuari per tal de que utilitzi més l'aplicació.
5. Crear una bona campanya de màrqueting a les xarxes socials per donar a conèixer l'aplicació.

Tots aquests objectius s'engloben en un objectiu principal que és el de solucionar el problema plantejat, és a dir donar una alternativa a aquells viatgers que es trobin a Barcelona i que estiguin intentant organitzar la seva jornada de forma fàcil, eficient i fent ús del transport públic de la ciutat.

1.3 Contextualització

En aquest projecte tractem una variació del problema del viatger canadenc. Hi han diferents solucions per aquest problema creades per empreses com Google o Apple. Aquestes aplicacions s'ajuden de la teoria de grafs i els algoritmes de cerca per tal de trobar el millor itinerari. Més endavant profunditzarem en l'explicació d'aquestes tècniques.

1.4 Actors implicats

Ara detallarem els actors implicats en el projecte, és a dir, aquelles persones o organitzacions que poden estar interessar-se per el projecte.

1.4.1 Desenvolupador, dissenyador i beta tester

Les tasques de desenvolupador, dissenyador i beta tester seran realitzada per mi mateix, ja que soc la única persona que porta a terme aquest projecte.

1.4.2 Tutor del projecte

El tutor d'aquest projecte és el Guillem Godoy Balil, el seu paper és el de supervisar que el projecte compleixi el calendari estipulat i que s'assoleixin els objectius marcats. A més a més, pot ajudar i guiar al desenvolupador per tal de realitzar el projecte.

1.4.3 Desenvolupadors de mapes

Les persones o empreses que desenvolupen aplicacions de mapes poden estar interessades en les conclusions d'aquest projecte, ja que s'estudiaran i compararan els comportaments dels dos algoritmes més utilitzats per calcular itineraris.

1.4.4 Usuaris

Considerarem com a usuaris totes aquelles persones que es beneficiïn del producte final d'aquest projecte. De fet, al penjar aquest codi a Google Code com a software lliure i amb la seva comercialització gratuïta a la App Store, qualsevol pot ser un usuari final d'aquest projecte.

2 Estat de l'art

2.1 El problema del viatger canadenc

El problema del viatger canadenc va ser definit per descriure una situació molt comú entre els viatgers al Canadà. Un cop els viatgers es troben en una intersecció, mira quina de les dues carreteres es troba en millors condicions i decideix per on seguir. En un graf això es tradueix en un pes específic per cada aresta, per tant cada cop que ens trobem un vèrtex del graf podem veure el cost total acumulat fins aquell moment. L'objectiu es trobar el millor camí, normalment el de cost mínim, per arribar d'un vèrtex inicial A fins al vèrtex destí B. [1][4][5][9]

El problema del viatger canadenc és un bon exemple per estudiar alguns conceptes d'optimització. Es un problema d'optimització estocàstica, en el que alguns detalls del graf d'entrada (el pes de les arestes) són desconeguts a l'hora de resoldre el problema. Per tant, ens trobem davant d'un problema que s'adapta, és a dir, les decisions es van prenent sobre la marxa basant-se en la informació que es després de cada pas. [1][4][5][9]

En el nostre cas el pes de cada aresta serà el cost temporal que suposarà anar d'un vèrtex a un altre, és a dir, la seva representació a la vida real seria el cost d'anar de la parada 'A' fins la parada 'B' (tant metro com bus i per parades consecutives).

2.2 Algoritmes de cerca

Els algoritmes de cerca tracen el camí més curt entre dos punts. És una variació pràctica de la resolució del problema del viatger canadenc. Aquesta àrea de recerca es basa sobretot en l'algoritme de Dijkstra per trobar el camí mínim en un graf amb pesos.

Bàsicament, un algoritme de cerca "navega" per un graf començant per un vèrtex i explora els seus vèrtexs adjacents fins que troba el vèrtex final, generalment amb el propòsit de trobar el camí més curt.

Alguns algoritmes de cerca, com per exemple el BFS³, trobaran un camí si tenen el temps suficient, altres algoritmes, que “exploren” el graf, tendeixen a trobar la solució en menys temps. Això es degut a que altres algoritmes com l’A* es guien d’una heurística per valorar quina és la millor opció i no generar totes les opcions possibles. [3][4][5]

En el nostre cas només estudiarem el comportament dels algoritmes de Dijkstra [2] i A* [5]. Aquest algoritmes s’ajuden de la programació dinàmica o la heurística per tal de “podar” el camí i estalviar-se visitar tot el graf quan realment no cal. Aquest algoritmes triguen en el cas pitjor $O(|A|^4 \cdot \log(|V|^5))$. [10]

2.2.1 Dijkstra

L’algoritme de Dijkstra, publicat l’any 1959 per el científic Edsger Dijkstra, és un algoritme de cerca que resol el problema del camí més curt per un graf amb tots els pesos positius[2]. Aquest algoritme s’utilitza normalment per problemes d’encaminament.

Per un vèrtex inicial donat, l’algoritme troba el camí de menor cost (per exemple el camí més curt) entre aquest vèrtex i qualsevol altre vèrtex del graf. En el nostre cas ens interessarà més la variació de l’algoritme en la que es vol el millor camí d’un vèrtex ‘A’ fins a un vèrtex ‘B’ i que deixa d’explorar el graf un cop ha trobat la millor solució per aquest camí. És a dir, en el nostre cas cada vèrtex serà una parada de la xarxa de metro o de bus i les arestes seran les connexions directes entre dues parades consecutives d’una línia, el pes d’aquestes arestes serà el temps necessari per anar d’un vèrtex a l’altre.

L’algoritme de Dijkstra en el cas pitjor triga un temps de $O(|V|^2)$. La implementació feta amb una cua de prioritat ordenada de major a menor té un temps en el cas pitjor de $O(|A| + |V| \cdot \log(|V|))$.

³ Breadth-first Search

⁴ Nombre d’arestes

⁵ Nombre de vèrtexs del graf

Aquest és l'algoritme de cerca de camins més ràpid en grafs dirigits arbitraris que tenen pesos positius.

Ara detallarem el procés de selecció i expansió per cada pas de l'execució:

1. Al vèrtex inicial li donem valor 0 i a la resta de vèrtexs del graf valor infinit.
2. Marquem tots els vèrtexs com a no visitats i al vèrtex inicial com l'actual.
Agafem la resta de vèrtexs del graf i els posem en un conjunt de vèrtexs no visitats que es propagarà per cada camí creat, és a dir, no serà un sol conjunt per tots els camins.
3. Per el vèrtex actual, considerem tots els seus veïns que estiguin al conjunt de vèrtex no visitats i calculem el pes fins a cadascun d'ells. Comparem els valors calculats per cada vèrtex veí i agafem el de menor valor. Si el valor del vèrtex de destí es superior al valor calculat actualment en aquest camí, substituïm el valor actual per el nou. Si no és el cas, deixem el valor actual i no ens expandim per aquest vèrtex.
4. Quan acabem d'explorar tots els veïns, marquem el vèrtex actual com a visitat i l'eliminem del conjunt de vèrtexs no visitats. Un vèrtex visitat no tornarà a ser visitat mai més per aquesta cerca (en una execució hi han tantes cerques com veïns té el vèrtex inicial).
5. Si el vèrtex de destí està marcat com a visitat o si el camí possible més curt, entre els vèrtexs no visitat, és infinit (es dona quan no hi ha cap connexió entre els dos vèrtexs) ens aturem. L'algoritme ha acabat.

Agafem el vèrtex no visitat amb valor actual mínim i el marquem com a vèrtex actual. Un cop arribats a aquest punt, tornem al tercer pas.

2.2.2 A*

En l'enginyeria informàtica, l'A* és un algoritme molt utilitzat per trobar el camí mínim en un graf. És molt utilitzat gràcies la seva velocitat i precisió. Va ser publicat l'any 1968 per tres membres del SRI⁶. [11]

En el millors dels casos el cost de l'algoritme A* serà igual al cost de l'algoritme de Dijkstra, per tant es preveu que el resultat de l'estudi sigui el d'acabar utilitzant l'algoritme de Dijkstra.

L'A* utilitza el BFS i troba el camí de cost mínim des d'un vèrtex inicial fins al vèrtex final desitjat. Mentre l'algoritme de A* s'expandeix pel graf, segueix el camí que s'espera resulti amb el menor cost o distància total, guardant una cua de prioritat ordenada de parts alternatives dels camins.

Utilitza una funció heurística del vèrtex 'x'⁷ per determinar l'ordre en el que la cerca visita els vèrtexs. La funció de cost és una suma de dues funcions:

1. El cost del camí ja expandit, que és la distància coneguda des del vèrtex inicial fins l'actual⁸.
2. El cost del camí que queda per recórrer, que és una estimació heurística acceptable de la distància des del vèrtex 'x' fins el vèrtex final⁹.

L'heurístic ha de ser admissible, és a dir, no ha de sobreestimar la distància fins l'objectiu. Per tant, per una aplicació com l'encaminament, $h(x)$ pot representar la línia recta fins l'objectiu, ja que es tracta de físicament la distància mínima entre dos punts. Si l'heurístic $h(x)$ satisfà aquesta condició, $h(x) \leq d(x,y)^{10} + h(y)$. [8][11]

⁶ Stanford Research Institute

⁷ Normalment denotat com a $f(x)$

⁸ Normalment denotat com a $g(x)$

⁹ Normalment denotat com a $h(x)$

¹⁰ Funció que retorna la distància entre dos punts

2.2.3 Optimitzacions de Dijkstra i A*

Tot i que la necessitat de optimitzar tant l'algoritme de Dijkstra com el d'A* només es dona per grafs realment grans, ens plantejarem possibles optimitzacions d'aquests algoritmes i mirarem el les possibles millores en l'execució.

Pel que fa al Dijkstra, podem fer dues cerques concurrentment que comencin des de l'origen i el destí del camí que volem trobar[6]. Aquesta optimització ens pot donar un millor temps, ja que els dos camins iniciats es trobaran en el camí mínim.

Pel que fa a IA* podríem aconseguir millors resultats que amb el Dijkstra. Tot i que aquesta solució és més complexa. En aquest cas, es proposa marcar uns vèrtexs com a referències i calcular les distàncies entre ells. Amb l'ajuda d'aquests punts d'interès, s'assoleixen uns molt bons resultats[7].

Cal dir que aquestes optimitzacions donen molts bons resultats, com es demostra a l'estudi realitzat per Windows, per la cerca de camins en cotxe. Per tant, es coherent afirmar que en el nostre cas la diferència serà tant petita que potser és imperceptible o ni es dona una millora real. A la **figura 1** es poden observar les execucions d'aquestes optimitzacions[7].

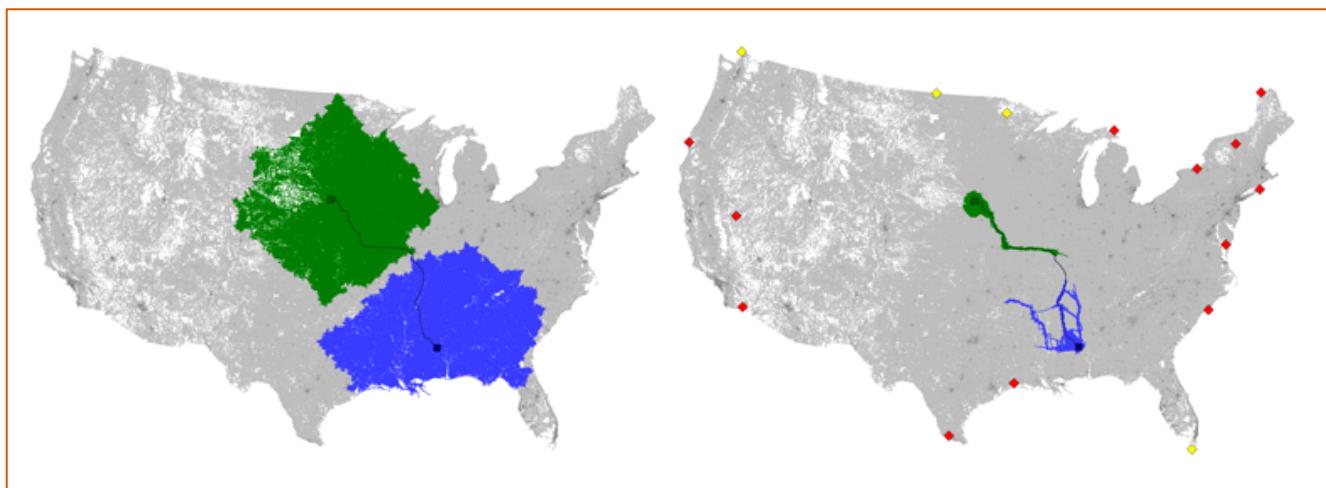


Figura 1: A l'esquerra podem veure la optimització feta a l'algoritme de Dijkstra i a la dreta la optimització de l'algoritme A*.

2.3 Disseny d'aplicacions mòbils

En el món de les aplicacions mòbils el disseny és molt important. La majoria de projectes d'aquest sector inverteixen més del 60% del temps en dissenyar la aplicació amb l'ajuda de dissenyadors gràfics. Per tant, aquesta part és molt important i se li haurà de dedicar tot el temps necessari per tal d'assolir aquell disseny que compleixi els nostres objectius: interfície intuïtiva, simple i concisa[13].

Per tal d'assolir un bon disseny l'equip de disseny i jo ens recolzarem en diferents vídeos i guies creades per Apple de com s'ha de dissenyar una aplicació per iOS[12][13][14][15].

2.4 Conclusions

Per tant, sembla que la millor solució serà l'algoritme de Dijkstra, ja que ens trobem davant d'un graf molt petit en el que, a priori, les optimitzacions sembla no aportaran molts beneficis. A més a més, caldrà escollir molt bé el criteri amb el que calcular el pes que se li dona a cada aresta per tal de trobar el camí mínim desitjat.

3 Definició de l'abast

3.1 Abast

Per trobar la millor solució possible calen fixar uns passos molt detallats abans de començar la elaboració del codi.

Primer s'estudiarà la codificació GTFS¹¹ creada per Google per tal de poder llegir i analitzar les dades proporcionades per la Generalitat de Catalunya i per el proveïdor dels serveis a la ciutat de Barcelona, TMB¹².

Un cop analitzades les dades es generarà un graf amb les xarxes de metro i bus i es farà un anàlisi i un estudi per tal de trobar el millor algoritme de cerca de camins.

Un cop creat el graf s'estudiarà l'ús de versions adaptades tant de l'A* com de Dijkstra i es valorarà quin dels dos algoritmes dóna millor resultat en cadascun dels casos, prenent com a resultat de l'experiment tant el temps d'execució com el nombre de nodes i arestes que els algoritmes visitin. Es posaran casos crítics amb situacions especials com, per exemple, festivitat o nocturnitat. A més a més, es ponderaran les condicions climàtiques, entre d'altres, per tal de trobar no només la solució més ràpida sinó també la més còmode.

Per últim, es farà una anàlisi de la influència en els resultats del nombre de punts d'interès que s'incloguin en els grafs, és a dir, s'analitzarà si la crescuda del nombre de punts d'interès fa créixer o no el cost computacional de la cerca.

Tot això formarà part d'una aplicació per iOS¹³ amb la que s'inclouran elements com el disseny i la inscripció, entre d'altres.

Finalment, cal dir que degut a la manca de temps, el projecte no inclourà el càcul de direccions al carrer, és a dir, el projecte només es centrarà en el transport públic.

¹¹ General Transit Feed Specification

¹² Transports metropolitans de Barcelona

¹³ Sistema operatiu per els dispositius portàtils d'Apple.

3.2 Possibles obstacles

Ara analitzarem els possibles obstacles que puguin sorgir durant la realització del projecte i les possibles solucions que es prendran per tal de minimitzar el seu efecte en el procés.

3.2.1 Errors en el codi

Un dels errors més habituals a l'hora de realitzar un projecte de software és l'error de programació. En un projecte amb una part computacional tant complexa cal minimitzar el nombre d'errors, ja que algun petit error pot generar resultats molt diferents als esperats. Per tal de intentar minimitzar el nombre d'errors es crearan una sèrie de proves a realitzar a cada nova versió del codi per tal d'assegurar que els resultats són els esperats.

3.2.2 Calendari

Degut a que el temps per realitzar aquest projecte és bastant limitat (3 mesos), caldrà fixar un calendari molt estricte amb fites a complir cada setmana i amb les respectives reunions amb el director del projecte per tal que validi la feina.

3.2.3 Canvi en el format de les dades

Tot i que el format GTFS s'ha estandarditzat, les dades compartides pels serveis de transports públics amb Google utilitzen de forma diferent alguns camps de les dades que són opcionals i, per tant, cal crear un codi flexible a aquesta variabilitat de les dades.

3.3 Metodologia i rigor

Com ja s'ha comentat, el temps disponible per fer el projecte és breu i caldrà que s'apliquin uns mètodes de treball de caràcter àgil i estricto. Existeixen moltes metodologies de treball d'aquest estil que s'apliquen a l'elaboració de projectes en els quals hi ha un grup de persones implicat. Tot i que aquest no és el cas, es poden aprofitar algunes metodologies per tal d'elaborar totes les tasques.

3.3.1 Cicles de desenvolupament curts

Mitjançant l'ús d'una metodologia de treball amb cicles curts (amb fites que s'han d'assolir cada setmana), es garanteix una millor visió real de l'estat del projecte i de si es troba o no dins del calendari marcat a l'inici del projecte.

3.3.2 Feedback del client

Tot i que aquest projecte no té un client real, es considerarà al director del projecte com el client final i es mantindran una sèrie de reunions periòdiques per tal de que hi hagi un feedback constant sobre l'estat del projecte i el que es desitja i així es puguin solucionar tots els problemes amb la màxima brevetat possible.

3.3.3 Desenvolupament guiat per proves

Ja s'ha esmentat la dificultat que poden suposar els errors de programació, ja que poden arribar a condicionar greument els resultats obtinguts. Per tal d'intentar minimitzar-ho, s'utilitzarà una metodologia de desenvolupament guiat per proves per tal d'intentar trobar els errors el més ràpid possible i així minimitzar els errors, ja que si es troben més tard quan el projecte ja ha avançat, poden suposar la revisió d'una part molt gran del projecte.

Per tant, podem concloure que s'utilitzarà la metodologia Scrum per la realització del projecte.

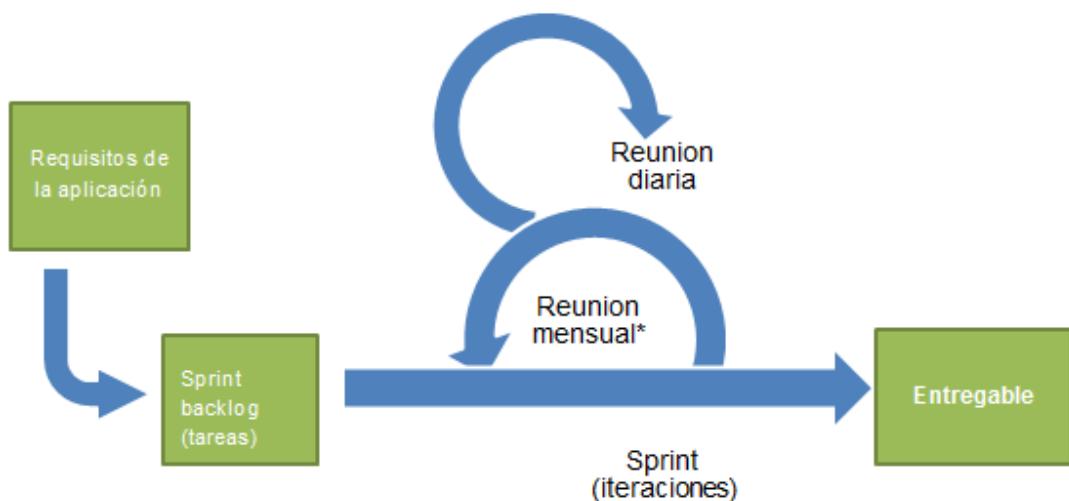


Figura 2: En aquesta imatge podem veure la estructura de la metodologia Scrum.

3.4 Eines per el desenvolupament

Per aquest projecte es farà servir el servidor SVN¹⁴ gratuït Google Code que permet oferir el codi sota una llicència pública, que sigui visible i que qualsevol usuari se'l pugui descarregar. Per tal de realitzar la tasca de disseny de l'aplicació s'utilitzarà la web 'fluid.com' per fer els mockups i la part de programació i implementació de la interfície es farà amb el Xcode proporcionat per Apple. Finalment, per les tasques de debugatge s'utilitzaran l'iPhone 6 i l'iPad Air.

A més a més, es farà servir el calendari d'OSX i iOS per tal de marcar unes fites a assolir cada setmana degut al poc temps disponible.

3.5 Seguiment del projecte

Mitjançant el registre que queda als servidors de Google Code amb totes les versions del codi i les reunions periòdiques amb el director del projecte, que estaran documentades amb les respectives actes de reunió, es farà un seguiment detallat del procés d'elaboració del projecte.

¹⁴ Subversion

4 Planificació temporal

4.1 Programa

La duració aproximada del projecte és de 4 mesos i mig, del 8 de setembre d'enguany fins al 25 de gener del 2015.

Cal destacar que la duració del projecte és estimada i, degut a la metodologia de treball escollida (Scrum), es pot veure afectada per algunes desviacions eventuals, a les quals farem referència més endavant.

4.2 Pla de projecte

4.2.1 Planificació del projecte (Fita inicial)

Aquesta part del projecte és en la que ens trobem actualment i consta de aquestes parts:

1. Definició de l'abast
2. Planificació temporal
3. Gestió econòmica i sostenibilitat
4. Contextualització i bibliografia (inclou l'estat de l'art)

4.2.2 Anàlisi i disseny del projecte

L'objectiu d'aquesta part és analitzar detalladament el projecte i especificar qui és el millor disseny final.

Pel que fa a l'anàlisi, caldrà que es fixin quins són els objectius a assolir en aquest projecte, els requisits, les funcionalitats i els casos d'ús. A més a més, s'aprofunditzarà en l'estat de l'art mitjançant un anàlisi de la competència i de la tecnologia utilitzades.

En quant al disseny, es crearà l'arquitectura del software, com per exemple el graf de connexions de l'aplicació, i la interfície gràfica "UI"¹⁵. Això implicarà plasmar els coneixements adquirits durant el grau agafant elements de l'enginyeria del software i la computació.

¹⁵ User Interface

4.2.3 Descripció de les tasques

A l'utilitzar una metodologia àgil basada en el prototipatge, no es marcaran unes tasques estrictes que es vagin realitzant iterativament, sinó que cada setmana es definiran els objectius a assolir adaptant el procés a possibles desviacions o problemes que puguin sorgir. Igualment, es poden definir unes tasques genèriques que sí que tenen un ordre estricte degut a les pre-condicions que existeixen entre elles.

Totes les tasques que es detallaran a continuació estan seccionades en una primera fase d'anàlisi, una segona de disseny, una tercera d'implementació i finalment una de proves. Lògicament les seccions que conformen les tasques, tenen una relació de dependència entre si que fan que tinguin un ordre establert i no es poden realitzar en paral·lel. Les tasques genèriques podrien definir-se de la següent manera:

0. Realització de la fita inicial.
1. Configuració de l'entorn de treball (Xcode, iPhone, llicència de desenvolupador).
2. Anàlisi de les dades GTFS i creació d'un lector d'aquestes.
3. Creació del graf amb l'ús de les dades descarregades al primer pas.
4. Parametrització del graf amb diferents condicions fixades per l'usuari.
5. Implementació de funcions addicionals com l'estat sense connexió i que poden ser d'utilitat pel client final.
6. Redacció de la memòria, l'annex i la documentació.

4.2.4 Fita final

A la fita final es presentarà la memòria que inclourà tots els prototips obtinguts durant el procés de desenvolupament de l'aplicació. A més a més, es lliurarà un apartat d'annexos amb la documentació de tot el codi, les actes de reunió que es faran durant el procés de la fita de seguiment, una guia d'ús de l'aplicació, una explicació del procés que cal per crear una aplicació iOS, des de la creació fins a la comercialització, i tots els documents rellevants per entendre correctament el funcionament de l'aplicació. També cal mencionar que es distribuirà l'aplicació a l'App Store de iOS.

4.3 Duració aproximada

Tasca	Duració aproximada
Planificació del projecte (Fita inicial)	90
Anàlisi i disseny	60
Configuració de l'entorn	10
Anàlisi de les dades	30
Creació del graf	160
Parametrització del graf	60
Funcions addicionals	60
Fita final (redacció de la memòria,...)	40
Total	520

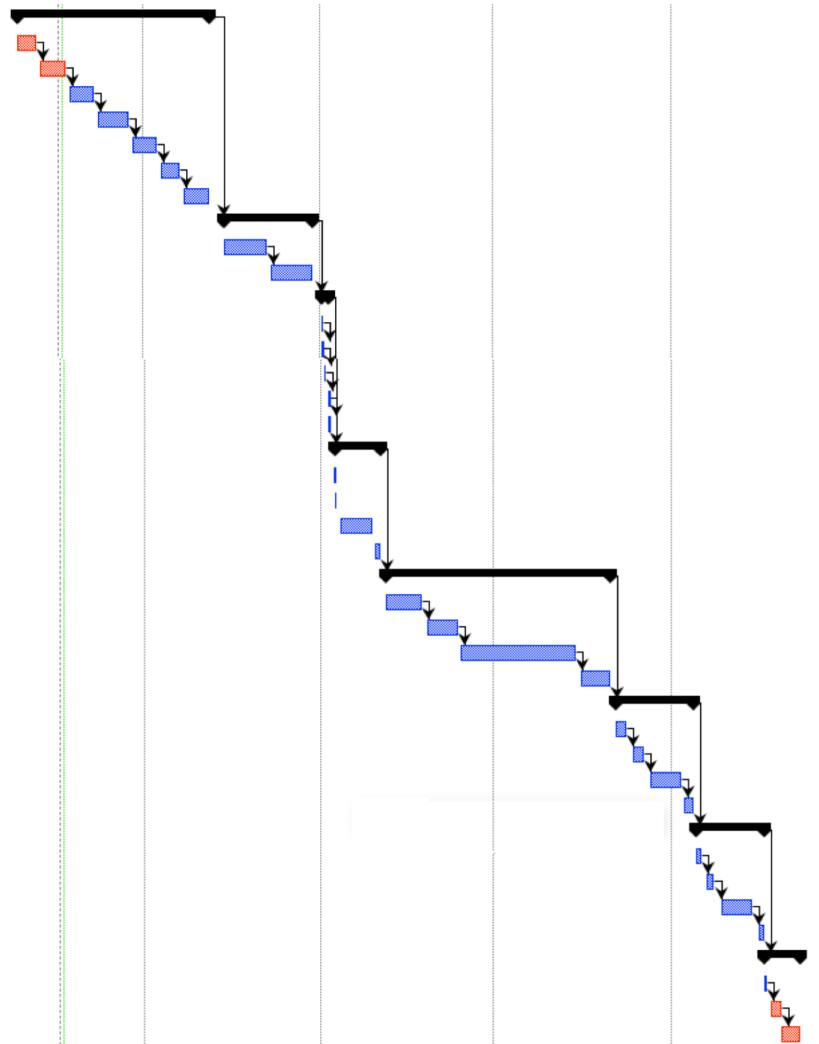
4.4 Valoració d'alternatives

Durant la realització del projecte poden sorgir diferents desviacions del pla original que s'intentaran analitzar i gestionar per tal que l'impacte sobre la planificació del projecte sigui el menor possible. Això s'aconseguirà gràcies a la metodologia àgil utilitzada (Scrum) ja que al fixar-se fites de forma periòdica (10-15 dies) permetrà gestionar aquestes desviacions amb facilitat i rapidesa.

Quant a les tasques, les hores fixades per a cada una poden variar i, alhora, poden solapar-se'n dues. Al final de projecte es podrà fer un seguiment de com s'han realitzat les tasques gràcies a les actes de reunió que es faran a l'inici i final de cada tasca (algunes reunions de finalització d'una tasca coincidiran amb el començament d'una nova).

4.5 Diagrama de Gantt

1	■ Fita Inicial	9/09/14 10:00	12/10/14 17:00
2	■ Definició de l'abast	9/09/14 10:00	12/09/14 17:00
3	■ Planificació temporal	13/09/14 10:00	17/09/14 17:00
4	■ Gestió econòmica i sostenible	18/09/14 10:00	22/09/14 17:00
5	■ Presentació preliminar	23/09/14 10:00	28/09/14 17:00
6	■ Contextualització i bibliografia	29/09/14 10:00	3/10/14 17:00
7	■ Revisió de les competències	4/10/14 10:00	7/10/14 17:00
8	■ Documentació presentació fita final	8/10/14 10:00	12/10/14 17:00
9	■ Anàlisi i disseny del projecte	15/10/14 10:00	30/10/14 17:00
10	■ Anàlisi	15/10/14 10:00	22/10/14 17:00
11	■ Disseny	23/10/14 10:00	30/10/14 17:00
12	■ Configuració de l'entorn	1/11/14 10:00	2/11/14 14:00
13	■ Xcode	1/11/14 10:00	1/11/14 12:00
14	■ Documentació Objective-c	1/11/14 12:00	1/11/14 14:00
15	■ Configuració iPhone/iPad	1/11/14 16:00	1/11/14 18:00
16	■ Compte desenvolupador	2/11/14 10:00	2/11/14 12:00
17	■ Descàrrega manual GTFS	2/11/14 12:00	2/11/14 14:00
18	■ Anàlisi dades GTFS més lec	3/11/14 10:00	11/11/14 12:00
19	■ Anàlisi	3/11/14 10:00	3/11/14 12:00
20	■ Disseny	3/11/14 12:00	3/11/14 16:00
21	■ Implementació	4/11/14 10:00	9/11/14 22:00
22	■ Proves	10/11/14 10:00	11/11/14 12:00
23	■ Creació del graf	12/11/14 10:00	21/12/14 10:00
24	■ Anàlisi	12/11/14 10:00	18/11/14 16:00
25	■ Disseny	19/11/14 12:00	24/11/14 22:00
26	■ Implementació	25/11/14 12:00	15/12/14 12:00
27	■ Proves	16/12/14 10:00	21/12/14 10:00
28	■ Parametritzar el graf	22/12/14 10:00	4/01/15 22:00
29	■ Anàlisi	22/12/14 10:00	24/12/14 10:00
30	■ Disseny	25/12/14 10:00	27/12/14 10:00
31	■ Implementació	28/12/14 10:00	2/01/15 20:00
32	■ Proves	3/01/15 10:00	4/01/15 22:00
33	■ Funcions addicionals	5/01/15 10:00	17/01/15 10:00
34	■ Anàlisi	5/01/15 10:00	6/01/15 10:00
35	■ Disseny	7/01/15 10:00	8/01/15 10:00
36	■ Implementació	9/01/15 22:00	15/01/15 8:00
37	■ Proves	16/01/15 10:00	17/01/15 10:00
38	■ Fita final	17/01/15 10:00	23/01/15 17:00
39	■ Tancar projecte	17/01/15 10:00	17/01/15 18:00
40	■ Redactar memòria, annex, ..	18/01/15 10:00	20/01/15 10:00
41	■ Presentació	20/01/15 10:00	23/01/15 17:00



4.6 Recursos

Per a la realització del pla de projecte proposat, s'utilitzaran diferents eines de hardware i de software.

4.6.1 Hardware

Les eines de hardware que s'utilitzaran són:

- Macbook Pro Retina 13"
- iPhone 6
- iPad Air



4.6.2 Software

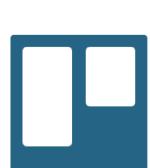
Les eines de software que s'utilitzaran són:

- OSX Yosemite
- iOS 8
- Xcode 6
- Microsoft Office 2011 (Mac)
- Trello
- Project Libre



**Project
Libre™**



 **Trello**

5 Gestió econòmica i sostenibilitat

5.1 Gestió econòmica

5.1.1 Consideracions i comentaris

Per tal de desenvolupar aquest projecte, s'utilitzaran tots els elements detallats a les entregues anteriors, que tenen un cost. Així, en aquest document es proporciona l'estimació del cost del projecte tenint en compte recursos humans, hardware, software i les seves llicències, i els costos indirectes.

Per tal de no superar el pressupost previst, s'actualitzaran les dades després de cada tasca. Per tant, el pressupost final contindrà les hores reals que s'han treballat i el preu final. S'haurà de portar el control amb cura, ja que cal que el projecte es realitzi dins dels terminis establerts.

Els elements sobre els que calculem el pressupost sorgeixen o estan derivats de tasques presents al diagrama de Gantt¹⁶.

5.1.2 Pressupost de recursos humans

Aquest projecte només serà portat a terme per una persona que haurà de ocupar-se dels 4 rols presents al projecte: cap de projecte, dissenyador, enginyer de software i beta tester. A la següent taula detallarem el nombre d'hores que caldran per dur a terme les tasques de cada rol i el valor econòmic de les hores invertides:

Rol	Hores	Preu per hora	Preu total
Cap de projecte	130 h	50€/h	6.500€
Dissenyador	60 h	35€/h	2.100€
Enginyer de software	260 h	35€/h	9.100€
Beta tester	60 h	30€/h	1.800€
TOTAL	520		19.500€

¹⁶ Apartat 4.5 de la pàgina 20

5.1.3 Pressupost de hardware

Per tal de dur a terme la implementació del projecte, sobretot el hardware, calen una sèrie d'elements de hardware per les tasques de documentació, implementació i proves. El preu de les reparacions poden ser variables però com a molt seran d'uns 250€ de mitjana per cada dispositiu.

Producte	Preu	Unitats	Vida útil	Amortització
MacBook Pro retina 13"	1.329€	1	5 anys	99,68€
iPhone 6	699 €	1	3 anys	87,38€
iPad Air	479€	1	3 anys	59,88€
TOTAL	2.507€			246,94€

5.1.4 Pressupost de software

També caldran eines de software, que seran utilitzades a totes les tasques del projecte. Totes les actualitzacions de software són gratuïtes.

Producte	Preu	Unitats	Vida útil	Amortització
Microsoft Office Mac	270€	1	3 any	33,75€
Project Libre	0,00 €	1	5 mesos	0,00€
Photoshop	700\$	1	3 anys	68,13 €
Illustrator	700\$	1	3 anys	68,13 €
Xcode	0,00€	1	-----	0,00€
iOS SDK	0,00€	1	-----	0,00€
OSX Yosemite	31,00€	1	3 anys	0,00€ ¹⁷
TOTAL	2.507€			170,01€

5.1.5 Pressupost de llicències

Per a la realització del projecte no només calen recursos en forma de hardware o software sinó que, a més a més, també calen una sèrie de llicències per tal de poder desenvolupar el producte i poder-lo comercialitzar.

Producte	Preu	Unitats	Vida útil	Amortització
iOS Developer Program	75,68€/any	1	1 any	28,38€
TOTAL	75,68€/any			28,38€

¹⁷ Inclòs al MacBook Pro Retina 13".

5.1.6 Despeses indirectes

En tot projecte d'informàtica també apareixen unes despeses indirectes derivades de l'ús d'electricitat o paper, entre d'altres.

Producte	Preu	Unitats	Cost aproximat
Electricitat	0,07€/kWh	65.500 kWh	4.585€
Paper	29,03€/pack	1 pack	29,03€
TOTAL	75,68€/any		4.614,03€

5.1.7 Pressupost total

Finalment, ajuntem totes les parts del pressupost per tal de veure el cost estimat total del projecte.

Concepte	Cost aproximat
Recursos humans	19.500€
Hardware	246,94€
Software	170,01€
Llicències	28,38€
Costos indirectes	4.614,03€
TOTAL	24.559,36€

El cost total del pressupost demostra que econòmicament és competitiu. Però, cal mencionar que es podria realitzar en menys temps si s'utilitzessin APIs¹⁸ públiques com les de Google Maps per el càlcul dels itineraris.

5.1.8 Control de desviacions

El principal problema que pot sorgir i que pot afectar el pressupost final és la desviació temporal en alguna de les tasques del projecte. Per tal d'intentar compensar aquestes desviacions s'utilitzarà el diagrama de Gantt per reorganitzar el temps que s'utilitzarà per cada tasca i així poder acabar el projecte en el temps esperat i amb el pressupost pactat. Es considera que degut a l'especificació que s'ha fet de les tasques, no es donarà un escenari en què la desviació temporal sorgeixi en tasques consecutives i, per tant es podrà acabar a temps el projecte.

¹⁸ Application Programming Interface

5.2 Àrea social

Aquest projecte es troba dins del món del software i més concretament, al món de les aplicacions mòbils. El mercat de les aplicacions mòbils no és molt gran a Espanya però cada cop té més importància. Aquest projecte pretén doncs impulsar el mon de les aplicacions mòbils a Barcelona (anomenada MWC¹⁹), Catalunya i Espanya.

Tot i que el producte final està destinat a turistes i no a la gent de Barcelona, aquest projecte pot aportar una bona alternativa a altres serveis com Google Maps. D'altra banda, aquest producte incideix en un dels sectors més importants de Barcelona, el sector dels serveis, però no es considera que pugui suposar cap problema al sector sinó tot el contrari, pot ajudar a millorar-lo i a facilitar la visita dels turistes a la nostra ciutat. S'espera doncs que l'usuari final de l'aplicació vegi millorada la seva visita i experiència a la ciutat de Barcelona.

5.3 Àrea ambiental

Pel que fa a l'àrea ambiental, aquest projecte no té una gran incisió, ja que al ser un projecte només de software les úniques despeses que poden afectar al medi ambient són les indirectes²⁰. Principalment perquè per obtenir tant la llum com el paper calen uns processos de fabricació previs que sí que deixen petjada al medi ambient.

5.4 Puntuació sostenibilitat

A la versió final de projecte, es proporcionarà una taula amb les puntuacions del projecte, però ara donarem les puntuacions per la part del projecte que ocupa aquesta entrega.

¿Sostenible?	Económica	Social	Ambiental
Planificación	8	6	8

¹⁹ Mobile World Capital

²⁰ Apartat 5.1.6 a la pàgina 24

6 Bibliografia

- [1] Bar-Noy, A., & Schieber, B. (1991, March). The canadian traveller problem. In Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms (pp. 261-270). Society for Industrial and Applied Mathematics
- [2] Edsger W. Dijkstra. A Discipline of Programming. Prentice-Hall, 1976.
- [3] Wilson, A. R. & Keil, C. F. The MIT Encyclopedia of The Cognitive Sciences. Massachusetts Intitute of Technology, 1999.
- [4] Cormen, T. H. & Leiserson, C. E. & Rivest, R. L. & Stein, C. *Introduction to Algorithms (3rd Edition)*, The MIT Press , 2009.
- [5] Sedgewick, R. *Algorithms in C++, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching and Graph Algorithms (3rd Edition)*, Addison-Wesley , 2001.
- [6] Chang, J. (2009, July 7). *Making The Shortest Path Even Quicker*. Retrieved from <http://research.microsoft.com/en-us/news/features/shortestpath-070709.aspx>
- [7] Goldberg V. A. & Kaplan H. & Werneck F. R. (2005, October). *Reach for A**: *Efficient Point-to-Point Shortest Path Algorithms*. Retrieved from <http://research.microsoft.com/pubs/60764/tr-2005-132.pdf>
- [8] Goldberg V. A. & Harrelson C. (2004, July). *Computing the Sortest Path: A* Search Meets Graph Theory*. Retrieved from <http://research.microsoft.com/pubs/64511/tr-2004-24.pdf>
- [9] Evdokia, N. & David R. K. (2008). *Route Planning under Uncertainty: The Canadian Traveller Problem*. Retrieved from <http://www.aaai.org/Papers/AAAI/2008/AAAI08-154.pdf>

- [10] Madrigal C. A. (2012, September 6). *How Google Builds Its Maps*. Retrieved from <http://m.theatlantic.com/technology/archive/2012/09/how-google-builds-its-maps-and-what-it-means-for-the-future-of-everything/261913/>
- [11] Roughgarden, T. (2012). *Design and Analysis of Algorithms*. Retrieved from <http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms>
- [12] Stern M. (2014, June). *Design Intuitive User Experiences*. Retrieved from <https://developer.apple.com/videos/wwdc/2014/?id=211>
- [13] Cavanaugh, J. & Roth, R. (2014, June). *Making a Great First Impression With Strong Onboarding Design*. Retrieved from <https://developer.apple.com/videos/wwdc/2014/?id=230>
- [14] Bernstein, T. J. & Dong, L. & Missig, J. & Hauenstein, M. (2014, June). *Prototyping: Fake It Till You Make It*. Retrieved from <https://developer.apple.com/videos/wwdc/2014/?id=223>
- [15] Apple Inc. (2014). *iOS Human Interface Guidelines*. [iBooks 1.1 OSX]. Retrieved from <https://itunes.apple.com/es/book/ios-human-interface-guidelines/id877942287?mt=11>