



UNIVERSITAT
POMPEU FABRA

ESCOLA SUPERIOR POLITÈCNICA
ESTUDIS D'INFORMÀTICA

*Projecte Fi de
Carrera*

**QuesTInSitu mobile:
aplicació per a rutes
de preguntes
geolocalitzades**

Fernando García García

Curs 2010-2011

ENGINYERIA TÈCNICA
EN INFORMÀTICA
DE SISTEMES

Directora:
PATRICIA SANTOS

QuesTInSitu *mobile*: aplicació per a rutes de preguntes geolocalitzades

Fernando Garcia Garcia

Projecte Fi de Carrera
Director/a: Patricia Santos
Juny del 2011
Enginyeria Tècnica en Informàtica de Sistemes
Universitat Pompeu Fabra

*A Alan Arisa,
siguis on siguis,
mai t'oblidaré.*

*En especial, gràcies a Patricia Santos
pel seu constant suport i la seva manera de fer,
a David Pérez per la incansable ajuda a la part
tècnica, junts han estat el millor tutor possible.*

Resum

En plena era de la informació, les noves tecnologies s'han posat també al servei de l'ensenyament. Per tal de proveir a l'estudiant dels mètodes més útils i eficients per donar suport al seu aprenentatge, han sorgit eines cada cop més acurades amb la intenció de proveir d'una manera robusta tot allò que fins fa poc només es podia fer a les aules. Així va néixer el concepte d'*e-learning*.

QuesTInSitu és una eina concebuda dins d'aquesta àrea que permet crear preguntes geolocalitzades sobre mapes de *GoogleMaps* i organitzar-les com a rutes (qüestionaris), sobre el mapa que es desitgi del món. Aquest projecte ofereix la possibilitat de poder realitzar físicament l'activitat creada per QuesTInSitu mitjançant terminals mòbils amb connexió 3G i GPS. Per això, s'ha dissenyat un portal web adaptat, QuesTInSitu *mobile*, que permet realitzar les gimcanes geolocalitzades dissenyades prèviament. Malgrat aquest projecte està contingut en l'àrea del *e-learning*, ofereix un llarg ventall de possibilitats d'ús; publicitat o turisme en són alguns exemples.

Resumen

En plena era de la información, las nuevas tecnologías se han puesto también al servicio de la educación. Con la intención de proveer al estudiante de los métodos más útiles y eficientes para dar soporte a su proceso de aprendizaje, han surgido herramientas cada vez mejor diseñadas con la intención de brindarle de una manera robusta todo aquello que hasta hace poco tiempo solo se podía hacer en las aulas. Con este objetivo nace el concepto del *e-learning*.

QuesTInSitu es una herramienta concebida dentro de esta área que permite crear preguntas geolocalizadas sobre mapas de *GoogleMaps* y organizarlas como rutas (cuestionarios), sobre el mapa que se desee del mundo. Este proyecto ofrece la posibilidad de poder realizar físicamente la actividad ofrecida por QuesTInSitu mediante terminales móviles con conexión 3G y GPS. Por este motivo se ha diseñado un portal web adaptado, QuesTInSitu *mobile*, que permite realizar yincanas geolocalizadas diseñadas previamente. Aunque este proyecto está contenido en el área del *e-learning*, ofrece un amplio abanico de posibilidades de uso; publicidad o turismo son algunos ejemplos.

Abstract

In the information age, new technologies are at the service of education. With the intention of providing the student with the most useful and efficient method to support the learning process, better tools are emerging and have been designed in order to provide a robust system that until now could only take place in the classroom. This is the basis of the e-learning concept.

QuesTInSitu is an e-learning tool that can create geolocated questions using GoogleMaps. Later you can organize the questions using the map that you want around the world. This project offers the possibility of physically doing the activities offered by QuesTInSitu, using 3G and GPS mobiles. For this reason we have designed an adapted web, QuesTInSitu mobile, which allows geolocated activities. Although this project is contained in the area of e-learning, it also offers a wide range of possible uses including advertising and tourism.

QuesTInSitu Mobile URL: <http://learn3scenario.upf.edu:8080/QTISM/>

Índex

1. Introducció.....	9
1.1 El context.....	9
1.2 Descripció del projecte.....	9
1.3 Els objectius.....	10
1.3.1 La localització geogràfica de l'usuari.....	10
1.3.2 El tractament de la informació rebuda de QTIS.....	11
1.3.3 La visualització de les dades.....	11
1.3.4 Avaluació de l'eina.....	11
1.4 Planificació del projecte.....	11
1.4.1 Documentació del projecte.....	12
1.4.2 Selecció i estudi de les eines a emprar.....	12
1.4.3 Anàlisi del disseny de l'aplicació.....	13
1.4.4 Implementació de l'aplicació.....	13
1.4.5 Avaluació de l'aplicació.....	14
1.5 Estructura de la memòria.....	19
2. Estat de l'art.....	21
2.1 L'e-learning.....	21
2.2 Informació geogràfica i tecnologia.....	22
2.2.1 Sistema d'informació geogràfica (SIG).....	22
2.2.2 Tècniques utilitzades.....	23
2.2.3 Sistemes de coordenades.....	24
2.3 Mètodes per a la localització.....	27
2.4 Discussió i anàlisi de tres eines SIG basades en la localització del usuari.....	29
2.4.1 <i>MScap</i>	29
2.4.2 <i>Foursquare</i>	30
2.4.3 <i>GoogleLatitude</i>	31
2.4.4 <i>QuesTInSitu</i>	31
2.4.5 Discussió de les funcions de QTISM.....	32
2.5 L'HTML 5.....	32
2.5.1 <i>Geolocation API Specification</i>	32
2.5.2 Funcionament.....	33
2.5.3 Privacitat.....	34
2.5.4 Mètodes.....	34
2.5.5 Compatibilitat del navegador.....	38
2.6 Mètodes de càlcul de la distància entre dos punts de la Terra.....	38
2.6.1 El teorema de Pitàgores.....	39
2.6.2 Fórmula en coordenades polars per a un model de Terra plana.....	39
2.6.3 Llei dels cosinus per trigonometria esfèrica.....	40
2.6.4 La fórmula de Haversin.....	40
3. Desenvolupament del projecte.....	43
3.1 Model casos d'ús.....	43
3.2 Anàlisi de requeriments.....	44
3.2.1 Requeriments del usuari.....	44
3.2.2 Requeriments funcionals.....	44
3.2.3 Requeriments no funcionals.....	45
3.3 Solució proposta.....	45
3.3.1 Descripció.....	46
3.3.2 Disseny dels elements de QTIS i QTISM.....	49
3.3.3 Comunicació entre els elements de QTIS i QTISM.....	49
3.3.4 Diagrama de flux del funcionament.....	50
3.3.5 Funcions més importants de QTISM.....	52
3.4 Avaluació QTISM.....	55
3.4.1 Context educatiu de la prova d'avaluació.....	55

3.4.2 Preparació de la activitat d'avaluació.....	55
3.4.3 Anàlisi dels resultats	56
3.4.4 Proposta de millores i treball futur.....	58
4. Conclusions	59
5. Bibliografia	61
Annex 1: Qüestionari d'avaluació.....	65
Annex 2: Documentació HTML5.....	67
Millores HTML5	67
<i>getCurrentPosition</i>	68
<i>PositionOptions Interface</i>	69
<i>Coordinates Interface</i>	69
Annex 3: Captures del codi.....	71

1. Introducció

Aquest capítol descriurà el context on es troba ubicada l'eina que es desenvoluparà, així com la tecnologia relacionada. D'altra banda, també es formularan els objectius d'aquest projecte i la planificació de les tasques. Finalment, s'explica com s'ha estructurat aquesta memòria.

1.1 El context

QuesTInSitu és una aplicació desenvolupada dins el marc de l'*e-learning* que consisteix en una eina de creació de preguntes localitzades basades en *GoogleMaps*. Conceptualment és molt senzill el seu ús. L'usuari pot especificar punts concrets sobre el mapa, i associar-los un event, ja sigui una pregunta, una fotografia o un so. D'aquesta manera, es pot anar constituint una ruta per qualsevol indret. A més a més, tenint en compte que cada pregunta s'associa a una coordenada GPS combinant aquesta eina amb un sistema basat en localització, es pot arribar a realitzar la ruta programada amb QuesTInSitu de forma presencial amb un mòbil. La idea es que, quan l'usuari s'apropi lo suficient a les coordenades on hi ha un event, aquest ha d'aparèixer per pantalla sent visualitzat, escoltat o contestat.

Aquest projecte consisteix a realitzar un mòdul d'ampliació per a l'aplicació web QuesTInSitu, d'ara en endavant, QTIS. El seu autor, David Pérez Calle, ho descriu de la següent manera:

'Aquesta és una eina d'autoria que permet crear preguntes geolocalitzades sobre mapes de GoogleMaps. Les preguntes segueixen l'especificació IMS Question & Test Interporability (QTI), i són gestionades per el motor de QTI newApis. L'usuari pot crear preguntes geolocalitzades i organitzar-les com a rutes (qüestionaris), sobre el mapa que desitgi del món. També s'ofereix la possibilitat de poder realitzar l'activitat de respondre a les preguntes geolocalitzades mitjançant mòbils 3G. Per això s'ha fet una aplicació especialment dissenyada per a mòbil. Els usuaris poden respondre les preguntes i veure la puntuació.' [1]

El mòdul farà possible utilitzar QTIS, desde un mòbil de manera que reconeixi la posició d'aquest i llenci els events geo-localitzats quan l'usuari es trobi en la posició correcte.

1.2 Descripció del projecte

L'autor de QuesTInSitu en el projecte reconeix alguna de les mancances de l'aplicació i que li agradaria millorar:

'També caldria crear un sistema de geolocalització diferent al que fa servir actualment QuesTInSitu com es MScape, ja que ens delimita la capacitat de distribució de l'eina ja que només és compatible amb dispositius mòbils amb sistema operatiu Windows Mobile, per aquest motiu s'hauria de transportar en un altre plataforma com Google Latitude [...] que fa servir un altre tipus de tecnologia i seria compatible per a qualsevol dispositiu mòbil 3G que hi han actualment en el mercat.' [1]

Tal com especifica el seu autor, per a la realització de l'activitat en un ambient mòbil mitjançant un terminal amb GPS, l'eina depèn d'un *software* propietari de la companyia *HP* anomenat *MScap*e que a l'estat de l'art s'explicarà amb més detall. Aquí és on entra en escena aquest projecte. Aquest mòdul, hauria de permetre captar la posició geogràfica del terminal mitjançant coordenades de latitud i longitud a temps real, és a dir hauria de tenir el que anomenem usualment un GPS. D'altra banda, també hauria de ser capaç de comparar aquestes coordenades actuals amb les requerides per l'aplicació QTIS per tal d'enviar la informació o events específics relacionats amb aquella posició, al terminal fent ús de la tecnologia 3G.

Per això serà necessari, en primer pla l'estudi de les opcions actuals relacionades amb posicionament global, que veurem a l'apartat *2.4 Discussió i anàlisi de tres eines SIG basades en la localització del usuari*. Cal destacar que per tal de saber amb que comparem les coordenades actuals del nostre terminal, i així poder transmetre l'event correcte associat a la posició correcte del usuari, l'eina QTIS ens hauria de proveir d'una taula semblat a la següent:

Coordenades geogràfiques	Event associat
Longitud: 2.2045084, Latitud: 41.9466589	Text
Longitud: 3.9875085, Latitud: 42.0987653	Imatge
Longitud: 4.4645670, Latitud: 44.2345678	Link
Longitud: 2.5803865, Latitud: 42.0976324	Àudio
Longitud: 3.2332496, Latitud: 43.5839654	Altres

Figura 1. Estructura del fitxer a llegir subministrat per QTIS

Per tal de facilitar aquest procés, sabem que QTIS és capaç de condensar tota la informació de la gimcana en un sol fitxer XML amb estructura similar a la *figura 1*. Aquesta taula prové de totes les coordenades específiques associades a l'event concret. Tenint això en compte, seria necessari que el mòdul de geolocalització pogués llegir aquest fitxer i consultar-lo quan fos necessari per indicar el següent punt o d'altra banda, que pogués emmagatzemar tota aquesta informació mitjançant una lectura completa per a ser tractada internament, depenent de la implementació.

Per tal de satisfer les exigències de l'aplicació QTIS, serà necessari l'estudi de tecnologies de geolocalització en la web similars a *MScap*e [2], *GoogleLatitude* [3] o d'altres per veure quin és l'ús i el tractament que es pot fer de les dades provinents de QTIS i com mostrar-les d'una manera visualment eficient i atractiva en el marc dels dispositius mòbils.

1.3 Els objectius

En primera instància, per a analitzar el conjunt d'objectius a fixar, hauriem de dividir el mòdul en dues parts importants; la primera farà referència als objectius que s'han d'assolir en la part de geolocalització, la segona farà referència al tractament de la informació proporcionada per QTIS, i una tercera que guardarà relació amb tot allò relacionat amb la interacció del usuari amb l'aplicació.

1.3.1 La localització geogràfica de l'usuari

Un dels primers objectius en aquest àmbit es que s'ha d'intentar en la mesura de lo possible realitzar aquesta tasca de forma que el **codi** sigui **obert**. D'altra banda, hauria de tenir la **màxima compatibilitat amb tot tipus de dispositius mòbils** de qualsevol tipologia i marca,

per això s'hauria de fer un petit estudi per triar quin tipus d'aplicació seria l'idònia per satisfer aquestes dues necessitats.

També és indispensable que el **posicionament tingui** una **precisió** suficient alta per a no produir problemes en el funcionament i la **fiabilitat** suficient per a que el seu funcionament sigui continuat i sense interrupcions. No obstant això, si ens trobem en un radi pròxim, s'hauria de permetre rebre el nou event al dispositiu. Per tant, un objectiu molt clar en aquest sentit seria dotar l'aplicació d'un sistema per **calcular la proximitat del objectiu**. Atès que es depèn d'una aplicació ja desenvolupada, hauria de ser totalment **compatible en el tipus de coordenades utilitzades per QTIS**, és a dir que facin servir les mateixes unitats de mesura de longitud i latitud, per tal d'establir una homogeneïtzació.

1.3.2 El tractament de la informació rebuda de QTIS

Aquesta part serà la més important. És necessari que l'**aplicació sigui** de **codi obert** i que a la vegada totalment independent o **encapsulada** de l'aplicació web QTIS. Per tant, s'haurà de buscar la manera de transmetre les dades que ens proporciona QTIS per a que apareguin al terminal. Tanmateix, quan rebem la informació que fa referència a la ruta, aquesta s'ha de tractar d'una manera rigorosa i amb una **gestió d'errors** eficient, ja que estem parlant d'un sistema de localització a temps real.

D'altra banda, s'haurà de **llegir la informació rebuda** de l'aplicació tenint en compte el rang d'error que pot tenir la localització GPS. Mai serà possible trobar-se a les mateixes coordenades que indica l'aplicació, ja que la tecnologia GPS no disposa d'una precisió absoluta.

1.3.3 La visualització de les dades

Tenint en compte que el mòdul s'executarà sobre un terminal mòbil, també seria important **adequar l'eina** a una distribució i dissenys centrats en les característiques dels **dispositius mòbils**. Així doncs, s'hauria de fer especial èmfasi en aspectes tant importants com la **visualització clara** de la informació, la grandària de la pantalla o la velocitat de l'aplicació per a millorar l'experiència del usuari. Així doncs, seria convenient observar, mitjançant test als usuaris, com reaccionen aquests a determinades disposicions de la informació.

1.3.4 Avaluació de l'eina

Finalment, i una vegada **creada l'eina**, es realitzarà una primera presa de contacte amb els usuaris per tal de detectar errors, **millorar l'eina** i fer una **avaluació final** amb els usuaris.

1.4 Planificació del projecte

La planificació del projecte és una tasca molt important, així com la divisió de les etapes que el conformen, per després, poder establir terminis d'entrega en aquestes. Per això és necessari aquest redactat abans de començar tan sols a buscar la informació necessària i el corresponent calendari setmanal de treball (veure *figura 2*) i establiment d'objectius associats a dates d'entrega. Aquesta planificació es realitzarà seguint els següents passos enumerats a continuació.

1.4.1 Documentació del projecte

Des de el primer moment en que decideixo realitzar aquest projecte i començo a buscar informació, veig la necessitat d'adquirir conceptes teòrics sobre la vessant del *e-learning* i el posicionament GPS. Alguns d'aquests conceptes els enumero a continuació:

- Conceptes bàsics sobre l'*e-learning*, així com les característiques que el defineixen. També les seves bases, principis i tendències.
- La contribució del *e-learning* als resultats individuals i col·lectius.
- Els avantatges de la xarxa com a vehicle del ensenyament i els coneixements que millor s'adapten a ser impartits en ell.
- Com aconseguir la motivació del interessat.
- Futur del *e-learning*.

D'altra banda caldrà documentar-se sobre els següents aspectes del posicionament GPS:

- Conceptes bàsics sobre posicionament GPS, així com les característiques que el defineixen. També les seves bases, principis i tendències.
- Documentar-me sobre els recursos i eines disponibles per a la realització de la implementació, o més concretament els llenguatges idonis.
- Esbrinar el format que millor s'adapti a l'aplicació, si nativa, o en format web, depenent del compliment dels objectius descrits anteriorment.
- Cercar a la xarxa, projectes similars de geolocalització d'events, com *GyPSii*, *CloudMade* o els ja esmentats *GoogleLatitude* i *MScape*.
- Documentar-me les característiques de les aplicacions natives o no, amigables als dispositius mòbils, així com un suport cartogràfic per complementar l'aplicació, com *GoogleMaps* o *OpenStreetMap*.
- Estudiar les maneres existents d'interacció amb els mòduls GPS dels terminals mòbils, per tal d'esbrinar les coordenades de localització.

Finalment, en referència al disseny de l'aplicació i la visualització de les dades:

- Indagar en la mesura de lo possible, mètodes estàndards utilitzats en el disseny i disposició de les dades en medis mòbils mostrats d'una manera adient per facilitar la comprensió d'aquestes [4].

Aquesta divisió de la documentació es necessària tant a l'explicació dels passos com en el procediment de recerca real, tal com es pot observar al calendari setmanal a l'etapa de documentació del projecte.

1.4.2 Selecció i estudi de les eines a emprar

Durant el desenvolupament del projecte seran necessàries tot un conjunt d'eines per a realitzar, sobretot la programació i el disseny de l'aplicació. Aquesta part facilita la implementació, es per això que a la *figura 2*, el calendari setmanal, se li ha atorgat un període de cinc setmanes:

- Triar i aprendre el/s llenguatge/s necessari/s per a poder obtenir un resultat eficient en l'aplicació a desenvolupar.
- Una vegada estudiada la possibilitat del format a desenvolupar, caldrà determinar l'eina de desenvolupament, IDE, més apropiada per aquell llenguatge.
- Triar la forma en que realitzem la petició de coordenades als mòduls GPS dels terminals mòbils.

- Estudi d'alguns estàndards de localització, com el de W3C, basat en navegadors web sobre programació HTML5, o altres.
- Estudi d'un estàndard de visualització de contingut per a terminals mòbils, com per exemple el impulsat, també, per W3C.
- Familiaritzar-me amb l'API del suport cartogràfic, si s'escau i estudiar els requeriments de *Open Geospatial Consortium*, la institució que s'encarrega de l'estandardització de serveis espacials.
- Observar el funcionament de l'aplicació QTIS per tal d'esbrinar la millor manera d'establir la comunicació necessària per intercanviar events i coordenades. O si es el cas, la forma de tractar la informació generada per QTIS.

És molt important que una vegada triat mètode de desenvolupament, ens assegurem de la seva precisió de coordenades i localització respecte a la posició real, ja que s'hauria d'intentar obtenir una precisió mínima per a la localització del terminal.

1.4.3 Anàlisi del disseny de l'aplicació

Per a poder realitzar una implementació estructurada, considero oportú dedicar tres setmanes a fer un anàlisi dels requeriments futurs que pot tenir el projecte i d'alguna manera, endreçar-los per tal que la implementació sigui més metòdica i no es prenguin decisions del disseny sobre la marxa que posteriorment puguin condicionar els resultats.

Per tant, per a fer un primer disseny de l'aplicació, aplicarem conceptes relacionats amb la metodologia del disseny contextual. Vegem-los:

- Fer una modelització que ajudi a visualitzar el treball a realitzar i que mostri les distincions importants. Un dels models que s'utilitzarà al disseny serà el de flux, que mostra la comunicació i la coordinació entre els diferents elements.
- D'altra banda, per tal d'oferir un model entenedor i flexible d'allò que es vol fer, farem servir també part de la metodologia 'disseny centrat en l'ús'; investigant els usuaris per establir rols i tasques, així com models de casos d'ús.
- Finalment, culminarem aquesta part del disseny de l'aplicació, amb una proposta de disseny en paper, un model abstracte dels continguts de diversos espais d'interacció del sistema.

1.4.4 Implementació de l'aplicació

Per tal d'implementar l'aplicació, és necessari abans de res, pensar en les parts essencials del procés per tal de definir aquelles funcions indispensables. Algunes d'elles poden ser:

- Funció que obre i llegeix la informació de QTIS.
- Algoritme de calcula la proximitat al radi del objectiu/event.
- Algoritme que esbrina les coordenades de la posició actual i les mapeja.
- Funció que compara les coordenades actuals amb les del event.
- Funció que mostra els events per pantalla.
- Ordenació i visualització de la informació.

Aquesta tasca pot esdevenir la més complexa de totes, i per aquest motiu s'han establert deu setmanes com a període d'implementació al calendari setmanal, cadascuna d'elles amb objectius més específics, com funcions més essencials i concretes.

1.4.5 Avaluació de l'aplicació

Després de la implementació de l'aplicació, serà necessari realitzar un test amb usuaris reals conjuntament amb l'aplicació QTIS, per tal detectar possibles errors en la implementació i/o a la visualització de la informació. És de vital importància detectar aquells possibles problemes de comprensió que poden sorgir a causa d'una mala tria de la disposició de la informació. Tanmateix, també serà necessari fer èmfasi en que l'aplicació funcioni correctament, i esbrinar aquelles millores que poden facilitar la tasca a l'usuari.

La situació ideal per a fer aquest estudi d'usabilitat és mitjançant l'experiència *in situ* del usuari realitzant una gimcana i en un context educatiu real, en aquest cas concret, aquesta prova es va realitzar al parc natural de Sant Llorenç del Munt i va ser coordinada per Patricia Santos. La primera impressió que té l'usuari de la interfície i el *feedback* que li transmet cadascuna de les parts en referència a la seva funció és molt útil de cara a l'avaluació.

Una vegada identificats els errors, s'han de classificar i prioritzar per importància i, en l'informe, llençar unes recomanacions per tal de solucionar-los. Finalment, és important ser capaç de veure després de la implementació, si s'han acomplert els objectius que es van fixar en primera instància o si, pel contrari, n'han esdevingut de nous conforme es desenvolupava el projecte. Per aquest motiu, serà necessari dedicar a aquesta part, tal com indica el calendari setmanal amb les etiquetes '*Avaluació de l'aplicació*', unes dues setmanes.

Setmana	Tasca a desenvolupar	Etapa	Termini	Event
25-31 Octubre	Adquisició i redactat de coneixements sobre bàsics sobre <i>e-learning</i> .	<i>Documentació del projecte</i>	_____	_____
1-7 Novembre	Adquisició de coneixements bàsics en l'àmbit del posicionament GPS.	<i>Documentació del projecte</i>	_____	_____
8-14 Novembre	Redactat dels coneixements relacionats amb el posicionament GPS.	<i>Documentació del projecte</i>	_____	_____
15-21 Novembre	Cercar eines similars a aquesta i el format idoni per realitzar l'aplicació.	<i>Documentació del projecte</i>	_____	_____
22-28 Novembre	Cercar diferents tecnologies i recursos per a l'obtenció de coordenades.	<i>Documentació del projecte</i>	26 de Novembre	Entrega del formulari de registre del projecte.
29 Novembre 5 Desembre	Cercar les característiques de les aplicacions en el format triat i un suport cartogràfic adient.	<i>Documentació del projecte</i>	_____	_____
6-12 Desembre	Redactat de totes aquelles decisions sobre la tecnologia i eines a emprar.	<i>Documentació del projecte</i>	_____	Final del primer trimestre.
13-19 Desembre	Redactat de totes aquelles decisions sobre la tecnologia i eines a emprar.	<i>Documentació del projecte</i>	19 de Desembre	Entrega del redactat de la primera part del projecte
20-26 Desembre	_____	_____	_____	_____
27 Desembre 2 Gener	Triar el llenguatge de desenvolupament, així com el seu IDE	<i>Selecció i estudi de les eines</i>	_____	_____
3-9 Gener	_____	_____	_____	_____
10-16 Gener	Estudi dels d'estàndards: W3C, OGC... Testejar l'API del suport cartogràfic	<i>Selecció i estudi de les eines</i>	_____	_____
17-23 Gener	Estudiar la manera d'establir comunicació amb l'aplicació QTIS i el mòdul GPS	<i>Selecció i estudi de les eines</i>	_____	_____
24-30 Gener	Comprovació de les limitacions i eficiència de les eines utilitzades.	<i>Selecció i estudi de les eines</i>	_____	_____
31 Gener 6 Febrer	Memòria de totes aquelles decisions que s'han pres per a la selecció de les eines.	<i>Selecció i estudi de les eines</i>	6 de Febrer	Entrega del redactat de la segona part del projecte.

7-13 Febrer	Fer servir part de la metodologia del 'disseny contextual' com la modelització.	<i>Anàlisi del disseny</i>	_____	_____
14-20 Febrer	Modelar rols d'usuari, tasques i establir casos d'ús.	<i>Anàlisi del disseny</i>	_____	_____
21-27 Febrer	Redactat de la documentació relacionada amb el disseny i la visualització de la info.	<i>Anàlisi del disseny</i>	27 de Febrer	Entrega del redactat de la tercera part del projecte.
28 Febrer 6 Març	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
7-13 Març	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
14-20 Març	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
21-27 Març	_____	_____	_____	_____
28 Març 3 Abril	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	Final del segon trimestre.
4-10 Abril	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	5 d' Abril	Entrega del informe de progrés.
11-17 Abril	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
18-24 Abril	_____	_____	_____	_____
25 Abril 1 Maig	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
2-8 Maig	Implementació. Redacció de la documentació.	<i>Implementació de l'aplicació</i>	_____	_____
9-15 Maig	Redacció de la documentació de la implementació de l'aplicació (i)	<i>Implementació de l'aplicació</i>	_____	_____
16-22 Maig	Redacció de la documentació de la implementació de l'aplicació (ii)	<i>Implementació de l'aplicació</i>	22 de Maig	Entrega del redactat de la quarta part del projecte.
23-29 Maig	_____	_____	_____	_____

30 Maig 5 Juny	Petit estudi d'usabilitat mitjançant l'avaluació experimental del disseny.	<i>Avaluació de l'aplicació</i>	_____	_____
6-12 Juny	Anàlisi dels resultats d'avaluació i redactat de conclusions i millores respecte el disseny original.	<i>Avaluació de l'aplicació</i>	_____	_____
13-19 Juny	_____	<i>Tancament del projecte</i>	19 de Juny	Entrega del redactat de la cinquena part del projecte.
20-26 Juny	Preparació de la presentació del PFC	<i>Presentació</i>	_____	_____
27 Juny 3 Juliol	Presentació del PFC	_____	_____	Memòria impresa i presentació PFC

Figura 2. Taula de Planificació setmanal.

*Aquesta planificació ha de ser estricte però a la vegada flexible als imprevistos que segur poden sorgir. No obstant això, és molt important l'acompliment dels terminis que s'han fixat, per tal de portar el treball de redacció de memòria al dia i de retruc, la implementació de l'aplicació.

1.5 Estructura de la memòria

La memòria comença amb una introducció al treball realitzat en el projecte, on s'explica breument en què consisteix el projecte. En primer lloc, a la introducció del capítol 2. *Estat de l'Art* hi ha el context històric on es descriu l'estat de la tecnologia al començar el projecte i quines circumstàncies han habilitat el que es pugui realitzar. També hi ha una breu explicació dels objectius que m'he anat marcant juntament amb la meva tutora i que s'han anat succeint. I la planificació del projecte amb els aspectes claus, en els que té fonament el projecte.

En una segona part de la memòria parlo de l'estat de l'art. Durant aquesta part s'explica sobre la tecnologia, en concret dels diferents temes que engloba el meu projecte i que cal tractar com la tecnologia educativa. Aquí explico les característiques de termes com que vol dir: tecnologia educativa, estàndard educatiu o sistema d'informació geogràfica (SIG) entre molts altres. I exposo per a que aquestes són importants per al meu projecte i la forma de que en faig ús dels recursos que em proporcionen. També explico una mica com funcionen i com interpreta la informació.

A més a més, parlem sobre els sistemes d'informació geogràfica (veure apartat 2.2 *Informació geogràfica i tecnologia*), explicant en què consisteix, per a que són útils i les possibilitats que proporcionen a l'eina, gràcies a les iteracions que es poden produir dins del mapa. A l'apartat 2.3 *Mètodes per a la localització*, també es parla dels tipus de mapes, dels diferents sistemes de geolocalització teòrics i les diferents aplicacions que existeixen per a servir aquest tipus de sistemes, com *GoogleMaps*, i com funcionen.

A continuació, es fa un anàlisi de les diferents eines SIG basades en la localització del usuari, a l'apartat que porta per nom 2.4 *Discussió i anàlisi de tres eines SIG basades en la localització del usuari*. Més concretament, s'analitzen *MScape*, *FourSquare*, *GoogleLatitude*, i *QuesTIInSitu*.

Donats aquests punts, passo a exposar en un altre apartat, el 2.5 *L'HTML 5*, que tracta sobre tots els aspectes tècnics. Inicialment faig una descripció de les propietats de l'eina creada i posteriorment, faig un anàlisi dels requeriments que es necessiten i que són implícits amb tot el software emprat. En aquesta línia, també apareix un apartat de perfil més tècnic, com és el 2.6 *Mètodes de càlcul de la distància entre dos punts de la Terra*, on s'expliquen les diferents metodologies per aquest fi, així com la emprada a la implementació.

Al llarg del capítol 3. *Desenvolupament del projecte* defineixo els tipus d'usuaris que poden tenir lloc i els casos d'ús que podem realitzar amb cada rol (veure apartat 3.1 *Model casos d'ús*). A continuació, es fa èmfasi en l'anàlisi dels requeriments a l'apartat 3.2, tant del usuari, com funcionals i no funcionals. Finalment, la solució proposta, així com l'explicació del disseny i comunicació dels diferents elements de l'eina es tracta al llarg de l'apartat 3.3 *Solució proposta*. Allà, documento la base de dades i les diferents classes amb un resum de les pantalles més significatives en format fitxa. A continuació, tot l'apartat 3.4 *Avaluació de QTISM* està dedicat al context educatiu de la prova d'avaluació, la preparació d'aquesta, l'anàlisi dels resultats i les propostes de millores i treball futur.

Com a últim apartat es pot trobar el 4. *Conclusions*, on apareixen totes aquelles reflexions obtingudes de la realització d'aquest projecte. També es disposa d'un apartat bibliogràfic per a consultar figures i conceptes extrets d'altres documents, així com un annex on es poden trobar els qüestionaris que s'han realitzat per avaluar l'aplicació.

2. Estat de l'art

En aquest capítol s'explicarà l'estat en que es troba la tecnologia existent i que ha permès la realització del projecte, així com una explicació detallada dels conceptes necessaris per entendre el que es treballa.

Primerament es indispensable contextualitzar el marc d'aquest projecte, s'explica doncs el concepte d'*e-learning*, els diferents àmbits en que s'estableix i algunes de les tecnologies que empra.

A continuació, es comencen a detallar els requeriments de QTISM, la part mòbil de QuesTInSitu i l'aplicació final que s'obtindrà d'aquest projecte, i a investigar les diferents parts. Com un d'ells és la necessitat de mostrar cartografia, es fa especial èmfasi en els Sistemes d'Informació Geogràfica (SIG), així com els sistemes de coordenades associats. Això es molt útil al llarg del desenvolupament del projecte, ja que molt sovint es manipulen sistemes de coordenades. Tot seguit s'expliquen els diferents mètodes per a la localització existents, necessari per tal d'esbrinar la posició del usuari en temps real, i així saber on es troba en cada moment reflectint-ho sobre la cartografia.

Com els conceptes de cartografia i posició esmentats anteriorment, molt sovint van de la mà en moltes aplicacions i sens dubte en aquesta també, s'aprofundeix en algunes d'aquestes eines SIG, conceptualment similars.

Finalment, per tal d'esbrinar la posició d'un usuari, es va trobar que l'API de geolocalització d'HTML5 permetia aquesta possibilitat. Serà necessari, per tant, aprofundir en l'ús de l'API, les seves funcions, mètodes, compatibilitat i limitacions. Tanmateix, tota aquesta part necessita de mètodes de càlcul per a poder comprovar la proximitat del usuari amb l'event en qüestió. Per aquest motiu es realitza un petit estudi de precisió dels diferents algorismes, fet que facilita la tria del més idoni a implementar. En aquest cas ha estat l'algoritme basat en la fórmula de Haversine.

2.1 L'e-learning

Avui dia, vivim en un món dominat per les noves tecnologies, no havent-hi sector que s'hagi lliurat de l'aportació de la ciència a aquest canvi; Indústria, comerç, navegació, I+D, agricultura o ensenyament, són alguns exemples de sectors que han sofert una forta revolució a causa de les tecnologies de la informació. Aquestes noves tecnologies han possibilitat millorar l'eficiència dels processos, l'automatització de tasques o l'accés a recursos compartits gràcies a la creació d'una xarxa mundial, Internet.

El sorgiment de Internet ha estat una gran aportació al sector de l'ensenyament, i sense el qual conceptes com l'*e-learning*, no haguessin estat possibles. És fàcil de veure que una de les bases d'aquest concepte, és que la tecnologia facilita que l'educació pugui ser a distància i completament virtual, fet que posa de manifest la necessitat d'una xarxa de comunicacions molt sòlida per a poder donar un servei estable i de qualitat. Així doncs, si es milloren o adequen les xarxes o processos de comunicació (ja sigui GPS, telefonia...) amb l'objectiu de donar suport a aquest sector, és apropiat afirmar que aquest projecte que realitzo s'ubica dins aquest camp.

No obstant això, el naixement del *e-learning* no ha estat paral·lel al sorgiment d'Internet, cal recordar doncs l'ús de formats com el DVD o el CD-ROM, la videoconferència o altres que ja queden lluny, però que van ser les primeres aportacions en aquest camp. D'altra banda, amb sorgiment i combinació de conceptes tals com la Web 2.0, les xarxes socials o les enciclopèdies lliures com *Wikipedia*, han dotat a la població d'una eina massiva de coneixement col·lectiu [4]. Això, sumat a la gran accessibilitat que proporciona la xarxa ha permès el sorgiment del que avui dia coneixem com *Tecnologia Educativa, comunment més coneguda com a e-learning*.

L'*e-learning* [5] es preocupa de donar suport tecnològic a l'educació mitjançant la creació d'eines per tal de millorar el model docent, doncs gràcies a les seves característiques afavoreix que a l'individu li sigui més fàcil aconseguir el coneixement que necessita de forma puntual, adaptant-se al ritme i horaris que requereixi, encara que es trobi físicament allunyat. La informació que es brinda és precisa, personalitzada i de fàcil actualització i distribució. També aquest coneixement prové de l'opinió intel·lectual d'un grup i no es condiona a la d'un únic docent. S'afavoreix el trànsit i intercanvi de coneixement de forma ràpida i eficient, també ens dota d'una capacitació per al desenvolupament de tasques en comú, en el cas de la nostra eina poder dotar a l'usuari d'aquesta informació personalitzada allà on estigui.

El context més comú d'ús del *e-learning* ha estat l'ordinador, enguany dispositius com els terminals mòbils, *smartphones*, que inclouen serveis com el GPS, càmera, Internet o *bluetooth*, són cada vegada més comuns en l'àmbit educatiu. És aquí on aquest PFC entra en joc, combinant els serveis de posicionament de terminals en context educatiu [6].

2.2 Informació geogràfica i tecnologia

És clar però, que també podríem contextualitzar-lo en un ambient de cartografia i posicionament global, que al cap i a la fi, seran els conceptes que s'utilitzaran per a satisfer les necessitats del projecte, així com la seva implementació.

El Sistema de Posicionament Global o GPS sorgí l'any 1964, i actualment és una constel·lació de 32 satèl·lits artificials uniformement distribuïts en un total de 6 òrbites. Aquests satèl·lits orbiten la terra a una alçada de 20.000 km i recorren dos òrbites completes al dia. Cada satèl·lit transmet senyals de radio a la Terra amb informació relativa a la seva posició. Podem rebre aquesta informació mitjançant receptors que descodifiquen les senyals de diversos satèl·lits simultàniament combinant les seves informacions per a calcular la posició del receptor a la Terra, és a dir citar les seves coordenades de latitud i longitud amb precisió variable [7].

Amb el sorgiment, l'any 1854 dels sistemes d'informació geogràfica (SIG) i amb el seu posterior desenvolupament a l'actualitat, aquestes dues tecnologies han convergit per dotar l'usuari de la capacitat d'integrar elements diversos, editar, analitzar, compartir o mostrar informació mitjançant un conjunt de referències geogràfiques. A partir d'aquest tipus de sistema l'usuari pot realitzar interaccions, i així obtenir informació de forma iterativa per a resoldre problemes complexos de planificació i gestió [8].

Quan aquests dos conceptes varen convergir, sorgí el que avui dia coneixem com a GPS: triangular la posició actual del usuari i ser capaç de situar-ho en un mapa. A partir d'aquí, poder gestionar una ruta punt a punt, ser capaç de situar en un mapa múltiples terminals i tot allò que se'ns pugui acudir. Aquí es on entra en joc el mòdul que desenvoluparem i que explicarem més endavant. Però a grans trets, ha de ser capaç de detectar la posició del usuari i, si es adient, retornar-li alguna cosa.

En aquest apartat s'explicarà tot allò relacionat amb l'obtenció i tractament de la informació geogràfica provinent d'un *software*, així com les tipologies de representacions de dades i sistemes de coordenades.

2.2.1 Sistema d'informació geogràfica (SIG)

Es tracta d'una eina de gestió de la informació espacial on els usuaris poden fer cerques definides, analitzar la informació espacial, i editar-ne les dades. Aquestes dades van relacionades amb referències geogràfiques, ja sigui associades a un mapa, o simplement mitjançant coordenades geogràfiques [8].

Els sistemes d'informació geogràfica tenen usos en la gestió de recursos, la planificació de despeses, l'avaluació de l'impacte mediambiental, plans de desenvolupament, la cartografia o la planificació de rutes. Un sistema d'informació geogràfica pot convertir informació digital, que no es troba en forma de mapa, de manera que pugui ser tractada i visualitzada de forma integrada. Per exemple, pot integrar imatges de satèl·lit obtingudes per teledetecció que incloguin informació sobre poblacions.

El funcionament conceptual d'un SIG és simple, ja que funciona com una base de dades amb informació geogràfica associats a un mapa mitjançant un identificador. Així doncs si es localitza l'identificador al mapa, es pot accedir als atributs emmagatzemats a la base de dades, i si pel contrari, es cerca la informació, es pot trobar la localització d'aquesta en el mapa.

Aleshores qualsevol event o objecte associat a un SIG comparteix les següents característiques [8]:

- La seva localització concreta sobre la superfície terrestre.
- La descripció de la seva posició i forma geomètrica en dos o tres dimensions.
- Informació alfanumèrica que el qualifica i classifica.
- Relacions espacials amb els altres elements.

No obstant això, existeixen diverses tasques amb diferents graus de complexitat que poden ser resoltes per un SIG [8]:

- Localització: D'un event concret i els seus atributs.
- Condició: Compliment de les condicions imposades pel sistema.
- Tendència: Comparació de situacions temporalment o espacialment diferents.
- Ruta: Càlcul de rutes òptimes.
- Pautes: Detecció de pautes especials o concretes.
- Models: Generació automàtica de models basats en fenòmens simulats.

En el mòdul de geolocalització, QuesTInSitu *mobile*, implementarem l'ús de la **localització** d'events i les seves preguntes associades, així com el compliment d'unes certes **condicions** imposades pel sistema, com estar en una situació pròxima o haver respòs la pregunta anterior. Finalment, també s'intentarà fer un càlcul de **ruta**, o com a mínim, donar unes mínimes indicacions per arribar al següent event, ja sigui l'aplicació de geolocalització o QTIS.

2.2.2 Tècniques utilitzades

Tenint en compte que els SIGs representen els objectes d'un món real, podem dividir la seva abstracció de dues maneres [9].

- **Discretització:** Són tots aquells objectes fàcilment identificables en un mapa, com per exemple un edifici.
- **Valor continu:** Són tots aquells objectes de formes més abstractes, com modulacions del terreny o rius.

D'altra banda, també podem dividir la representació de les dades en dos models [9] amb avantatges i inconvenients segons l'ús que se li doni al SIG (veure *figura 3*):

- **Vectorial:** És un mètode que permet mantenir les característiques geomètriques de les figures. El interès de les representacions dels objectes és la gran precisió de la localització d'aquests sobre l'espai. Aquests fenòmens són discrets i estan vinculats a una fila de la base de dades que descriu els seus atributs, de manera que es poden fer servir per a la creació d'un mapa, comparacions o altres.

Per tal de modelar els objectes del món real, s'utilitzen els tres elements geomètrics bàsics: El punt (0D), la línia (1D) i el polígon (2D), així com les seves combinacions.

- **Raster:** És un mètode utilitzat en la representació de les dades per tal d'emmagatzemar, processar i visualitzar la informació geogràfica mitjançant la divisió en files i columnes formant una malla regular. Cada cel·la emmagatzema la seva localització i el seu valor temàtic. Les estructures raster no emmagatzemen de forma explícita la topologia, discretitzen la realitat contínua en unitats de dimensions concretes.

Com és de suposar en aquest model, a mida que augmentem la mida de les cel·les, menor és la precisió resolució de la representació del espai geogràfic. També, degut als seus avantatges, com el fàcil emmagatzemament d'imatges digitals o la seva facilitat en la superposició d'imatges és un model òptim per, per exemple, aplicacions tan comunes com *Google Maps* o *OpenStreetMap*.

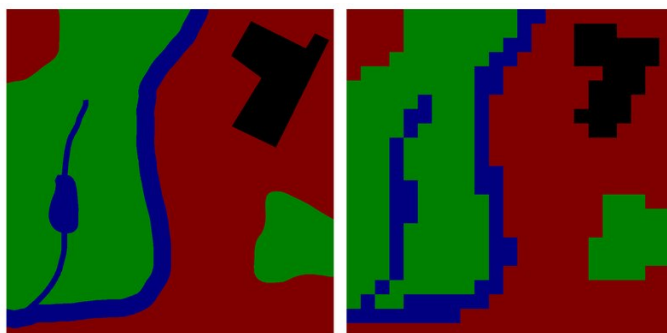


Figura 3. A l'esquerra, el model vectorial, a la dreta, el model raster [8].

2.2.3 Sistemes de coordenades

Aquest concepte té a veure amb el fet de dotar una posició en l'espai, en aquest cas la Terra, d'un valor únic amb el qual es pot identificar, sent molt útil en la tasca de desenvolupar bona part del projecte, ja que haurem de treballar amb sistemes de coordenades de diferents tipus, cadascun d'ells amb una precisió associada, depenent del model triat per tal de representar la superfície de la Terra com:

- Esfera: Esfera perfecta que no té en compte la forma el·líptica de la Terra.
- El·lipsoide: Cos de forma el·líptica que no té en compte la superfície de la Terra.
- Geoide: El·lipsoide rugós que referència la superfície desigual de la Terra.

A continuació, explicarem els diferents sistemes de coordenades utilitzats als diferents sistemes d'informació geogràfica [8] i que seran necessaris posteriorment:

Sistema de coordenades cartesianes

Per convenció, l'origen del sistema de coordenades cartesianes és el punt (0,0,0). Les coordenades cartesianes es defineixen per dos eixos (X,Y) quan es tracta de fixar punts en el pla i per tres (X,Y,Z) quan es vol fer això en l'espai. Es consideren eixos mútuament perpendiculars que es tallen en l'origen. Les coordenades d'un punt qualsevol vindran donades per les projeccions de la distància entre el punt i l'origen sobre cadascun dels eixos. S'ha d'entendre per

tant, que aquest model de representació, parteix de la base d'una Terra esfèrica o, en el millor dels casos en forma d'el·lipsoide i en cap cas ens assegura la precisió desitjada per la nostre aplicació i en general, per cap aplicació de posicionament. No obstant això, és la base del sistema que pren la Terra com un geoide.

Coordenades geogràfiques

Són els paràmetres que determinen la posició d'un punt de la superfície terrestre respecte a les línies de referència: l'equador terrestre i un meridià inicial o primer meridià, que prenem per conveni i que, generalment, és el meridià que passa per l'observatori de *Greenwich*, d'aquí el seu nom. Definim les magnituds que especifiquen longitud i latitud [10]:

- La longitud: La longitud d'un punt P de la superfície terrestre, és l'arc d'equador comprès entre el punt d'intersecció del meridià inicial amb l'equador i el punt d'intersecció del meridià local de P amb l'equador, comptat seguint el camí més curt i mesurat de 0° a 180° cap a l'est o cap a l'oest a partir del meridià inicial.
- La latitud: La latitud del punt P , és l'arc del meridià local de P comprès entre l'equador i P , mesurat de 0° a 90° a cada hemisferi a partir de l'equador (veure la figura 4).

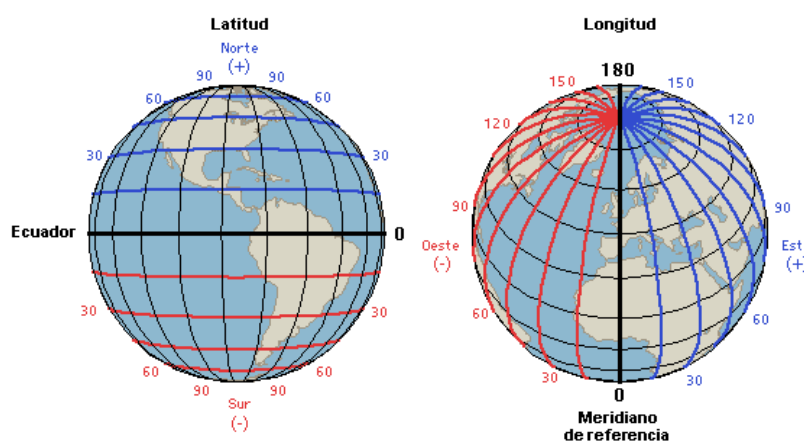


Figura 4. Representació física de la latitud i la longitud [22].

Degut a la forma d'el·lipsoide de la Terra, l'extensió d'un grau de longitud o de latitud és diferent en diferents punts geogràfics, per la qual cosa podem definir sistemes de coordenades més precisos que tenen en compte la desviació de l'el·lipsoide real terrestre respecte a l'esfera teòrica dels primers sistemes de coordenades. Per això, en els nous sistemes, definim la latitud com l'angle entre la vertical a l'el·lipsoide terrestre traçada pel punt i el pla de l'equador, i es manté la definició de longitud. La posició geogràfica d'un punt es completa en especificar l'altitud. La latitud i la longitud poden mostrar-se en tres formats equivalents (veure la figura 5):

- Graus Decimals, *Decimal Degree* (DD). Exemple: 41.333- 106..500.
- Graus:Minuts, *Degree Minute* (DM). Exemple: 41:20- 106:30.
- Graus:Minuts:Segons, *Degree Minute Second* (DMS). Exemple: 41:20:00- 106:30:00.

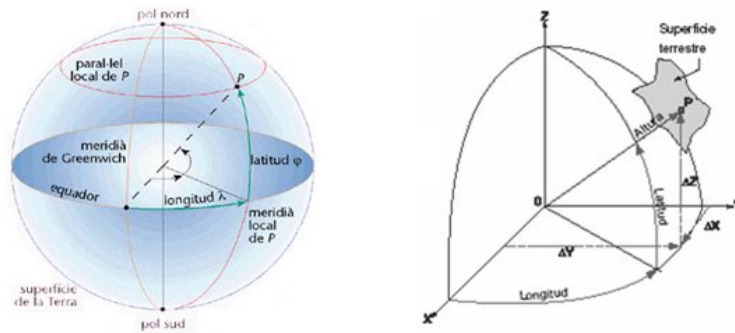


Figura 5. Pas de magnitud latitud i longitud, a l'ús addicional de l'altitud.

Veiem alguns altres models de representació:

Sistema de coordenades polars

Les coordenades (veure *figura 6*) polars es defineixen per un eix que passa per l'origen, anomenat eix polar. La primera coordenada és la distància entre l'origen i el punt considerat, mentre que la segona és l'angle que formen l'eix polar i la recta que passa per tots dos punts [10]:

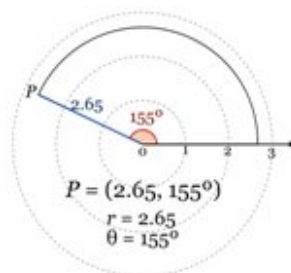


Figura 6. Coordenades polars [23].

Sistema de coordenades cilíndriques

El sistema de coordenades cilíndriques (veure *figura 7*) és una generalització del sistema de coordenades polars, al que s'afegeix un tercer eix de referència perpendicular als altres dos. La primera coordenada és la distància que hi ha entre l'origen i el punt, la segona és l'angle que formen l'eix i la recta que passa per tots dos punts, mentre que la tercera és la coordenada que determina l'altura del cilindre [10]:

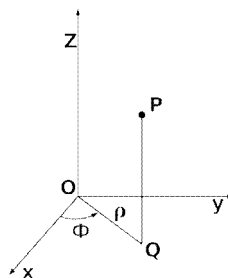


Figura 7. Coordenades cilíndriques [24].

Sistema de coordenades esfèriques

El sistema de coordenades esfèriques (veure *figura 8*) està format per tres eixos mútuament perpendiculars que es tallen en l'origen. La primera coordenada és la distància entre l'origen i el punt, essent les altres dues els angles que cal girar per arribar a la posició del punt [10]:

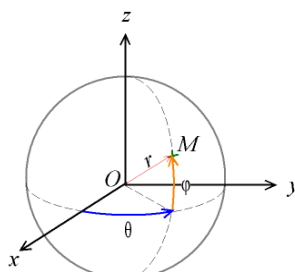


Figura 8. Coordenades esfèriques [24].

2.3 Mètodes per a la localització

Existeixen dos mètodes primaris per a l'obtenció de les coordenades geogràfiques d'un usuari depenent de la tecnologia utilitzada, que són els següents:

Geolocalització IP: [11] Mitjançant la IP del terminal, l'usuari es capaç de localitzar la ciutat on es troba ubicat aquell terminal. En instància final però, és molt probable que s'identifiqui el node de la central telefònica que dona connexió a Internet.

El avantatges d'aquest mètode són una alta disponibilitat a causa de ser una detecció per part del servidor, és a dir, que cap dada és facilitada per l'usuari, sinó que el servidor localitza la posició d'aquest internament.

No obstant això, els inconvenients són molt grans, ja que té una precisió a nivell de ciutat, és molt més costos i produeix, en ocasions, falsos positius.

La triangulació: [11] Bàsicament, consisteix a establir quatre distàncies de referència de la nostra posició respecte a quatre satèl·lits. Mitjançant aquestes quatre distàncies, podem afirmar amb seguretat que, en el punt on hi hagi una intersecció de les quatre línies és el punt a l'espai on ens trobem. Dins aquesta tipologia de localització, podem trobar tres variants:

- Receptor GPS: Requereix l'ús d'un receptor GPS, ja sigui integrat a un terminal mòbil o de forma externa. Per exemple: Tom-Tom.
- Wifi: No requereix l'ús d'un receptor GPS. Mitjançant una xarxa sense fils podem establir la posició del usuari de forma precisa en àrees urbanes i llocs de poca recepció de senyal GPS com interiors. Per exemple: Ipod Touch.
- Antena de telefonia mòbil, 3G, GPRS o altres: Tanmateix, es pot realitzar el mateix procediment mitjançant antenes de telefonia mòbil.

Està clar que aquests dos últims procediments, es realitzen mitjançant el intercanvi d'informació del terminal amb uns servidors. És a dir, tenint emmagatzemada aquella posició GPS de la xarxa Wifi o antena de telefonia mòbil, podem aproximar la situació d'un terminal sense necessitat del GPS, amb un error relatiu, però d'una manera molt ràpida (veure *figura 9*).

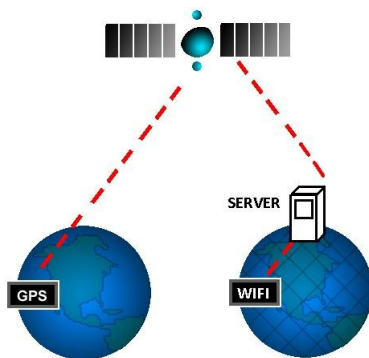


Figura 9. Model GPS vs model Wifi [25].

Amb la fusió d'aquests 3 models de triangulació, sorgeix el que s'anomena com A-GPS. A causa de les dificultats del GPS convencional de proporcionar posicions precises en condicions de baixa senyal, ja sigui dintre d'edificis, sota d'arbres o envoltats d'estructures d'altura que atenuen la senyal que rep el receptor, s'han fusionat els models descrits anteriorment. Atès que quan un receptor GPS s'encén és possible que no sigui capaç de descarregar la informació dels satèl·lits fins que no trobi una senyal clara, procés que en ocasions pot trigar més d'un minut, sorgeix la idea de transferir aquestes dades mitjançant un mitjà totalment diferent per tal d'agilitzar aquest procés:

Un A-GPS [12] per tant, pot solucionar aquests problemes de dues maneres:

- Mitjançant un servidor d'assistència en línia: Aquest model té la necessitat de tenir una connexió de dades activa en el moment de la comanda mitjançant l'ús d'una xarxa GPRS, 3G, GSM, Wifi o similar. Per tant, només mitjançant la torre de telefonia mòbil podem esbrinar la ubicació aproximada de l'usuari. A més a més, el servidor extern proveeix de dades orbitals al telèfon, fent-lo apte per a la connexió a satèl·lits que d'altra manera no podria.
- Sense el servidor d'assistència en línia: Aquest model fa servir les mateixes dades que l'anterior, però descarregades prèviament, de forma manual o automàtica quan s'ha tingut connexió. Aleshores, aquest fitxer s'emmagatzema, i té una validesa d'uns dies, fins que la informació esdevé obsoleta. És a dir, obté la seva posició aproximada mitjançant torres de telefonia mòbils, i mitjançant el fitxer obtingut prèviament, fa la connexió a satèl·lits que d'altra manera no podria.

En ambdós casos, aquests sistemes GPS, utilitzen les dades obtingudes del servidor extern per combinar-les amb la informació d'una antena de telefonia mòbil i esbrinar quins satèl·lits té sobre ell. A més a més, el mòdul GPS obtindrà a la vegada, informació sobre els satèl·lits que disposa el nostre terminal, per tal de fer la posada en marxa molt més ràpida i precisa (veure figura 10). Alguns terminals funcionen tant en un model com a l'altre depenent de si la connexió de dades està activa o no.

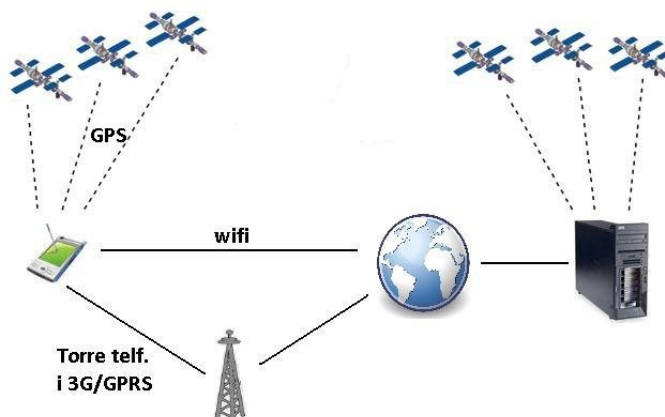


Figura 10. Diagrama de funcionament del A-GPS.

2.4 Discussió i anàlisi de tres eines SIG basades en la localització del usuari

A continuació, es detallaran tres eines SIG que han estat analitzades per tal d'entendre millor les avantatges i limitacions que poden oferir. D'aquest petit estudi, es podrà extreure quines funcionalitats poden ser adequades per al mòdul que es desitja implementar.

2.4.1 *M*Scape

Ha estat la primera eina que s'ha provat en una experiència real amb QTIS per Barcelona. Per tant, ha estat la primera en la que ens hem fixat per tal d'establir una xarxa de projectes similars, com a mínim conceptualment, al nostre. No obstant això, *MediaScape*, va molt més enllà.

*M*Scape (veure figura 11) és una eina d'escriptori que té un mòdul per a mòbil que permet crear rutes interactives utilitzant dispositius mòbils. Aquesta plataforma és bastant flexible en el sentit que incentiva el seu ús a desenvolupadors a crear alguna cosa més que jocs, com guies d'informació de punts d'interès, informació relacionada amb llocs concrets i d'altres. *M*Scape realitza una localització GPS amb l'usuari del videojoc i fa aparèixer, en el joc, events per a que l'usuari interactuï amb ells, i el joc en general, movent-se de lloc en lloc.

Aquest software [2] és propietari de la companyia *Hewlett Packard*, per tant no és lliure. Està pensat per a dispositius mòbils 3G amb connexió per satèl·lit, es troba disponible per a la seva descarrega i ús únicament sobre el sistema operatiu per a mòbils: *Windows Mobile*. Això és un inconvenient molt gran, ja que el software està subjecte a una companyia i sobretot, a un únic tipus de plataforma, fet que limita el seu ús a aquest. Seguint en aquesta línia, podem suposar algun inconvenient més, relacionat amb la connexió GPS a interiors, on poden existir zones d'ombra on la cobertura per satèl·lit sigui deficient i l'aplicació no pugui rendir com s'esperava. En aquest sentit es molt útil la connexió A-GPS, que guarda relació amb l'obtenció de mapes i dades mitjançant 3G, si la connectivitat GPS és limitada. Aquest sistema actualment, és el implementat per l'iPhone 3G, que explicarem més endavant.

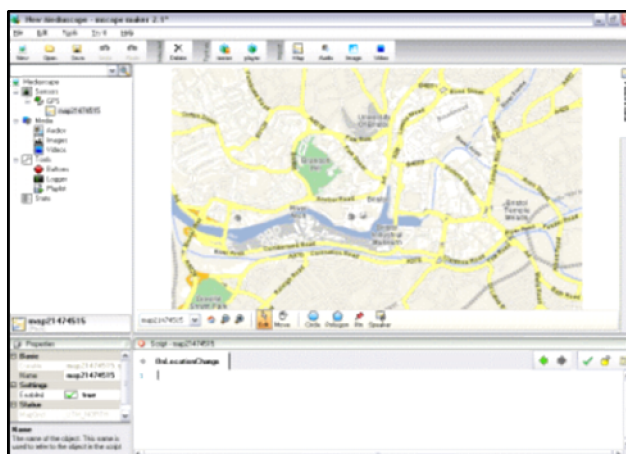


Figura 11. Captura del editor de mapes de *M*Scape.

També s'utilitza per a crear experiències interactives basades en vídeo, àudio, imatges o text, on l'usuari mitjançant el moviment activa els events que a de rebre associats a unes coordenades geogràfiques, justament allò que es vol implementar en aquest projecte. Tanmateix, l'aplicació sobre QTIS funciona de la següent manera: L'aplicació mitjançant el dispositiu on es troba instal·lada et va controlant la teva posició cada x segons. Un cop que entres en una de les zones dels mapes que té editats, sorgeix una alerta i et retorna el feedback que hagi configurat.

2.4.2 Foursquare

Foursquare [13] és una xarxa social que ha aconseguit en molt poc temps una massa crítica d'usuaris, en gran part per l'estratègia utilitzada per a fomentar l'activitat d'aquests. A diferència de l'objectiu de comunicació que persegueixen *Facebook* i *Twitter*, *Foursquare* està basat en la geolocalització web aplicada a les xarxes socials.

Bàsicament, permet als seus usuaris avaluar locals o emplaçaments compartint recomanacions i opinions sobre aquests amb els seus contactes, acció que s'anomena *check-in*. Per fer-ho, implementa el mètode de localització web explicat anteriorment i que s'implementarà per a l'aplicació *QTIS mobile*. D'aquesta manera, mitjançant totes aquelles eines de que disposa el terminal (GPS, torres de telefonia, 3G) permet determinar la posició exacte de l'usuari. Seguidament, l'usuari serà l'encarregat d'introduir la informació de l'emplaçament si no ha estat avaluat anteriorment, o fer constar la seva opinió si ja ho ha estat.

No obstant això, l'aplicació està dotada amb un sistema de puntuació que fomenta l'ús en forma de videojoc mitjançant la competició. Així, l'usuari rep punts i/o medalles cada vegada que visita llocs determinats (un bar, una tenda, un aeroport, un hotel, un museu...). A més a més, també es reben bonificacions per donar d'alta emplaçaments no registrats o valoracions a l'estranger.



Figura 12. Captura de les opinions d'un local a *Foursquare* [26].

Cal destacar que l'ús del sistema de geolocalització no només es limita a la valoració i puntuació d'emplaçaments; la xarxa social es capaç de mostrar publicitat d'una manera selectiva dependent de la ubicació del usuari.

2.4.3 GoogleLatitude

Aquesta eina desenvolupada per *Google*, és conceptualment molt senzilla. Permet, previ consentiment del usuari, localitzar als teus amics registrats al servei, ja sigui mitjançant un dispositiu mòbil o no. D'aquesta manera les seves posicions són situades sobre un mapa interactiu que amplia funcionalitats, ja que a més de veure la situació dels contactes, que compartiran un status, també disposaran d'un sistema de trucades, missatges o xat per a comunicar-se.

És tan fàcil com ingressar al servei amb una compte de *Google* i de forma automàtica es mostren tots aquells contactes associats a la compte amb la finalitat que puguin ser convidats i es permeti compartir la ubicació amb ells. Aleshores, es pot executar mitjançant dues opcions; el dispositiu mòbil o l'ordinador. Si es fa amb el primer es necessària una connexió 3G o GPRS, amés de l'aplicació *GoogleMaps* [3].

2.4.4 QuesTInSitu

Fins ara QuesTInSitu per tal de proveir-se d'algunes de les funcionalitats esmentades a les aplicacions anteriors, feia ús de Mscapce com mòdul de localització per a dispositius mòbils. Per tant, existeix una independència entre ambdós mòduls; el de creació de rutes, i el de localització (veure la *figura 13*). QTISM permetrà resoldre aquesta independència d'una aplicació de *software* propietari que tampoc s'ajusta a les necessitats de QTIS. Finalment, es podrà incorporar el *software* de localització al existent.

A continuació, s'explicarà les diferents tecnologies i funcionalitats que caldria incorporar a QTISM.

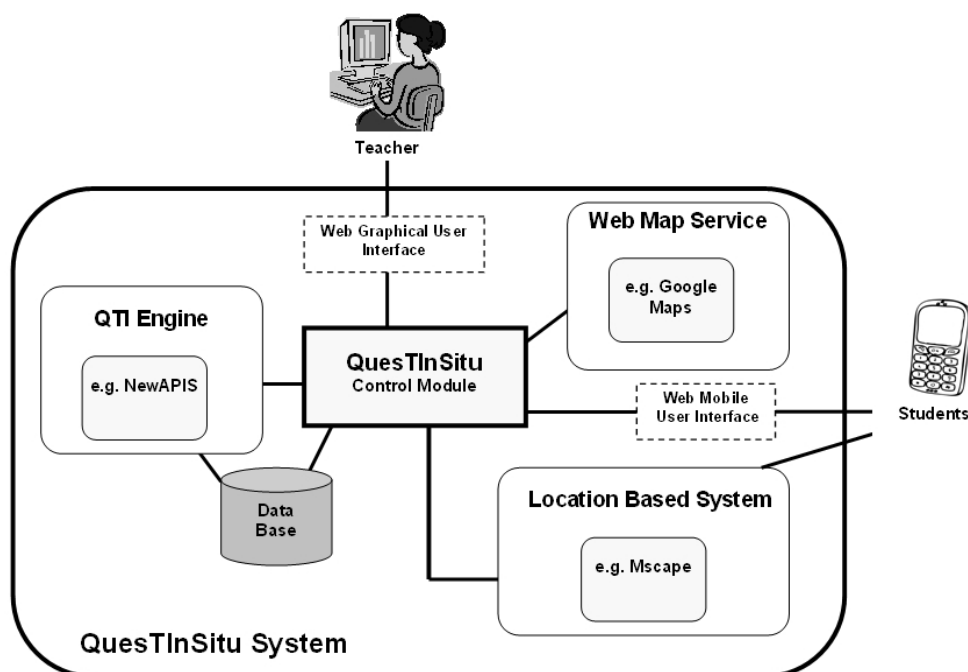


Figura 13. Esquema de la connexió de les diferents parts de QTIS [27].

2.4.5 Discussió de les funcions de QTISM

QuesTInSitu en la seva versió per a la creació de rutes fa servir l'API de *GoogleMaps* com a element de similitud a les aplicacions anteriors. No obstant, li manquen algunes de les característiques més importants de cara a augmentar la seva funcionalitat, que sí formen part de les eines anterior.

Una d'aquestes característiques és l'execució en terminals mòbils. Aquesta característica és molt important per tal de maximitzar els usos i els contextos d'utilitat de l'eina. Si observem les aplicacions anteriors, totes permeten aquesta execució sobre plataformes mòbils, inclús sobre diferents tipus de sistemes operatius per a terminals. Al ser una aplicació relacionada amb la localització, és natural que sigui executable sobre dispositius de mà. Aquest mateix requisit, dona peu a la necessitat d'una funció de localització, que ja incorporen tant *Foursquare* com *GoogleLatitude*. Sens dubte és una de les millors funcionalitats que podria oferir QuesTInSitu, així es podria detectar la posició real de l'usuari en un context educatiu realitzant l'activitat programada prèviament.

Tanmateix, es podria extreure l'essència de la web 2.0, que són les xarxes socials i més concretament, la comunicació entre usuaris, tal com ocorre a les aplicacions abans explicades. Per això, en un segon pla, pot ser interessant permetre que els usuaris puguin mantenir una comunicació durant l'activitat, d'aquesta manera s'incrementa l'experiència d'aquests.

Per últim, la necessitat d'un mapa com el que s'incorpora a *GoogleLatitude*, és pràcticament indispensable, així com un panell de direccions que sigui capaç de conduir l'usuari des de la seva posició actual fins a la fita a trobar.

2.5 L'HTML 5

Després d'estudiar d'una manera més acurada les possibilitats d'interacció amb el GPS o A-GPS del terminal, vaig poder comprovar que la cinquena revisió d'HTML, HTML5, incorporava unes API's que podien resultar molt profitoses per a la tasca d'obtenció de les coordenades d'un terminal mòbil. Per això, serà necessari que parlem del context que envolten aquestes API's, la filosofia o intenció d'aquesta cinquena revisió i les limitacions de la tecnologia.

Aquesta especificació fou publicada per W3C (*World Wide Web Consortium*) l'any 2009. W3C es un consorci internacional que produeix recomanacions per a la *World Wide Web*. Aquest consorci està format per URL (*Uniform Resource Locator*), HTTP (*HyperText Transfer Protocol*) i HTML (*HyperText Markup Language*), que són les principals tecnologies utilitzades per la Web [14].

HTML 5 es la cinquena revisió del llenguatge HTML. Aquesta revisió especifica, per una banda, el llenguatge HTML vist fins ara, una nova variant HTML5 i l'XHTML5. Aquest fet ha implicat el desenvolupament conjunt dels dos llenguatges en paral·lel per part del consorci regulador W3C.

Algunes de les millores més importants d'aquesta especificació es troben a l'annex 2 apartat *Millores d'HTML5*. A continuació però, es començarà a detallar aquella més important de cara al projecte, que ha estat la *Geolocation API Specification*.

2.5.1 *Geolocation API Specification*

Una de les API's més importants i que analitzarem més exhaustivament per a comprovar la seva eficiència, serà l'API de geolocalització anomenada en anglès *Geolocation API Specification*

[15]. Aquesta API bàsicament, permet a l'aplicació web obtenir la posició geogràfica del usuari mitjançant un conjunt de funcions ja implementades que permeten llençar una petició (*quest*) al navegador per tal que aquest ens retorni la informació desitjada. Podríem dir, que el navegador li demana al terminal que li faciliti la seva posició de la manera que pugui i amb la precisió que es pugui permetre. D'aquesta manera, es pot obtenir posicions, potser no massa acurades, d'ordinadors connectats a xarxes Wifi, terminals amb accés a Internet sense mòdul GPS, com *Ipods* i similars, i terminals mòbils amb xarxa 3G i mòdul GPS o A-GPS.

No obstant això, cal destacar que malgrat aquesta API i moltes altres han esdevingut un estàndard adoptat pel consorci W3C, han estat desenvolupades conceptualment per *Google*, dins un projecte de software lliure anomenat *Gears* que ja formen part d'HTML5, avui dia corrent en la seva versió 0.5. Google ho descriu així:

'Gears és un projecte de software lliure que permet afegir noves funcions als navegadors web per tal de crear aplicacions més potents.'

2.5.2 Funcionament

El funcionament és conceptualment senzill. Primerament, l'algoritme detecta si el navegador té implementada la API de geolocalització, si es així, crida a la funció necessària per a obtenir les coordenades actuals. Malgrat no coneixem el procediment intern que es realitza a nivell de xarxa, el podem aproximar segons la tipologia d'accés del terminal:

Si accedim al servei mitjançant un ordinador connectat a una xarxa Wifi, a través d'aquesta obtindrem la seva posició relativa. Aquest procediment es realitza utilitzant una base de dades de *Google* (*Google Location Services*) que conté gran quantitat d'informació relativa a les posicions de les xarxes sense fils d'arreu del món (veure *figura 14*). La informació d'aquesta base de dades ha estat recopilada mitjançant un LIDAR, *Laser Imaging Detection and Ranging* instal·lat en el *Google-car* quan recorria les ciutats per tal d'escanejar punts d'accés a Internet i antenes de telefonia.

La informació que s'envia és la següent:

- L'adreça IP del teu ordinador.
- Informació sobre els punts d'accés sense fils més propers.
- Identificador de client aleatori assignat per Google que caduca a les dues setmanes.



Figura 14. Diagrama del funcioament de l'API de geolocalització.

D'aquesta manera, podem aproximar la posició de la nostra xarxa sense fils en comptes de la del l'equip. De la mateixa manera, qualsevol aparell similar per exemple: *IpodTouch* o *Ipad Wifi*; també realitza el mateix procediment.

Si accedim al servei mitjançant un dispositiu 3G o GPRS, l'aproximació de les coordenades es realitza triangulant la posició de l'antena 3G. Tenint en compte que les coordenades de les antenes de telefonia són conegudes i es poden emmagatzemar en una base de dades i depenent de la intensitat de la senyal rebuda, es pot deduir la posició del terminal. Si a més a més, disposem d'un dispositiu amb GPS o A-GPS, aquest rebrà la petició i ens dotarà d'unes coordenades de major precisió [17].

Si l'objectiu és fer una aplicació de màxima accessibilitat per a tot tipus de plataformes, no podem oblidar que amb només la tecnologia 3G o GPRS ja seria possible fer tot el conjunt d'operacions necessàries per a que, tant el mòdul de localització com l'aplicació QTIS, funcionessin d'una manera eficient. Seria necessari per tant, esbrinar si també es pot potenciar l'aplicació amb el mínim ús de diferents tecnologies, que no sempre s'incorporen als dispositius mòbils.

2.5.3 Privacitat

Gairebé tots els navegadors poden informar a les web que visitem on ens trobem físicament, per tal de facilitar aquella informació més rellevant per a nosaltres. No obstant això, aquesta API d'HTML5 presenta dubtes degut a la major oportunitat de seguiment que ofereix i a la gran quantitat de dades que pot ser capaç de recol·lectar i emmagatzemar mentre l'usuari està connectat. És a dir, el problema existent no són les vulnerabilitats que pot oferir el sistema, que no les té, sinó la facilitat que tindran altres persones o organitzacions d'accedir sense autorització a grans quantitats de dades personals, abans inexistents. Naturalment, aquestes dades seran oferides per la pròpia persona fent ús de les diferents funcionalitats que ens brinda aquest nou estàndard [15].

D'altra banda, s'ha de considerar que a major desenvolupament d'una tecnologia d'aquest tipus, sempre hi ha un major perill i és per això, que el consorci regulador W3C estableix que aquesta API no pot fer ús de la seva funcionalitat a no ser que l'usuari ho permeti expressament [15]:

'Cap agent no autoritzat expressament per l'usuari pot transmetre informació de localització a llocs web.'

2.5.4 Mètodes

L'API de geolocalització permet a l'aplicació web:

- Obtenir la posició actual de l'usuari, fent servir el mètode *getCurrentPosition*.
- Veure si la posició del usuari ha canviat en el temps, fent servir el mètode *watchPosition*.
- Obtenir la última posició coneguda, fent servir la propietat *lastPosition*.

Aquesta API pot proporcionar la posició estimada del usuari fent servir diferent nombre de recursos, és a dir, pot fer servir el mòdul GPS del terminal, si en té, o fer servir proveïdors basats en servidors com els explicats anteriorment. Per a especificar quin proveïdor volem fer servir en els dos mètodes descrits, cal modificar el paràmetre *PositionOptions*. A més a més, és possible implementar el nostre propi proveïdor de localització mitjançant el *Geolocation API network protocol* [15].

Geolocation Interface

L'objecte *Geolocation* [15] és utilitzat pels *scripts* per determinar, mitjançant la crida, la informació de localització associada al dispositiu. Aquesta informació es adquirida mitjançant un algoritme específic, que després citarem, que crea un objecte amb les dades corresponents.

A més a més, els objectes que implementen la interfície del navegador també han d'implementar la interfície del *NavigatorGeolocation* com s'observa a la *figura 15 i 16*. Aleshores, una instància de *NavigatorGeolocation* s'obté mitjançant l'ús de mètodes específics en una instància del *Navigator*.

```
interface Geolocation {
    void getCurrentPosition(in PositionCallback successCallback,
        in optional PositionErrorCallback errorCallback,
        in optional PositionOptions options);

    long watchPosition (in PositionCallback successCallback,
        in optional PositionErrorCallback
        errorCallback,
        in optional PositionOptions options);

    void clearWatch(in long watchId);
};
```

Figura 15. *Interface NavigatorGeolocation i Geolocation* [15].

```
interface PositionCallback {
    void handleEvent(in Position position);
};

interface PositionErrorCallback {
    void handleEvent(in PositionError error);
};
```

Figura 16. *Interface PositionCallback i PositionErrorCallback* [15].

Degut al poc ús del mètode *getCurrentPosition*, la documentació relativa a la seva implementació es pot trobar a l'annex 2 a l'apartat *getCurrentPosition*.

watchCurrentPosition

El mètode *watchCurrentPosition* [15] pot tenir un, dos o tres arguments per tal d'obtenir la ubicació actual del dispositiu. Si ha tingut èxit, el *successCallback* serà invocat per tal de que mitjançant la operació *handleEvent*, retorni a l'objecte que l'ha invocat la posició actual del dispositiu. Si per contra no ha pogut esbrinar la ubicació, serà invocat el *errorCallback* amb un objecte *PositionError* nous per tal de reflectir la raó de l'error.

Fins aquí el procediment és pràcticament idèntica al mètode *getCurrentPosition*. Però aquest mètode aleshores, continua monitoritzant la posició del dispositiu i invocant de forma reiterada la crida quan la posició canvia. Aquesta operació continuarà fins que es cridi el mètode *clearWatch* amb l'identificador corresponent.

La implementació del mètode *watchCurrentPosition* ha d'executar el següent nombre de passos enumerats a continuació:

1. Realitzar el següents passos previs:
 - a. Si *SuccessCallback* té un valor *null* es considera que la conversió a *PositionCallback* ha fracassat i no s'efectuaran la resta de passos.
 - b. Si el paràmetre *PositionOptions* està present i el seu atribut *maximumAge* està definit com un valor no negatiu, s'assignarà aquest valor a la variable interna *maximumAge*. Si està definit com un valor negatiu o no està especificat, s'assignarà a la variable interna un zero.
 - c. Si el paràmetre *PositionOptions* està present i el seu atribut *timeout* està definit com un valor no negatiu, s'assignarà aquest valor a la variable interna *timeout*. Si està definit com un valor negatiu, s'assignarà a la variable interna un zero, i si no està especificat un infinit.
 - d. Si el paràmetre *PositionOptions* està present i el seu atribut *enableHighAccuracy* està definit, s'assignarà aquest valor a la variable interna *enableHighAccuracy*. En qualsevol altre cas aquesta variable estarà a *false*.
2. Si un objecte *Position* guardat a la memòria cache té una edat que no supera el valor de la variable *maximumAge* i està disponible, s'invocarà el *successCallback* per tal d'introduir-lo com un paràmetre.
3. Es registren els events rebuts del sistema que indiquen una modificació de la posició del dispositiu (p.e. escoltant els canvis en la senyal de telefonia).
4. Iniciar una operació d'adquisició d'ubicació, amb la possibilitat de prendre el valor de la variable *enableHighAccuracy*.
5. Realitzar els següents passos en l'adquisició:
 - a. Si en timer encara no esta funcionant, s'activa un timer que després d'un nombre de milisegons especificats, pel valor de la variable *timeout*. Seguidament s'invocarà el *errorCallback* amb el nou objecte *PositionError* i el seu atribut *code* tindrà el valor del *timeout*, i saltarà al pas 6.
 - b. Si l'operació d'adquisició es realitza amb èxit donant una nova posició abans que el *timeout* expiri, s'han de realitzar els següents 2 passos:
 - i. Cancel·lar el *timer* pendent. Cal destacar que el *timer* serà restablert quan l'algoritme salti enrere per tornar a fer els passos d'adquisició.
 - ii. Si la nova posició difereix prou de l'anterior, s'invoca el *successCallback* amb el nou objecte *Position* que representa el resultat de la operació d'adquisició.
 - c. D'altra manera, si l'adquisició de la localització reporta un error abans que el *timeout* expiri, s'invoca al *errorCallback* amb l'objecte *PositionError* nou i el seu atribut *code* tindrà el valor de *Position_Unavailable*.
6. Espera per a rebre un nou event del sistema. Quan el rep, salta als passos d'adquisició altre cop.

Cal destacar que tant el mètode *getCurrentPosition* com el *watchCurrentPosition*, la seva implementació mai invoca el *successCallback* sense haver obtingut el permís del usuari per a fer-ho. És més, la implementació d'això, hauria d'obtenir el permís del usuari abans tan sols de realitzar qualsevol dels passos descrits anteriorment. D'altra banda, en el cas que l'usuari denegi el permís, el *errorCallback* serà invocat amb el seu atribut *code* amb valor *Permission_Denied* com si un error s'hagués produït. Aquest temps que es consumeix en obtenir el permís del usuari no està inclòs en el *timeout*, l'atribut del paràmetre *PositionOptions*, sinó que s'aplica només a l'operació d'adquisició.

El mètode *clearWatch* té un sol argument. Quan aquest mètode és cridat, comprova el valor de l'argument *watchId* que ha rebut per tal de veure si correspon a algun procés començat anteriorment. Si correspon, el procés amb aquesta ID s'atura, i si no correspon, el mètode no realitza cap acció.

PositionOptions Interface

Els mètodes descrits anteriorment, accepten el paràmetre *PositionOptions* [15] com a tercer argument de la seva crida. Tots els seus atributs són opcionals i s'han d'especificar a l'hora de crear-la tal com a la figura 17.

```
interface PositionOptions {
    attribute boolean enableHighAccuracy;
    attribute long timeout;
    attribute long maximumAge;
};
```

Figura 17. *Interface Position Options* [15].

L'atribut *enableHighAccuracy* pot dotar a l'aplicació d'un índex de millors resultats però a costa d'un temps de resposta superior i un major consum d'energia. La finalitat d'aquest atribut es poder donar servei a aplicacions que no requereixen gran precisió i així evitar l'ús de proveïdors de geolocalització que poden consumir una gran quantitat d'energia, com un GPS. Això és molt interessant per a dispositius alimentats amb bateries, i que sense dubte s'haurà de prendre en consideració a la hora de la implementació. Finalment, si el paràmetre *PositionOptions* no apareix als mètodes, aquest atribut serà *false* per defecte.

La resta de la documentació relativa a l'ús dels atributs del mètode *PositionOptions Interface* es troba a l'annex 2 a l'apartat que porta el mateix nom.

Position Interface

Aquesta *interface* [15] és la portadora de la informació geogràfica retornada per la API tal com s'observa a la figura 18.

```
interface Position {
    readonly attribute Coordinates coords;
    readonly attribute DOMTimeStamp timestamp;
};
```

Figura 18. *Interface Position* [15].

L'atribut *coords* conté les coordenades geogràfiques amb la seva precisió associada, així com l'altitud i la velocitat. I l'atribut *timestamp* és el moment en que la posició ha estat adquirida.

PositionError Interface

L'atribut *code* [15] retorna un nombre enter corresponent a la següent llista de significats vistos a la figura 19:

- Valor 1: PERMISSION_DENIED. Indica que l'usuari no dona permís per executar la localització a l'API.

- Valor 2: PERMISSION_DENIED. Indica que la posició no pot ser determinada a causa de la fallada dels proveïdors de localització.
- Valor 3: TIMEOUT. Indica que el temps d'espera ha finalitzat abans d'adquirir la localització.

L'atribut *message* retorna un missatge que descriu detalladament l'error que ha trobat.

```
interface PositionError {
  const unsigned short PERMISSION_DENIED = 1;
  const unsigned short POSITION_UNAVAILABLE = 2;
  const unsigned short TIMEOUT = 3;
  readonly attribute unsigned short code;
  readonly attribute DOMString message;
};
```

Figura 19. *Interface PositionError* [15].

2.5.5 Compatibilitat del navegador

Per últim, cal esmentar el rang de compatibilitats que té l'API de geolocalització, que a mesura que passa el temps, serà implementada per tots els navegadors. Actualment però, té una compatibilitat amb els següents navegadors [18]:

- Safari 4.0+.
- Chrome 5.0+.
- Firefox 3.5+.
- Opera 10.6+.

I els sistemes operatius per *Smartphones* com:

- Android 2.0+.
- iPhone 3.0+.
- BlackBerry OS 6.

Internet Explorer no suporta l'API de geolocalització de forma directa. No obstant això, es pot executar mitjançant l'aplicació *GoogleGears* que, com hem comentat anteriorment, és de codi obert i funciona sobre sistemes operatius com *Windows*, *Mac*, *Linux*, *Windows Mobile* o *Android*. Aquesta és una de les funcions que proporciona aquest *software*, que no és exactament la mateixa que l'API de geolocalització, però té la mateixa finalitat. Malgrat això, la opció més interessant és l'ús d'aquesta API i del format de portal web per a la seva execució.

2.6 Mètodes de càlcul de la distància entre dos punts de la Terra

Durant les gimcanes geolocalitzades establertes mitjançant QTIS sorgeix la necessitat d'esbrinar la distància entre la posició del usuari i la del objectiu o event a seguir. El motiu és molt senzill: si l'usuari sap que ha d'anar a una posició concreta del mapa identificada, per exemple, per les coordenades (40.420088, -3.688810), serà gairebé impossible que pugui arribar a aquesta posició específica i exacte per tal de rebre l'event i en conseqüència no el rebrà.

Cal aleshores, establir un radi d'uns metres al voltant de l'event a cercar, per poder observar si la posició del usuari es similar a la ubicació del event i així, donar per vàlida la posició d'aquest i permetre que rebi l'event per pantalla. Per tal de solucionar aquest problema, es poden posar en pràctica diferents teoremes de càlcul de distàncies. A continuació, s'explicaran aquells que s'han valorat amb l'objectiu final de ser implementats, així com les avantatges o inconvenients que oferien sobre els altres.

Tanmateix, cal destacar que en cap mètode es tenen en compte les diferències d'elevació, així com les irregularitats del terreny, ja que aquests factors poden augmentar enormement la complexitat de la implementació de l'algoritme triat i no aporten una precisió addicional notable.

2.6.1 El teorema de Pitàgores

El teorema de Pitàgores estableix que en un triangle rectangle, el quadrat de la longitud de la hipotenusa és igual al sumatori de les longituds al quadrat dels dos catets. Si es considera a i b com a catets i c com la hipotenusa, s'obté [19]:

$$c^2 = a^2 + b^2$$

Prenent una distància menor a 20 Km i les coordenades cartesianes del punt $P_1(x_1, y_1)$ i $P_2(x_2, y_2)$ es pot aplicar el teorema:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Cal destacar però que la distància, degut a la curvatura de la Terra, tindrà uns errors de l'ordre de:

- Menys de 30 metres per a latituds inferiors a 70 graus.
- Menys de 20 metres per a latituds inferiors a 50 graus.
- Menys de 9 metres per a latituds inferiors a 30 graus.

Aquests errors tenen en compte els efectes de convergència dels meridians i la curvatura dels paral·lels en la distància en un pla expressada en les mateixes unitats que les coordenades. A més a més, si les coordenades no estan en un sistema de referència cartesià, el cost computacional de transformar-les a cartesianes i utilitzar a continuació el model de terra plana de Pitàgores pot excedir el cost computacional de fer servir un altre model més precís, fet que ens suposa un gran inconvenient, ja que les coordenades que rebem de la funció *getPosition* no són en forma cartesiana.

Vegem com seria el pseudocodi:

$$d = \text{sqrt}((X2 - X1)^2 + (Y2 - Y1)^2)$$

2.6.2 Fórmula en coordenades polars per a un model de Terra plana

Aquesta nova forma permet reduir els errors màxims respecte l'ús del teorema de Pitàgores en latituds elevades (allà on la curvatura de la Terra és més pronunciada) o grans distàncies. Realment, és una opció molt interessant per a realitzar el càlcul necessari i la seva implementació és força trivial. Vegem la fórmula en coordenades polars [19]:

$$R \cdot \sqrt{\left(\frac{\pi}{2} - x_1\right)^2 + \left(\frac{\pi}{2} - x_2\right)^2 - 2 \cdot \left(\frac{\pi}{2} - x_1\right) \cdot \left(\frac{\pi}{2} - x_2\right) \cdot \cos(y_2 - y_1)}$$

Per exemple, s'obté el mateix error a 80 graus de latitud quan la distància és de 30 Km, que a 82 graus de latitud quan la distància és de 15 Km, fins hi tot quan la latitud és de 84 graus amb distància de 9 Km. No obstant això, els resultats acostumen a ser millors que amb l'aproximació pitagòrica donat que a 88 graus de latitud i amb una distància de 20 Km, l'error pot ser de com a molt 20 metres.

Vegem com seria el pseudocodi:

```

a = pi/2 - lat1
b = pi/2 - lat2
c = sqrt(a^2 + b^2 - 2 * a * b * cos(lon2 - lon1))
d = R * c
    
```

2.6.3 Llei dels cosinus per trigonometria esfèrica

Tot i que aquesta fórmula és matemàticament exacte [19] i podria constituir una opció viable per calcular la distància entre coordenades, resulta poc fiable per distàncies petites. Aquest fet s'ha de tenir molt en compte per a la funció implementada, ja que l'arc del cosinus presenta un mal comportament pels nostres propòsits prop del origen. Això és pot veure a la figura 20.

cos (5 graus)	0.996194698
cos (1 grau)	0.999847695
cos (1 minut)	0.9999999577
cos (1 segon)	0.999999999882
cos (0.05 segon)	0.9999999999971

Figura 20. Taula de rang de variació de la fórmula cosinus.

Si es considera l'ús d'aquestes operacions amb la limitació de 7 xifres significatives després de la coma, es pot observar que no es possible diferenciar el cosinus d'unitats més petites que un minut, fet que perjudica el resultat final.

Vegem el pseudocodi:

```

a = sin(lat1) * sin(lat2)
b = cos(lat1) * cos(lat2) * cos(lon2 - lon1)
c = arccos(a + b)
d = R * c
    
```

2.6.4 La fórmula de Haversin

La fórmula de Haversin [20] és una de les equacions més importants per a la navegació astronòmica, ja que permet calcular la distància entre dos punts de la Terra coneixent les seves longitud i latitud. No obstant això, com sabem, la Terra no té forma d'el·lipsoide ni d'esfera. La Terra té una forma pròpia anomenada geoid, ja que en un el·lipsoide o esfera no es té en

compte depressions, mars, rius i muntanyes, és a dir, el seu relleu diferenciat. A la pràctica però, es prendrà com a referència la forma d'una esfera per a facilitar els càlculs [21]:

$$\text{haversin} \left(\frac{d}{R} \right) = \text{haversin} (\varphi_1 - \varphi_2) + \cos (\varphi_1) \cdot \cos (\varphi_2) \cdot \text{haversin} (A\lambda)$$

Sent:

- R és el radi de l'esfera, en aquest cas de la Terra.
- d és la distància entre dos punts.
- φ_1 és la latitud del punt 1 en radiants.
- φ_2 és la latitud del punt 2 en radiants.
- $A\lambda$ és la diferència de la longitud dels punts 1 i 2 en radiants.

Per tal d'aïllar la distància d, cal arribar a la forma d'equació que ve a continuació. Per fer-ho, s'ha de considerar que:

$$\text{haversin} \left(\frac{d}{R} \right) = h$$

Així es podrà aïllar i esbrinar la distància d mitjançant la fórmula de *haversin* inversa o la funció *arcsin*:

$$d = R \cdot \text{haversin}^{-1}(h) = 2R \cdot \arcsin(\sqrt{h})$$

Si *haversin* té la igualtat trigonomètrica següent:

$$\text{haversin}(\theta) = \sin^2 \left(\frac{\theta}{2} \right) = \left(\frac{1 - \cos(\theta)}{2} \right)$$

I ho substituïm a la fórmula de Haversin, obtenim:

$$\sin^2 \left(\frac{d}{2R} \right) = \sin^2 \left(\frac{\varphi_1 - \varphi_2}{2} \right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2 \left(\frac{A\lambda}{2} \right)$$

Si ara considerem que $a = \sin^2 \left(\frac{d}{2R} \right)$, obtenim:

$$a = \sin^2 \left(\frac{\varphi_1 - \varphi_2}{2} \right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2 \left(\frac{A\lambda}{2} \right)$$

I per consegüent l'inversa de Haversin:

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{(1-a)})$$

Quan això es programa es fa servir la funció *atan2()* que deriva de la funció *atan()* per tal de simplificar aquells casos en que el denominador és zero i poden portar a conflicte, així d es pot tractar sense ambigüitats en tots els quadrants.

Finalment només caldrà fer:

$$d = R \cdot c$$

S'ha d'anar molt en compte quan es fan servir aquestes fórmules, ja que s'ha d'assegurar que h no excedeix 1, és a dir la distància d és correcte si $0 \leq h \leq 1$. S'ha de destacar però, que això només succeeix en punts antipodals.

Els punts antipodals són aquells punts oposats l'un de l'altre en una esfera: Imaginem una esfera travessada per una recta, en aquells dos punts on es produeix la intersecció de la recta amb la superfície de l'esfera són punts simètricament oposats a l'esfera, fet que la seva distància estableix el diàmetre d'aquesta.

Per últim, cal esmentar que aquesta fórmula permet una aproximació de la distància real a la Terra, ja que aquesta no és una esfera perfecta (s'assembla més a un el·lipsoide, tot i que és un geoide) i el seu radi pot variar de 6356 Km de radi als pols, fins a 6378 Km a l'equador, fet que obligarà a establir un radi aproximat a aquests dos. Típicament, això no presenta masses problemes alhora de calcular punts poc distants, com és el cas de la l'aplicació QTIS, que estableix recorreguts de no més de 5 Km, però d'aquesta manera es procedeix d'una manera més òptima i elegant a calcular les distàncies. Així doncs, aquesta és la solució que s'implementa en el sistema per tal d'establir un radi entre la posició del usuari i l'event.

3. Desenvolupament del projecte

En aquest apartat es desenvoluparan tots aquells temes relacionats directament amb l'eina de geolocalització. S'explicarà com s'ha constituït l'eina, ja sigui visualment, com a la part de programari, i més concretament, s'explicaran totes aquelles decisions que s'han pres alhora d'implementar la interfície, les funcionalitats d'ubicació geogràfica i tot allò relacionat amb la connexió amb el portal QTIS.

3.1 Model casos d'ús

En el model de casos d'ús (veure *figura 21*) podem observar el conjunt de funcions que podem executar amb QTISM des de el punt de vista del rol d'estudiant. Aquest usuari pot realitzar diferents funcions abans de fer el *login*, com per exemple consultar les opcions disponibles o actualitzar la posició mitjançant el moviment.

Una vegada fet el *login*, es pot accedir a la ruta desitjada, més concretament consultar cada event de la ruta o interactuar amb el mapa. Si interactuem amb el mapa, el podem ampliar o reduir, canviar la seva tipologia o desplaçar-lo. Si pel contrari consultem l'event podem respondre la pregunta, sentir el so o veure la imatge.

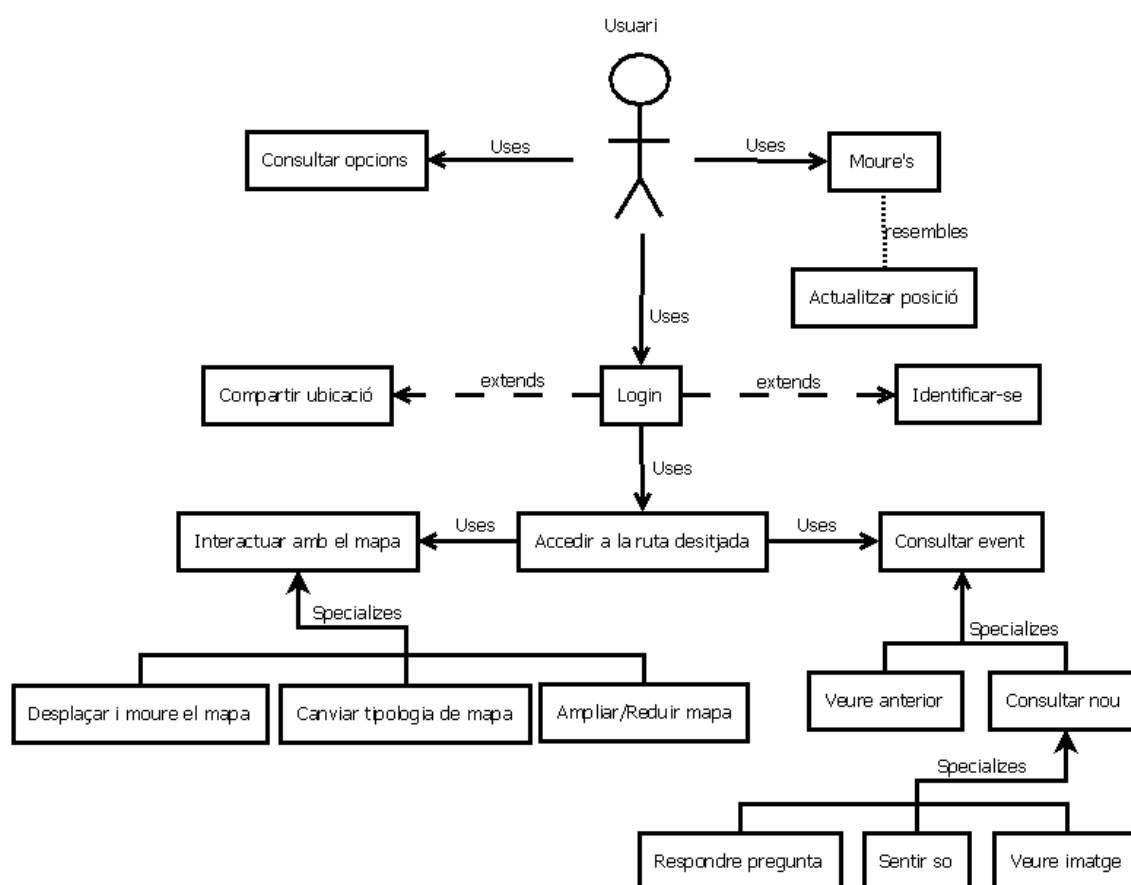


Figura 21. Model de casos d'ús de QTISM.

3.2 Anàlisi de requeriments

Desenvolupar un sistema significa que s'ha de poder construir primerament, mitjançant la seva descripció, en un segon pla, s'ha de poder observar els efectes que pot provocar sobre els usuaris i l'entorn i finalment, arribar a la seva implementació. Per això és molt important establir uns requeriments que ens permetin descriure i especificar el sistema.

Aquest anàlisi de requeriments ens permet també, incloure els serveis que l'usuari necessitarà, així com les restriccions a les que es veurà sotmès el conjunt. A més a més, aquest anàlisi permet identificar els potencials problemes, dividir les funcionalitats, i poder planificar cadascun dels elements per separat, fet que ens ajudarà a l'hora de la implementació.

Aquesta llista de requeriments és molt possible que a mesura que anem definint el sistema s'hagi d'ampliar, però és important definir-la en primera instància.

3.2.1 Requeriments del usuari

En aquest primer apartat, s'explicaran acuradament totes les funcionalitats que s'ha de donar la possibilitat de realitzar a l'usuari. S'ha d'entendre com a usuari, en primera instància (pot existir un rol d'observador o monitor), un alumne que està realitzant una prova consistent en la resolució de les preguntes que se li formulen. Cal per tant, centrar-ne els objectius d'aquest usuari per tal d'especificar els requeriments d'aquest, i així també, poder esbrinar quines dificultats i facilitats troba en el funcionament i disseny.

Els requeriments del usuari són els següents:

- Caldrà que l'usuari es registri i/o s'identifiqui al sistema QTIS.
- L'usuari podrà accedir al sistema mitjançant el *link* pertinent de QTIS que el portarà al sistema de geolocalització.
- L'usuari ha de poder veure on es troba constantment i d'una manera actualitzada sobre un mapa de *GoogleMaps*.
- L'usuari ha de poder rebre un event i per tant ha de poder veure'l correctament, sempre i quan es trobi en una ubicació idònia.
- Tant si l'event que l'usuari ha rebut ha obtingut un *feedback* satisfactori, com si no, s'ha de poder veure-ho i passar al següent.
- Una vegada l'usuari ha respòs una pregunta, rebut un so o vist una imatge, el sistema li ha de poder assenyalar el camí a seguir, ja sigui mitjançant un missatge o sobre el mapa.

3.2.2 Requeriments funcionals

En aquest segon apartat, s'explicaran acuradament totes aquelles funcionalitats que el sistema ha de ser capaç de brindar a l'usuari d'aquest. Per tant, es defineix d'una forma més detallada les possibles transformacions que el sistema realitza sobre els 'senyals d'entrada' per tal de produir 'senyals de sortida'. Els requeriments funcionals seran aquelles necessitats que el sistema haurà d'implementar.

- El sistema ha de ser capaç de llegir el fitxer que tindrà com a contingut, totes aquelles ubicacions d'un event, emmagatzemar-les i poder tractar-les segons la ubicació del usuari.

- El sistema ha de permetre transformar la ubicació del usuari en coordenades per a la comparativa amb el fitxer d'events.
- S'ha de poder llençar un event a l'usuari si aquest es troba a la posició correcte o dins un radi aproximat mitjançant una comparativa de coordenades.
- El sistema ha d'establir una connexió a la compte de QTIS associada a un usuari d'aquest per tal de respondre preguntes, veure fotografies o escoltar sons.
- Tot el sistema ha de ser executable en 'qualsevol' tipus de terminal mòbil, per tal d'oferir la màxima compatibilitat.
- Tanmateix, ha de ser funcional en qualsevol tipologia de navegador, a excepció de *Internet Explorer*.

3.2.3 Requeriments no funcionals

En aquest tercer i últim apartat, s'expliquen tots aquells requeriments no funcionals, és a dir, aquells que venen imposats per la tecnologia o llenguatge emprats i que, d'alguna manera, poden limitar el sistema. Cal per tant, tenir-ho molt en compte a l'hora d'implementar el prototipus. D'altra banda però, s'ha d'intentar minimitzar aquestes restriccions en la mesura de lo possible, com per exemple, dotant al sistema d'una màxima compatibilitat o d'una programació en codi obert.

- L'ús del llenguatge de programació HTML 5 ens facilitarà sobretot la implementació de la part d'ubicació geogràfica mitjançant l'API de geolocalització, serà per tant, convenient adaptar-se coma mínim a l'estàndard HTML.
- El sistema combinarà diversos llenguatges com HTML, Javascript o PHP si s'escau, per tal d'accedir a la base de dades QTIS.
- El temps d'aprenentatge de la interfície, no hauria de ser gaire gran gràcies a l'automatització de les tasques a realitzar, però podria portar uns minuts.
- Serà necessària e imprescindible una connexió de dades 3G o GPRS al terminal per a realitzar totes les accions.
- De la mateixa manera, tot i que no serà requisit imprescindible, serà necessari que el terminal estigui dotat amb un sistema de GPS, fet que millorarà d'una forma òptima la precisió de la ubicació del usuari en tot moment.
- Hi haurà una gestió d'errors per tal d'assegurar a l'usuari que l'acció desitjada s'ha realitzat correctament i sinó, com solucionar-la, sobretot a la part de geolocalització d'aquest.

3.3 Solució proposta

La solució proposta finalment (veure *figura 22*), consisteix en un portal web adaptat per a terminal mòbils que s'estructura en tres apartats: Mapa, events i suport, que s'explicaran a continuació. D'aquesta manera s'aconsegueix maximitzar el rang de dispositius compatibles i permetre al terminal la càrrega d'una web molt lleugera, fet que millorarà l'eficiència del sistema.

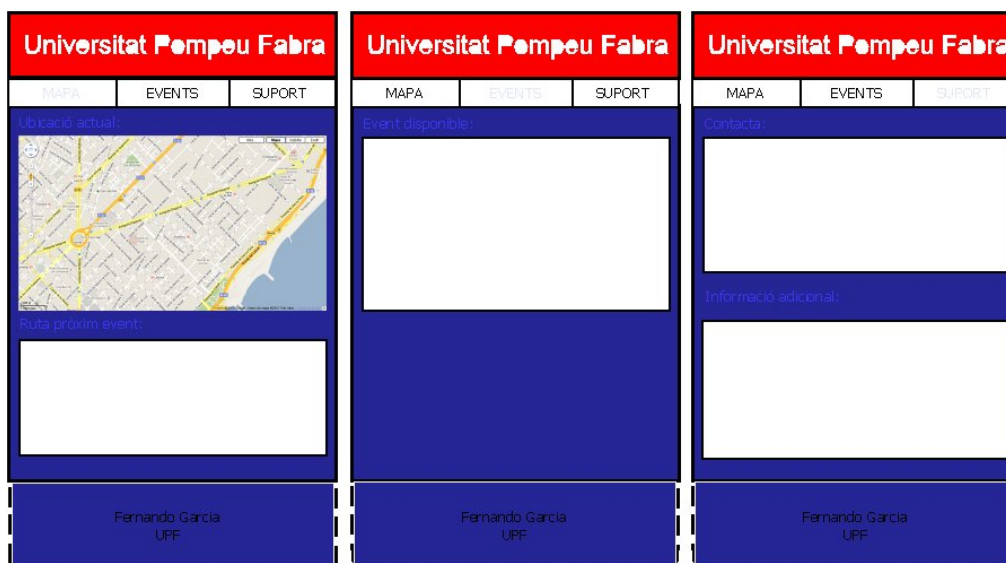


Figura 22. Primer disseny visual de QTISM.

3.3.1 Descripció

En aquest apartat s'explicarà les diferents seccions que conformen QTISM en detall i per separat, una vegada s'ha realitzat el *login* (veure la *figura 23*) i la seva posterior validació.



Figura 23. Pàgina de *login* de QTISM.

El mapa

La secció del mapa és la més important. En ella s'orquestra tota l'activitat de l'eina. Podem:

- Consultar el mapa (veure *figura 24*).
- Comprovar aquells events visitats i l'actual mitjançant les icones.

- Fer servir les indicacions de la ruta a seguir des de la ubicació del usuari fins a l'event actual.
- Veure la posició actual del terminal, que només es pot obtenir amb permís explícit d'aquest (veure *figura 25*). Com es pot apreciar a la imatge aquesta petició es duu a terme tant bon punt es carrega la pàgina principal, fet que permet carregar el corresponent mapa amb la ubicació del usuari.

D'altra banda, és també en aquesta pàgina on es produeix el càlcul, de forma transparent per al usuari, que permet redirigir-lo a l'event específic de forma automàtica per tal de contestar la pregunta. El motiu pel qual es va decidir fer servir aquest mètode fou que l'usuari només pogués veure la pregunta quan es trobés a una posició concreta. A més a més, es va pensar en tot moment en poder brindar a la posició per tal de mantenir l'usuari informat a temps real de on es troba en tot moment, així com les ubicacions anteriors d'events passats i l'actual.

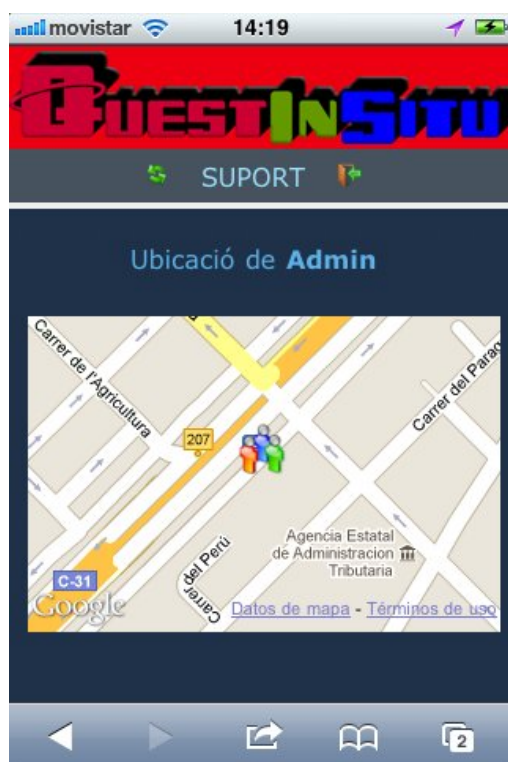


Figura 24. Posició actual del usuari al mapa (a l'esquerra).

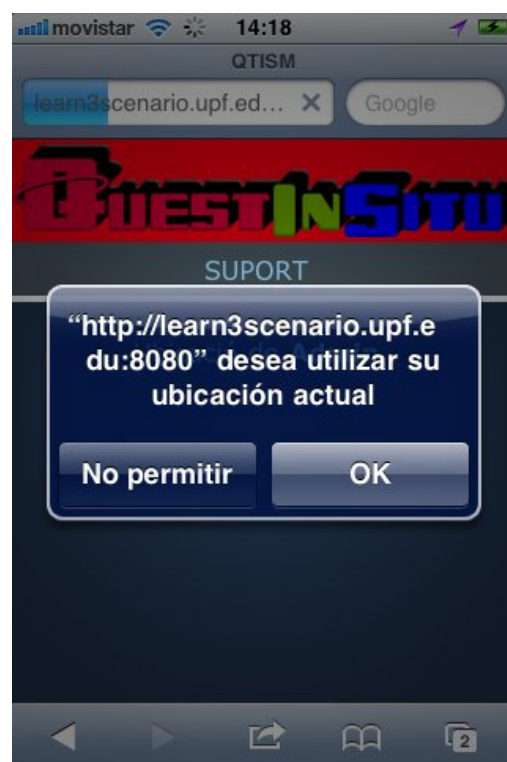


Figura 25. Petició de localització del usuari per part de QTISM (a la dreta).

A més a més, com s'ha comentat abans, a la part inferior de la pàgina, sota el mapa es pot veure la ruta de carrers a seguir fins a l'event actual (veure *figura 26*). Aquesta ruta està calculada per a ser realitzada a peu, amb indicacions de noms de carrers. Absolutament totes les parts de l'aplicació han estat pensades per a ser vistes a dispositius mòbils: tamany de lletra, distribució en columna i absència d'imatges en són alguns exemples. Tanmateix, tota l'eina ha estat feta per adaptar-se a qualsevol tipologia de mapa de qualsevol terminal i també té inhabilitada l'opció d'escalat, ampliació o reducció, de tot el seu contingut, a excepció del mapa que sí ho permet.

En ocasions la diferencia de terminals, aporta lleugeres diferències a la interfície. Un exemple d'això ho trobem en l'absència de botons de control en el mapa si, com a les imatges, l'aplicació s'executa sobre *iPhone*. Però si és executada en un terminal *HTC*, apareixen controls addicionals per tal de modificar el tamany d'aquest. En aquest cas, la plataforma d'*Apple* permet realitzar-ho amb la clàssica pessigada.

Cal esmentar la tipologia d'ícones que apareixen. Si la pregunta ja ha estat contestada, apareix en aquella coordenada una bandera de color verd, si pel contrari és un event encara no visitat, apareixerà una estrella de color groc tal com es pot apreciar a la figura 27. Una vegada s'hagin visitat tots els events, apareixerà el corresponent missatge de fi de ruta.

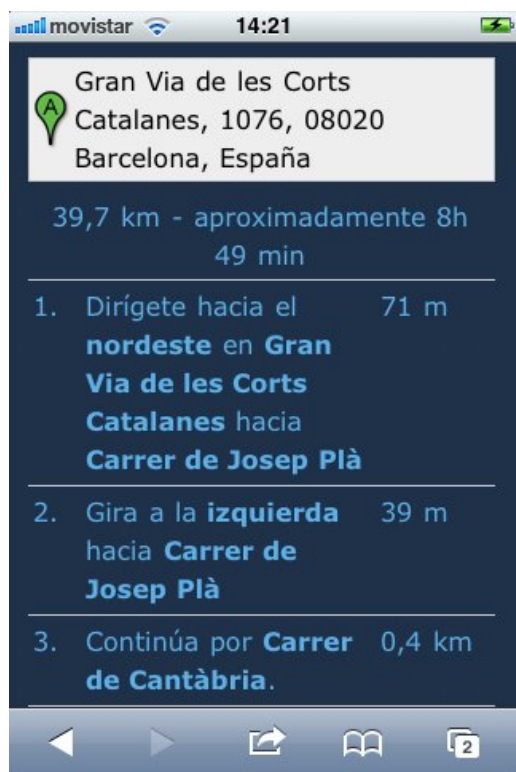


Figura 26. Panell d'indicació de carrers fins l'event actual (a l'esquerra).



Figura 27. Exemple de ruta amb l'event actual i els completats (a la dreta).

Els events

Gràcies a QTIS, aquesta part no ha estat necessari treballar-la massa, ja que aquesta eina ens ha donat la possibilitat de mostrar la pregunta corresponent mitjançant un *iframe* de la pàgina i un *scroll*. Aquesta part de l'aplicació només és visible quan es realitza la redirecció a l'event corresponent, i permet la seva contestació, retornant un *feedback* a la mateixa posició on estava l'*iframe*.

Suport

Aquesta part ha estat pensada per futures ampliacions de QTISM, com un xat de comunicació entre els usuaris d'una ruta concreta, manuals d'ús, així com un possible contacte directe amb l'administrador del sistema. No obstant això, en aquest treball només ha estat plantejat com a una eina a desenvolupar en un futur i que podria aportar unes funcionalitats addicionals al sistema.

3.3.2 Disseny dels elements de QTIS i QTISM

En aquest apartat s'explicarà la distribució i comunicació dels diferents elements que conformen QTIS i QTISM. Així com les seves funcions generals i el paper que desenvolupen en el sistema. Una bona comunicació és fonamental per a la integritat de l'eina, ja que s'ha de preservar una bona coherència de les dades en les seves diferents transformacions per tal d'obtenir bons resultats.

La millor manera per dur a terme aquesta correcta comunicació i distribució, es un espai web que pugui interactuar amb el portal QTIS, les diferents API necessàries i fins hi tot amb bases de dades. D'aquesta manera, el portal esdevé la peça d'interconnexió d'aquests elements encapsulats que conformen una unitat per si mateixos, facilitant possibles ampliacions i dotant el sistema d'una coherència en el disseny. En la figura 28 es mostra com estan distribuïts els diferents mòduls d'eines i com es comuniquen entre si amb el portal web.

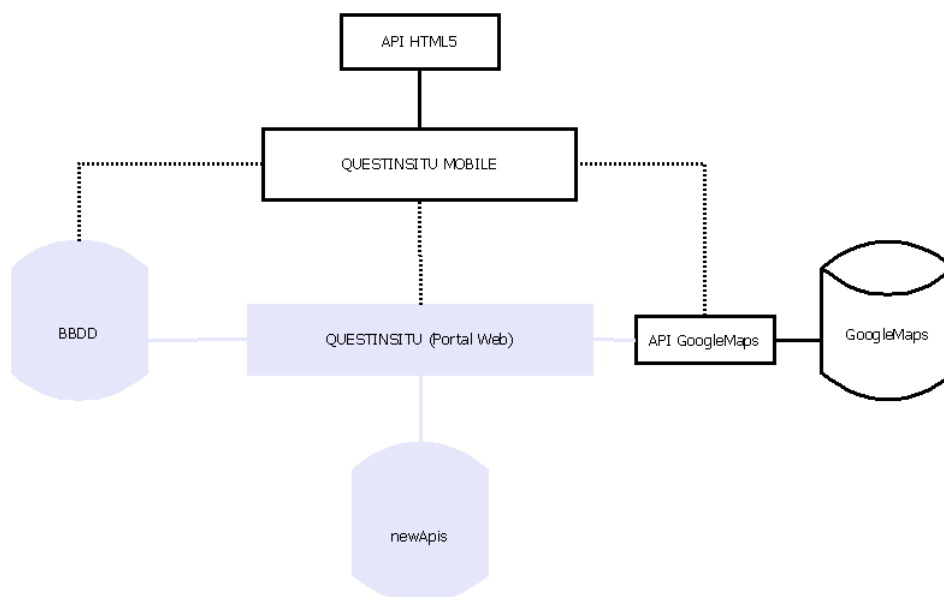


Figura 28. Esquema del disseny i la comunicació dels diferents mòduls de QTIS.

3.3.3 Comunicació entre els elements de QTIS i QTISM

En aquest apartat s'explicarà com es produeix la comunicació entre els diferents elements del sistema i mitjançant quins procediments es connecten cadascun d'ells.

- **Comunicació entre QTIS i QTISM:** La premissa bàsica per realitzar aquesta comunicació ha estat la independència completa d'ambdós portals. Per això, aquesta comunicació es realitza mitjançant tecnologia XML. Cada ruta creada al portal QTIS genera el seu XML corresponent (veure figura 29) que es recollit per QTISM per al seu processament.

Primerament, en aquest arxiu figura l'identificador del test, així com el número d'events o preguntes que conté. A continuació, entre les etiquetes *question_item*, que n'hi ha una per a cada event, apareix l'identificador de la pregunta relacionada amb el seu corresponent *id*. Això es molt útil de cara a realitzar la petició de la pregunta a QTIS per a ser mostrada per pantalla. Posteriorment, entre les etiquetes *coord*, es troba el conjunt de coordenades associades a aquell event o pregunta en l'ordre latitud i longitud. Per tal de llegir aquesta informació correctament, QTISM conté una classe *java* dins el paquet *coreservlets* anomenada *lecturaEntradaXML.java* que analitzarem a l'apartat 3.3.5 *Funcions més importants de QTISM*.

```

<test>
<question_item>
<id>22</id>
<coord>41.40328994689847, 2.19400405884564886</coord>
</question_item>
<question_item>
<id>23</id>
<coord>41.40328994223777, 2.1940040588378906</coord>
</question_item>
<question_item>
<id>24</id>
<coord>41.402388625924175, 2.1947765350341797</coord>
</question_item>
<question_item>
<id>25</id>
<coord>41.402710526042696, 2.195420265197754</coord>
</question_item>
<question_item>
<id>26</id>
<coord>41.40322556291566, 2.1959567070007324</coord>
</question_item>
</test>

```

Figura 29. Estructura del XML d'una ruta.

- **Comunicació entre QTISM i la Base de dades:** Aquesta comunicació es necessària, en primer lloc, per tal d'identificar els usuaris. Per això al paquet coreservlets també compta amb dues classes, una per a fer el *Login* i una altre per a fer *Logout*, que son *BeanCheckUsers.java* i *LogOUT.java*, respectivament. Aquestes classes realitzen diferents consultes a la base de dades fent ús també de la llibreria *jdbc (Java DataBase Connectivity)*.
- **Comunicació entre QTISM i l'API de GoogleMaps:** Aquesta és, probablement, la comunicació conceptual més complexa i prova d'això, són els nombrosos arxius i funcions en *javascript* que es comuniquen amb aquesta API. En el següent apartat es fa un anàlisi més exhaustiu de les seves funcions.
- **Comunicació entre QTISM i l'API d'HTML 5:** Malgrat el complexa procediment que hi ha darrere de la localització de la posició del terminal mòbil, la petició d'aquesta activitat es realitza mitjançant un conjunt de crides a funcions de l'API de localització integrada a la majoria de navegadors. En aquest aspecte, el procediment pel qual s'obté la posició del usuari es completament irrellevant. Únicament cal tenir en compte, la comunicació que hi ha en el procediment de crida a les funcionalitats de l'API.

3.3.4 Diagrama de flux del funcionament

Per tal d'entendre d'una forma més acurada els la finalitat de les funcions del sistema, així com el desenvolupament dels processos interns fins a completar una ruta sencera, s'ha procedit a la composició d'un diagrama de flux amb tot allò més rellevant (veure *figura 30*). Tanmateix, s'exposa un diagrama del disseny de les classes (veure *figura 31*) que resulta d'utilitat abans d'explicar algunes de les funcions més importants de QTISM.

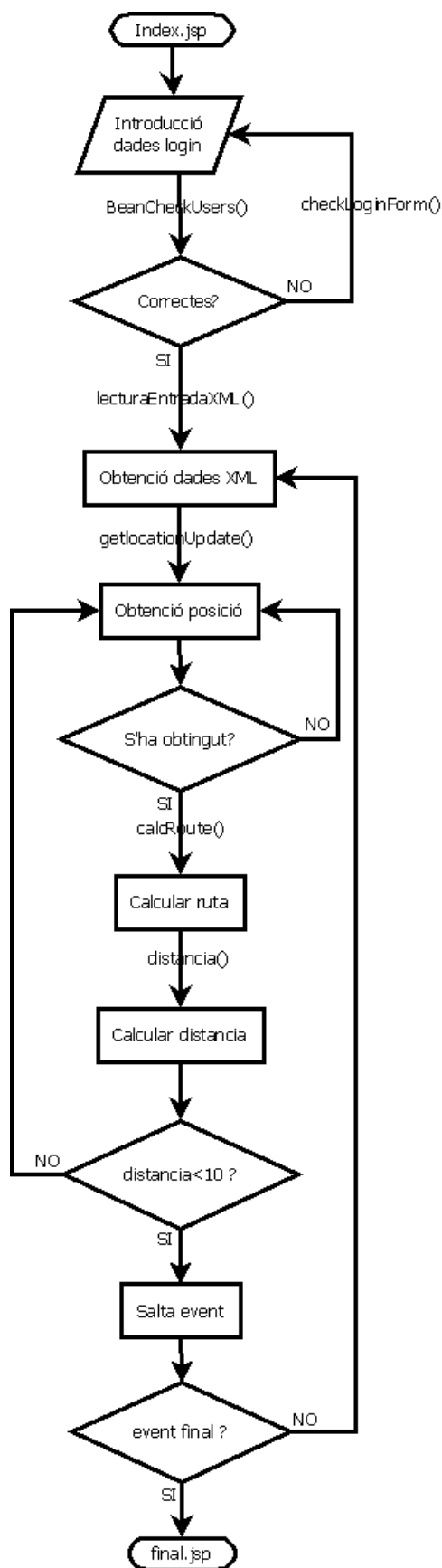


Figura 30. Diagrama de flux.

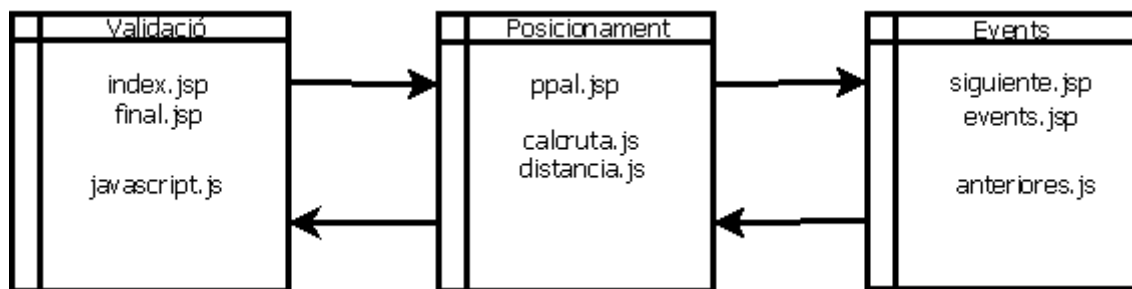


Figura 31. Disseny abstracte de les classes de QTISM.

3.3.5 Funcions més importants de QTISM

A continuació, s'exposaran algunes de les funcions més importants per al funcionament de QTISM. Les funcions es detallaran en format fitxa per tal de que sigui molt més entenedor en els següents punts: ubicació, funció i utilitat d'aquesta.

Funció: getLocationUpdate

Aquesta funció està ubicada a l'arxiu *ppal.jsp* en el contingut web de la pàgina i es podria considerar l'artífex de la localització d'usuaris (veure *figura 32*). Al cridar-la, i si el navegador té implementada l'API de localització, executa la funció *inicializar* cada vegada que l'usuari varia la seva posició, o com a màxim, cada 30 segons. D'això últim se n'encarrega *watchPosition*.

```

function getLocationUpdate(){
    if(navigator.geolocation){
        var options = {timeout:30000, enableHighAccuracy: true};
        var watchID = navigator.geolocation.watchPosition(inicializar, errorHandler, options);
    }else{
        alert("Ho sentim, el teu navegador no suporta geolocalització!");
    }
}
  
```

Figura 32. Estructura de la funció *getLocationUpdate*.

A més a més, també es poden especificar altres opcions, com per exemple *enableHighAccuracy*, que permet l'obtenció més acurada d'una posició mitjançant l'ús addicional del mòdul GPS. D'altra banda, si no es possible especificar la posició del terminal, es cridarà la funció *errorHandler* (veure l'annex 3 *figura 38*).

Funció: errorHandler

Aquesta funció es troba ubicada dins l'arxiu *calcruta.js* i, únicament es cridada quan no es possible especificar la posició del terminal. Si l'atribut *err.code* té com a valor 1, es produeix un error degut a que l'usuari nega el permís d'accés a la seva posició. Si pel contrari l'atribut té valor 2, vol dir que no ha estat possible determinar la posició de l'usuari degut a un error intern.

Com es pot observar, la funció (veure l'annex 3 *figura 38*) rep com a paràmetre l'error que ha ocorregut, amb tots els seus atributs, el més important *code*.

Funció: inicializar

Aquesta funció, juntament amb *getLocationUpdate*, és la més important. Aquí es realitza tota la lògica del sistema i la crida a funcions per a la delegació de tasques concretes, com el càlcul de distàncies o la gestió de l'addició d'icones al mapa. Per això, es necessari dividir l'explicació de les tasques que realitza per parts i d'una manera una mica més acurada.

El primer procediment que es realitza, es la comprovació del final de ruta. Com es pot observar, la funció *inicializar* rep com a paràmetres la posició esbrinada per *getLocationUpdate*. Mitjançant una condició ens assegurem que la variable *aux*, que actua com a índex de l'event actual, no sigui igual al nombre d'events de la ruta. Si fos així, significaria que s'ha arribat a l'últim event d'aquesta i per tant, caldria redirigir l'usuari a la pàgina *final.jsp* per a indicar-ho (veure l'annex 3 *figura 39*). A continuació, si es la primera vegada que s'executa la funció, es realitza una petició a l'API de *GoogleMaps* per tal de crear un mapa centrat a la posició relativa a la ubicació del usuari definit a la variable *myOptions*. També es defineixen atributs com la tipologia de mapa, el zoom i els controls a mostrar.

Si pel contrari no es la primera vegada que s'executa la funció, es retira el marcador que indica la posició del usuari, per tornar a carregar-lo a la posició actual d'aquest. D'aquesta manera, s'evita la recàrrega del mapa cada vegada que s'executa la funció, i només es recarrega el marcador (veure l'annex 3 *figura 40*). Però, com es defineix el vector de marcadors i com es mostren sobre el mapa amb ordre? Doncs, abans d'executar qualsevol línia de codi de les abans esmentades, es realitza la lectura del XML amb la ruta. Una vegada s'ha llegit, es crea una matriu amb tota la informació de la següent manera (veure l'annex 3 *figura 41*).

D'aquesta manera a la primera columna de la matriu, es troba l'identificador de la pregunta corresponent. A la segona i tercera columnes, es troba la latitud i longitud on es troba ubicada aquesta pregunta, respectivament. I, a la columna final, la icona associada a aquest event. Una vegada es té aquesta informació, només cal crear un vector de marcadors de *GoogleMaps*, on s'afegiran en ordre amb la finalitat de ser inserits al mapa quan es desitgi. Primerament, però, només seran creats i afegits al vector *marcadores* (veure l'annex 3 *figura 42*).

D'aquesta manera, es dota el marcador d'una posició (latitud i longitud), un títol i una icona identificadora. Així, es podrà afegir cada marcador al mapa, recorrent el vector, quan es desitgi. A continuació, es mesura la distància amb la funció *distancia* explicada més endavant i es comprova, mitjançant la condició que es mostra a la *figura 33*, si cal recalcular la ruta de direccions mitjançant *calcRoute* perquè ha aparegut un nou event. A més a més, si la distancia es menor a la especificada a la condició, es mostra l'event per pantalla.

```

if (j==1) {
    j = 0;
    eliminar();
    calcRoute();
}

if ((d<=(15/1000)) && (j==0)) {
    location.href= "events.jsp?ID="+eventos[aux][0];
}
}

```

Figura 33. Estructura de les condicions de mostra d'events.

Funció: calcRoute

Aquesta funció bàsicament defineix el panell de direccions que apareix a *ppal.jsp* i està continguda dins l'arxiu *calcruta.js*. La seva funció es recalculer la ruta de carrers que ha de seguir l'usuari amb el seu terminal des de la seva posició fins a la posició de l'event actual (veure l'annex 3 figura 43).

Per això es necessari conèixer la posició inicial, en aquest cas *start*, que serà la del usuari i també la posició final, *end*, que serà la de l'event a trobar. Aquestes variables, juntament amb alguna opció, com el fet que es calculi per a fer a peu, permet realitzar la petició a l'API de *GoogleMaps* que retornarà el *directionsDisplay* mostrat per pantalla.

Funció: distancia

La funció distancia està ubicada a l'arxiu *distancia.js* i té com a finalitat oferir la longitud, en kilòmetres, entre l'usuari i l'event actual. Mitjançant la fórmula de *Haversin*, explicada a l'apartat 2.6.4, s'estableix un radi d'interacció entre el terminal i l'event que permet definir quan es desitja que aparegui l'event per pantalla. D'aquesta manera, si la distància entre l'usuari i l'event es inferior a 15 metres es redirigirà a la pàgina *events.jsp* on apareix el *iframe* de la pregunta associada a l'event (veure la figura 34).

```
function distancia (){
    var PI = 3.141592653589793238462643383279502884197169399375105820974944;
    var lat1 = latitude;
    var lon1 = longitude;
    var lat2 = latitude2;
    var lon2 = longitude2;

    var R = 6372;
    var dLat = ((lat2-lat1)*PI)/180;
    var dLon = ((lon2-lon1)*PI)/180;

    var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
            Math.cos((lat1*PI)/180) * Math.cos((lat2*PI)/180) *
            Math.sin(dLon/2) * Math.sin(dLon/2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    d = R * c;
}
```

Figura 34. Estructura de la funció *distancia*.

Funció: anteriores

Aquesta funció, ubicada a l'arxiu *anteriores.js*, permet reformular els marcadors que corresponen a un event anterior i per tant, que ja han aparegut per pantalla. D'aquesta manera, es pot distingir aquells events ja contestats mitjançant banderes, d'aquells que encara estan per contestar amb estrelles. Per això, es necessari crear un vector que conté els marcadors anteriors, tal com s'observa a l'annex 3 figura 44.

3.4 Avaluació QTISM

L'avaluació de l'eina és un dels punts més importants del projecte, doncs gràcies a les avaluacions podem trobar errors de codi, característiques difícils d'entendre, i veure el nivell d'acceptació per part dels usuaris de les propietats actuals. A partir d'aquestes avaluacions podem millorar el nostre projecte, per així obtenir un producte final de molt més robust. Això és molt important, ja que la qualitat del producte determinarà l'èxit que pugui tenir l'eina.

A QTISM l'avaluació és una part indispensable ja que aquesta tipologia d'eina es centra en la facilitat de compressió i ús d'aquesta per part del estudiant. A més però, també ho és pel tipus de tecnologia que empra. Una tecnologia amb un grau de robustesa elevat, assegurant el seu òptim funcionament en una multitud de circumstàncies. Com el context d'avaluació és educatiu, es important que l'eina no dificulti el procés d'aprenentatge, i que el seu ús no faciliti la pèrdua de concentració sobre l'objectiu principal de la tasca.

3.4.1 Context educatiu de la prova d'avaluació

La prova d'avaluació de l'eina QTISM es va poder provar en un context educatiu real dins l'assignatura de botànica de primer curs dels estudis de Biomedicina impartits per la UPF. Gràcies a la mediació de Patricia Santos, la tutora d'aquest projecte, amb Elisabeth Moyano, professora del grup de botànica, es va poder desenvolupar la prova al parc natural de Sant Llorenç del Munt.

Aquesta activitat opcional per als alumnes té com a objectiu motivar als estudiants en l'estudi de la botànica en ambients naturals. Per això aquesta activitat s'havia realitzat anys anteriors en aquest mateix entorn sense cap tecnologia similar a QTISM. Enguany, es va decidir mantenir la ruta ja coneguda, però modificant la tecnologia emprada. Per això va ser necessari dissenyar una ruta de vuit preguntes amb QTIS, realitzar proves de cobertura a l'entorn, testejar l'aplicació abans que els alumnes per comprovar el correcte funcionament de l'aparició d'events, ajustar les coordenades i idear correctament les preguntes relacionades amb l'entorn natural. Finalment, van ser un total de set estudiants els que van realitzar la prova.

3.4.2 Preparació de la activitat d'avaluació

Per la preparació de l'activitat van ser necessari, en primera instància, el disseny de les preguntes, així com les respostes, mitjançant QTIS per part de la professora Elisabeth Moyano. A continuació, va ser indispensable testejar el correcte funcionament de les preguntes a l'entorn on s'anava a desenvolupar la prova. Per tal de comprovar-ho, es va haver d'anar al context real i comprovar *in situ* com apareixien els events, si apareixia el *feedback* corresponent a la pregunta i en definitiva, si es podia treure profit de QTISM en l'activitat.

Realitzant aquesta activitat vam poder comprovar les limitacions, importantíssimes, que hi havia a la ruta. La limitació més gran fou la cobertura. Aquesta era una necessitat essencial, ja que tot el sistema es basa en la cobertura 3G i de telefonia mòbil, per tal de carregar les pàgines de l'eina i encara més, per ubicar l'usuari amb una precisió adequada. El problema fou que la cobertura en el parc natural era present en uns 250 metres, per això en aquest espai es va presentar el repte de condensar el major nombre de preguntes possible, fet que fa requisit indispensable l'ordre de les preguntes i la màxima precisió en cadascuna per tal d'evitar un solapament del radi d'acció d'aquestes, i en conseqüència, la seva aparició seguida.

Malgrat els problemes que presentava el terreny, ja que QTISM està pensat per a ser utilitzat en ambients urbans, finalment, el resultat de la prova va ser molt satisfactori, però aleshores es va presentar un nou problema. Aquest fou com realitzar la ruta a les zones sense cobertura i sense

QTISM. La solució va ser el ús de *QR Codes*, solució que no s'analitzarà en el context d'aquest PFC.

3.4.3 Anàlisi dels resultats

Al llarg de l'activitat l'estudiant ha pogut observar tots els problemes i mancances que presenta QTISM, és per això que una vegada acabada l'activitat l'estudiant havia de complimentar un qüestionari d'avaluació (veure annex 1). En aquest qüestionari s'intenta aprofundir en els diferents aspectes de l'eina, esbrinar l'experiència del estudiant a la prova, la utilitat de QTISM i el seu comportament. El tractament dels resultats s'ha fet de forma anònima i si és necessari fer referència a algun d'aquests usuaris, seran numerats de l'1 al 7.

Mitjançant el qüestionari i les valoracions dels estudiants, s'ha pogut extreure tot un seguit de millores a implementar, tant en l'àmbit de funcionalitat, com en el d'usabilitat. Primerament, es demanava una valoració en la dificultat d'ús. En segona instància, si l'estudiant havia patit algun problema, permetia indicar-ho. D'altra banda, també preguntava sobre algun dels aspectes més controvertits del disseny per tal d'esbrinar si estaven ben situats, com per exemple, les indicacions de la ruta a seguir sota el mapa. Finalment, es demana tant una valoració general, com una opinió en el cas d'utilitzar QTISM en un context de ciutat.

Limitacions i aspectes a millorar

En la pregunta d'indicació de problemes que han sorgit durant l'activitat podem trobar un de comú en gairebé tots els estudiants, però de difícil solució: la cobertura. Tanmateix, apareix de forma repetida un altre problema: llegir el *feedback* de resposta d'una pregunta. Aquest problema deriva de l'anterior ja que si a causa de la mala cobertura la resposta no arriba al servidor, aquest no és capaç d'emetre el *feedback*, o si ho ha fet, també existeix la possibilitat que la xarxa no hagi permès entregar-lo i en conseqüència, visualitzar-lo al terminal. A més a més, el tipus de *feedback* que s'ha transmès no depèn de QTISM, sinó de QTIS.

Tot i que, com s'ha esmentat anteriorment, l'eina no necessita grans millores d'usabilitat, ha sobtat la detecció d'un problema relacionat amb la **visualització del panell d'indicacions**. Aquest panell apareix sota el mapa i permet a l'usuari seguir les indicacions des de la seva posició actual fins a l'event. El problema ha estat que no s'ha detectat aquesta funcionalitat per part del estudiant a causa de la necessitat de fer *scroll* a la pantalla del terminal. Concretament en aquesta experiència no resultava de gran utilitat, ja que era un sol camí a seguir, però pot ser molt útil en una experiència per la ciutat. Dels 7 estudiants, únicament 1 ha sigut capaç de veure-ho, per això cal retocar aquest espai, ja sigui indicant-ho explícitament amb text o implícitament en la visualització.

Aspectes positius

Les valoracions de QTISM respecte a la **facilitat d'ús**, han estat força favorables; 3 dels 7 estudiants han atorgat a QTISM un 5 sobre 5 en la valoració de la facilitat en l'ús, 3 li han atorgat un 4 i tan sols 1 estudiant un 3. Això és molt positiu, ja que ens indica que la part d'usabilitat requereix molt poques millores. Tanmateix, s'ha intentat que l'estudiant valori l'activitat segons si li ha agradat o no (veure *figura 35*).

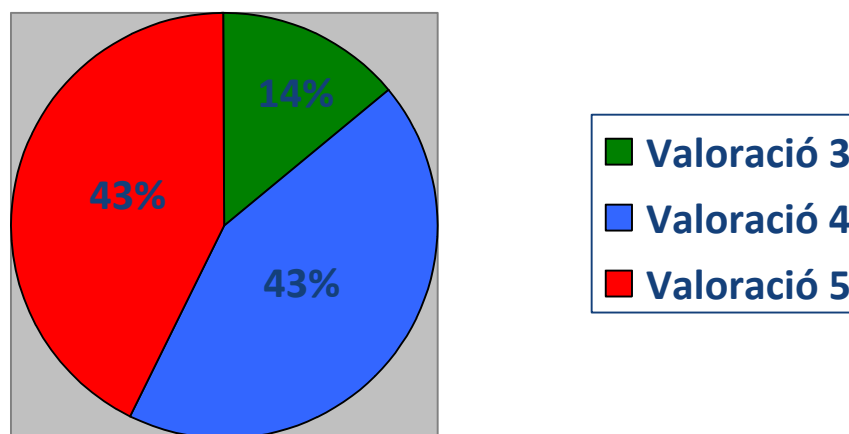


Figura 35. Gràfica en tant per cent de les puntuacions obtingudes per QTISM d'1 fins a 5.

Seguidament, es pregunta si a l'estudiant li agradaria **realitzar una activitat similar en un context urbà** i cal destacar que tots els estudiants han respòs afirmativament, fins i tot es pot observar que d'alguns entenen que l'eina es més potent en un entorn d'aquest tipus i afirmen que els hi ha agradat el dinamisme de l'activitat, que les preguntes apareguin espontàniament i amb una relació tan estreta amb l'entorn. Finalment, es demana a l'estudiant que valori si l'experiència ha estat satisfactòria, quins aspectes li han agradat més i quins menys. Aquí també podem trobar opinions molt interessants (*veure figura 36*):

L'usuari 1 diu: 'És una manera diferent i divertida de conèixer les ciutats amb motivació.'

L'usuari 2 diu: 'Seria interessant per esbrinar algunes curiositats de la ciutat.'

L'usuari 3 diu: 'M'ha agradat que les preguntes saltin espontàniament quan estem a prop de la zona on es troba la planta.'

L'usuari 4 diu: 'Si es fa en un entorn on pugui funcionar correctament, amb cobertura, és una eina que t'ajuda a fixar-te en l'entorn i a aprendre coses noves.'

L'usuari 5 diu: 'M'ha agradat participar en una activitat docent on s'utilitza i veure les possibles aplicacions, però no m'ha convençut del tot per les dificultats de connexió.'

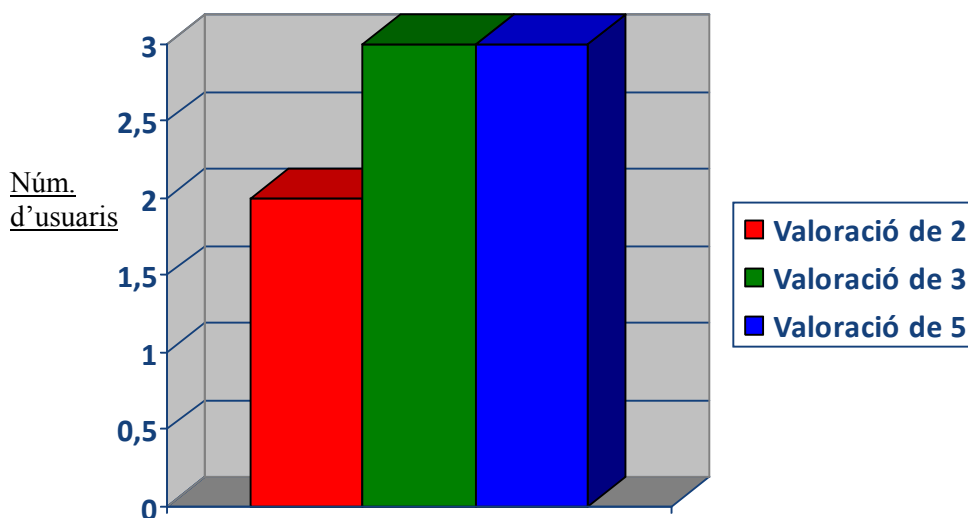


Figura 36. Opinió de l'usuari respecte a QTISM.

3.4.4 Proposta de millores i treball futur

A continuació, es detallaran els aspectes que cal millorar en QTISM tenint en compte els resultats obtinguts després de la prova amb els usuaris finals. Degut a que el projecte té una durada determinada, no ha estat possible implementar els canvis. No obstant això, en aquest apartat s'explica com un programador podria incloure'ls.

Primerament, cal solucionar el problema de la **visibilitat del panell de direccions**. Això millorarà l'experiència del usuari, sobretot en un context de ciutat. També seria molt profitós afegir una pàgina intermèdia entre la de *login* i la principal que permetés a l'usuari **seleccionar** quina **ruta** desitja carregar a QTISM, ja que actualment només en permet una a la vegada. Això no presenta masses complicacions, però es necessari l'obtenció dels noms de les rutes generades per QTIS i que siguin mostrades com una llista d'elements a seleccionar.

Per tal de solucionar el problema de la visibilitat del panell de direccions, es podria col·locar en una posició superior que entri mínimament a la zona de visibilitat principal, ja que malgrat quedi tallat, es podrà intuir que continua. Una altre forma, pot ser l'addició d'algun indicador visual que permeti intuir-ho, com per exemple un *scroll*. D'altra banda, per solucionar la part de selecció de ruta, la meua proposta seria la creació d'una funció capaç de llegir la quantitat i el nom dels arxius XML generats per QuesTInSitu. Una vegada fet això, generar de forma dinàmica la pàgina de selecció de rutes oferint un llistat amb aquestes.

Una altre proposta, per tal de tenir en compte les valoracions del usuaris, és la millora de la cobertura en la mesura de lo possible. Això no depèn de l'eina, sinó de la xarxa, però el fet de minimitzar el pes dels diferents elements, pot ajudar a millorar la velocitat de càrrega i transferència de les dades. Un exemple, pot ser el canvi del fons de la pantalla, compost per la repetició d'imatges, el *logo* o la millora en termes d'eficiència de l'algoritme.

Per últim, la part que requereix més dedicació i pot aportar una major funcionalitat a l'eina, és la potenciació del **sistema de comunicació**. QTISM està dotat amb una part no desenvolupada anomenada *suport*, on dona peu a integrar un xat que millori la comunicació entre usuaris d'una gimcana i amb l'administrador de l'eina. D'aquesta manera, a més de d'interactuar amb l'entorn, els usuaris també ho poden fer entre ells. No obstant això, aquesta idea no ha arribat a materialitzar-se, però ha deixat una porta oberta a la seva implementació. Per tal de fer-se una idea, la última versió de QTISM incorpora un petit xat en flash com a *iframe*.

4. Conclusions

En aquest apartat s'explica què s'ha extret de la realització d'aquest projecte final de carrera, tant en l'aspecte personal, com en la tècnica. Tanmateix, es fa una reflexió sobre les possibilitats que permet la tecnologia emprada per a futurs projectes relacionats amb l'àmbit de la localització web.

Al llarg de l'elaboració, he pogut comprendre la importància de la planificació de la feina i com s'ha de gestionar aquesta. La realització d'un projecte de llarga durada és una tasca difícil, dividint la feina en terminis i realitzant recerca dels materials i llocs de consulta per a poder realitzar totes les fases planificades i així arribar a aconseguir els terminis concretats. Durant el desenvolupament s'ha aprofundit en el coneixement dels sistemes d'informació, el seu plantejament, disseny i avaluació dins el marc de l'*e-learning*. Per tal de realitzar aquesta tasca correctament, s'han posat en pràctica conceptes relacionats amb l'àmbit acadèmic, més concretament amb assignatures com *Taller de Software*, *Cartografia* o *Disseny d'interfícies*.

Amb l'elaboració d'aquesta eina, crec que s'assoleixen les bases per a la realització de gimcanes fent ús de QuesTInSitu com a *software* de creació de rutes, propòsit inicial de l'eina. Per tant, es potencia clarament, permetent l'experiència de l'usuari tant en el disseny de la ruta, com en el desenvolupament d'aquesta. Des del punt de vista pedagògic i de l'alumne, penso que aquest projecte és interessant per tal de desenvolupar el raonament autònom en un entorn diferent, potenciar les ganes d'aprendre en àrees que sovint requereixen un estudi de camp, com la botànica, la urbanització, el turisme, el coneixement de la història, l'art o la publicitat.

No obstant això, per tal que a l'usuari li sigui profitós l'ús de l'eina, aquesta ha estat dissenyada pensant en les seves necessitats, ja sigui simplificant al màxim el seu ús, o minimitzant el temps d'aprenentatge. D'aquesta manera s'estimula la motivació de l'usuari pel contingut de la matèria mitjançant el mètode i no pas la frustració d'un entorn d'aprenentatge caòtic i desconegut.

D'altra banda, una de les primeres conclusions tècniques que s'han extret són les limitacions actuals de la tecnologia de localització web o l'API *Geolocation Specification*. Mitjançant aquest projecte s'ha posat a prova la precisió de les deteccions i s'ha pogut concloure que la tecnologia emprada, primer, ha estat la correcta i, segon, té molt poques limitacions de funcionalitat en entorns semblants. En definitiva, és una tecnologia en ple auge, fàcil d'usar i de potenciar. En aquest cas, considero que el concepte de localització cada cop més, anirà prenent força a la xarxa, com en el seu moment ho varem fer les xarxes socials.

Aquest projecte s'ha dissenyat amb la intenció que fos el punt de trobada entre diferents tecnologies ja existents per tal de crear un espai comú amb una funcionalitat específica. Bàsicament, s'ha intentat actuar com un pont entre l'API de localització, l'API de *GoogleMaps* i QuesTInSitu. Cal esmentar que s'ha aconseguit amb èxit. A més a més, manté una certa independència del portal QuesTInSitu, de on només s'agafa l'XML de la ruta corresponent. Això permet el funcionament d'una forma autònoma.

Conseqüència de lo anterior, és que he pogut assolir un nivell de programació en entorns HTML, CSS, *Javascript* i *Java*, així com en *software* de cartografia, més concretament mitjançant l'ús de l'API de *GoogleMaps*. Tanmateix l'optimització del portal per a terminals mòbils ha estat també un repte, ja que només s'ha de realitzar aquesta tasca en l'aspecte visual, sinó que a causa de les necessitats de precisió i lleugeresa de l'eina que imposen els terminals mòbils, s'ha optimitzat en gran mesura el codi existent. En aquest sentit, s'ha hagut de valorar mètodes alternatius per tal de facilitar les operacions a dur a terme. Així, no només s'ha procurat el bon funcionament, sinó també que sigui eficient.

Finalment, cal esmentar que les proves amb els usuaris han estat molt satisfactòries perquè hem pogut esbrinar i identificar tots aquells problemes relacionats tant amb el disseny de l'eina, com en el funcionament d'aquesta, identificar les limitacions de funcionament en condicions extremes i comprovar la robustesa del *software*.

5. Bibliografia

INTRODUCCIÓ:

[1] David Pérez Calle. QuesTInSitu: Aplicació per a crear preguntes QTI geolocalitzades. Juny del 2010.

ESTAT DE L'ART:

[2] HP. HP *Mscape*. Visitat el 8 de Novembre del 2010. <http://www.hpl.hp.com/mediascapes>

[3] Google. *GoogleLatitude*. Visitat el 10 de Juny del 2010. <http://www.google.com/latitude/intro.html>

[4] Huang, Y. M., Lin, Y. T., & Cheng, S. C. (2009). Effectiveness of a mobile plant learning system in a science curriculum in taiwanese elementary education. *Computers & Education*, 54(1), 47-58.

[5] Jorge A. Mendoza. E-learning, el futuro de la educación a distancia. 10 de Juny del 2010. Visitat el 12 d'Octubre del 2010.

<http://www.informaticamilenium.com.mx/paginas/mn/articulo78.htm>

[6] Naismith, L., Lonsdale, P., Vavoula, G., & Sharples, M. (2004). *Literature review in mobile technologies and learning* No. 11. University of Birmingham: Futurelab.

[7] Wikipedia. Sistema de Posicionament Global (GPS). Visitat el 12 d'Octubre del 2010. http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global

[8] Wikipedia. Sistema de Información Geográfica (SIG). Visitat el 12 d'Octubre del 2010. http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica

[9] Ricardo Rapallo Fernandez. Raster vs Vectorial. Visitat el 12 d'Octubre del 2010. <ftp://ftp.fao.org/TC/TCA/ESP/pdf/rapallo/Bloque%20I.II.pdf>

[10] Dagoberto Salazar. Índice general. 13 de Setembre de 2006. Visitat el 3 de Novembre del 2010. <http://nacc.upc.es/tierra/node4.html>

[11] Julio Jaime. Como saca mi posición. Visitat el 3 de Novembre del 2010. <http://d3ny4ll.blogspot.com/2010/09/como-carancho-saca-mi-posicion.html>

[12] Wikipedia. GPS Asistido. Visitat el 12 d'Octubre del 2010. http://es.wikipedia.org/wiki/GPS_Asistido

[13] Rosa Jiménez Cano. *FourSquare* habla castellano. 18 de Febrer del 2011. Visitat el 22 de Febrer del 2011. http://www.elpais.com/articulo/tecnologia/FourSquare/habla/castellano/elpeputec/20110218elpeputec_3/Tes

[14] Ian Hickson, Google. W3C. HTML5. Ampliat el 25 Maig del 2011. Visitat al Març del 2011. <http://www.w3.org/TR/html5/>

[15] Andrei Popescu, Google. W3C. Geolocation API Specification. Ampliat el 10 de Febrer del 2010. Visitat al Abril del 2011. <http://dev.w3.org/geo/api/spec-source.html>

[16] Google. Gears Improving Your Web Browser. 2008. Visitat al Març del 2011. http://code.google.com/intl/es-ES/apis/gears/api_geolocation.html

- [17] Odín. Where 2.0 – Location on the Web. 2009. Visitat al Maig del 2011. <http://www.slideshare.net/rsarver/where-20-2009-location-on-the-web?type=presentation>
- [18] Enrique Tech. HTML5, geolocation y Google Maps API V3. 27 d'Agost de 2010. Visitat el 16 de Maig del 2011. <http://enrique-tech.blogspot.com/2010/08/html5-geolocation-y-google-maps-api-v3.html>
- [19] Robert G. Chamberlain. GIS: Cálculo de distancias sobre la Tierra. 16 de Noviembre de 2007. Visitat el 23 d'Abril del 2011. <http://franchu.net/2007/11/16/gis-calculo-de-distancias-sobre-la-tierra/>
- [20] aNieto2k. Calcula la distancia entre dos puntos geográficos con Javascript. 2007. Visitat el 23 d'Abril del 2011. <http://www.anieto2k.com/2007/12/11/calcula-la-distancia-entre-dos-puntos-geograficos-con-javascript/>
- [21] Movable Type Scripts. Calculate distance, bearing and more between Latitude/Longitude points. Visitat el 23 d'Abril del 2011. <http://www.movable-type.co.uk/scripts/latlong.html>
- [22] Ramon Gonzalvo Mourelo. Latitud, Longitud, Meridianos y Paralelos. 1 de Diciembre del 2010. Visitat el 12 de Diciembre del 2010. <http://enlacespolivalentes.blogspot.com/2010/12/latitud-distancia-angular-entre.html>
- [23] Wikipedia. Coordenadas polares. Visitat el 24 de Noviembre del 2010. http://es.wikipedia.org/wiki/Coordenadas_polares
- [24] Urbipedia. Sistema de coordenadas. Visitat el 24 de Noviembre del 2010. http://www.urbipedia.org/index.php/Sistema_de_coordenadas
- [25] Radu Poenaru. GPS vs aGPS vs WiFi vs GSM. Visitat el 7 de Diciembre del 2010. <http://www.radupoenaru.com/gps-vs-agps-vs-wifi-vs-gsm-localization/>
- [26] Matías S. Zavia. Foursquare. 30 d'Abril del 2010. Visitat el 22 de Febrer del 2011. <http://www.xatakamovil.com/aplicaciones/foursquare-su-millon-de-usuarios-y-su-prometedor-futuro>
- [27] Patricia Santos, Mar Pérez-Sanagustín, Davinia Hernández-Leo, Josep Blat. QuesTInSitu: from tests to routes for assessment in situ activities (en estat de revisió a la revista Computers & Education).

ANNEXES

Annex 1: Qüestionari d'avaluació

Preguntes sobre l'aplicació mòbil de QuesTInSitu (usada sol en el 1er tram)

1. Valora QuesTInSitu *mobile*: (1 – molt difícil d'utilitzar – 5 molt fàcil d'utilitzar)

1 2 3 4 5

2. Indica si has tingut problemes, marcant l'opció si cal, amb alguna d'aquestes funcionalitats:

Entendre les icones del mapa []

Fer zoom o treure zoom al mapa []

Respondre preguntes []

Llegir el feedback de resposta d'una pregunta []

Llegir informació per pantalla []

Altres []

Cap problema, l'aplicació és clara i concisa []

Comenta breument quin/s ha/n estat el/s problemes, si cal,:

3. T'has adonat que sota el mapa sortia un missatge textual indicant una ruta? Per a què creus que serveix?

4. T'agradaria utilitzar QuesTInSitu *mobile* per a fer una activitat de ruta de preguntes per la ciutat?

Si []

No []

Explica breument, perquè?

5. Valora QuesTInSitu *mobile*: (1 – no m'ha agradat gens – 5 m'ha agradat molt)

1 2 3 4 5

Justifica la resposta: (*exemple: m'ha agradat la funcionalitat que servia per....., però no m'ha agradat que ...*)

Annex 2: Documentació HTML5

Aquest annex tracta alguns dels mètodes esmentats a l'apartat 2.5 L'HTML5 que formen part de l'estàndard però no s'han utilitzat directament en la implementació del projecte.

Milliores HTML5

Algunes de les millores del HTML han estat les etiquetes:

- Article, section: Indicar parts d'un article.
- Datalist: Completar automàticament amb una llista de valors.
- Details, legend: Mostrar/ocultar text sense *javascript*.
- Dialog, dt, dd: Per escriure conversacions.
- Canvas: Mostrar gràfics.
- Figure, figcaption: Associen un contingut multimèdia a un títol o llegenda.
- Keygen: Genera una clau criptogràfica per enviar-la.
- Output, input: Per indicar als formularis si són dades d'entrada o sortida.
- Progress, meter: Dona la barra de progrés d'una tasca.
- Ruby, rp, rt: Indica el text escrit en alfabet orientals.
- Video: Per mostrar vídeos sense necessitat de cap mena de *plugIn*.

De la mà del d'aquest nou estàndard, han sorgit un conjunt de nous recursos, API's, que permeten al dissenyador dotar la web de noves funcionalitats més fàcilment. Citarem alguns exemples:

- API de posicionament global per a dispositius que ho suporten.
- API de comunicació bidireccional entre pàgines.
- API per a treballar *offline* que permet descarregar tots els continguts necessaris.
- API per a fer *Drag & Drop*, és a dir, arrossegar i deixar anar un element en un espai determinat.
- API *WebWorkers*. Consisteix a poder realitzar execucions de codi en paral·lel per aquelles aplicacions de llarga durada.
- API *Storage*, que permet realitzar un emmagatzemen local mitjançant bases de dades en SQL (en *SQLite*) o amb emmagatzemen d'objectes per aplicació o per domini web (*Local Storage* i *Global Storage*).
- A més a més, es disposa d'una base de dades que possibilita fer consultes SQL.

Finalment, s'està treballant en recurs anomenat *System Information API* que permet a la Web, tenir accés al *hardware* de baix nivell; fitxers de memòria, CPU, ports USB, càmeres o micròfons, una opció que podria augmentar les possibilitats exponencialment, però amb molts problemes de seguretat.

getCurrentPosition

El mètode *getCurrentPosition* [15] pot tenir un, dos o tres arguments per tal d'obtenir la ubicació actual del dispositiu. Si ha tingut èxit, el *successCallback* serà invocat per tal de que mitjançant la operació *handleEvent*, retorni a l'objecte que l'ha invocat la posició actual del dispositiu. Si per contra no ha pogut esbrinar la ubicació, serà invocat el *errorCallback* amb un objecte *PositionError* nous per tal de reflectir la raó de l'error.

La implementació del mètode *getCurrentPosition* ha d'executar el següent nombre de passos enumerats a continuació:

1. Realitzar el següents passos previs:
 - a. Si *SuccessCallback* té un valor *null* es considera que la conversió a *PositionCallback* ha fracassat i no s'efectuaran la resta de passos.
 - b. Si el paràmetre *PositionOptions* està present i el seu atribut *maximumAge* està definit com un valor no negatiu, s'assignarà aquest valor a la variable interna *maximumAge*. Si està definit com un valor negatiu o no està especificat, s'assignarà a la variable interna un zero.
 - c. Si el paràmetre *PositionOptions* està present i el seu atribut *timeout* està definit com un valor no negatiu, s'assignarà aquest valor a la variable interna *timeout*. Si està definit com un valor negatiu, s'assignarà a la variable interna un zero, i si no està especificat un infinit.
 - d. Si el paràmetre *PositionOptions* està present i el seu atribut *enableHighAccuracy* està definit, s'assignarà aquest valor a la variable interna *enableHighAccuracy*. En qualsevol altre cas aquesta variable estarà a *false*.
2. Si un objecte *Position* guardat a la memòria cache té una edat que no supera el valor de la variable *maximumAge* i està disponible, s'invocarà el *successCallback* per tal d'introduir-lo com un paràmetre, sense efectuar la resta de passos.
3. Si el valor de la variable *timeout* és zero, s'invocarà el *errorCallback* amb l'objecte *PositionError* nou i el seu atribut *code* tindrà el valor del *timeout*, sense efectuar la resta de passos.
4. Iniciar una operació d'adquisició d'ubicació, amb la possibilitat de prendre el valor de la variable *enableHighAccuracy*.
5. Activar un timer que després d'un nombre de mil·lèssegons especificats, pel valor de la variable *timeout*, acabarà cancel·lant qualsevol comanda d'adquisició de localització vinculada amb aquesta instància de passos. Seguidament s'invocarà el *errorCallback* amb l'objecte *PositionError* nou i el seu atribut *code* tindrà el valor del *timeout*, sense efectuar la resta de passos.
6. Si la operació s'ha completat amb èxit abans que el *timeout* expiri, es cancel·larà el *timer* pendent i s'invocarà el *successCallback* amb un nou objecte *PositionError* que reflectirà el resultat de l'operació d'adquisició, sense efectuar la resta de passos.
7. Si la operació falla abans que el *timeout* expiri, es cancel·larà el *timer* pendent i s'invocarà el *errorCallback* amb l'objecte *PositionError* nou i el seu atribut *code* tindrà el valor de *Position_Unavailable*.

PositionOptions Interface

A continuació, l'atribut *timeout* especifica el màxim temps en mil·l·segons que ha d'esperar el mètode *getCurrentPosition* o *watchPosition* fins que el *successCallback* corresponent sigui invocat. Si la posició no s'ha adquirit abans que el *timeout* finalitzi, encara que no hagin altres errors, aleshores s'invoca el corresponent *errorCallback* amb l'objecte *PositionError* que contindrà a l'atribut *code* el valor *timeout*. A més a més, si el paràmetre *PositionOptions* no apareix als mètodes, l'atribut *timeout* tindrà valor infinit, o si té un valor negatiu, serà considerat zero.

Es de suposar que si fem servir el mètode *getCurrentPosition*, el *errorCallback* s'invocarà com a molt una única vegada. D'altra banda si fem servir el *watchPosition*, el *errorCallback* s'invocarà repetides vegades: El primer *timeout* fa referència al moment que ha estat cridat el mètode, és a dir el moment en que s'ha obtingut el permís de l'usuari. Els *timeout* posteriors fan referència al moment en que la implementació determina que hi ha hagut un canvi de posició en el dispositiu, i per tant, s'ha d'adquirir una nova posició.

Per últim, l'atribut *maximumAge* indica que l'aplicació acceptarà una posició emmagatzemada a la memòria cache d'edat no superior a un temps especificat, si no la pot aconseguir, n'adquirirà una de nova. D'altra banda, Si l'atribut té valor zero, l'aplicació intentarà obtenir una nova posició, i si per contra té valor infinit, aquesta obtindrà la posició de la memòria cau independentment de la seva edat. I si el paràmetre *PositionOptions* és omès dels mètodes, el valor per defecte de l'atribut serà 0, malgrat es doni el cas que sigui negatiu.

Coordinates Interface

Els atributs *latitude* i *longitude* [15] són les coordenades geogràfiques en graus decimals. A més a més, tenim els següents atributs addicionals tal com s'observa a la *figura 37*:

```
interface Coordinates {
    readonly attribute double latitude;
    readonly attribute double longitude;
    readonly attribute double altitude;
    readonly attribute double accuracy;
    readonly attribute double altitudeAccuracy;
    readonly attribute double heading;
    readonly attribute double speed;
};
```

Figura 37. Interface Coordinates [15].

- *Altitude*: Denota l'altura de la posició en metres per sobre de l'el·lipsoide. Si no es pot proporcionar, el valor serà *null*.
- *Accuracy*: Denota la precisió de la longitud i la latitud en metres. Mai serà un nombre negatiu.
- *AltitudeAccuracy*: Denota la precisió de l'altitud en metres. Si no es pot proporcionar, el valor serà *null* i mai un nombre negatiu.
- *Heading*: Denota la direcció del moviment del dispositiu en graus decimals en forma $0^\circ \leq \text{heading} \leq 360^\circ$, tenint en compte el sentit horari prenent com a referència el nord. Si no es pot proporcionar, el valor serà *null* i si el valor de l'atribut *speed* és zero (dispositiu estàtic), aleshores serà de valor NaN.
- *Speed*: Denota la velocitat actual del dispositiu en metres partit per segon (m/s). Si no es pot proporcionar, el valor serà *null* i mai un nombre negatiu.

Annex 3: Captures del codi

A continuació, es mostren les captures de pantalla d'algunes funcions secundàries.

```
function errorHandler(err) {
  if(err.code == 1) {
    alert("Error: Accés denegat!");
  }else if( err.code == 2) {
    alert("Error: Posició desconeguda!");
  }
}
```

Figura 38. Estructura de la funció *errorHandler*.

```
function inicializar(position) {

  if (aux==entradaXML[0]) {
    location.href="final.jsp";
  }
}
```

Figura 39. Estructura de la funció *inicializar*.

```
if (j==1) {

  var myOptions = {
    zoom: 16,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    streetViewControl: false,
    mapTypeControl: false
  };

  map = new google.maps.Map(document.getElementById("map"), myOptions);
}

if (j==0) {
  marcador.setMap(null);
}

marcador = new google.maps.Marker({
  position: myLatLng,
  map: map,
  icon: 'imatges/personas.png'
});
```

Figura 40. Estructura de la funció *inicializar*: Càrrega del mapa i marcador.


```

for (i=0;i<entradaXML[0];i++)
{
    eventos[i]=new Array(4);

    eventos[i][0] = entradaXML[(i*3)+1];
    eventos[i][1] = entradaXML[(i*3)+2];
    eventos[i][2] = entradaXML[(i*3)+3];
    eventos[i][3] = "imatges/estrella.png";
}

```

Figura 41. Estructura de la matriu d'events.

```

var marcadores = new Array(entradaXML[0]);

for (var i = 0; i < entradaXML[0]; i++){

    marcadores[i] = new google.maps.Marker({
    position: new google.maps.LatLng(eventos[i][1], eventos[i][2]),
    //map: map, Creem el marcador però no l'afegim al mapa
    title: ' ',
    icon: eventos[i][3]
    });
}

```

Figura 42. Creació del vector marcadors de *GoogleMaps*.

```

function calcRoute() {

var start = myLatLng;
var end = evento;
var request = {
    origin:start,
    destination:end,
    travelMode: google.maps.DirectionsTravelMode.WALKING
};

var directionsService = new google.maps.DirectionsService();

directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
        directionsDisplay.setDirections(response);
    }
});
}

```

Figura 43. Estructura de la funció *calcRoute*.

```

function anteriores(){

    var anterior = new Array(aux);

    for (var i = 0; i < aux; i++){

        anterior[i]=new google.maps.Marker({
        position: new google.maps.LatLng(eventos[i][1], eventos[i][2]),
        map: map,
        title: ' ',
        icon: "imatges/bandera.png"
        });

        anterior[i].setMap(map);
    }
}

```

Figura 44. Estructura de la funció *anteriores*.

