



UNIVERSITAT
POMPEU FABRA

ESCOLA SUPERIOR POLITÈCNICA
ESTUDIS D'INFORMÀTICA

Projecte Fi de Carrera

QuesTInSitu: mejora del sistema de autoría y móvil para la creación, realización y monitorización de rutas abiertas de preguntas geolocalizadas

David Pérez Calle

Curs 2012-2013

ENGINYERIA
EN INFORMÀTICA

Director:
Patricia Santos Rodríguez

Agradecimientos

Quiero dejar testimonio y dedicar mi proyecto final de carrera en primer lugar a mi familia pues sin ella no hubiera llegado hasta aquí. También quiero dedicárselo muy en especial a mi tutora Patricia Santos Rodríguez por su gran apoyo todos estos últimos años y por su gran trabajo incondicional durante la elaboración de este proyecto, siendo la mejor tutora que haya podido desear. También agradecer a todos los miembros del grupo de investigación de tecnologías interactivas (GTI) de la Universitat Pompeu Fabra por colaborar en los diferentes ámbitos del proyecto, tanto de soporte técnico, logístico, como así también el personal, participando en las diferentes experiencias orquestadas para mostrar la herramienta.

Finalmente quiero agradecer también el apoyo de las siguientes instituciones:

This work has been supported by the scholarship Colab 2011-2012 by Agència de Gestió d'ajuts Universitaris i de Recerca de la Generalitat de Catalunya

Este trabajo ha sido realizado gracias al apoyo de la beca Colab 2011-2012 por la Agència de Gestió d'ajuts Universitaris i de Recerca de la Generalitat de Catalunya

Aquest treball ha estat possible gràcies a la beca Colab 2011-2012 de l'Agència de Gestió d'ajuts Universitaris i de Recerca de la Generalitat de Catalunya

This work has been supported by the Interactive Technology Group of the Universitat Pompeu Fabra (GTI), thanks to a grant from Department of Collaboration Information and Communication Technologies (DTIC) of UPF.

Este trabajo ha sido realizado dentro del Grupo de Tecnologías Interactivas de la Universitat Pompeu Fabra (GTI), gracias a la beca de colaboración del Departamento de Tecnologías de la Información y Comunicación (DTIC) de UPF.

Aquest treball ha estat realitzat dins el Grup de Tecnologies Interactives de la Universitat Pompeu Fabra (GTI), gràcies a la beca de col·laboració del Departament de Tecnologies de la Informació i Comunicació (DTIC) de UPF.

This work has been supported by the Spanish Science and Innovation Ministry, EEE Project TIN2011-28308-C03-03.

Este trabajo ha sido apoyado por el Ministerio de Innovación y Ciencia, EEE Project TIN2011-28308-C03-03.

Aquest treball ha estat recolzat per el Ministeri d'Innovació i Ciència, EEE Project TIN2011-28308-C03-03.

*This work has been supported by the Spanish Science and Innovation Ministry, Learn3
TIN2008-05163.*

*Este trabajo ha sido apoyado por el Ministerio de Innovación y Ciencia, Learn3 TIN2008-
05163.*

Aquest treball ha estat recolzat per el Ministeri d'Innovació i Ciència, Learn3 TIN2008-05163.

Muchas gracias a todos.

Resum

Aquest projecte té com a objectiu estendre el treball realitzat amb l'eina QuesTInSitu, que s'emmarca dins del món de les Tecnologies de la Informació i Comunicació (TIC) i concretament en l'àrea d'*E-Learning*. Es presenta una aplicació Web, QuesTInSitu, la qual és una eina d'autoria que permet crear preguntes geolocalitzades a sobre de mapes de *Google Maps*. Aquestes preguntes segueixen l'especificació *IMS Question & Test Interporability (QTI)* i són gestionades pel motor de *QTI NewApis*.

L'usuari pot crear preguntes geolocalitzades i organitzar-les com a rutes (qüestionaris) sobre un mapa de qualsevol punt del món. Per una altra banda, s'ofereix la possibilitat de respondre a les preguntes geolocalitzades mitjançant mòbils 3G gràcies a una aplicació especialment dissenyada per a dispositius mòbils on els usuaris poden respondre les preguntes i veure la puntuació.

Aquest PFC presenta nous aspectes de millora sobre l'eina ja existent, com per exemple: un nou sistema de monitorització, un nou sistema de rutes, noves funcionalitats tant de l'aplicació mòbil com de l'aplicació web entre d'altres. Aquests nous aspectes s'han avaluat dins de nous escenaris educatius.

Resumen

Este proyecto se enmarca dentro del mundo de las Tecnologías de la Información y Comunicación (TIC) y concretamente en el área del *E-Learinng*. Se presenta una aplicación Web, QuesTInSitu, la cual es una herramienta de autoría que nos permite crear preguntas geolocalizadas sobre mapas de *Google Maps*. Estas preguntas siguen la especificación *IMS Question & Test Interoperability (QTI)* y están gestionadas por el motor de *QTI NewApis*.

El usuario puede crear preguntas geolocalizadas y organizarlas como rutas (cuestionarios) sobre un mapa de cualquier punto del mundo. Por otra parte, se ofrece la posibilidad de responder a las preguntas geolocalizadas mediante móviles 3G gracias a una aplicación especialmente diseñada para dispositivos móviles donde los usuarios pueden responder, preguntar y ver la puntuación.

Este PFC presenta nuevos aspectos de mejora sobre la herramienta ya existente, como por ejemplo: un nuevo sistema de monitoreo, un nuevo sistema de rutas, nuevas funcionalidades tanto de la aplicación móvil como de la aplicación web entre otros. Estos nuevos aspectos se han evaluado dentro de nuevos escenarios educativos.

Abstract

This project is part of Information and Communication Technologies (ICT) world and in particular the *E-Learning* area. It presents a Web application QuesTInSitu, it is an authoring tool to create geolocated questions on *Google Maps*. The questions follow the *IMS Question & Test Interoperability (QTI)* specification and are managed by the engine *NewApis QTI*.

The user can create geolocated questions and organize them as routes (questionnaire) on a map as anywhere of the world. In other hand, we offer the possibility to answer the geolocated questions using 3G Mobiles with a specially designed for Mobile phones application where users can answer questions and see the score.

This project presents new aspects of PFC improves on existing tools, such as a new monitoring system, new route system, new features both the mobile application and Web application among others improves. These new aspects are evaluated in new educational scenarios.

Índice

1. Introducción	11
1.1 Contexto y justificación del trabajo	11
1.2 Contexto Tecnológico	12
1.3 Motivación y propósitos del proyecto	13
1.4 Objetivos del proyecto	14
1.5 Puntos de trabajo	14
1.5.1 Mejora sistema de geolocalización de QTISM	14
1.5.2 Mejorar sistema monitorización de la aplicación web	14
1.5.3 Creación de un sistema para poder escoger rutas	15
1.5.4 Poder escoger no contestar una pregunta con el dispositivo móvil	15
1.5.5 Contestar preguntas con el dispositivo móvil de forma no secuencial	15
1.5.6 Mejorar la visualización en el dispositivo móvil del mapa	15
1.5.7 Poder generar archivos XML para encapsular datos en la aplicación Web	15
1.6 Planificación del proyecto	16
1.7 Breve descripción de los demás capítulos de la memoria	23
2. Estado del arte	26
2.1 M-Learning	26
2.2 Aprendizaje fuera del aula: Smartphones + GPS + Internet	28
2.2.1 GPS en <i>smartphones</i>	29
2.2.2 Sistemas de información geográfica (SIG)	32
2.2.3 Google Maps	34
2.2.4 HTML5	37
2.2.5 IMS QTI	38
3. Diseño, implementación, arquitectura y uso de QuesTInSitu	41
3.1 Diseño y casos de uso	41
3.2 Arquitectura e implementación	48
3.2.1 Elementos de la aplicación	48
3.2.2 Interconexión de elementos	52
3.2.3 Funciones más importantes de QTIS	57
3.2.4 Funciones más importantes de QTISM	64
3.2.5 Procesado de la información mediante XML	68
3.3 Base de datos	70
3.3.1 Documentación de la base de datos	70
3.3.2 Base de datos de <i>NewApis</i>	70
3.3.3 Base de datos de QTIS y QTISM	73
3.4 Interfaz y clases relacionadas	75
3.4.1 Secciones más importantes de QTIS	75
3.4.2 Pantallas más importantes de QTISM	89
3.5 Actividades realizadas con QuesTInSitu en 2012	96
4. Conclusiones	102
5. Trabajo futuro	105
5.1 Futuras mejoras generales para la aplicación	105
5.2 Futuras mejoras técnicas para la aplicación	108
6. Bibliografía	110
7. Anexos	115
7.1 Manual de Instalación	115
7.1.1 Manual de Instalación de QTIS	115
7.1.2 Requisitos para la instalación de QTIS	115
7.1.3 Pasos a seguir para la instalación de QTIS	115
7.2 Manual de Instalación de QTISM	117
7.2.1 Requisitos para la instalación de QTISM	117
7.2.2 Pasos a seguir para la instalación de QTISM	118
7.3 Manual para obtener una <i>Key</i> de <i>Google Maps</i>	119

1. Introducción

1.1 Contexto y justificación del trabajo

QuesTInSitu es una herramienta de autoría que nos permite crear preguntas geolocalizadas sobre mapas de *Google Maps*. Estas preguntas siguen la especificación *IMS Question & Test Interoperability (QTI)* y están gestionadas por el motor de *QTI NewApis*.

El usuario puede crear preguntas geolocalizadas y organizarlas como rutas (cuestionarios) sobre un mapa de cualquier punto del mundo. Por otra parte, se ofrece la posibilidad de responder a las preguntas geolocalizadas mediante móviles 3G gracias a una aplicación especialmente diseñada para dispositivos móviles donde los usuarios pueden responder, preguntar y ver la puntuación.

La idea principal es realizar rutas de preguntas geolocalizadas con propósitos de *assessment* (evaluación). Así, al acceder a una zona donde hay una pregunta a contestar, esta aparece en el dispositivo para que el usuario la responda. La consecución de este tipo de preguntas constituye una ruta, la cual se puede tomar como un examen. Mientras, el profesor puede monitorizar la actividad y comprobar in situ los resultados que van obteniendo los alumnos [1].

Este proyecto parte como continuación de dos proyectos finales anteriores realizados dentro del Grupo de Tecnologías Interactivas (GTI) de la UPF. Por una parte, el proyecto final de carrera que realicé en mis estudios de Ingeniería Informática Técnica en Sistemas, presentado en el año 2010 con el título: “QuesTInSitu: Aplicació per a crear preguntes QTI geolocalitzades” [2]. Por otra parte, también extiende el trabajo realizado en el proyecto de Ferran García García, presentado en el año 2011 y centrado en la aplicación de QTISM (versión móvil de QuesTInSitu) [3].

QuesTInSitu se ha evaluado en contextos reales mediante la orquestación de varias actividades, con diferentes institutos y eventos como ferias o premios (dentro del marco de las investigaciones del GTI). Algunos de los resultados obtenidos demostraron que la aplicación presentaba algunas carencias y era necesario dar soporte a algunos casos (como por ejemplo: monitorear la actividad en tiempo real, o editar una ruta abierta de preguntas, así como mejorar algunos aspectos de visualización en la interfaz móvil). También es necesario subrayar que respecto el año 2010, han salido un abanico de nuevas tecnologías y posibilidades que, adaptadas al proyecto, pueden dar soporte a los problemas encontrados y han sido también motivo por el cual reemprender el desarrollo de la herramienta.

Este proyecto parte con la idea de dar soporte a la evaluación de conocimiento in situ, mediante el uso de preguntas geolocalizadas y los beneficios que nos pueden aportar las nuevas tecnologías, en este caso los *smartphones* y las redes de Internet 3G. Estas tecnologías permiten orquestar actividades al aire libre de forma más eficiente, tanto para los alumnos como para el profesor, respecto a cómo lo venían haciendo habitualmente.

1.2 Contexto Tecnológico

Este proyecto está enmarcado en un momento histórico, donde las redes de telecomunicación han evolucionado a entornos abiertos y los dispositivos se han adaptado a estas nuevas situaciones. Esto ha provocado que los dispositivos de telecomunicación se hayan transformado tanto física como digitalmente incorporando nuevos Gadgets como antenas de recepción de datos para poder conectarse a dichas redes, o la masiva proliferación de aplicaciones para este tipo de dispositivos. Este fenómeno ha provocado una revolución en este campo tecnológico, que ha promovido la proliferación de nuevas herramientas soportadas con estas nuevas posibilidades como ejemplo este mismo proyecto.

Otra revolución también se produce en la aparición de nuevos conceptos como puede ser el de *m-learning*. El cual tiene como tarea dar soporte mediante los nuevos dispositivos tecnológicos móviles al mundo de la educación, como puede ser el caso de la herramienta que aquí se presenta. Este concepto nos posibilita poder diferenciar un nuevo tipo de aplicaciones con un fin específico y diferenciado con su propio campo de trabajo.

En la actualidad nos encontramos en una época donde la sociedad demanda nuevas formas de acceso a la información y de aprendizaje muy diferentes a las utilizadas tradicionalmente, lo que ha propiciado una revolución en el ámbito de la educación soportada tecnológicamente. Dado que estas nuevas formas de aprendizaje necesitan también de formas de evaluar actividades docentes. A través de este proyecto se quiere presentar una herramienta que utilice estas nuevas tecnologías (tecnología 3G y de geoposicionamiento) para el ámbito de la evaluación *in situ*.

Con el uso de las TIC (Tecnologías de la información y la comunicación) en entornos educativos y de aprendizaje, aparecen instituciones para reglamentar y estandarizar los métodos más adecuados para poder construir las futuras herramientas educativas. Un ejemplo es el *IMS Global Learning Consortium* [13]. Su principal tarea, es la de estandarizar modelos computacionales, basados en métodos de soporte al aprendizaje... En el área de soporte a la evaluación, IMS propone el estándar IMS QTI [13], dedicado a regular procesos de evaluación en plataformas tecnológicas mediante el uso de test. Este estándar, QTI, es el utilizado en QesTInSitu para representar computacionalmente los test y preguntas creados por la aplicación.

También es necesario destacar los apreciables avances de otras tecnologías que utiliza QesTInSitu como son los sistemas tecnológicos referentes a la cartografía, también conocidos como SIG (Sistema de Información Geográfica), la aparición revolucionaria de sistemas como la aplicación Google Maps (junto la posibilidad de poder desarrollar software gracias a su API) y el hecho que los dispositivos móviles tengan también GPS y esto posibiliten su posicionamiento en el espacio.

1.3 Motivación y propósitos del proyecto

Tal y como se indica en el apartado 1.1 este proyecto tiene como punto de partida los proyectos realizados por Ferran García García en 2011 y el que realicé en los estudios de Ingeniería Técnica de Informática de Sistemas presentado en 2010.

Gracias a este hecho, este proyecto puede partir con dos herramientas ya construidas como son *QuestInSitu* (QTIS), relacionada con toda la operativa desde dispositivos de sobremesa, y *QuestInSitu Mobile* (QTISM), su versión para dispositivos móviles.

La aportación principal de este proyecto final de carrera es la de continuar un trabajo que se entendía que aún no estaba completado pese al trabajo realizado anteriormente. Por ello se ha querido tratar campos que totalmente nuevos y que son el propósito de este proyecto:

- **Tracking de datos:** Un propósito muy antiguo y pilar fundamental de este proyecto era poder ser capaces de rastrear y memorizar la actividad realizada con los dispositivos móviles en tiempo real. En el proyecto actual se utiliza un sistema de rastro que refleja la realidad de la actividad, de modo que los profesores puedan monitorizar el progreso de los estudiantes.
- **Poder manejar las rutas:** Se ha intentado mejorar aspectos como la edición de rutas pudiendo escoger entre siendo abiertas en vez de en un orden específico. Ya que este punto fue señalado por los profesores, en evaluaciones anteriores, diciendo que preferían que las preguntas se pudieran contestar en el orden deseado por los estudiantes. Esos cambios se han visto reflejados tanto en QTIS como QTISM
- **Mejora interfaz:** Muchas pantallas y presentaciones debían de adaptarse a los nuevos propósitos y mejoras que se creían necesarias, lo cual se ha mejorado en este proyecto.
- **Encapsulación de datos:** Los datos en la aplicación anterior se gestionaban de manera poco estandarizada. En este nuevo proyecto los datos se encapsulan en determinados scripts para así agilizar el uso a los administradores de la herramienta. Con ello se ofrece un abanico de automatismos que mejoran mucho este escenario y también se mejoran las estructuras de datos que se guardan con estándares como ficheros “.xml”.

Las evaluaciones realizadas posteriormente al desarrollo de la herramienta anterior, nos dieron a ver que sería bueno dotar a la herramienta de nuevas posibilidades cómo puede ser que las rutas en el dispositivo móvil se pudieran escoger de forma dinámica y autónoma, poder realizar las rutas sin la necesidad de que estas se tengan que resolver de forma correlacionada o mejoras de la interfaz del mapa, así como tener un control sobre la actividad realizada en tiempo real, entre otras necesidades que fueron surgiendo a medida que avanzó este proyecto. Con ello pretendemos potenciar la herramienta y hacerla aún más atractiva tanto para los estudiantes como para los profesores. Las mejoras conseguidas en este proyecto y las intenciones de este se recogen en los apartados posteriores, *1.4 Objetivos del proyecto* (propósitos de este proyecto) y *1.5 Puntos de trabajo* (mejoras realizadas respecto al trabajo anterior).

Estos han sido algunos de los aspectos por los cuales se ha decidido reprendre el trabajo ya realizado y mejorarlo para obtener una herramienta mejor.

1.4 Objetivos del proyecto

En este apartado se explican las intenciones que se tienen con este proyecto.

Durante el período en que se realizó el primer proyecto en 2010 hasta la actualidad se ha ido mejorando y evolucionando la herramienta según las conclusiones obtenidas con las pruebas con usuarios y con las nuevas tecnologías que se han añadido para obtener un propósito mejorado de herramienta educativa.

Después de estos dos trabajos, se ha visto que aún era necesario ampliar la herramienta con nuevas funcionalidades que se detallan en el apartado siguiente (1.5 Puntos del trabajo). Estos objetivos nacen como justificación al interés que puede promover incluir estas novedades para el proyecto y el impacto que estas pueden tener, aportando nuevas mejoras generales, como puede ser mejores modelos de persistencia o geolocalización, así también objetivos específicos como añadir nuevas características a la interfaz de QTISM (menús, nuevos iconos, ...)

1.5 Puntos de trabajo

A continuación, se detallan los puntos marcados como objetivos para mejorar la herramienta y que se pretenden poder solventar con este proyecto y que hemos citado en cuatro grupos en el apartado anterior 1.4.

Realizar tracking de datos:

1.5.1 Mejora sistema de geolocalización de QTISM

La mejora más importante y que ha propiciado otras mejoras, se produce en los sistemas de geolocalización, ya que con esta nueva versión, el sistema es capaz de memorizar las diferentes posiciones y usuarios que están interactuando, siendo capaz de localizar al usuario exactamente en qué punto se encuentra y qué recorrido ha realizado desde que ha encendido la aplicación. Todo esto se realiza mediante la memorización de todos los TICs en un archivo “.txt”, que va escribiendo todos estos datos para su posterior trato.

1.5.2 Mejorar sistema monitorización de la aplicación web

En la versión anterior del aplicativo de QTIS para Web la aplicación monitorizaba la posición de los usuarios según si habían o no respondido alguna pregunta dando como posición la misma posición de la pregunta. Esto ya no es así, ahora la geolocalización es en tiempo real, donde el usuario va enviando su geoposición y esta se va registrando en un fichero que se puede *renderizar* en un mapa de monitorización de la web de QTIS.

Nuevas posibilidades de poder manejar las rutas:

1.5.3 Creación de un sistema para poder escoger rutas

En la versión anterior del aplicativo de QTIS para dispositivos móviles (QTISM) la aplicación tenía que tener una ruta ya predispuesta por el administrador y preparada específicamente para una actividad en concreto. Con esta actualización, es el propio usuario que puede escoger que actividad o ruta desea realizar de forma automática, sin que sea necesaria la intervención de un administrador.

La antigua versión también tenía el inconveniente de que la actividad o ruta era para todos los dispositivos móviles, mientras que con la actualización las actividades son individualizadas y los diferentes usuarios pueden realizar diferentes actividades a la vez.

Nuevas mejoras interfaz:

1.5.4 Poder escoger no contestar una pregunta con el dispositivo móvil

En la versión anterior del aplicativo de QTIS para dispositivos móviles (QTISM) cuando aparecía una pregunta a contestar tan sólo se tenían dos posibles opciones, poder contestar la pregunta o dejarla sin contestar, lo que imposibilitaba poder finalizar la ruta. Con esta nueva opción el usuario, si no sabe la pregunta y no desea restar puntos, puede escoger no contestarla.

1.5.5 Contestar preguntas con el dispositivo móvil de forma no secuencial

La versión anterior del aplicativo de QTIS para dispositivos móviles (QTISM) sólo se permitía un solo recorrido para realizar la ruta o actividad, obligando al usuario a seguir un orden preestablecido de la ruta. Con esta nueva versión el usuario no está obligado a seguir ningún orden y al acercarse a un punto caliente de la ruta donde se encuentra una pregunta la puede contestar en ese momento si lo desea.

1.5.6 Mejorar la visualización en el dispositivo móvil del mapa

Respecto a la versión anterior del aplicativo de QTIS para dispositivos móviles (QTISM) se han añadido nuevos iconos para determinar si una pregunta está pendiente de resolver o si ya ha sido resuelta y de forma favorable, desfavorable o obviada por el usuario.

Nuevas mejoras en encapsulación de datos:

1.5.7 Poder generar archivos XML para encapsular datos en la aplicación Web

Una de las últimas mejoras del aplicativo de QTIS para dispositivos móviles (QTISM) es que ahora la aplicación permite poder salvar los datos en archivos XML sobre orden de rutas y preguntas de una forma más regularizada.. Esto permite guardar la configuración de una ruta de

forma que el profesor puede editar con más exactitud y conocimiento del entorno y así generar una ruta más atractiva y eficaz.

1.6 Planificación del proyecto

La planificación del proyecto se llevó a cabo desarrollando los puntos que se enuncian a continuación:

1.6.1 Documentación sobre el proyecto

La documentación generada dentro del proyecto es de vital importancia para el mismo, en especial en proyectos de cierta complejidad, de larga duración o donde intervienen equipos numerosos de personas. Su correcta gestión evita duplicar innecesariamente los documentos, y permite dar a conocer a los integrantes del equipo datos como por ejemplo: cuál es la última versión, donde se encuentra localizada, cómo se instala, donde obtener los recursos,...., etc. Pero la documentación de un proyecto no solo es importante para la vida del mismo.

Su utilidad trasciende más allá de la conclusión del proyecto que la generó, sirviendo como referencia y base de conocimiento ante nuevos proyectos de similar naturaleza, o cuando cambie alguno de los integrantes del equipo de trabajo.

En esta primera fase inicial de construcción del proyecto adquiere un papel muy importante, puesto que se trata de la base sobre la cual se basaran todos los conocimientos para desarrollar el proyecto y elaborar las múltiples funcionalidades que puede ofrecer. Algunos de los pasos más importantes de este apartado son:

- Aprender conceptos teóricos y básicos para el desarrollo de la herramienta como el concepto *E-Learning*, *M-Learning*, el mundo de las tecnologías TIC, o saber cómo funciona un estándar de aprendizaje como QTI.
- Buscar en la red proyectos de la misma temática, para evaluar su funcionalidad y los recursos que utilizan, como los portales web: *www.panoramio.com* (web de diseño mapas de *Google Maps* con imágenes) o *www.wikiloc.com* (web de diseño de mapas de *Google Maps* que contiene rutas).
- Comparar la situación del mercado, realizar estudios sobre las funcionalidades que ofrecen las otras aplicaciones móviles del mercado y que puede aportar a la nuestra de diferente y beneficiosa. Para ello sólo es necesario observar el software que se ofrece en sitios de descargar de aplicaciones para *smartphones* como el market de Android o Apple Store [14] de la empresa Apple en este sentido.
- Analizar y comparar el proyecto con otros proyectos que se han realizado en la universidad de temática similar. Mediante la búsqueda y evaluación de los trabajos realizados años anteriores por otros alumnos y de los trabajos realizados por grupos de investigación de la universidad en concreto del Grupo de Tecnologías iterativas (GTI).
- Documentarme los recursos útiles para realizar la herramienta.

Este punto de la planificación se llevó a cabo al inicio del proyecto como punto de salida.

1.6.2 Conocimiento y estudio de las herramientas

Durante el desarrollo del proyecto hay que dedicar tiempo de este al aprendizaje y uso de todas las herramientas que utiliza el proyecto de forma ampliada y de las que se hace referencia en el apartado anterior.

Algunas de las herramientas y aplicaciones más importantes requeridas para el desarrollo del proyecto son las siguientes:

- Escoger un IDE para el desarrollo de la herramienta (**Eclipse para *Dynamic Web Project***).
- Determinar qué tipo de estándar se usará para la enseñanza. En este caso el más adecuado para la herramienta es **IMS Question and Test Interoperability (QTI)** [13].
- Descubrir las opciones que proporciona un servidor como **NewAPIS** para servir los diversos test del mapa.
- Aprender la **API de Google Maps**, para poder gestionar los mapas de la forma más interesante para la herramienta.
- Conocer que son los **Servlets**, el lenguaje **JSP** y **JavaScript** para implementar el portal web con información dinámica.
- Saber cómo confeccionar una **base de datos** para tener toda la información relacionada tanto la del portal web como la del servidor *NewApis*.

Este apartado se lleva a cabo posteriormente del apartado 1.3.1 (Documentación sobre el proyecto). Durante la producción de material útil para la confección de la herramienta es muy importante crear borradores y hacer pruebas para conocer todas las posibilidades que podemos obtener de las herramientas editoras como por ejemplo Eclipse, ya que no tener un buen conocimiento de las herramientas puede repercutir en la calidad final del resultado.

La calidad del producto es muy importante ya que puede determinar si nuestro proyecto tendrá éxito en el futuro.

1.6.3 Determinar los objetivos

Como se ha explicado detalladamente en los apartados 1.4 y 1.5, este punto ha sido clave en la planificación y elaboración del proyecto, pues han sido el conjunto de pasos los que han ayudado a ir proponiendo los pasos en la elaboración de la aplicación punto por punto y en un tiempo determinado. Este punto se ha estado realizando durante el transcurso de todo el proyecto, pues durante la confección del mismo, se han ampliado partes o se ha tenido que hacer otros pasos para llegar a la solución correcta.

1.6.4 Análisis de requerimientos y diseño

En este apartado se desarrollan y llevan a estudio las necesidades que pueden requerir los diferentes roles de usuarios y así determinar el diseño más adecuado para desarrollar determinada tarea. Por otra parte se debe evaluar la herramienta mediante criterios de uso, para determinar si las funcionalidades y la forma de mostrar la información, el diseño, la accesibilidad a la información son adecuadas para los usuarios. Este apartado se desarrolla una vez se han determinado todos los objetivos a desarrollar.

Unos ejemplos de requerimientos y diseños aplicados a este proyecto pueden ser:

- Un gestor de rutas en el mismo mapa, para facilitar la edición de las rutas con los test de forma más accesible.
- Se determinó la necesidad de crear un sistema de monitorización útil para el monitor de la actividad de forma intuitiva y que fuera verdaderamente in situ.
- Poder cargar rutas desde la aplicación móvil y poder realizar las etapas de la ruta de forma ordenada o desordenada, según como determine el creador de la ruta.
- Diseñar un sistema de monitorización mucho más interactivo con la incorporación de nuevos marcadores, más información y mostrar una información más actualizada y real de la situación de la actividad que se esté desarrollando en la plataforma.

Esta parte de los requerimientos y diseño del proyecto se llevarán a cabo una vez se realicen las pruebas de la parte a evaluar con los usuarios, las cuales determinan si la propuesta es correcta o si es necesario aplicar nuevas mejoras para satisfacer las necesidades que requiere el usuario y la herramienta.

QuesTInSitu ha sido evaluado en diferentes momentos de su creación en evaluaciones periódicas a modo local para probar sus funcionalidades.

1.6.5 Redacción de la documentación del proyecto

La documentación del proyecto es una de las partes más importantes ya que da a conocer a las demás personas el proyecto. Una vez se tiene alcanzada una parte del proyecto y contrastada hay que documentar esta parte de la forma más fácil, para que los usuarios puedan trabajar con ella. Por eso se ha tenido que planificar la forma y agrupación del código en los diferentes paquetes de código según su relación y planificar una memoria que explique el proyecto.

1.6.6 Fases elaboración del proyecto

La elaboración del proyecto se determina en cuatro grandes etapas más una presentación final. Estas son: fase de análisis, fase de diseño, fase de implementación, fase de testeo y entrega del proyecto. A continuación se detalla como se ha enfocado el trabajo en cada una de estas etapas:

Fase de análisis

- Propuesta de temas.
- Selección y argumentación (negociación).
- Activación de conocimientos previos sobre el tema.
- Concreción del que se quiere conocer.
- Elaboración de hipótesis sobre el contenido de aprendizaje.

Fase de diseño

- Toma de decisiones sobre la distribución de tareas.
- Puesta en común de conceptos.
- Estructuración y la comunicación de la información.

Fase de implementación

- Se inicia el proceso de investigación.
- Búsqueda y selección de fuentes de información.
- Se establecen criterios de selección e interpretación de la información.
- Formulación de nuevas preguntas e hipótesis.
- Exploración de las nuevas preguntas e hipótesis.
- Relación con otros problemas.
- Realización de las diversas actividades y propuestas.

Fase de puesta en marcha y testeo final

- Representación del proceso de elaboración del conocimiento que ha seguido.
- Evaluación de lo aprendido.
- Conexión con nuevos temas o proyectos

Una vez definida la planificación para cada etapa, se exponen las operaciones a realizar en cada una de ellas siguiendo la planificación ideada.

Fase de análisis

- Análisis de requerimientos.
- Listar los requerimientos funcionales del sistema.
- Listar los requerimientos de usuarios.
- **Entregable:** BBDD de productos.
- **Entregable:** Lista de requerimientos.
- Revisión de los requerimientos del cliente.
- **Entregable:** Análisis de los requerimientos.
- Documentación.
- Analizar la BBDD.
- Analizar los sistemas ya existentes.
- Definir los lenguajes de programación del sistema.
- **Entregable:** Documentación.
- Aprendizaje de Tecnología.
- Estudio API de *Google Maps*.
- Estudio del lenguaje JSP.
- Estudio del lenguaje de Java.
- Estudio del lenguaje de JavaScript.
- Estudio del lenguaje de MySQL.
- Estudio del lenguaje de HTML.
- Estudio del lenguaje DAO.
- Estudio de los dispositivos 3G.
- Reunión con el tutor.

Fase de diseño

- Diseño del modelo lógico del sistema.
- **Entregable:** Modelo lógico.
- Diseño del modelo físico del sistema.
- **Entregable:** Modelo físico.
- Diseño de la aplicación web.
- Diseño de la interfaz.
- Diseño de la BBDD.
- Entregable: Modelo ER de la BBDD.
- Reunión con el cliente.
- Diseño de los bloques.
- Diseño de la aplicación móvil.
- **Entregable:** *Mock-up* de la aplicación móvil.
- Reunión con el tutor.

Fase de implementación y testeo

- Módulo Páginas web.
- Creación de plantillas web.
- Implementación de la lógica de la web.
- Implementación de la web a diferentes idiomas.
- **Entregable:** Página web.
- Módulo de BBDD.
- Creación de BBDD.
- Inserción de la Tabla Rutas.
- Inserción de la tabla Tests.
- Inserción de la tabla Usuarios.
- Inserción a la BBDD de *NewApis*.
- **Entregable:** Bases de datos.
- Módulo de paquetes de código nuevo.
- Implementación del diseño.
- Implementación de la lógica del bloque.
- **Entregable:** Blogs.
- Módulo de implementación de la aplicación móvil.
- Creación de *Web Services*.
- **Entregable:** Publicar versión operativa para todos los dispositivos.
- Reunión con el tutor.

Fase de puesta en marcha y testeo final

- Módulo web.
- Sistema con la aplicación Web.
- **Entregable:** web.
- Módulo del Bloque.
- Sistema Bloques nuevo código.
- **Entregable:** Bloques.
- Módulo Base de Datos.
- Sistema Base de datos.
- **Entregable:** Base de datos.
- Experiencia.
- Dar de alta a los usuarios.
- Interactuar con la aplicación.
- **Entregable:** Documentación de la experiencia.

Entrega de todo proyecto

- Documentación.
- **Entregable:** Programa.
- **Entregable:** BBDD implementada.
- **Entregable:** Documentación.
- Presentación/exposición al tribunal.

1.6.7 Equipo de trabajo

Para la elaboración del proyecto se disponen de tres diferentes agentes, que son los siguientes:

- **Tutor:** Se encargará de guiar al alumno en las tareas de elaboración del proyecto, marcándole plazos y periodos de entrega. Así como ayudándole con las tareas del proyecto referenciándole fuentes de información que pueda consultar.
- **Estudiante:** Encargado de realizar el proyecto tanto la parte técnica referida a la programación de la herramienta como en su posterior documentación, para finalmente concluir en una exposición final en un tribunal de evaluación.
- **Alumnos y profesores:** Sujetos que prueban el funcionamiento de la aplicación y permiten certificar que esta se puede utilizar en escenarios educativos reales.

1.6.8 Riesgos del proyecto

Por lo que respecta a la tecnología, se usará una programación en lenguajes comúnmente conocidos, debidamente estandarizados y con una buena estabilidad. En cuanto al Hardware, la universidad puede proveer varios servidores en un tiempo asequible, con un Hardware necesario similar a lo requerido o superior y además se dispone de equipos propios con los que ya se puede empezar a poder programar, como ordenadores portátiles.

En caso de destrucción (por incendio, inundación, etc...) de los servidores, el sistema se podrá restaurar en poco tiempo gracias a copias de seguridad que se dispondrán en otros servidores, CDs, copias locales, etc. Para integrar este sistema en nuestra interfaz se deberá conocer cada API. Por tanto, no se puede garantizar que ésta esté libre de errores y que no se pueda hacer el envío. Para ello se ha elaborado un documento técnico para poder dar el conocimiento necesario a futuros programadores para poder proseguir con el trabajo realizado en este proyecto.

Por lo tanto se puede llegar a la conclusión que los riesgos de la planificación no habrán problemas importantes para realizar el proyecto durante las fases hasta la entrega del Hardware.

1.7 Breve descripción de los demás capítulos de la memoria

En este apartado se hará un breve resumen de la temática de los diferentes apartados que componen la memoria del proyecto:

- **Introducción:** En este apartado se pretende explicar de dónde parte el proyecto, en qué contexto tecnológico se ubica y porque se justifica el trabajo realizado. En los diferentes subapartados se explica que trabajo se ha realizado con anterioridad y que trabajo se pretendía hacer y cuáles son las mejoras que se han conseguido. Teniendo en cuenta el enfoque tomado, plazos y requerimientos y estudio sobre la materia. Este apartado trata de realizar una especie de Proyecto de Ingeniería para optimizar el proyecto y obtener un mejor resultado del mismo.
- **Estado del Arte:** Este apartado se centrará en el estado de la tecnología actual y la aparición de nuevas herramientas y conceptos útiles para el desarrollo del proyecto. En concreto se hablará de la tecnología *M-Learning*, ya que la aplicación QTISM es la herramienta que más ha mejorado durante el transcurso de este proyecto. También hay que destacar conceptos teóricos como *Learning*, sin comentar puestos ya expuestos y focalizados en proyectos anteriores como el proyecto presentado en 2010 [3].
- **Diseño, implementación, arquitectura y uso de QuesTInSitu:** Aquí se detalla el cuerpo del proyecto tanto interiormente, refiriéndose a como están construidas las bases de datos y como se relacionan, como explicar la estructura del código, en que paquetes están distribuidos y que funcionalidades tienen cada uno. También se explica la parte visible del proyecto que incluye toda la parte de presentaciones que recibe el cliente. En este apartado se explican las pantallas más importantes de QTIS y QTISM y que operaciones se puede realizar en cada una y de qué manera.

- **Conclusiones:** En este apartado se explican unas breves conclusiones sobre la experiencia y resultados obtenidos en la realización del proyecto.
- **Trabajo futuro:** Una vez realizado el proyecto, se han detectado nuevos problemas y posibilidades que serían buenos añadir en futuros trabajos y que se citan en este apartado.
- **Anexos:** Apartado no propiamente de la memoria, donde se publica un manual para la instalación de todo el software para su puesta en marcha.

2. Estado del arte

A continuación se ofrece una introducción de la situación actual de las nuevas técnicas de docencia. En este caso se trata el aprendizaje de nuevos conceptos como puede ser *M-Learning* y el aprendizaje con *GPS* fuera del aula.

2.1 M-Learning

Hasta hace poco los ordenadores eran los únicos dispositivos utilizados para poder realizar actividades de test soportadas tecnológicamente. El término inglés utilizado para referirse al ámbito que investiga el uso de los ordenadores para realizar actividades de evaluación es el de *Computer Assisted Assessment (CAA)* [15], en español Evaluación Asistida por Ordenador [5]. Uno de los aspectos más investigados es el de utilizar la tecnología para poder diseñar y crear nuevos tipos de preguntas que aumenten la interacción del usuario con la información y que a la vez permitan capturar más datos del usuario. La tendencia es la de integrar elementos y funcionalidades de la web 2.0 para la creación de nuevos tipos de preguntas. Hoy en día los investigadores de éste ámbito están comenzando a estudiar el uso de otros dispositivos tecnológicos como móviles, PDAs, consolas, y periféricos como lectores NFC, GPS y antenas *bluetooth* entre otros. Este tipo de doctrina está cogiendo tal relevancia que usuarios de tipo sénior son capaces de utilizarla, como se puede ver en una de las pruebas que se realizó con QuestInSitu con este tipo de usuarios (ver imagen *Figura 2.1.A M-Learning con usuarios senior*). Esta prueba se realizó con investigadores del grupo de investigación GTI de la Universitat Pompeu Fabra dentro del proyecto Life2.0 - CIP ICT PSP-2009-4-270965 [16].



Figura 2.1.A M-Learning con usuarios senior

El uso de las Tecnologías de la Información y Comunicación (TIC) en la educación permite el diseño y realización de nuevos escenarios educativos. El proyecto QuesTInSitu centra su atención en el uso de la tecnología para apoyar actividades innovadoras de tipo test. El proyecto QuestInSitu es un aplicación educativa que reúne tecnologías como: móvil, servicios web 2.0 como *Google Maps* y aplicaciones como *GPS* para crear rutas de preguntas geolocalizadas.

El uso de dispositivos móviles con propósitos educativos (*Mobile Learning*) está siendo cada vez más aceptado. La ventaja más conocida del uso de estos dispositivos en la educación es el hecho que permiten aprender en cualquier lugar y en cualquier momento, permitiendo que las escuelas exploren nuevos espacios de aprendizaje. Al utilizar dispositivos como los teléfonos móviles, se pueden diseñar actividades de aprendizaje en espacios como un parque, un museo, o las calles de una ciudad.

Teniendo en cuenta estos aspectos QuesTInSitu permite crear preguntas tipo test y geolocalizarlas sobre un mapa de *Google Maps*. Las preguntas pueden ser interactuadas de dos formas distintas: utilizando la aplicación web e interactuando con los mapas (evaluación in virtual situ) o usando teléfonos móviles con conexión a Internet y GPS y respondiendo a las preguntas siguiendo una ruta (evaluación in situ). Con ella se pueden diseñar recursos educativos atractivos que fomenten el aprendizaje y desarrollo de conocimiento y habilidades relacionados con el aprendizaje in situ. QuestInSitu ha sido utilizado en una experiencia real con un grupo de alumnos de 2º de Bachillerato y su profesora para la realización de una actividad in situ.

El concepto de *M-Learning*, ha provocado diversidad de opiniones, sus defensores confían en la posibilidad de poder combinar movilidad y aprendizaje, lo cual puede ser muy positiva para los estudiantes mientras que sus retractores simplemente ven el mismo sistema educativo pero con dos doctrinas alineadas con los atributos únicos de aprendizaje móvil uniendo movilidad al método [7].

Por otra parte, la definición de aprendizaje móvil puede destacar los atributos únicos (movilidad, actividad orquestada, aprendizaje interactivo, valor añadido al aprendizaje, ...) que lo posicionan dentro del aprendizaje informal y no formal. Estos atributos colocan el aprendizaje móvil muy en desacuerdo con la enseñanza formal con sus cursos, evaluaciones, y las escuelas, y con el seguimiento y evaluación de los regímenes. Esto plantea la preocupación por la naturaleza de cualquier despliegue a gran escala y sostenida y el grado en que los atributos únicos de aprendizaje móvil se pueden perder o comprometer, ya que el *M-Learning* puede dar a confusión entre informal y no formal y empeorar el método educativo en vez de mejorarlo como es su propósito.

En cuanto a aprendizaje móvil en un contexto más amplio, tenemos que reconocer que los dispositivos móviles, personales e inalámbricos están transformando radicalmente las nociones sociales de discurso y conocimiento y son responsables de las nuevas formas de arte, el empleo, la privación del lenguaje, el comercio, así como el aprendizaje. Con un mayor acceso popular a la información y al conocimiento en cualquier lugar, en cualquier momento, el papel de la educación, tal vez sobre todo formal, es desafiado y las relaciones entre la educación, la sociedad y la tecnología son ahora más dinámicas que nunca.



Figura 2.1.B Combinación móvil con actividad in real situ

Como se defiende anteriormente, es curioso y beneficioso la combinación que el alumno puede tener de lo que percibe con sus propios sentidos ya sean: vista, oído, olfato,... Y como esto le beneficia para poder interpretar de mejor manera el feedback que recibe de información procedente de la pantalla de su dispositivo móvil [5] y [6]. Como se puede ver en la figura anterior (ver imagen Figura 2.1.B Combinación móvil con actividad in real situ), y así obtener estos beneficios para un mayor aprendizaje y poder relacionar de manera superior el concepto con lo tangible. Todo esto es posible gracias a los beneficios que nos aporta la tecnología *M-Learning*.

2.2 Aprendizaje fuera del aula: Smartphones + GPS + Internet

En la última década han irrumpido los llamados teléfonos de tercera generación que son dispositivos de transmisión de voz y datos a través de telefonía móvil mediante la tecnología de datos UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles). Esto permite, que los antiguos dispositivos de telefonía ahora puedan ser posicionados con mayor facilidad, gracias a la red de datos que incorporan. Esto ha permitido, que estos dispositivos empiecen a incorporar nuevas herramientas como GPS, que ha provocado que estos dispositivos tengan otra funcionalidad, ya que el flujo de información I/O, tanto de entrada como de salida convierte estos dispositivos en fuentes de consulta y información dinámica como podemos ver en los actuales *tablets*, los cuales no dejan de ser un dispositivo móvil con otro enfoque en su concepto comercial y así reemplazar los antiguos cuadernos de texto.

La llegada de los *smartphones* con GPS han capacitado que lugares como un edificio histórico, una flor en caso de botánica,...etc, se conviertan ahora en una aula de estudio para los alumnos. Esto es posible a la combinación dual de los *smartphones* con la tecnología GPS. Por ejemplo la arquitectura software de nuestra aplicación QuestInSitu está formada por diversos módulos. Uno de ellos es el motor que permite interpretar y ejecutar las preguntas y test de QTI. Este motor, *NewAPIS*, permite *renderizar* una pregunta en formato IMS QTI [13] cuando las funciones de HTML5 geoposicionan al alumno cerca de una pregunta que se encuentra gracias al mapa de *Google Maps*, lo cual permite interactuar con esta. En la siguiente figura (ver imagen Figura 2.2.A Aporte nuevas tecnologías a la movilidad), se puede apreciar que gracias a la llegada de nuevas tecnologías a los dispositivos de telefonía, estos pueden derivar en una revolución que va más allá de la comunicación y que abarca más campos como la educación.



Figura 2.2.A Aporte nuevas tecnologías a la movilidad

La imagen anterior se ha extraído de un artículo producido por componentes del grupo de investigación GTI de la Universitat Pompeu Fabra: Santos, P., Pérez-Sanagustín, M., Hernández-Leo, D. & Blat, J. (2012). *Routes of geolocated questions in formal and informal learning contexts. Bulletin of the IEEE Technical Committee on Learning, 14(4), 21-23* [6].

Esto es posible gracias a la llegada de nuevas tecnologías como: la incorporación de GPS en *smartphones*, el auge de las Tecnologías de la información y la comunicación (TIC), actualización de nuevos lenguajes como HTML5, actualización de APIs con nuevas funcionalidades, etc. ... A continuación se detallan de forma más precisas las tecnologías que se han enunciado anteriormente.

2.2.1 GPS en *smartphones*

Los primeros GPS tienen como origen la navegación marítima, donde antiguamente los navegantes iban navegando a través de la costa buscando puntos de referencia como accidentes geográficos o poblados para ir posicionándose en la ruta y así poder seguir su camino hasta el puerto de destino. El posicionamiento ha sido vital durante el transcurso de la existencia de la humanidad y se ha aplicado a múltiples campos. Con la incorporación de GPS en *smartphones* se nos abre una nueva puerta para poder aplicar geoposicionamiento en otros ámbitos como en la educación gracias al *M-Learning*. A continuación se explica qué técnica utilizan los *smartphones* para posicionarse y cómo es útil para la herramienta desarrollada. Los teléfonos actuales tienen básicamente dos métodos de geoposicionamiento [20]:

- El primer método consiste por triangulación IP. Cada terminal tiene una IP que lo identifica y gracias a la infraestructura actual de antenas de telefonía móvil la señal de un teléfono móvil puede ser recogida por tres o más torres, en el mejor de los casos, que conforman una celda, permiten poder triangular nuestra señal. Desde un punto de vista más matemático, esto es posible, teniendo la distancia a cada una de las antenas de telefonía que conforman la celda en la que nos encontramos, lo que permite que se pueda calcular la ubicación aproximada de nuestro *smartphone* en relación con los tres puntos de referencia (antenas de telefonía). Este cálculo geométrico se aplica en el caso de los teléfonos celulares, ya que sabemos la ubicación de las torres de celulares que reciben la señal del teléfono, y se puede estimar la distancia del teléfono de cada una de las torres de antenas, basándose en el tiempo que transcurre entre el momento en que la torre envía un ping a la IP de nuestro teléfono y este recibe el ping y lo contesta a cada

una de las antenas. Este proceso se detalla en la siguiente figura (ver imagen Figura 2.2.1.A Técnica geoposicionamiento triangulación IP).

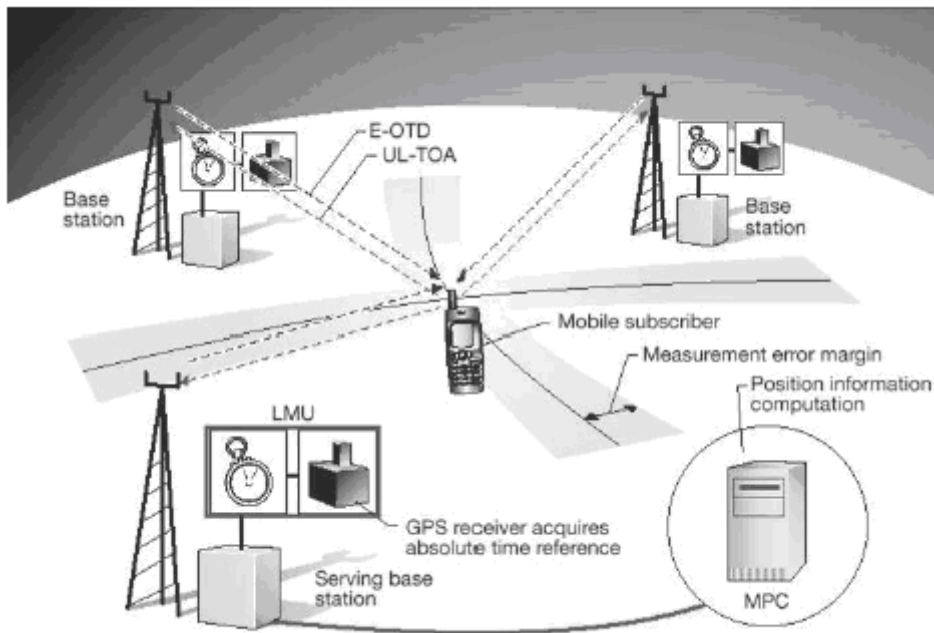


Figura 2.2.1.A Técnica geoposicionamiento triangulación IP

- El segundo método consiste en aprovecharse del chip que incorporan los últimos modelos de telefonía que nos proporcionan una geolocalización más exacta, aunque su coste energético es mayor y difícilmente asumible para dispositivos que carecen de autonomía energética como son las baterías de los *smartphones*.

El sistema GPS tiene por objetivo calcular la posición de un punto cualquiera en un espacio de coordenadas (x,y,z). Cuando se desea determinar esta posición, el dispositivo que se utiliza para ello localiza automáticamente como mínimo cuatro satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. En base a estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las señales, es decir, la distancia al satélite. Por triangulación calcula la posición en que éste se encuentra y, conociendo las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o las coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que desde tierra sincronizan a los satélites (ver imagen *Figura 2.2.1Bb Técnica geoposicionamiento GPS*). El único inconveniente de esta tecnología es cuando el dispositivo móvil se encuentra en espacios cerrados o de difícil acceso al satélite.

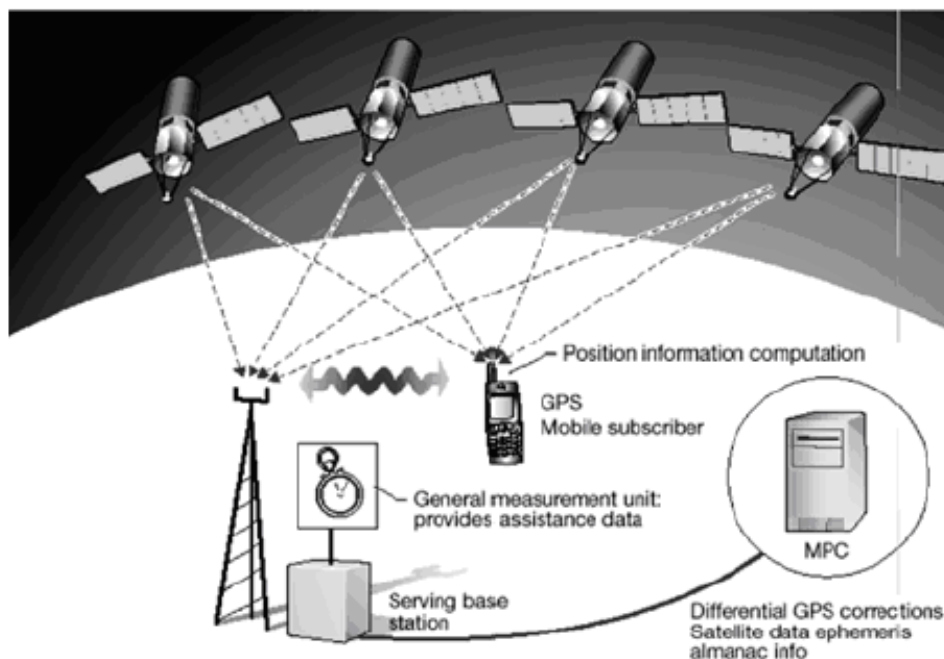


Figura 2.2.1.B Técnica geoposicionamiento GPS

2.2.1.1 Indoor Positioning System (IPS)

Actualmente existe una nueva tecnología llamada IPS (*Indoor Positioning System*) y que se presenta como alternativa al GPS, que es capaz de solventar la problemática que tiene el GPS de no poder geoposicionar en espacios cerrados y que se dispondrá de ella en futuros terminales. Con ellos se solucionan los problemas en calles estrechas, como cuando se usa la aplicación desarrollada, por ejemplo, en cascos antiguos, al haber poca conectividad con el satélite. Esta nueva tecnología proporcionará un mayor rendimiento de la herramienta.

La mayoría de los sistemas IPS [22], también se basan en el concepto de triangulación de señales para posicionar objetos, simplemente cambian los satélites, por otro tipo de dispositivos cuya posición es conocida. Entre las posibles soluciones podemos encontrar sistemas basados en:

- **WiFi:** Dado que hoy en día, existen multitud de dispositivos móviles que implementan tecnología WIFI, podemos obtener una solución basada en puntos de acceso inalámbricos situados en posiciones conocidas que actuarían a modo de satélites, y que se encargarían de enviar a un sistema central, información acerca de las señales intercambiadas con el dispositivo móvil para que sea capaz de posicionar un elemento.
- **Bluetooth:** Se trata de la misma idea, usar receptores *bluetooth* en posiciones conocidas y dispositivos *bluetooth* móviles. Todo el sistema estará gestionado por una capa software superior.
- **RFID:** Es un sistema basado en *tags* o etiquetas RFID, y lectores RFID, que al pasar un *smartphone* cerca de ellos, estos le informan de datos como la posición en la que se encuentra (ver imagen Figura 2.2.1.1.A Dispositivo RFID). Existen dos tipos de

dispositivos RFID. Por un lado dispositivos pasivos, los cuales no tienen ningún tipo de batería, y su funcionamiento esta basando en que los lectores RFID emiten un señal que induce una pequeña corriente a las etiquetas las cuales empiezan a funcionar. Por otro lado, tenemos los dispositivos activos en los que no se necesita ningún tipo de alimentación pero se ha de estar relativamente cerca del lector.



Figura 2.2.1.1.A Dispositivo RFID

Nota: Este tipo de tecnología en un futuro se podría aplicar a QuesTInSitu para poder hacer actividades en espacios cerrados, ya que en la actualidad no es posible por temas de cobertura orquestar actividades en este tipo de escenarios.

2.2.2 Sistemas de información geográfica (SIG)

Servir mapas en Internet se ha convertido en una funcionalidad muy popular en los últimos años. Actualmente existen una gran cantidad de aplicaciones que utilizan mapas, utilizando diferentes lenguajes de programación. Para estandarizar toda esta multiplicidad de herramientas y agruparlas en un único estándar existe la organización internacional *Open Geospatial Consortium* (OGC) [17]. Esta institución es la que se ha convertido como referente en el campo de servicios geospaciales. Muchas de las especificaciones de *Open Geospatial Consortium* (OGC) [17] se han convertido como ISO estándar para mapas. Los mapas que propone esta organización deben cumplir estos requisitos [19]:

- **GetCapabilities:** El servicio de interfaz geográfica que devuelve información sobre las capacidades que tiene la interfaz.
- **GetMap:** El servicio de interfaz geográfica que devuelve una imagen de un mapa según los parámetros que se le pidan. Un ejemplo aplicado a la herramienta QuesTInSitu sería la posibilidad de poder definir parámetros del mapa, para determinar un tipo de configuración determinada.
- **GetFeatureInfo:** El servicio de interfaz geográfica que devuelve al usuario información a partir de puntos de coordenadas del mapa pulsados por el usuario. Por ejemplo cuando se hace clic en la herramienta de este proyecto sobre una pregunta y puedes trabajar con ella.

La representación de datos mediante los Sistemas de información geográfica (SIG) son objetos del mundo real como ciudades, carreteras, bosques,..., etc. Estos objetos se pueden dividir de dos maneras diferentes [19]:

- **Objetos discretos:** Son todos los objetos fácilmente reconocidos en el mapa como puede ser una casa.
- **Objetos continuos:** Este tipo de objetos tienen formas más abstractas. Un ejemplo podría ser la depresión del terreno.

Este tipo de objetos tanto los discretos como los continuos, nos pueden ayudar a la hora de crear una pregunta o representar un tipo de interacción como se puede ver en la aplicación móvil. La forma que tienen los mapas digitales de guardar la información es muy importante para la presentación de la información a mostrar al usuario. Existen dos tipos de formas de mapas los vectoriales y los de *raster* [19]:

- **Mapas vectoriales:** Son mapas formados dibujado a partir de objetos geométricos independientes como segmentos, polígonos o arcos, entre otros. Cada uno de ellos está definido por diferentes atributos matemáticos de forma, posición, color, etc. Una de las grandes ventajas de los mapas vectoriales es que se pueden modificar las dimensiones de la imagen sin tener problemas en escalar correctamente la presentación y no obtener como resultado imágenes deformadas, ya que en los mapas es fundamental tener una precisa relación entre la escala y la representación cartográfica. Para sistemas de geocalización la precisión es fundamental y por este motivo la gran mayoría de aplicaciones de este tipo usan mapas vectoriales, como los dispositivos GPS de los coches.
- **Mapas *raster*:** También conocido como mapa de bits. Es un fichero de datos donde se representa el mapa a partir del almacenamiento de colores en cada píxel. Este último modelo es el que utiliza *Google Maps* para establecer sus imágenes georeferenciadas. Aprovechar la disposición de los píxeles en la pantalla también es muy utilizado para hacer capturas de fotografías, video o programas como *Photoshop* de Adobe. También es el utilizado para la aplicación *QuestInSitu*.

A continuación se dispone de una figura para comprobar la diferencia de resultados entre los dos tipos de mapas, a partir de un mapa sencillo (ver imagen Figura 2.2.2.A Ejemplo sencillo de un mapa vectorial y uno bits) [19].

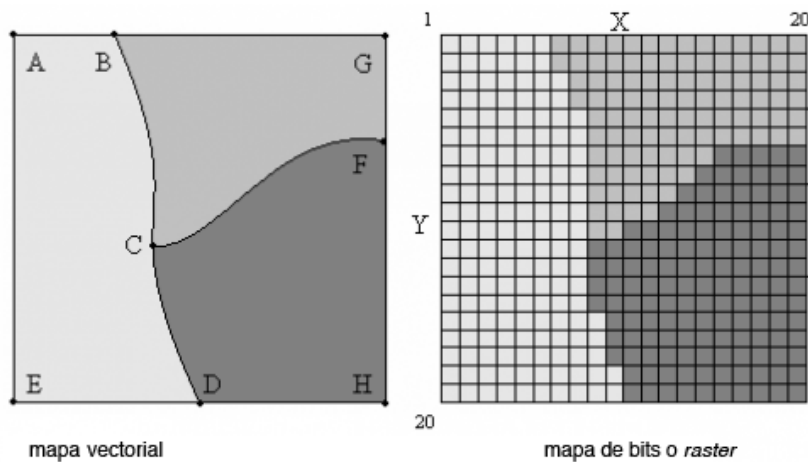


Figura 2.2.2.A Ejemplo sencillo de un mapa vectorial y uno bits

La tecnología que se cita en este apartado es muy útil ya que nos permite poder relacionar conceptos con mapas de forma digitalizada y que sean computables por máquinas de procesamiento. Esto nos permite poder generar iconos y actividades con los dispositivos móviles.

2.2.3 Google Maps

Google Maps [17] es una aplicación de código libre propiedad de la empresa Google y que forma parte de los Sistemas de información geográfica (SIG) comúnmente más utilizados. Esta aplicación ofrece imágenes vía satélite del planeta, combinadas en algunos casos con mapas de ciudades hechos con series de fotografías ortofotográficas para dar una calidad de imagen mejor, pues la resolución de estas imágenes se suelen hacer desde aviones proporcionando una resolución mayor que en las fotos por satélite. Un ejemplo podría ser la representación cartográfica de Cataluña facilitada por la Generalitat de Catalunya.

Este servicio ofrece aspectos muy interesantes a nivel computacional como poder desplazarse interactivamente por el mapa, adaptándose este a las diferentes imágenes de forma correcta durante el recorrido, la adaptación del mapa según las dimensiones de la ventana, una interfaz muy intuitiva y minimalista y poder elegir entre diferentes tipos de mapa como: vista de satélite, callejero o un híbrido entre ambas.

Cada recuadro de información cartográfica se almacena en un fichero donde su nombre indica su longitud y latitud, y su valor de zoom. *Google Maps* se basa en un cliente que hace peticiones para diseñar y editar mapas mediante Javascript y un servidor que es quien dispone de forma efectiva la información solicitada por el cliente.

Este tipo de peticiones con JavaScript vienen definidas por una API de *Google Maps* [17]. Esto se hace para poner en común la forma de solicitar al servidor los mapas y hacer llevaderas las funciones de editar mapas de una aplicación a todas las otras que utilicen este API.

Gracias a las ventajas anteriormente mencionadas como una gran interfaz y una gran eficiencia para poder realizar presentaciones mediante *Google Maps* se ha escogido este software para

realizar los mapas de la herramienta, ya que podían realizar todas las operaciones y presentaciones que se requerían para representar la información a evaluar con el alumno. El único inconveniente que se ha encontrado con *Google Maps* es que se debe especializar en su API para poder programar las presentaciones y las funciones de los mapas se pueden llevar también a los de *Google Maps*.

Con estos scripts se pueden llamar a funciones del API de *Google Maps* [17] las cuales devuelven presentaciones que se pueden incluir dentro de un código en formato JSP, y así poder incluir el mapa dentro de una plataforma Web para poder mostrarse posteriormente en un dispositivo móvil. A continuación se muestra un fragmento de *script* donde se llama a la API de *Google Maps* para *renderizar* un mapa (ver imagen Figura 2.2.3.A Diagrama de comunicación entre APIS con *Google Maps*):

```
<script type="text/javascript">
//
function load() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map"));
        map.setCenter(new GLatLng(37.4419, -122.1419), 13);
    }
}
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="246 456 796 473" data-label="Caption"><p>Figura 2.2.3.A Diagrama de comunicación entre APIS con <i>Google Maps</i></p></div><div data-bbox="161 500 586 517" data-label="Section-Header"><h3>2.2.3.1 Magnitudes de posicionamiento de <i>Google Maps</i></h3></div><div data-bbox="161 544 889 576" data-label="Text"><p>QuesTInSitu contiene un potente gestor de propiedades y se ha utilizado la tecnología <i>Google Maps</i> para ubicar todas las propiedades en un único mapa.</p></div><div data-bbox="161 602 889 650" data-label="Text"><p><i>Google Maps</i> dispone una API de amplia matriz que permite al usuario insertar las funciones más completas y de uso cotidiano de <i>Google Maps</i> en su propio sitio web y en sus propias aplicaciones, así como superponer sus propios datos sobre ellas.</p></div><div data-bbox="161 677 889 740" data-label="Text"><p>Para entender el funcionamiento de <i>Google Maps</i> hay que explicar cómo se interpretan los mapas. La tierra se divide en 360 ° y cualquier lugar de la tierra puede ser situado de forma precisa por la intersección del meridiano con un paralelo, es decir con dos magnitudes (latitud, longitud). Estas dos magnitudes las vemos de la siguiente forma:</p></div><div data-bbox="161 766 889 828" data-label="Text"><p><b>Latitud:</b> Indica la cantidad de grados de Norte a Sur desde el ecuador. Se calcula trazando una recta entre el centro de la esfera (punto O) hasta el punto donde nos encontramos (punto P), formando un ángulo (ángulo a) entre el Ecuador y el punto donde nos encontramos, como se puede ver en la siguiente figura (ver imagen <i>Figura 2.2.3.1.A Latitud</i>):</p></div><div data-bbox="847 935 876 952" data-label="Page-Footer"><p>35</p></div>
```

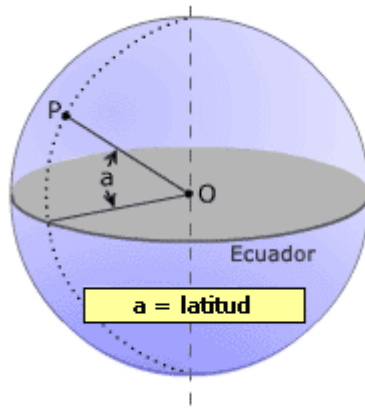


Figura 2.2.3.1.A Latitud

Longitud: Indica la cantidad de grados que hay entre el Este y el Oeste tomados desde el meridiano 0° (meridiano de Greenwich). Se calcula tomando el meridiano donde nos encontramos (punto P), y se busca la intersección con el ecuador. El ángulo conformado entre la intersección y la intersección entre el Ecuador y el meridiano 0° (meridiano de Greenwich), conforman un ángulo (ángulo b) si se trazan dos líneas hacia el punto central (punto O), tal y como se puede observar en la siguiente figura (ver imagen *Figura 2.2.3.1.B Longitud*):

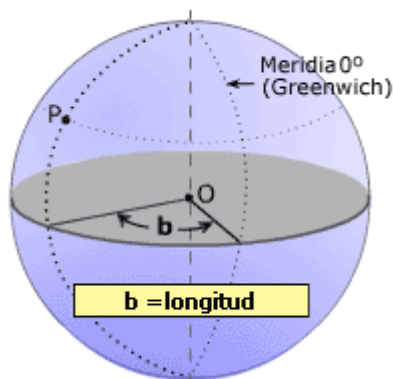


Figura 2.2.3.1.B Longitud

2.2.3.2 Key Google Maps

Para aprovechar al máximo los mapas de *Google Maps* y para poder integrarlos en la aplicación desarrollada, se ha tenido que generar una *Key* de la API de *Google Maps* (ver Anexos Manual apartado 7.2 para más información) y así poder generar mapas con contenido propio mediante los mapas funcionales de *Google Maps*. Esto permite insertar mapas de Google en las propias páginas web de dominio propio que contienen el proyecto. El único requisito para poder realizar este paso consta de disponer de una cuenta de Google, ya que la clave de la API se conecta a la cuenta de la compañía Google.

2.2.4 HTML5

Como se ha dicho en el apartado anterior 2.2.3 *Google Maps*, la API de *Google Maps* [17], permite mediante scripts poder tener un Sistema de información geográfica propio. En las últimas versiones se ha incorporado HTML5 que aportan grandes beneficios dentro del campo del *M-Learning*, ya que nos permite nuevas posibilidades de scripts con funcionalidades como poder acceder al sistema de geoposicionamiento del *smartphone*.

HTML5 se rige por los estándares de *World Wide Web Consortium* (W3C) [21], esto significa que está su acceso a todas las APIs, incluida la de *Google Maps*, están estandarizadas para que todas las aplicaciones móviles que utilizan este tipo de tecnología lo hagan de la misma manera. HTML5 todavía se encuentra en modo experimental, tal y como indica la misma W3C, aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

Al no ser reconocido HTML5 en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva de estos para poder disfrutar de todo el potencial que provee. Esto puede dar problemas según que dispositivo *smartphone* ya que puede no ser capaz de asimilar esta nueva tecnología y se ha de tener en cuenta en las evaluaciones de la herramienta.

A continuación se cita un ejemplo de la herramienta, en que utilizando funciones de HTML5 es capaz de comunicar a la API que quiere solicitar la posición del dispositivo (ver imagen Figura 2.2.4.A Conectarse API *Google Maps* mediante HTML5).

```
function getLocationUpdate(){  
  
    if(navigator.geolocation){  
  
        var options = {timeout:30000, enableHighAccuracy: true}; // timeout de 30  
segons  
        var watchID = navigator.geolocation.watchPosition(inicializar, errorHandler, options);  
  
    }else{  
        alert("Ho sentim, el teu navegador no suporta geolocalitzaci\xf3!");  
    }  
}
```

Figura 2.2.4.A Conectarse API Google Maps mediante HTML5

El resultado final es una web para dispositivo móvil que solicita la ubicación actual del *smartphone* y la coloca en un mapa. El resultado podría variar en algunos casos, según el navegador y el tipo de configuración que se haya definido.

2.2.5 IMS QTI

IMS QTI es un estándar de enseñanza, propuesto por el organismo internacional *IMS Global Learning Consortium* [13]. Este tipo de especificación permite el intercambio de preguntas, evaluaciones y resultados entre las diferentes herramientas que intervienen en el proceso de evaluación [8].

IMS QTI está formado por dos partes (ver imagen *Figura 2.2.5.A Estructura fichero IMS QTI*):

- Ítems
- *Assessments*

A continuación se muestra que contiene y que funcionalidad posee cada una de las dos partes del ítem:

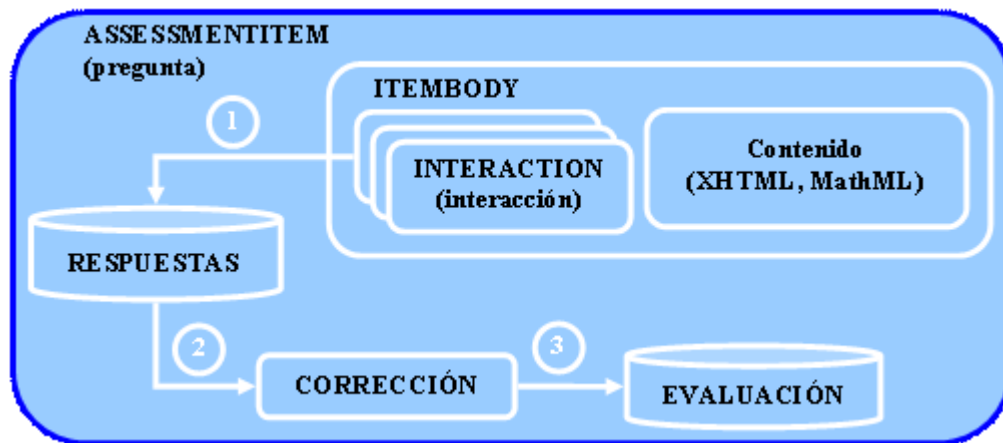


Figura 2.2.5.A Estructura fichero IMS QTI

Por una parte, tenemos los ítems donde definimos el tipo de pregunta a realizar (*Multiple Choice, Multiple Response, True/False, etc...*) y por otra parte los cuestionarios donde encontramos los test descritos por ficheros llamados *assessments* (enunciado, puntuación, feedback, etc...). Estos dos elementos se relacionan mediante una identidad única que interpreta el motor para relacionar las dos partes.

Esta especificación nos permite definir tres tipos de preguntas readaptando este estándar y asociándolos con una coordenada del mapa e imagen del lugar donde se encuentra, que son:

- **Multiple Choice:** Con este tipo de pregunta podemos crear preguntas con cuatro posibles opciones de las cuales sólo una es correcta.
- **Multiple Response:** Con este tipo de pregunta podemos crear preguntas con cuatro posibilidades de las que pueden ser correctas todas, algunas o ninguna de las respuestas propuestas.
- **True / False:** En este tipo de pregunta sólo se da como opción a la pregunta si es cierta o falsa.

Actualmente existe la versión IMS QTI v.2.1. [13] que es la que se utilizará para la comunicación entre el servidor y la especificación. Esta versión apareció en el año 2006 y, a diferencia de las anteriores, aporta informes complementarios sobre resultados obtenidos por los usuarios. En las distintas aplicaciones que utilizan esta versión se puede observar que una vez realizada la interacción se obtiene un *feedback* más completo con: el *score* de la pregunta, el porcentaje de preguntas acertadas i/o erróneas, y otros.

Estos resultados son un informe exclusivo de esta pregunta respecto al usuario que la haya realizado. En este caso, la especificación para saber si es correcta o incorrecta la respuesta se basa en el *tag ítemVariable*, el cual también permite saber las veces que esta pregunta ha sido realizada o cómo también saber si ya ha sido resuelta con *numAttempts* o *completionStatus*, que tratan con el motor gestor de la especificación, en este caso NewApis.

Para finalizar, decir que este estándar nos ofrece ventajas muy beneficiosas como la reutilización del modelo de datos, intercambiabilidad de formas y duración independiente, lo que facilita su tratamiento dentro del mundo computacional.

3. Diseño, implementación, arquitectura y uso de QuesTInSitu

3.1 Diseño y casos de uso

Los casos de uso se utilizan en el proceso de prototipado de sistemas basados en el Diseño Centrado en el Usuario. Los casos de uso nos sirven para detectar que necesidades debe soportar nuestro sistema.

Los casos de uso se pueden modelar a través del Lenguaje Unificado de Modelado UML (*Unified Modeling Language*), el cual a su vez se compone de muchas otras herramientas, básicamente diagramas, como: diagramas de clase, diagramas de secuencia, colaboración, transición de estados, diagramas de actividad, componentes y *deployment*, entre otros. Todas ellas son usadas a lo largo de las etapas o durante el ciclo de vida del proceso de desarrollo.

En el caso de QuesTInSitu la información sobre los roles y los casos de uso ha sido recogida a través de varios experimentos realizados dentro del marco de las investigaciones del grupo de investigación GTI de la UPF.

3.1.1 QTIS – Roles y casos de uso

Para los usuarios que entren en la aplicación QuesTInSitu mediante la web a QTIS, se han creado tres tipos de roles diferentes, que son (ver figura *Tabla 3.1.1.A* Tabla de Casos de uso):

- **Alumno (usuario):** Simula las capacidades que puede tener un alumno dentro de la aplicación QuesTInSitu. Puede tener acceso a los exámenes y puede crear transmitir los conocimientos aprendidos a otros alumnos. Este usuario se puede crear directamente en la herramienta.
- **Administrador (monitor):** Es el encargado de la integridad de la aplicación y de sus datos. También se ocupa de dar soporte al profesor con consultas y tareas que no puede realizar, un ejemplo puede ser: la creación de XML generadores de listas de todos los resultados o de usuarios.
- **Profesor:** Simula las capacidades (crear pregunta, crear ruta, monitorización de una ruta, monitorización posicionamiento de los grupos de alumnos en la ciudad, evaluación de actividades orquestadas,...) que puede tener un profesor dentro de la aplicación QuesTInSitu. También es el usuario encargado de proporcionar ayuda a los usuarios de forma continua en la herramienta y es el encargado de controlar el buzón online para dudas y problemas de los usuarios. Tiene acceso a opciones privadas de control y monitorización para poder gestionar los exámenes más adecuadamente. Este usuario sólo puede ser creado mediante otro administrador de la herramienta.

Acciones	Alumno (Usuario)	Profesor	Administrador
Crear preguntas Geolocalizadas	X	X	
Crear ruta de preguntas		X	
Crear un usuario en la aplicación	X	X	
Dar de baja un usuario	X	X	
Editar un usuario	X	X	
Monitorizar actividad orquestada		X	
Obtener puntos para responder preguntas		X	
Participar en una actividad orquestada geolocalizada	X	X	
Responder una pregunta geolocalizada aleatoria	X	X	
Responder una pregunta geolocalizada en el mapa	X	X	
Ver emails sobre dudas enviadas a QuesTInSitu		X	
Ver ranking mejores usuarios	X	X	
Ver listado de todos los usuarios	X	X	
Crear fichero XML de rutas			X
Crear fichero XML de scores			X
Salir de la aplicación	X	X	

Tabla 3.1.1.A Tabla de Casos de uso

Profesor:

- Interactuar con el mapa.
- Crear Preguntas geolocalizadas.
- Responder preguntas sobre el mapa (aleatoria o total).
- Responder Pregunta mediante un dispositivo móvil 3G.
- Visualizar el *feedback* desde el portal web o teléfono móvil (*score*, fotografías asociadas a las preguntas, indicaciones para ir al siguiente punto de la ruta, etc...).
- Monitorización de un escenario educativo.
- Crear Rutas y ver todas las disponibles.
- Añadir cuestionarios a las rutas directamente desde un mapa o también desde el listado de una tabla de rutas.
- Gestionar el correo de ayuda de la aplicación.

A continuación, en la *Figura 3.1.1.B* “Casos de uso profesor”, se muestra un esquema donde se recogen todas las iteraciones que puede hacer el usuario con rol administrador en QuesTInSitu (QTIS):

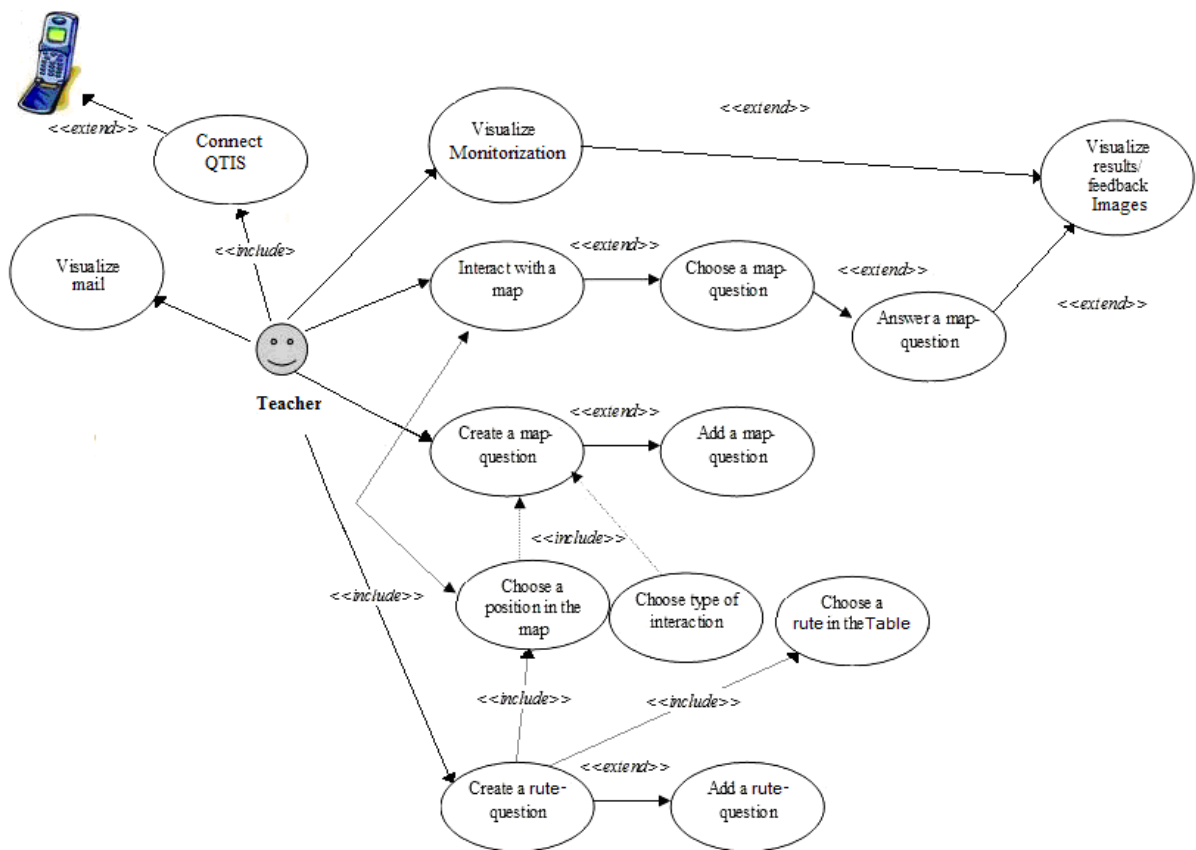


Figura 3.1.1.B Casos de uso profesor

Alumno (Usuario):

- Interactuar con el mapa.
- Responder preguntas sobre el mapa (aleatoria o total).
- Responder preguntas mediante un dispositivo móvil 3G.
- Visualizar el *feedback* desde el portal web o teléfono móvil (*score*, fotografías asociadas a las preguntas, indicaciones para ir al siguiente punto de la ruta, etc...).
- Salir de la aplicación.

A continuación, en la *Figura 3.1.1.C* “Casos de uso alumno”, se muestra un esquema donde se recogen todas las iteraciones que puede hacer el usuario con rol alumno en *QuesTInSitu* (QTIS):

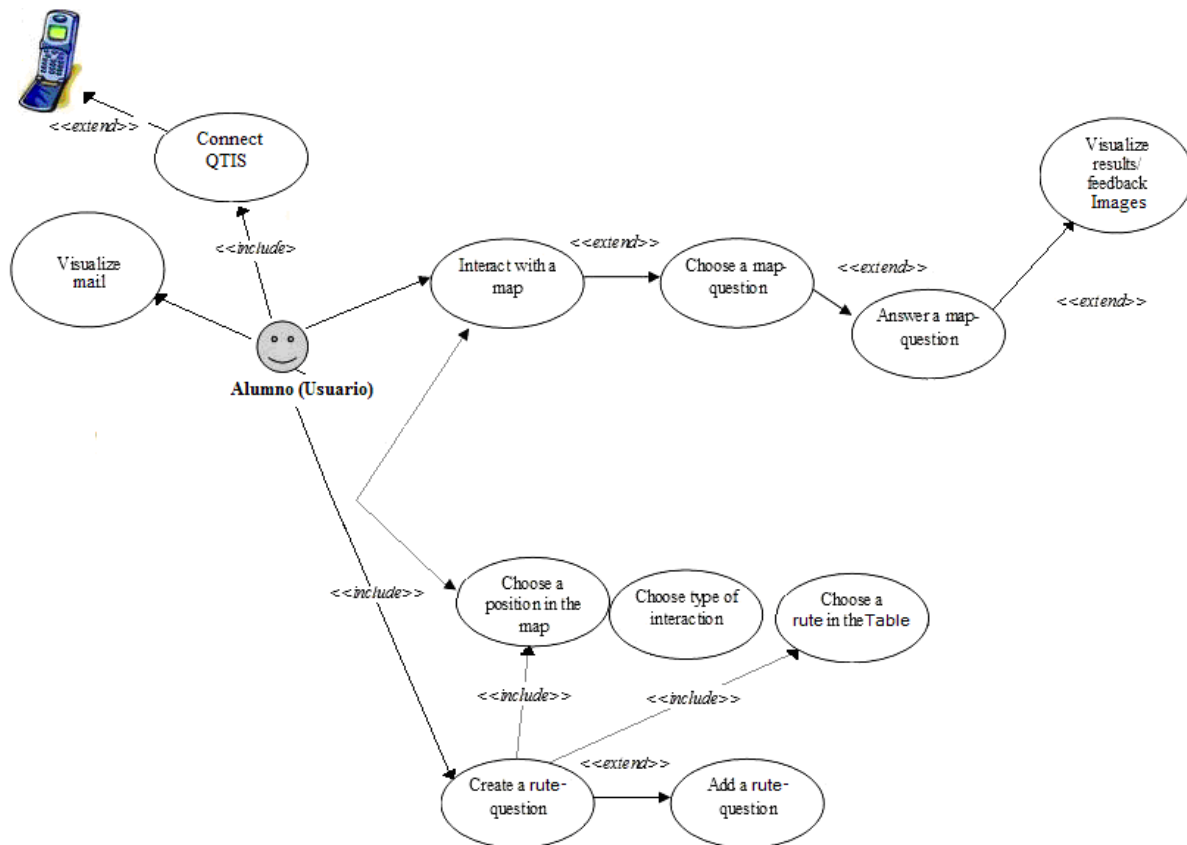


Figura 3.1.1.C Casos de uso alumno

Casos de uso generales para los dos roles (alumno y profesor):

- Darse de alta al sistema como usuario.
- Editar perfil.
- Darse de baja.
- Ver todos los tests de la aplicación.
- Contestar los test de la aplicación.
- Ver todos los usuarios de la aplicación.
- Consultar el manual de uso de la aplicación.

Administrador:

A continuación, ver *Figura 3.1.1.D* “Casos de uso administrador”, se muestra un esquema dónde se recogen todas las iteraciones que puede hacer el usuario con rol administrador en QuesTInSitu (QTIS):

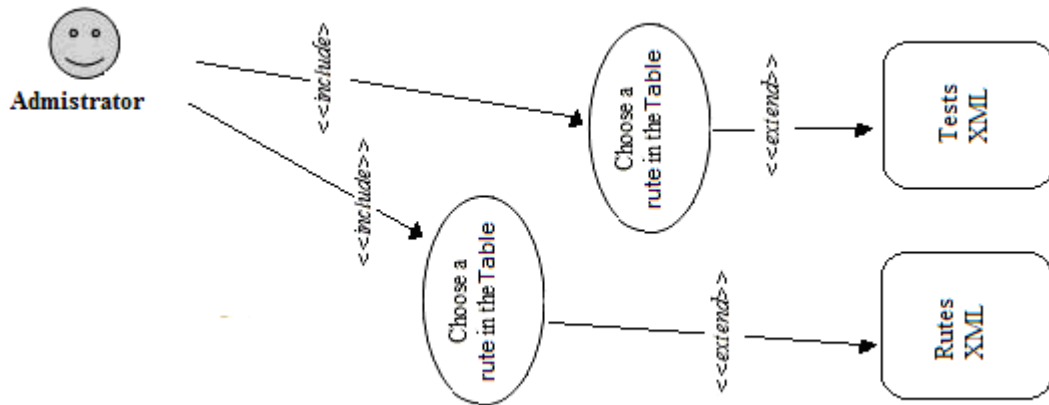


Figura 3.1.1.D Casos de uso administrador

Casos de administrador:

- Crear fichero resumen con todas las rutas en XML.
- Crear fichero resumen con todos los marcadores de los usuarios en XML.

3.1.2 QTISM – roles y casos de uso

Para la aplicación QuesTInSitu para usuarios que utilicen un dispositivo móvil a QTIS, se ha creado un único rol (ver imagen *Figura Tabla 3.1.2.A Tabla de Casos de uso en QTISM*):

Alumno (usuario): Este usuario podrá realizar todas las operaciones que nos ofrece la aplicación móvil.

Acciones	Alumno (Usuario)
Escoger ruta	X
Ver nuestra posición en el mapa	X
Interactuar con el mapa	X
Responder preguntas	X
Ver feedback de nuestras preguntas	X
Ver nuestro marcador en la ruta que estamos realizando	X
Seguir una ruta	X
Ver puntos calientes de la ruta en el mapa	X
Ver preguntas por contestar en el mapa	X
Ver preguntas contestadas en el mapa	X
Ver preguntas obviadas en el mapa	X
Salir de la aplicación	X

Figura Tabla 3.2.A Tabla de Casos de uso en QTISM

A continuación, en la Figura 3.1.2.B Casos de uso alumno en QTISM, hay un esquema donde se recogen todas las iteraciones que puede hacer el usuario con rol alumno usuario en QuesTInSitu Mobile (QTISM):

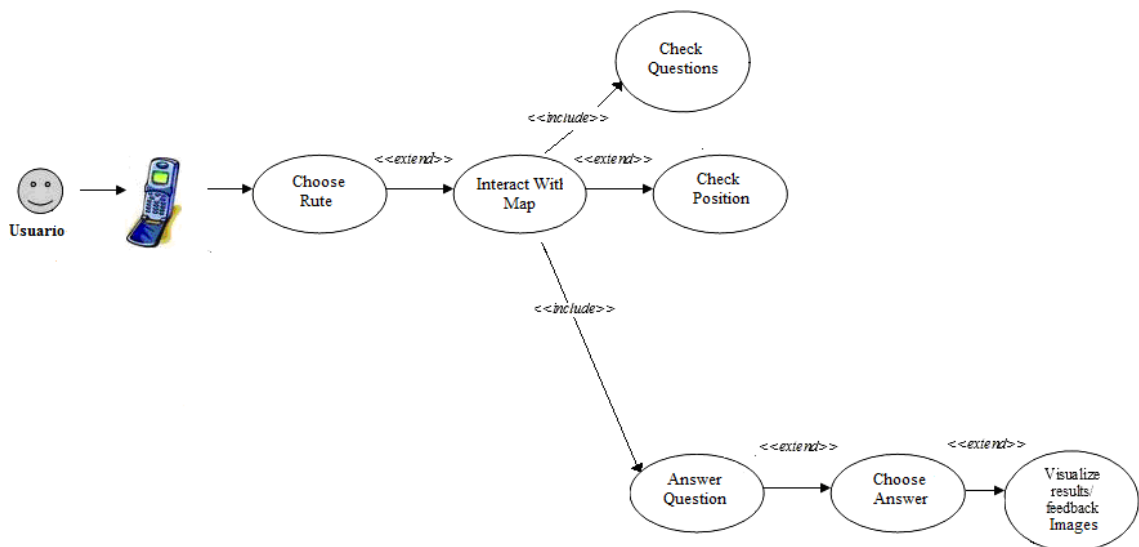


Figura 3.1.2.B Casos de uso alumno en QTISM

Usuario:

- Escoger ruta.
- Ver la posición propia en el mapa.
- Interactuar con el mapa.
- Responder preguntas.
- Ver feedback de las preguntas respondidas.
- Ver el *score* realizado en la ruta actual.
- Seguir una ruta.
- Ver puntos calientes de la ruta en el mapa.
- Ver preguntas por contestar en el mapa.
- Ver preguntas contestadas en el mapa.
- Ver preguntas obviadas en el mapa.
- Salir de la aplicación.

En este apartado se detalla el proceso de diseño e implementación de las nuevas funcionalidades de QuesTInSitu de forma más técnica., Entrando en detalle en su diseño, en la distribución arquitectónica de los elementos y formas y maneras de cómo se puede utilizar QuesTInSitu para así obtener su máximo potencial de las opciones que nos ofrece.

3.2 Arquitectura e implementación

A continuación se detalla toda la arquitectura del proyecto, describiendo sus diferentes elementos y la interconexión entre ellos.

3.2.1 Elementos de la aplicación

QuesTInSitu es un sistema que incorpora una herramienta de autoría (que facilita la edición y geolocalización de preguntas), un *player* (que muestra las preguntas tipo test sobre un mapa web de *Google Maps*) y un motor de preguntas tipo test geolocalizadas, conforme a la especificación IMS QTI (que interpreta este estándar y gestiona el procesado de las preguntas) [13].

Además se ha diseñado la aplicación QuesTInSitu para móviles, utilizando HTML5, que permite gestionar los recursos de geoposicionamiento del dispositivo móvil.

QuesTInSitu facilita la creación y puesta en marcha de rutas interactivas con fines educativos, conforme a currículos educativos o los intereses de instituciones como ayuntamientos, museos o en el ámbito del turismo (Ej. Editoriales de guías de viajes). Las rutas se gestionarán con ordenador y/o teléfonos móviles con GPS e Internet. Diseñada para el público, principalmente estudiantes y profesores con objetivos de educación formal y/o informal, que esté interesado en aprender sobre una zona geográfica (Ej. una ciudad) mientras responde a preguntas.

3.2.1.1 QTIS

A continuación (ver Figura 3.2.1.1. Estructura de QTIS) se explicara cada uno de estos módulos (gráfico general, estructura de módulos y sus características) del sistema QTIS.

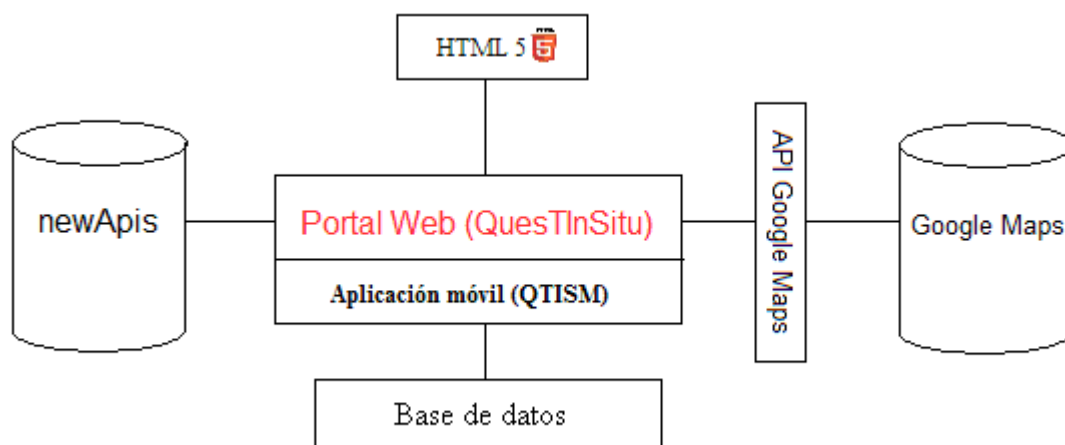


Figura 3.2.1.1. Estructura de QTIS

QTIS es la aplicación web proyecto enfocado a realizar las tareas de la actividad a través de terminales fijos, tales como ordenadores de sobremesa o portátiles. QTIS está orientado a la parte de edición de actividades (gestión preguntas, creación, usuarios,...), como también a la parte de monitorización. Esta aplicación está diseñada de forma modular en una plataforma donde se disponen diferentes elementos que, interconectados entre si, pueden ofrecer el servicio que ofrece la aplicación. A continuación se hará una breve descripción de la funcionalidad de cada módulo:

NewApis: Es la parte de la plataforma encargada de ofrecer un motor de *renderización* de preguntas, para que el módulo central de QTIS obtenga preguntas para poder ofrecer a los usuarios.

HTML5: Es la parte de la plataforma encargada de ofrecer servicio en la parte de ejecución en el cliente, mediante un dispositivo móvil, de la aplicación de QTISM. Una de las funcionalidades que se aprovecha de esta funcionalidad es acceder al GPS de los *smartphones* y así obtener las posiciones de los usuarios.

Base de datos: Base de datos en formato MySQL, que ofrece a la plataforma un lugar donde almacenar todos los datos para la gestión y desarrollo de la aplicación. Estos pueden ser resultados, datos de usuarios, datos de preguntas, ..., etc.

Google Maps: Aplicación web de la empresa Google que nos ofrece soporte de mapa, con los cuales podemos añadir contenido propio para generar nuestros mapas de rutas con sus preguntas respectivas.

3.2.1.2 QTISM

A continuación se explicará cada uno de estos módulos, gráfico general, estructura de módulos y sus características, del sistema QTISM.

QTISM es la aplicación móvil del proyecto, enfocada a realizar la conexión entre los dispositivos móviles y los estudiantes. Este software obtiene la posición de las preguntas geolocalizadas así como facilita la respuesta de las preguntas de forma móvil a los usuarios, ya que mientras interactúan con el ambiente pueden también trabajar con la aplicación. A continuación se muestra un gráfico de la arquitectura de QTISM:

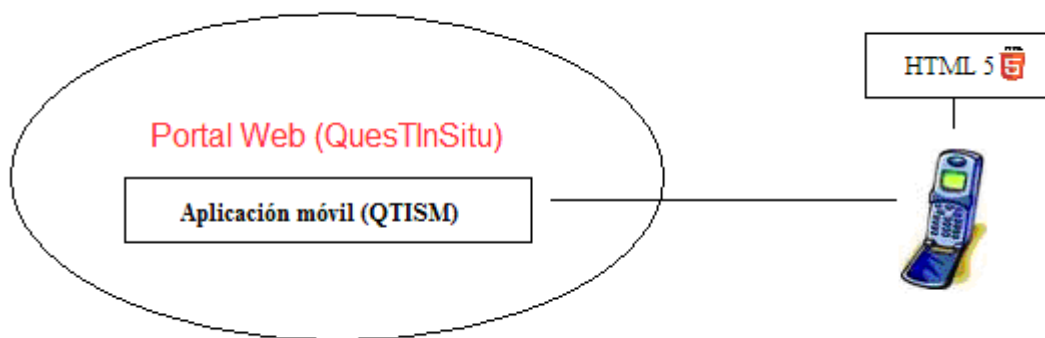


Figura 3.2.1.2.A Estructura de QTISM

Como se puede observar en el dibujo anterior (Figura 3.2.1.2.A Estructura de QTISM), los dos módulos, tanto el de QTIS como el de QTISM, están asociados a una misma estructura. La interfaz de QTISM, enfocada a dispositivos *smartphones* realiza llamadas tanto al GPS del dispositivo móvil como a la operativa de datos (ya sea de preguntas, usuarios, etc...) que tiene QTIS, haciendo así QTISM de puente entre los dos módulos, lo que economiza recursos y aprovecha el código que ya tiene construido QTIS.

3.2.1.3 NewApis

Apis, y en concreto su última versión *NewApis*, es un motor de código libre creado a partir de *Servlets*, una base de datos y el lenguaje *Java*. *NewApis* es un motor en forma modular que procesa archivos con formato *.XML*.

Este motor fue creado inicialmente por la Universidad de Stratchclyde y ha sido mejorado y utilizado por otras instituciones como la Universidad Pompeu Fabra para proyectos propios. Su funcionamiento se basa a partir de bocetos para servir presentaciones y respuestas procesadas correctamente en forma estructural siguiendo el formato *.XML*.

Gracias a su forma modular, este motor es independiente de la plataforma LMS (*Learning Management System*) en que se encuentre, haciéndolo transportable a cualquier tipo de plataforma. *NewApis* incluye una base de datos independiente de formato, para almacenar las interacciones realizadas por las diferentes sesiones de los usuarios (respuestas, puntos, veces intentado,..., etc.)

En la siguiente figura (ver imagen Figura 3.2.1.3.A Estructura funcionamiento motor *NewApis*), se puede ver un gráfico dónde se explica como APIS se integra dentro de un servidor de una aplicación con este motor para servir cuestionarios IMS QTI [13].

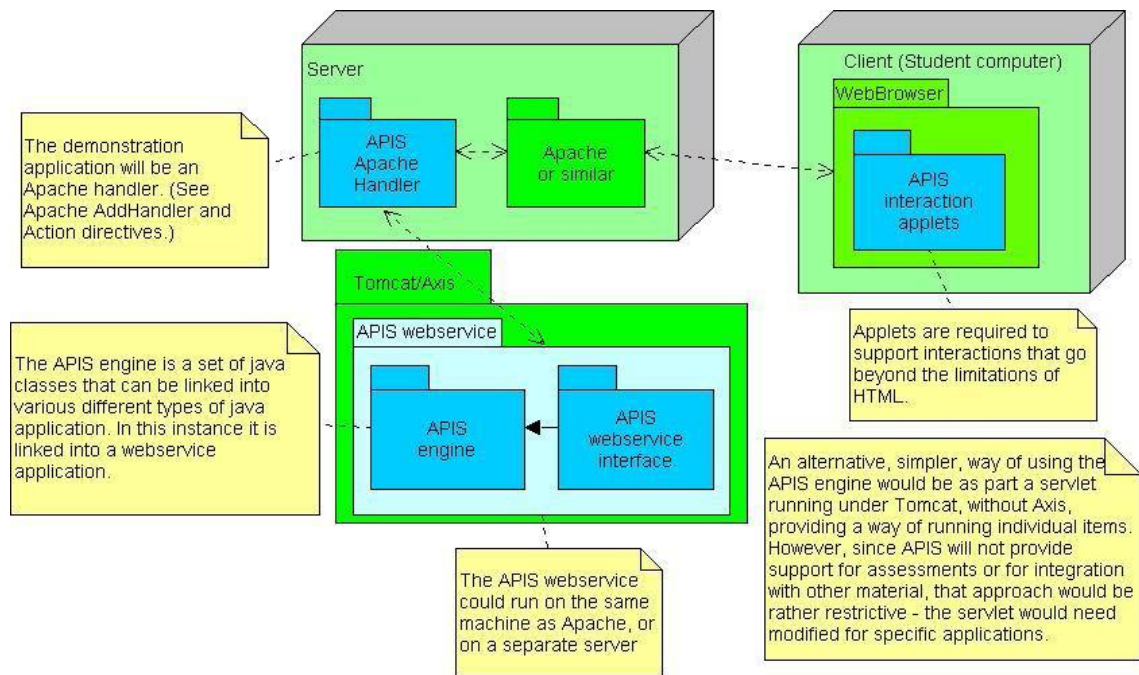


Figura 3.2.1.3.A Estructura funcionamiento motor NewApis

Como se puede observar, aunque el paquete del motor se encuentra incluido dentro del servidor, este es totalmente autónomo y podría ser externo o incluso podría estar dando servicio incluido desde una aplicación ya existente [9].

3.2.2 Interconexión de elementos

En este apartado se muestra un esquema con todos los elementos que se conectan en la aplicación y, a continuación, se expone una explicación breve de la forma que se realiza (ver imagen *Figura 3.2.2.A Gráfica interconexión entre elementos de QTI y QTISM*):

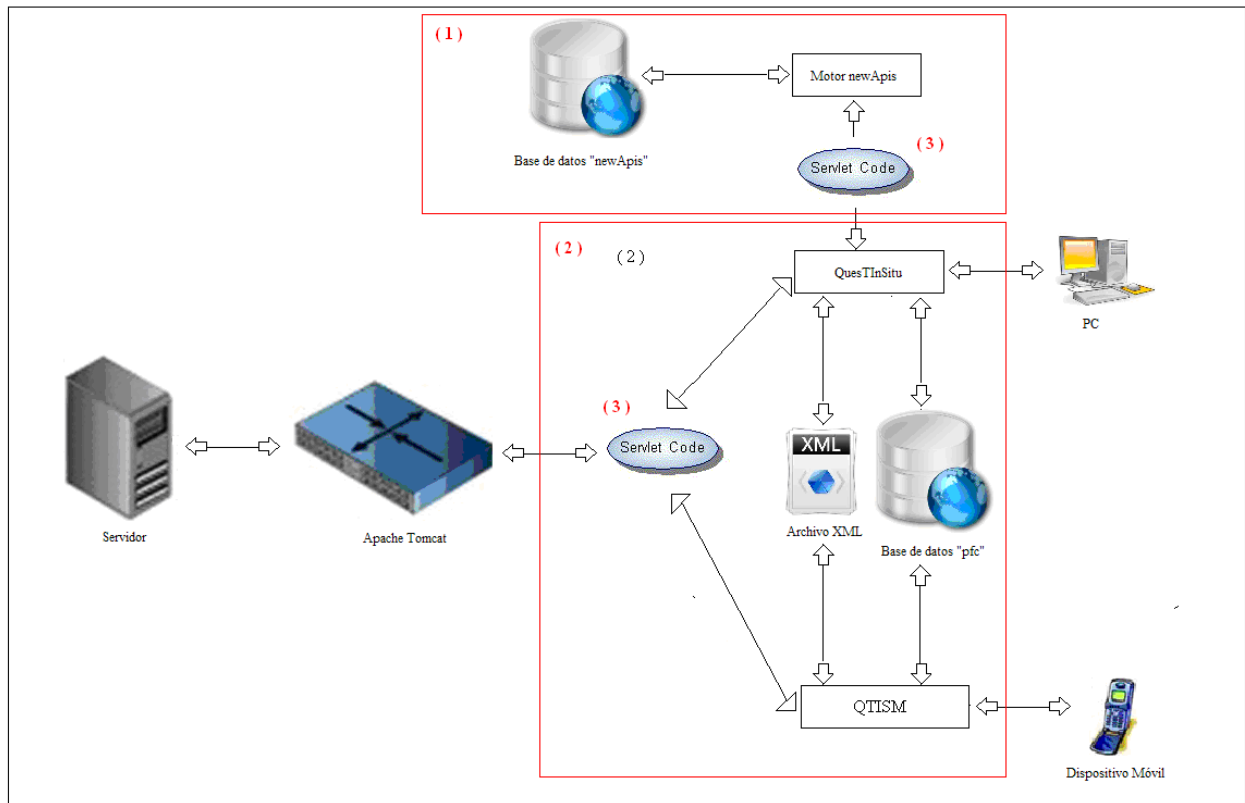


Figura 3.2.2.A Gráfica interconexión elementos entre QTI y QTISM

Inicialmente se tiene un servidor web, que alberga un servidor Apache Tomcat y los diferentes módulos: QuesTInSitu, QTISM y la base de datos. Una vez dispuestos los elementos, éstos se comunican entre ellos, compartiendo funcionalidades, para que, al final, se ofrezca el servicio de la aplicación desarrollada al cliente.

3.2.2.1 Conexión NewApis – QuesTInSitu (1)

La conexión entre NewApis y QuesTInSitu tiene como eje central la clase “RunQTITest.java” que se encuentra dentro del archivo de proyecto de QTIS, en el paquete de código “apisRun”. El objetivo de esta clase es hacer de puerto de peticiones y enlazar todo el tráfico que se genera entre el motor de NewApis con todo el código del proyecto provenga de QuesTInSitu o de peticiones más alejadas, como de QTIS. Esta clase se comporta como un mini puerto que encapsula las peticiones que pueden llegar siguiendo un formato en concreto: usuario, id_pregunta, base de datos a acceder, etc.... Una vez que recibe una petición coge estos parámetros y los hace llegar al motor de NewApis, el cual realiza la operación correspondiente y devuelve al usuario un *feedback* en formato XHTML. La clase “RunQTITest.java”, es también la encargada de adaptar la respuesta a la aplicación y hacerle llegar el *feedback*. Para configurar el acceso a la base de datos la clase “RunQTITest.java” utiliza un *parser* de

configuración para desglosar los datos de la base de datos de *NewApis*. Esta clase se encuentra también en el paquete el paquete de código “apisRun” con nombre “OwnQtiParser.java”.

Entrando más en detalle en el funcionamiento que realiza *NewApis*, este consiste en recibir un conjunto de ficheros QTI como entrada, en un fichero XML, que gestiona los parámetros del test y unos ficheros XML que contienen la información de los ítems (preguntas). *NewAPIS* procesa la información del XML mediante un *parser*. Para ello es necesario llamar al *parser* del motor y comunicarle tres parámetros de entrada:

- Usuario
- Identidad pregunta (id)
- Idioma

Una vez se indican estos parámetros, el motor ya conoce el usuario que está operando, el tipo de pregunta y el enunciado de ésta, con lo que el motor ya es capaz de realizar una respuesta y devolver una presentación (enunciado de la pregunta) correctamente estructurada a partir de los parámetros de entrada. En este momento el motor permanece a la espera de que el usuario haga una interacción y posteriormente un *submit* de la respuesta. Una vez el usuario responde la pregunta, el servidor recoge la interacción y la salva en su base de datos. A partir de los datos recogidos confecciona otra presentación (*feedback*) con la respuesta obtenida a la pregunta y su puntuación.

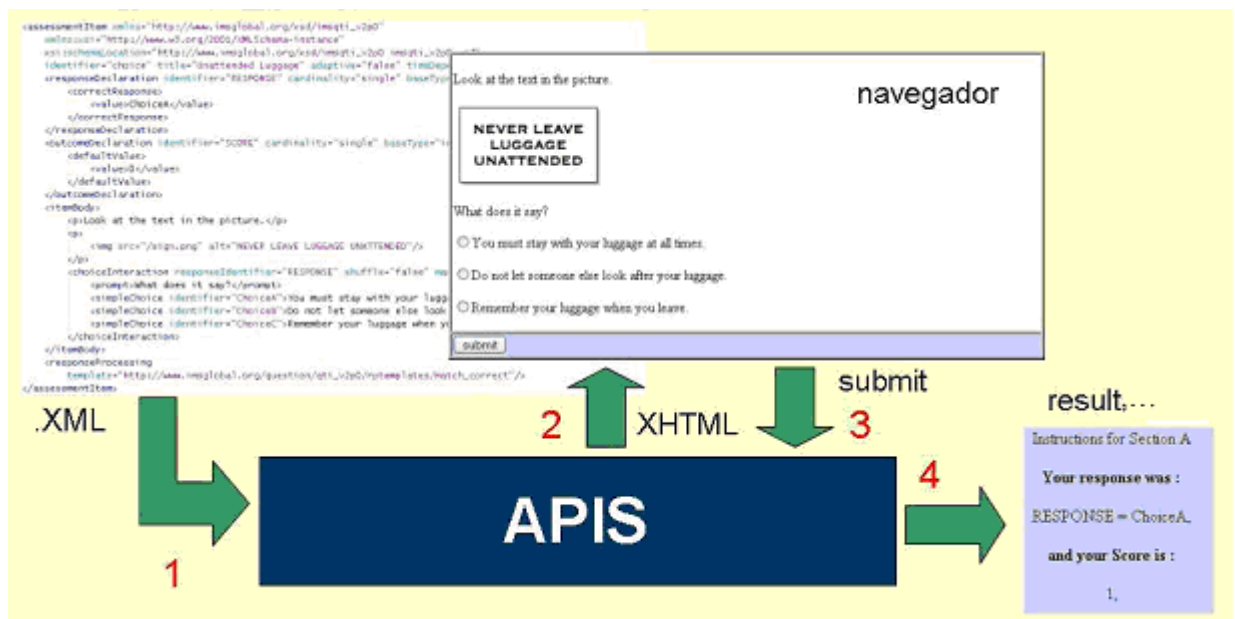


Figura 3.2.2.1.A Gráfica funcionamiento de renderizar una pregunta con motor *NewApis*

Como se puede ver en el ejemplo (ver imagen *Figura 3.2.2.1.A Gráfica funcionamiento de renderizar una pregunta con motor NewApis*), la interacción con el motor es muy sencilla, ya que solamente consta de tres pasos principales que son la llamada al servidor con sus parámetros de entrada, una primera *renderización* (dónde se muestra la pregunta) y, finalmente, una segunda *renderización* con el resultado de ésta y el *feedback* correspondiente. Estos pasos siempre se cumplen para cualquier tipo de pregunta que tome el motor como entrada.

Las preguntas que formula *NewApis* para realizar una pregunta están construidas mediante *items*. Cada pregunta que muestra la aplicación QuesTInSitu está contenida dentro de este *item* único en formato XML y que es independiente del motor *NewApis*. Cada uno de estos ficheros XML tienen su identidad correspondiente, tipo de interacción, enunciado, feedback, etc.... Estos ficheros XML son interpretados por el motor *NewApis* y crea una presentación para el usuario en forma de enunciado, con una presentación del tipo XHTML. Cuando el usuario hace *submit*, *NewApis* interpreta este XHTML y le devuelve al usuario otro XHTML con la respuesta corregida y su feedback correspondiente.

3.2.2.2 Conexión QuesTInSitu – QTISM (2)

La conexión entre QuesTInSitu y QTISM se realiza entre dos subconjuntos donde QuesTInSitu contiene funcionalidades que QTISM adopta y las utiliza de forma externa. Para ello esta conexión se hace valer de una comunicación que se realiza mediante archivos XML. Por ejemplo, Para que QTISM genere una ruta que posteriormente se *renderizará* en un mapa, se necesitan previamente los datos de dicha ruta. Estos datos son comunicados por QuesTInSitu mediante la generación de un archivo .XML (ver imagen *Figura 3.2.2.2.A* Archivo para *renderizar* una ruta en QTISM). Este archivo se guarda en la carpeta “...\QTISMXML” de QTISM, carpeta dónde se guardan los XML con los datos de la ruta que QTISM recoge con la clase “QTISMRequest.java” para su posterior muestreo en un mapa de *Google Maps*.

```
<test>
<question_item>
<id>42</id>
<coord>41.403933731950715, 2.193145751953125</coord>
</question_item>
<question_item>
<id>43</id>
<coord>41.403933731950715, 2.193145751953125</coord>
</question_item>
<question_item>
<id>44</id>
<coord>41.403933731950715, 2.193145751953125</coord>
</question_item>
<question_item>
<id>45</id>
<coord>41.40296804498943, 2.194347381591797</coord>
</question_item>
<question_item>
<id>46</id>
<coord>41.40339455850014, 2.1941328048706055</coord>
</question_item>
</test>
```

Figura 3.2.2.2.A Archivo para *renderizar* una ruta en QTISM

3.2.2.3 Funcionamiento presentaciones web servlets (3)

Para el desarrollo de esta aplicación se ha utilizado la estructura de *JSP/Servlets* para realizar la comunicación de presentaciones Web de nuestro proyecto. Esto significa que la aplicación está desarrollada en lenguaje Java y utiliza elementos como *Servlets*, JSP, JavaScript, Base de datos, *Beans*, etc... (Ver imagen *Figura 3.2.2.3.A Gráfica funcionamiento Servlets, JSP y jdbc*).

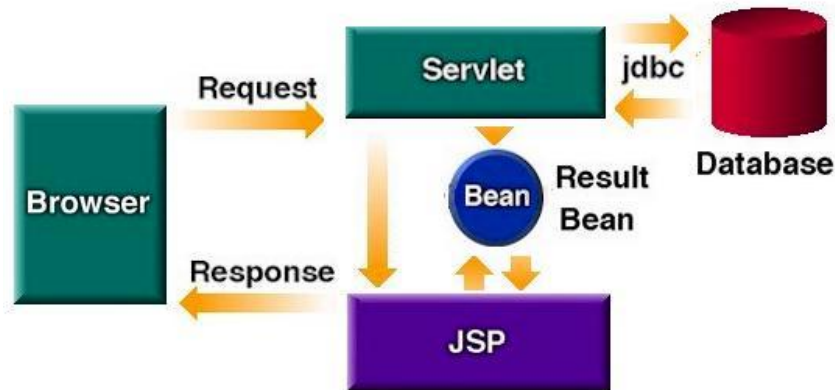


Figura 3.2.2.3.A Gráfica funcionamiento Servlets, JSP y jdbc

A continuación se hace una breve explicación de estas tecnologías:

Servlets: Son la respuesta de la tecnología Java a la programación CGI. Se trata de programas que se ejecutan en un servidor Web y permiten construir páginas Web al vuelo, cosa que es útil puesto que se basa en los datos enviados por el usuario, que cambian frecuentemente devolviendo una página previamente construida y luego actualizándola. Algunas de las ventajas que aportan son:

- **Eficiencia:** Con los *Servlets*, la máquina Virtual Java permanece arrancada y cada petición es manejada por un *thread* Java de peso ligero, en lugar de un pesado proceso del sistema operativo.
- **Potencia:** Los *Servlets* de Java permiten hacer fácilmente muchas cosas que son difíciles o imposibles con un CGI normal. Los *Servlets* pueden hablar directamente con el servidor Web como también pueden compartir los datos entre ellos, permitiendo realizar cosas útiles como almacenes de conexiones a bases de datos fáciles de implementar.
- **Portable:** Los *Servlets* están escritos en Java y siguen un API bien estandarizado. Consecuentemente, los *Servlets* escritos se pueden ejecutar sin modificarse en Apache, Microsoft IIS y Webster entre otros. Los *Servlets* están soportados directamente o mediante un *plug-in* en la mayoría de los servidores Web.

JSP: Es una tecnología que permite mezclar HTML estático, con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son estáticas, con la parte dinámica limitada a muy pocas localizaciones. Las características que nos ofrece JSP están disponibles también en *Servlets*, pero es preferible utilizar JSP puesto que estos son mucho más convenientes para escribir i/o modificar código HTML. Además, al separar el formato del contenido, se puede dividir las tareas en diferentes personas: los expertos en diseño de páginas Web pueden construir el HTML, mientras que los expertos programadores de

Servlets pueden insertar el contenido dinámico, tal y como se puede ver en la aplicación aquí presentada.

JavaScript: Este lenguaje puede generar HTML dinámicamente en el cliente, lo cual es una capacidad útil, pero hay que tener en cuenta que sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente. Con la excepción de las *Cookies*, el HTTP y el envío de formularios no están disponibles con JavaScript. Al ejecutarse en el cliente, JavaScript no puede acceder a los recursos en el lado del servidor, como bases de datos, catálogos, información de preguntas,..., Etc.

JDBC: es utilizado para enviar comandos SQL hacia una base de datos relacional como en nuestro caso MySQL. Este elemento constituye un “*driver*” que puede realizar las siguientes operaciones:

- Establecer una conexión con una BD
- Enviar sentencias SQL
- Procesar los resultados

En el proyecto se utiliza JDBC con la siguiente llamada de código Java:

```
Connection con = DriverManager.getConnection ( "jdbc:odbc:wombat", "login", "password");  
Statement stmt = con.createStatement();
```

La metodología de llamadas y procedimiento para conectarnos a la base de datos desde las funciones Java creadas se desarrolla en tres capas. La aplicación Java llama al *servlet* que se ocupa de la parte lógica de negocio y control de datos y éste contiene el JDBC que da acceso a MySQL, como se puede ver en la imagen a continuación (ver imagen Figura 3.2.2.3.B Modelo tres capas de jdbc):

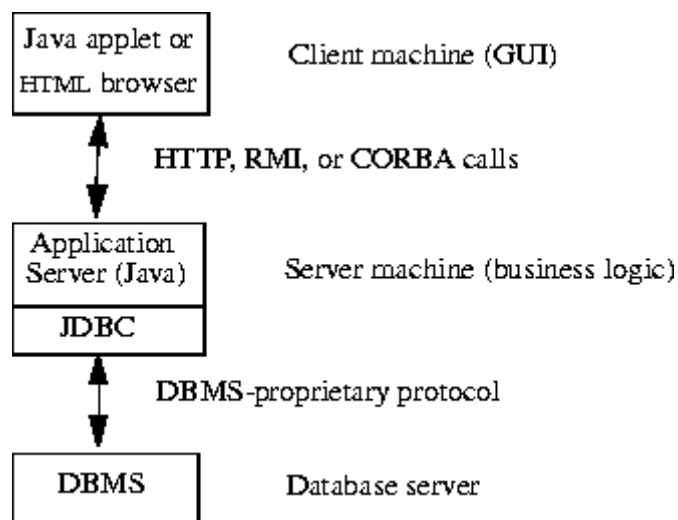


Figura 3.2.2.3.B Modelo tres capas de jdbc

3.2.3 Funciones más importantes de QTIS

Descripción actualizada de las clases más significantes del software teniendo en cuenta las nuevas creaciones y funcionalidades que se han creado en este tiempo, tanto en los proyectos de QTIS.

En este apartado se documentarán todos los paquetes y clases más importantes creados para la elaboración de la herramienta QuesTInSitu. En la página siguiente se puede ver la figura 3.2.3.1.A (Tablas base de datos de QTIS), donde se muestra los paquetes que se utilizan para la herramienta. Debajo de esta, se explicará la función de los más significativos.

Para la realización de este proyecto hemos tenido que añadir y modificar funciones de la aplicación. Un ejemplo son todas las funciones correspondientes con el menú gestor de rutas, las cuales han sido mejoradas para ofrecer un mejor funcionamiento del gestor de rutas, como por ejemplo las funciones “BeanDeleteRuteMap.java” o “BeanInsertRuteMap.java”. También se ha creado las funciones “HTTPRequestInfo.java” para mejorar la comunicación con la aplicación móvil y la función “BeandAdminResults.java” y “BeanAdminMenu.java” para la encapsulación de datos. Para la mejora del sistema de monitorización,

También se ha tenido que modificar paquetes de funciones completos, como es el caso del paquete de funciones Java “Gmaps”. Este paquete se ha tenido que ser modificar y reconstruir casi de nuevo, ya que ha variado gran parte de las funciones que incorpora, ya sea desde la obtención de datos de las funciones, la forma de tratar los datos y leerlos, hasta la forma de mostrar la información que recibe el usuario en las diferentes presentaciones JSP.

Package Resources																																	
<table border="1"> <thead> <tr> <th>NewApis Engine</th> </tr> </thead> <tbody> <tr> <td>adapters -package</td> </tr> <tr> <td>apis - package</td> </tr> <tr> <td>apis.expression - package</td> </tr> <tr> <td>nb_utils - package</td> </tr> </tbody> </table>	NewApis Engine	adapters -package	apis - package	apis.expression - package	nb_utils - package	<table border="1"> <thead> <tr> <th>coreservlets</th> </tr> </thead> <tbody> <tr> <td>BeanBaixaUser.java</td> </tr> <tr> <td>BeanChangeUserData.java</td> </tr> <tr> <td>BeanCheckUsers.java</td> </tr> <tr> <td>BeanDeleteRuteMap.java</td> </tr> <tr> <td>BeanInsertRuteMap.java</td> </tr> <tr> <td>BeanNewUser.java</td> </tr> <tr> <td>BeanRute.java</td> </tr> <tr> <td>CreateMail.java</td> </tr> <tr> <td>CreateRute.java</td> </tr> <tr> <td>DropTestRutePopUp.java</td> </tr> <tr> <td>GoInici.java</td> </tr> <tr> <td>InsertTestRutePopUp.java</td> </tr> <tr> <td>BeanAdminMenu.java</td> </tr> <tr> <td>BeanAdminResults.java</td> </tr> <tr> <td>DeleteTestRute.java</td> </tr> <tr> <td>CreateAdmin.java</td> </tr> </tbody> </table>	coreservlets	BeanBaixaUser.java	BeanChangeUserData.java	BeanCheckUsers.java	BeanDeleteRuteMap.java	BeanInsertRuteMap.java	BeanNewUser.java	BeanRute.java	CreateMail.java	CreateRute.java	DropTestRutePopUp.java	GoInici.java	InsertTestRutePopUp.java	BeanAdminMenu.java	BeanAdminResults.java	DeleteTestRute.java	CreateAdmin.java	<table border="1"> <thead> <tr> <th>apisRun</th> </tr> </thead> <tbody> <tr> <td>ConnectionQTI.java</td> </tr> <tr> <td>HTTPRequestInfo.java</td> </tr> <tr> <td>IdNameUser.java</td> </tr> <tr> <td>LoadTestFromSQL.java</td> </tr> <tr> <td>OwnQtiParser.java</td> </tr> <tr> <td>RandomXML.java</td> </tr> <tr> <td>RunQTITest.java</td> </tr> <tr> <td>SaveQTIScore.java</td> </tr> </tbody> </table>	apisRun	ConnectionQTI.java	HTTPRequestInfo.java	IdNameUser.java	LoadTestFromSQL.java	OwnQtiParser.java	RandomXML.java	RunQTITest.java	SaveQTIScore.java
NewApis Engine																																	
adapters -package																																	
apis - package																																	
apis.expression - package																																	
nb_utils - package																																	
coreservlets																																	
BeanBaixaUser.java																																	
BeanChangeUserData.java																																	
BeanCheckUsers.java																																	
BeanDeleteRuteMap.java																																	
BeanInsertRuteMap.java																																	
BeanNewUser.java																																	
BeanRute.java																																	
CreateMail.java																																	
CreateRute.java																																	
DropTestRutePopUp.java																																	
GoInici.java																																	
InsertTestRutePopUp.java																																	
BeanAdminMenu.java																																	
BeanAdminResults.java																																	
DeleteTestRute.java																																	
CreateAdmin.java																																	
apisRun																																	
ConnectionQTI.java																																	
HTTPRequestInfo.java																																	
IdNameUser.java																																	
LoadTestFromSQL.java																																	
OwnQtiParser.java																																	
RandomXML.java																																	
RunQTITest.java																																	
SaveQTIScore.java																																	
<table border="1"> <thead> <tr> <th>IOData</th> </tr> </thead> <tbody> <tr> <td>CreadordeTest.java</td> </tr> <tr> <td>CreadordeTestMC.java</td> </tr> <tr> <td>CreadordeTestMR.java</td> </tr> <tr> <td>IOSQL.java</td> </tr> <tr> <td>MyFileIO.java</td> </tr> <tr> <td>TestNumber.java</td> </tr> </tbody> </table>	IOData	CreadordeTest.java	CreadordeTestMC.java	CreadordeTestMR.java	IOSQL.java	MyFileIO.java	TestNumber.java																										
IOData																																	
CreadordeTest.java																																	
CreadordeTestMC.java																																	
CreadordeTestMR.java																																	
IOSQL.java																																	
MyFileIO.java																																	
TestNumber.java																																	
<table border="1"> <thead> <tr> <th>Gmaps</th> </tr> </thead> <tbody> <tr> <td>GSetLatLong.java</td> </tr> <tr> <td>GSQL.java</td> </tr> <tr> <td>Monitor.java</td> </tr> </tbody> </table>	Gmaps	GSetLatLong.java	GSQL.java	Monitor.java		<table border="1"> <thead> <tr> <th>DAO</th> </tr> </thead> <tbody> <tr> <td>DAO.java</td> </tr> <tr> <td>DAOAllTest.java</td> </tr> <tr> <td>DAOLogout.java</td> </tr> <tr> <td>DAOMail.java</td> </tr> <tr> <td>DAORank.java</td> </tr> <tr> <td>DAORute.java</td> </tr> <tr> <td>DAOTest.java</td> </tr> <tr> <td>DAOGLoad.java</td> </tr> <tr> <td>DAOUsuario.java</td> </tr> </tbody> </table>	DAO	DAO.java	DAOAllTest.java	DAOLogout.java	DAOMail.java	DAORank.java	DAORute.java	DAOTest.java	DAOGLoad.java	DAOUsuario.java																	
Gmaps																																	
GSetLatLong.java																																	
GSQL.java																																	
Monitor.java																																	
DAO																																	
DAO.java																																	
DAOAllTest.java																																	
DAOLogout.java																																	
DAOMail.java																																	
DAORank.java																																	
DAORute.java																																	
DAOTest.java																																	
DAOGLoad.java																																	
DAOUsuario.java																																	

Figura 3.2.3.1.A Tablas base de datos de QTIS

Los paquetes de la herramienta son:

- **Conjunto de paquetes *NewApis Engine***: Este conjunto de paquetes son los que conforman el motor de *NewApis*.
- ***ApisRun Package***: Este paquete contiene todas las clases Java necesarias para enlazar el portal web de la herramienta con el motor de *NewApis*.
- ***IODATA Package***: Paquete que contiene todas las clases Java para la confección de ficheros y los cuestionarios IMS QTI.
- ***Gmaps Package***: Paquete que contiene clases Java relacionadas con *Google Maps*.

- **Coreservlets Package:** Paquete de clases Java diversas para desarrollar acciones que debe realizar el portal.

- **DAO Package:** Estructura de clases Java para el acceso estructura a la base de datos, de varias funciones, que requieran alguna interacción con MySQL.

A continuación, se exponen algunas de las funciones más importantes del portal web y que requieren una supervisión específica dada su importancia en la funcionalidad del proyecto. Se detallarán las funciones en formato ficha (con los puntos: localización, captura y utilidad de la función) para que sea más sencillo de comprender.

Función: setDBProperties()

Localización: La función setDBProperties se encuentra contenida dentro de la clase OwnQtiParser.java del paquete aspiRun.

Captura imagen de la función (Figura 3.2.3.1.B Función: setDBProperties de QTIS):

```
/**
 * setDBProperties. Properties are read from from portal-ext.properties.
 *
 * @param theTest
 *         QTIv21test
 */
public void setDBProperties(final QTIv21test theTest) {
    try {
        final String driverClassValue = "com.mysql.jdbc.Driver";
        final String databaseUrlValue = "jdbc:mysql://localhost/tenc_apis";
        final String databaseUsernameValue = "root";
        final String databasePasswordValue = "root";
        theTest.linkTheTestWithDb(driverClassValue, databaseUrlValue, databaseUsernameValue, databasePasswordValue);
    } catch (Exception e) {
        System.out.println("Error amb la Base de Dades");
    }
}
```

Figura 3.2.3.1.B Función: setDBProperties de QTIS

Utilidad: Esta función permite configurar *NewApis* para que se pueda conectar a su base de datos “tenc_apis”.

Función: obrirConnexio()

Localización: La función obrirConnexio se encuentra contenida dentro de la clase DAO.java del paquete DAO.

Captura imagen de la función (Figura 3.2.3.1.C Función: obrirConnexio de QTIS):

```

public boolean obrirConnexio()
{
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost/pfc";
        con = DriverManager.getConnection(url, "root", "root"); //definim la connexio
        stmt = con.createStatement(); // creem un statement sobre la connexio
        return true;
    }
    catch(Exception e)
    {
        return false;
    }
}

```

Figura 3.2.3.1.C Función: *obrirConnexio* de QTIS

Utilidad: Esta función permite configurar el portal web para que se pueda conectar a su base de datos "pfc".

Función: doGet (Servlet RunQTITest.java)

Localización: La función doGet, es el método GET de la función RunQTITest.java que se encuentra dentro del paquete apisRun.

Captura imagen de la función (Figura 3.2.1.D *Función: doGet* de QTIS):

```

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    String Post;
    String Score;

    Score = "";

    Post = "";

    String Numero = request.getParameter("Numero");

    StringWriter result = new StringWriter();

    System.out.println("RESPONSE:" + request.getQueryString());

    //if (request.getQueryString() == null) {
    if (request.getQueryString().startsWith("&Numero")) {

        Post = "test" + Numero + ".xml";

    }
    else {

        Post = request.getQueryString();

    }
    ...
}

```

Figura 3.2.3.1.D Función: *doGet* de QTIS

Utilidad: Esta función es la más importante del proyecto ya que conecta el portal con el motor de *NewApis*, gestiona el *parser* del motor y las presentaciones de los cuestionarios IMS QTI [13].

Función: sendRenderedTestPage

Localización: La función sendRenderedTestPage, se encuentra dentro de la clase RunQTITest.java que se encuentra dentro del paquete apisRun.

Captura imagen de la función (Figura 3.2.3.1.E Función: sendRenderedTestPage de QTIS):

```
private StringWriter SendRenderedTestPage(String Post, String user) {

    StringWriter result = new StringWriter();
    String xml_url;
    String file = Post;

    if(file.startsWith("file")) {

        file = file.substring(file.lastIndexOf("%2F")+3, file.lastIndexOf("&identifier"));
        xml_url = "http://193.145.50.210:8080/QTIS/tests_servits/"+file;

    }
    else {
        xml_url = "http://193.145.50.210:8080/QTIS/tests_servits/"+file;
    }

    System.out.println("URL: " + file);
    //xml_url = "http://localhost:8080/Projwar/tests_servits/test1.xml";
    //xml_url = "http://localhost:1234/tests/"+file;

    //OwnQtiParser qti_parser = new OwnQtiParser(user, 0, result, xml_string, Post,"english");
    OwnQtiParser qti_parser = new OwnQtiParser(user, 0, result, xml_url, Post, "english");

    SAXParserFactory factory = SAXParserFactory.newInstance();
    factory.setValidating(false);
    factory.setNamespaceAware(true);

    String JAXP_SCHEMA_LANGUAGE = "http://java.sun.com/xml/jaxp/properties/schemaLanguage";
    String W3C_XML_SCHEMA = "http://www.w3.org/2001/XMLSchema";

    try {
        SAXParser parser = factory.newSAXParser();
        //parser.setProperty(JAXP_SCHEMA_LANGUAGE, W3C_XML_SCHEMA);
        parser.parse(xml_url,qti_parser);
    }catch(Exception e){
        e.printStackTrace();
        System.out.println("parser cannot be runned ..");
    }

    //retorna el render
    return result;
}
```

Figura 3.2.3.1.E Función: sendRenderedTestPage de QTIS

Utilidad: La función sendRenderedTestPage, es la función que indica al motor *NewApis*, según los parámetros recibidos, cuál es la cuestión IMS QTI [13] que debe *parsear* y tratar.

Función: load

Localización: La función load se encuentra en todos los *scripts* que deben crear un mapa de *Google Maps*.

Captura imagen de la función (Figura 3.2.3.1.F Función: load de QTIS):

```
function load() {

    if (GBrowserIsCompatible()) {

        //Instance of the map
        map = new GMap2(document.getElementById("mapa"));

        //Startup options
        map.setCenter(new GLatLng(41.38830608079951, 2.1587533940328285), 11, G_NORMAL_MAP);
        map.addControl(new GLargeMapControl());
        map.addControl(new GMapTypeControl());
        geocoder = new GClientGeocoder();

        obtePunts();
        //for (var i=0;i<punts.length;i++) {
        <%for (int i = 0; i < Integer.parseInt (GLoad.getMaxNumTest()); i++) { %>

            if( <%=GLoad.getRuteMark(i , Ruta)%> > 0 ) {
                afegeixMarcador(<%=i%>);
                puntscont = puntscont + 1;
            }

            <% }%>
        alert('Tests disponibles: ' + puntscont);
    }
}

window.onload=load;
```

Figura 3.2.3.1.F Función: load de QTIS

Utilidad: La función load, es la que indica al API de *Google Maps* la carga de un mapa. Esta función siempre está presente en cualquiera de los mapas de la aplicación a la hora de llamarlos.

Función: CreateAdmin.java

Localización: La función CreateAdmin se puede encontrar dentro del paquete “coreservlets”.

Captura imagen de la función (Figura 3.2.3.1.G Función: CreateAdmin de QTIS):

```
/**
 * function to change a user information
 * @author David Pérez Calle
 * @version $Revision: 1.0 $, $Date: 2010/06/01 $
 */
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    String Alias = request.getParameter("Alias");

    //Carreguem el driver JDBC
    try {

        Class.forName("com.mysql.jdbc.Driver").newInstance();
        //Creem la connexió
        Connection connexio = DriverManager.getConnection("jdbc:mysql://localhost/qtis?user=root&password=root");
        //Creem un objecte Statement mitjançant la connexió i la Query corresponent
        Statement sentencia = connexio.createStatement();

        String query = "UPDATE Users SET Administrador = '1' Where alias = '" + Alias + "'";

        int result = sentencia.executeUpdate(query);

        if (result != 1) {

            System.out.println("****");
        }

        else {

            PrintWriter out = response.getWriter();
            out.println("Admin Done");
        }
    }
}
```

Figura 3.2.3.1.G Función: CreateAdmin de QTIS

Utilidad: Esta función sirve para dar privilegios a los usuarios para que convertirse en administradores. Esta función se engloba dentro del bloque de funciones para administrar usuarios como BeanCheckUsers, BeanBaixaUser.java, BeanNewUser.java, etc....

Función: DeleteTestRute.java

Localización: La función DeleteTestRute se puede encontrar dentro del paquete “coreservlets”.

Captura imagen de la función (Figura 3.2.3.1.K Función: DeleteTestRute de QTIS):

```

/**
 * function to delete test from rute
 * @author David Pérez Calle
 * @version $Revision: 1.0 $, $Date: 2010/06/01 $
 */
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String IDRuta = request.getParameter("IDRuta");
    String IDTest = request.getParameter("IDTest");

    System.out.println("Ruta: " + IDRuta + " Test: " + IDTest);

    String Ruta = "";
    String Target = "";

    //Carreguem el driver JDBC
    try {

        Class.forName("com.mysql.jdbc.Driver").newInstance();
        //Creem la connexió
        Connection connexio = DriverManager.getConnection("jdbc:mysql://localhost/qtis?user=root&password=root");
        //Creem un objecte Statement mitjançant la connexió i la Query corresponent
        Statement sentencia = connexio.createStatement();

        String query = "SELECT t.Ruta as Rutas, r.nom as Target FROM test t, ruta r where r.id = '" + IDTest + "' and t.numero='" + IDRuta

        ResultSet result = sentencia.executeQuery(query);

        while (result.next()){

            Ruta = result.getString("Rutas");
            Target = result.getString("Target");
        }

    } catch (InstantiationException e) {} catch (IllegalAccessException e) {} catch (ClassNotFoundException e) {} catch (SQLException e) {}
}

```

Figura 3.2.3.1.K Función: DeleteTestRute de QTIS

Utilidad: Esta función sirve para dar de baja un test de una ruta específica.

3.2.4 Funciones más importantes de QTISM

Descripción actualizada de las clases más significantes del software teniendo en cuenta las nuevas creaciones y funcionalidades que se han creado en este tiempo, tanto en los proyectos de QTISM.

En este apartado se documentarán todos los paquetes y clases más importantes. Para ello se ha creado la herramienta específica QTISM. En la figura 3.2.4.1.A (Tablas base de datos de QTISM) se ven los paquetes que se utiliza la herramienta y la función de los más significativos.

Para este nuevo proyecto, se incorporan nuevas funciones de código y se añaden nuevas funciones y modificaciones a las funciones ya existentes. Un ejemplo en concreto, puede ser la incorporación de la función “lecturaEntradaXML.java”. Esta función nos permite cargar para la aplicación móvil una ruta y la manera que esta esta configurada (orden de las preguntas, cantidad, posición, etc...). Otra de las funciones que aparecen nuevas en este proyecto es la función “ChooseRuteMap.java” en combinación con “DAOAllRutes.java”. Estas nuevas funciones habilitan al usuario de smartphones para que este pueda escoger que ruta desea realizar dentro del catalogo disponible que dispone el aplicativo. También tenemos importantes modificaciones en funciones como “DAOGLoad.java”, que modifica la manera en que la aplicación carga los datos.

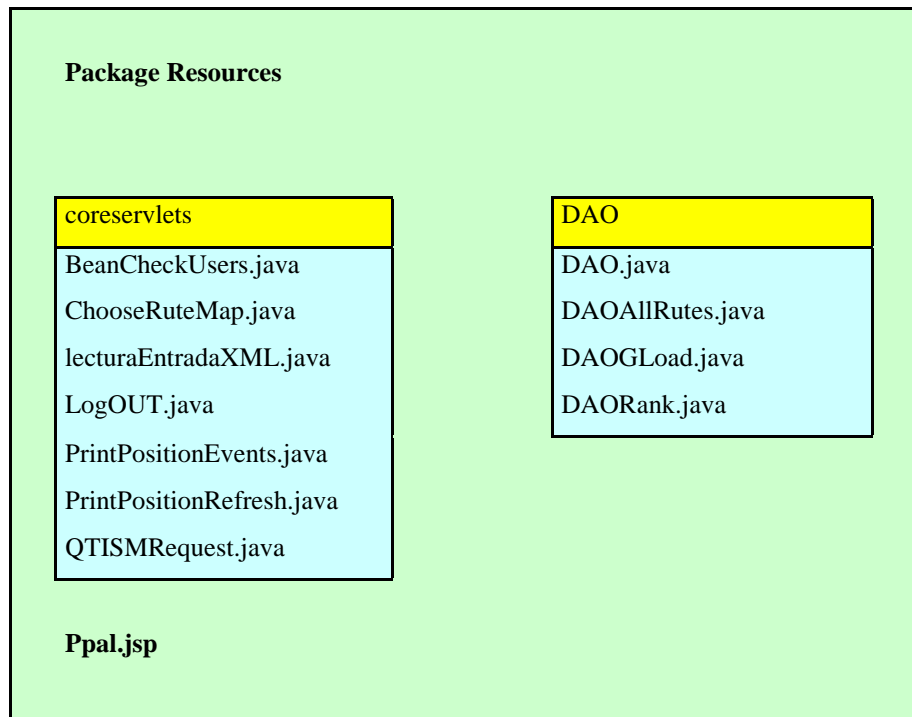


Figura 3.2.4.1.A Tablas base de datos de QTISM

Función: PrintPositionEvents.java

Localización: La función PrintPositionEvents se puede encontrar dentro del paquete coreservlets.

Captura imagen de la función (Figura 3.2.4.1.B Función: PrintPositionEvents de QTISM):

```

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String Alias = request.getParameter("Alias").toUpperCase();
    String Ruta = request.getParameter("Ruta").toUpperCase();
    String Cont = request.getParameter("Cont");
    String Latlong = request.getParameter("Latlong");
    String Hora = request.getParameter("Hora");
    String Min = request.getParameter("Min");
    String Seg = request.getParameter("Seg");

    Latlong.equals(Latlong.replace("(", ""));
    Latlong.equals(Latlong.replace(")", ""));

    MyFileIO File = new MyFileIO();
    File.stringToFile("E:/apache-tomcat-6.0.24/webapps/QTISM/Positions/" + Alias + ".txt", Latlong);
    File.stringToFile("E:/apache-tomcat-6.0.24/webapps/QTISM/Positions/" + Alias + "_Date.txt", "<user>" + Alias + "</user><pos>" + Latlong + "</pos><

    response.sendRedirect(response.encodeRedirectURL("ppal.jsp?"));
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}

```

Figura 3.2.4.1.B Función: PrintPositionEvents de QTISM

Utilidad: Sirve para memoriz las posiciones de los usuarios una vez han hecho *login* en la aplicación.

Función: lecturaEntradaXML.java

Localización: La función lecturaEntradaXML se puede encontrar dentro del paquete “coreservlets”.

Captura imagen de la función (Figura 3.2.4.1.C Función: lecturaEntradaXML de QTISM):

```

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String Alias = request.getParameter("Alias").toUpperCase();
    String Ruta = request.getParameter("Ruta").toUpperCase();
    String Cont = request.getParameter("Cont");
    String Latlong = request.getParameter("Latlong");
    String Hora = request.getParameter("Hora");
    String Min = request.getParameter("Min");
    String Seg = request.getParameter("Seg");

    Latlong.equals(Latlong.replace("(", ""));
    Latlong.equals(Latlong.replace(")", ""));

    MyFileIO File = new MyFileIO();
    File.stringToFile("E:/apache-tomcat-6.0.24/webapps/QTISM/Positions/" + Alias + ".txt", Latlong);
    File.stringToFile("E:/apache-tomcat-6.0.24/webapps/QTISM/Positions/" + Alias + "_Date.txt", "<user>" + Alias + "</user><pos>" + Latlong + "</pos><

    response.sendRedirect(response.encodeRedirectURL("ppal.jsp?"));
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}

```

Figura 3.2.4.1.C Función: lecturaEntradaXML de QTISM

Utilidad: Sirve para *renderizar* el XML que proporciona QTIS. Posteriormente se podrá interpretar esa información en un mapa de *Google Maps*, el cual puede verse desde un *tablet* o

smartphone. Esta operación, como se puede ver, se realiza mediante cargas de datos del XML y base de datos, combinando con bucles *for*.

Función: ppal.jsp

Localización: La presentación HTML “ppal.jsp” se puede encontrar dentro del directorio de presentaciones WEB-INF.

Captura imagen de la función (Figura 3.2.4.1.D Función: ppal de QTISM):

```
if(j==1){ //Eliminem la ruta fins l'event anterior per recalcular la nova
    j = 0;
    eliminar();
    //calcRoute();
}

<#for (int i = 0; i < Longitud_Ruta; i++){ %>

    narrow = distancia(eventos[<%=i%>][1], eventos[<%=i%>][2]);

    if ((narrow < 12)&&(GLoad.getValueMark(lecturaEntradaXML.lectura_ID(URL_XML, i), Alias)%>==5)&&(j==0)&&(session.getAttribute("CANCELADA")
    //Si la distancia entre l'event i l'usuari es menor a 15 metres salta l'event

        if ((narrow <= narrow)|| (narrowAUX=="")) { //agafem l'event mes proper
            narrowAUX = narrow;
            narrowID = <%=i%>;
        }
    }
    <%=i%>

    location.href= "events.jsp?ID=" + eventos[narrowID][0] + "&Latlong=" + myLatlng;
}

</script>
```

Line: 210

Figura 3.2.4.1.D Función: ppal de QTISM

Utilidad: Como se puede observar, las presentaciones en la aplicación QTISM contienen partes de código, ya que al ejecutarse en el dispositivo del cliente esta parte de código no se puede ejecutar desde el servidor. La presentación “ppal.jsp” es el núcleo de toda la aplicación de QTISM, donde se controla que ruta ha escogido el usuario, en que posición se encuentra, si tiene eventos cerca de la posición (configurado de forma predeterminada a 12 metros de radio de acción) y refrescar la posición de forma continua, entre otras acciones.

3.2.5 Procesado de la información mediante XML

Una de las mejoras más importantes que presenta este proyecto, es poder consolidar la información de una manera estandarizada. De esta manera conseguimos que esta sea más perdurable y más entendible para otras personas.

En el proyecto tenemos tres tipos de procesos distintos para guardar información dentro de XML, y que a continuación se presentan y detallan.

3.2.5.1 Generación de XML para sistema de monitorización

El sistema de monitorización ha mejorado con la incorporación de XML, que sirven para memorizar las posiciones de los alumnos y hora en que se encuentran durante el uso de la herramienta.

En primer lugar el usuario se conecta a QTISM mediante su dispositivo móvil y este llama al servlet PrintPosition para que le comunique su posición y hora en la que se encuentra el usuario. El servlet PrintPosition recoge esta información y la redacta en un archivo XML con una estructura definida. Posteriormente este archivo es leído por otro servlet llamado indexMonitor, que renderiza la información en un mapa de Google Maps ayudándose de su Api, para que esta información se pueda ver en el sistema de monitorización que dispone la versión web de QTI. A continuación se puede ver en la siguiente imagen (ver imagen Figura 3.2.5.1.A Modelo generación XML sistema de monitorización):

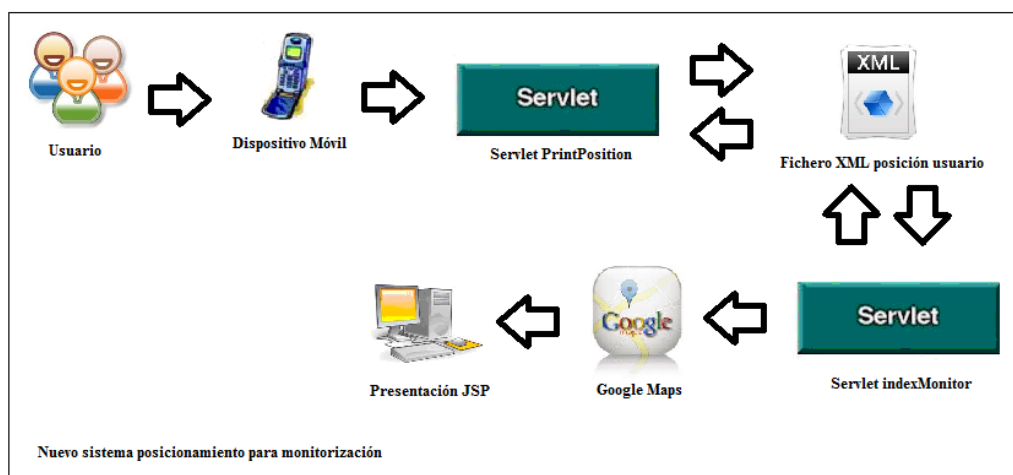


Figura 3.2.5.1.A Modelo generación XML sistema de monitorización

3.2.5.2 Generación de XML para encapsulación de datos

Los datos que utilizan la herramienta de QTIS, en este nuevo proyecto se pueden encapsular en formato XML. Esto permite poder contener información y generar informes muy útiles para el maestro. Ya que con un solo click, ahora puede disponer de resúmenes de datos de las actividades realizadas con la herramienta.

Esta generación se nutre de las interacciones que van haciendo los diferentes usuarios con la herramienta, mientras estos van realizando la actividad el servlet PrintPosition, va memorizando los pasos que van dando los usuarios. Esta información complementada con la que puede el servlet recoger de la propia base de datos, hace que pueda generar un informe en XML de los resultados de los usuarios en la actividad. A continuación se puede ver en la siguiente imagen (ver imagen Figura 3.2.5.2.A Modelo generación XML para encapsulación de datos):

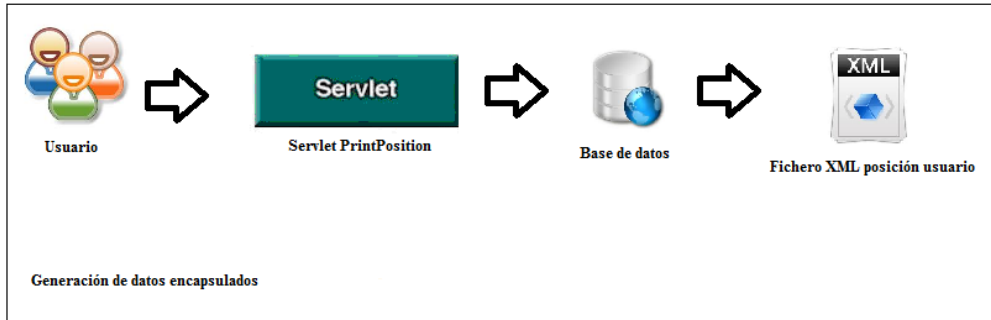


Figura 3.2.5.2.A Modelo generación XML para encapsulación de datos

3.2.5.3 Generación de XML para sistema de rutas

El último modelo para generar XML, es uno que en el anterior proyecto se detectó como muy importante para trabajo futuro, y que ha sido gran motivo para este nuevo trabajo. El sistema de rutas no es predefinido y el usuario puede seleccionar y hacer la ruta que desee sin necesidad del apoyo de un administrador.

Este proceso se lleva a cabo en la aplicación QTISM. El usuario mediante su dispositivo móvil, conecta con la aplicación QTISM. Una vez dentro el usuario le aparece un menú con las distintas rutas disponibles, este selecciona una de ellas y el dispositivo móvil mediante su red 3G se lo comunica a la herramienta. La herramienta dispone de un servlet que recoge la opción escogida por el usuario y consulta las propiedades en la base de datos (posiciones, numero de preguntas, id de las preguntas,...). Una vez que la herramienta conoce las propiedades, redacta un archivo XML único e individual con estas propiedades, que servirá para la herramienta como documento de seguimiento para conocer en que estado de la ruta se encuentra el usuario. A continuación se puede ver en la siguiente imagen (ver imagen Figura 3.2.5.3.A Modelo generación XML para nuevo sistema de rutas):

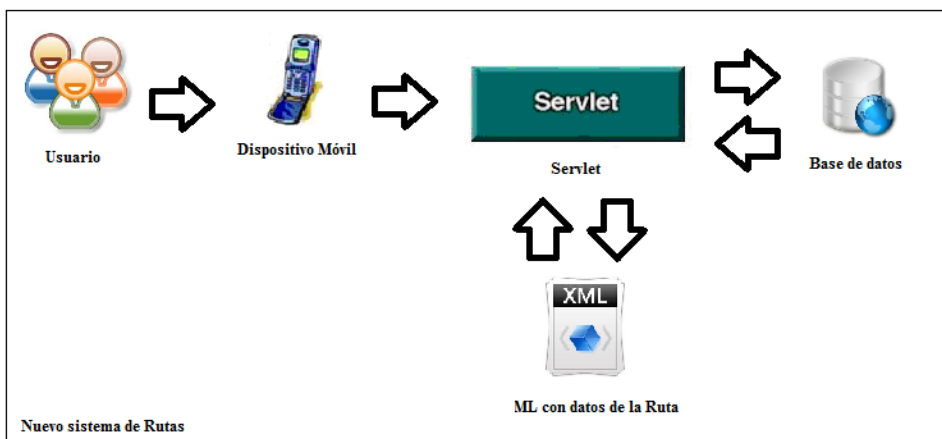


Figura 3.2.5.3.A Modelo generación XML para nuevo sistema de rutas

3.3 Base de datos

QuesTInSitu consta de dos bases de datos, divididas en dos bloques. Por un lado encontramos la base de datos de *NewApis* que da soporte al motor de *renderizar* preguntas de *NewApis*. Por otro lado encontramos la base de datos de QTIS, encargada de dar soporte en memoria a datos para las aplicaciones de QTIS y QTISM. Estas dos bases de datos están programadas en lenguaje MySQL.

3.3.1 Documentación de la base de datos

Descripción general de la base de datos mediante gráficos y detalles de las funcionalidades y capacidades de las tablas del proyecto, teniendo en cuenta las nuevas mejoras. Para ello se muestra un esquema de cómo esta compuesta la base de datos del proyecto (ver imagen *Figura 3.3.1.A Estructura base de datos del proyecto*):

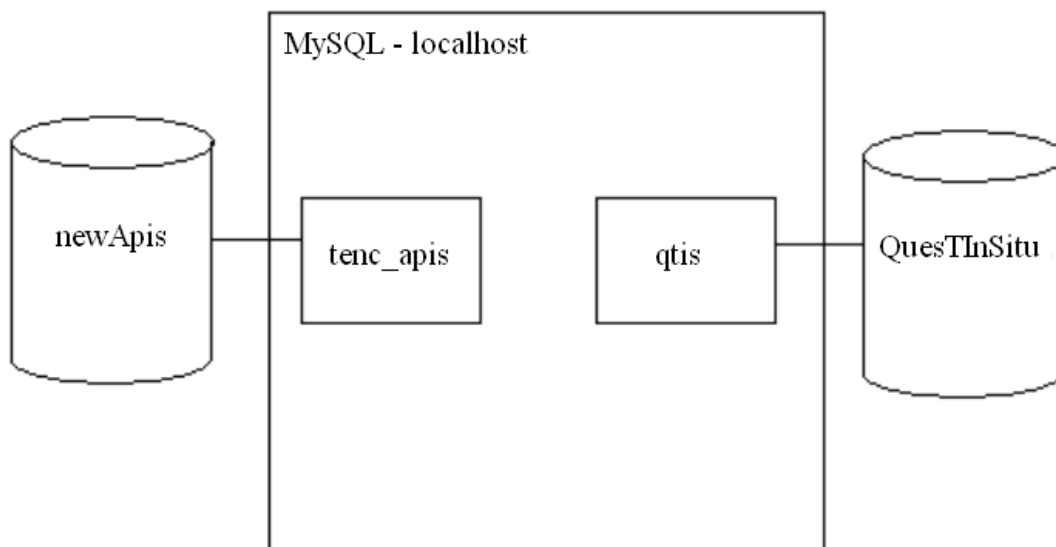


Figura 3.3.1.A Estructura base de datos del proyecto

3.3.2 Base de datos de *NewApis*

A continuación se muestra un gráfico con la estructura de la base de datos relacional de *NewApis*, donde se muestran diferentes tablas sus respectivos atributos (ver imagen *Figura 3.3.2.A Estructura base de datos de NewApis*):

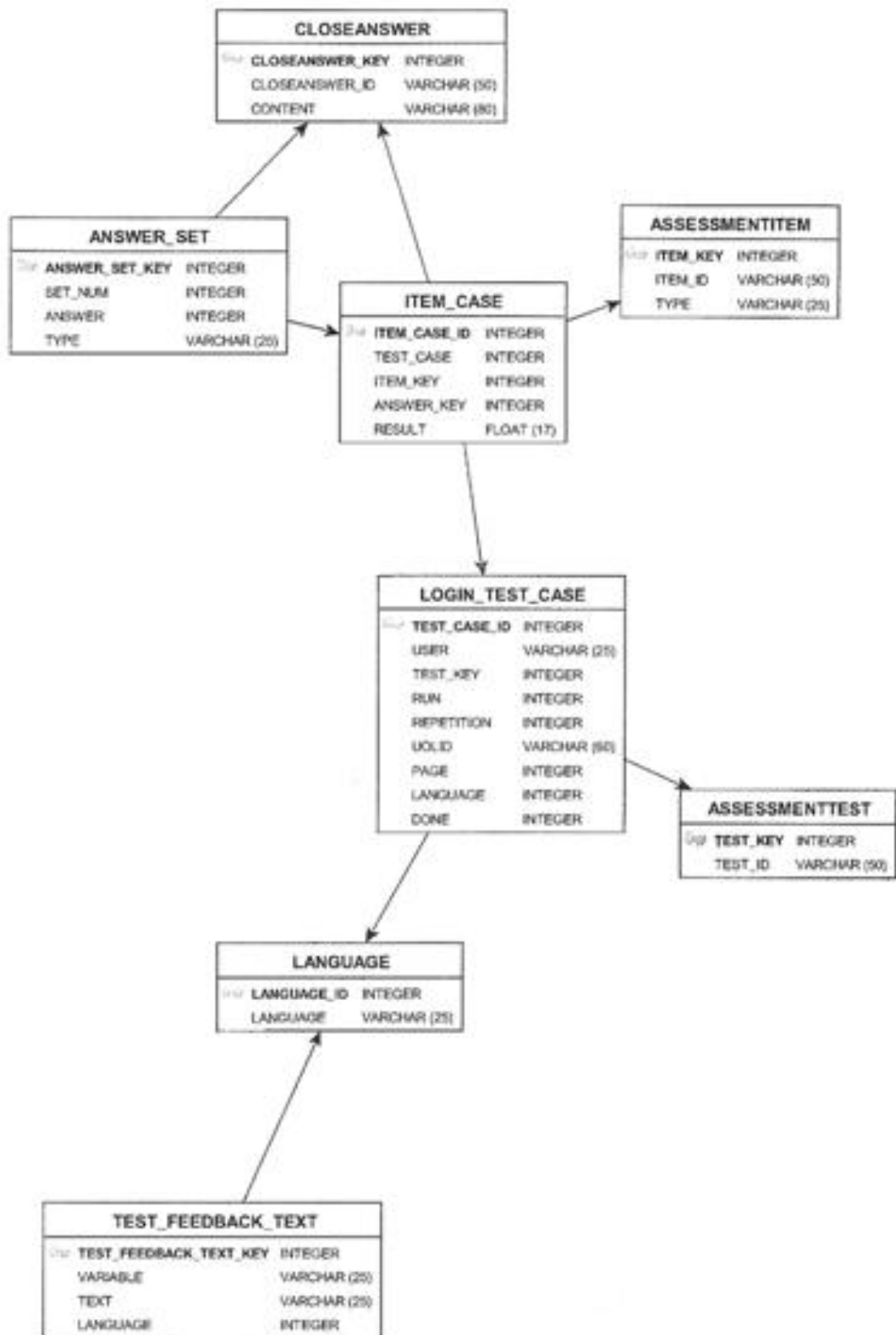


Figura 3.3.2.A Estructura base de datos de NewApis

Descripción de las tablas

A continuación se detalla una descripción de la misión que tiene que acometer cada tabla.

Closeanswer

Tabla que relaciona la opción escogida según la identidad del *ítem*, esta opción la relaciona con los registros de feedback de todos los test para así mostrar el enunciado resuelto.

Answer_set

Tabla que relaciona las opciones escogidas en cada iteración

Item_case

Registro general de las iteraciones realizadas con los puntos obtenidos.

Assessmentitem

Esta tabla contiene las identidades de los diferentes tipos de preguntas. Por ejemplo el tipo de pregunta *Multiple Choice* corresponde a la ID: AI-dcc54401-5035-4e57-a9ab-564755414f7a-4 '

Login test case

Tabla que mantiene un registro de los usuarios para ver qué preguntas han consultado.

Assessmenttest

Tabla que mantiene la correspondencia del orden de preguntas que se han ido respondiendo.

Language

Esta tabla contiene los lenguajes definidos que contiene el motor, que pueden ser catalán, castellano, inglés, alemán,..., etc.

Test feedback text

Tabla que contiene cadenas de caracteres para poder construir el feedback y datos del enunciado de los cuestionarios

3.3.3 Base de datos de QTIS y QTISM

La base de datos de QTIS y QTISM es de uso común, ya que así podemos beneficiarnos de no tener que repetir información y facilitamos la sincronización de datos entre las dos aplicaciones. Para ello se ha diseñado una base de datos que sea transportable y escalable para ambos casos.

La base de datos consta de seis tablas relacionadas, cada una de ellas con su temática y funcionalidad individual. Estas tablas están relacionadas teniendo en cuenta dos mesas principales, que son la tabla de usuarios (USERS) y la que contiene los test (TEST). A partir de estas dos tablas se relaciona el resto, como por ejemplo la mesa de control de interacciones (TESTUSERDONE) que toma el atributo "ALIAS" de la tabla de usuarios y "TEST" de la tabla de test.

Como ejemplo se muestra la distribución de la estructura de base de datos "qtis", construida con MySQL y que se ha usado como referencia para guardar los datos de la herramienta QuesTInSitu y QTISM, y que se muestra a continuación (ver imagen *Figura 3.3.3.A Estructura base de datos de QTIS*):

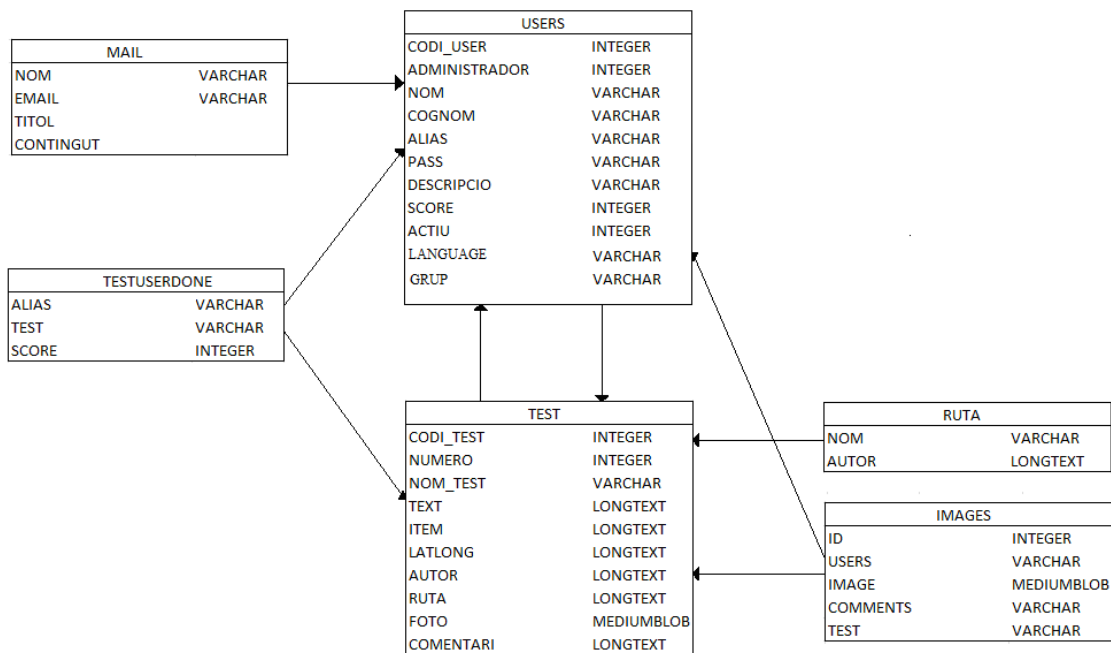


Figura 3.3.3.A Estructura base de datos de QTIS

A continuación se explicaran cuáles son las tablas que contiene esta base de datos "qtis" y sus principales características y funcionalidades:

- **USERS:** Tabla que contiene todas las propiedades de todos los usuarios del portal web, como los atributos: Nombre, Apellido, Administrador (tipo de rol del usuario), etc.... A partir de esta tabla se pueden hacer presentaciones como la pantalla de listar usuarios "Usuarios" o cualquier otro que tenga contenida información sobre ellos.

- **TEST:** Podemos comprobar que la tabla test es de las tablas que contienen más información, ya que contiene todos los test de la herramienta, con lo que conlleva un gran número de registros. Esta tabla tiene propiedades básicas para los test como los atributos: Número (identidad), Autor, LatLong (coordenadas del test), Ruta (ruta a la que pertenece el test), etc.... Mediante esta tabla podemos representar toda la información de las pantallas que contengan informes sobre los test. Un ejemplo puede ser la pantalla de todos los test listados "Tests".
- **MAIL:** Tabla que tiene la funcionalidad de gestionar todo el sistema de correo de la aplicación web, para dar soporte a los usuarios del mismo.
- **TESTUSERDONE:** Esta tabla tiene la funcionalidad de registrar todas las interacciones realizadas en cada maceta para cada usuario y la puntuación que ha obtenido en cada una de ellas. A partir de esta tabla se pueden hacer presentaciones como la pantalla de la herramienta con la puntuación de todos los usuarios "Ranking".
- **RUTA:** Tabla que contiene todas las rutas creadas en la aplicación. Esta tabla está relacionada directamente con el atributo ruta de la tabla test. Esta tabla realiza todas las presentaciones que contienen información sobre rutas, como la pantalla de "Gestor de rutas".
- **IMAGES:** Tabla que gestiona todas las imágenes de la aplicación. Esta tabla se relaciona con la tabla de test mediante los atributos ID (IMAGES) - NUMERO de (TEST). Esta tabla nos permite representar cualquier representación de pantalla que contenga una imagen tomada por la aplicación de telefonía móvil. Un ejemplo sería en la pantalla de responder cuestionarios y en los marcadores se hace clic en la pestaña de imágenes. En la Figura 3.3.3.B (Pantalla panel de imágenes de QTIS) se muestra un ejemplo de una pantalla de la herramienta, con un conjunto de imágenes memorizadas en la base de datos realizadas por uno de los grupos durante la experiencia del escenario educativo en el barrio de Les Corts.



Figura 3.3.3.B Pantalla panel de imágenes de QTIS

3.4 Interfaz y clases relacionadas

En este apartado se detallarán las pantallas más importantes del software de QTIS, subdividiendo cada apartado en una pantalla en concreto. En cada uno de las diferentes secciones también iremos destacando que funcionalidades nuevas se han añadido en este proyecto.

3.4.1 Secciones más importantes de QTIS

En este apartado se detallarán las pantallas más importantes del software de QTIS, subdividiendo cada apartado en una pantalla en concreto.

3.4.1.1 Pantalla: Crear

Esta pantalla (ver imagen Figura 3.4.1.1.A Pantalla crear preguntas de QTIS) es la encargada de construir las preguntas geolocalizadas. En ella se puede escoger el tipo de pregunta que se quiere construir a partir de las siguientes posibilidades: *Multiple Choice*, *Multiple Response*, *True / False*. También existe la opción de geolocalizar la pregunta sobre un punto del mapa y configurar el enunciado y respuestas según los conceptos que se quieran utilizar para hacer la evaluación, como se puede observar en el formulario con los diferentes campos que se exponen a continuación:

Creador de Preguntas

Si us plau seleccioneu un tipus de pregunta que volgueu crear i ompliu les dades que us demanem: admin

Multiple Choice
 Multiple Response
 Yes/No

M/Choice M/Response Yes/No

Pregunta tipus Multiple Choice

Enunciat de la pregunta...

Categoria

Resposta Correcta:

Localitza en el mapa la teva Pregunta:

Resposta1 Feedback:
Resposta2 Feedback:
Resposta3 Feedback:
Resposta4 Feedback:

(1) Aplicar Lloc (2) Crear Pregunta

Figura 3.4.1.1.A Pantalla crear preguntas de QTIS

3.4.1.1.2 Acciones posibles:

- Crear pregunta del tipo: *Multiple Choice, Multiple Response, True / False*.
- Configuración de los enunciados y posibles respuestas.
- Geoposicionar la pregunta sobre el mapa.
- Crear una pregunta geolocalizada IMS QTI [13] según el tipo: *Multiple Choice, Multiple Response, True / False*.
- Ampliar el localizador de cuestionarios (ver imagen Figura 3.4.1.1.2.A Pantalla geolocalizar preguntas de QTIS)

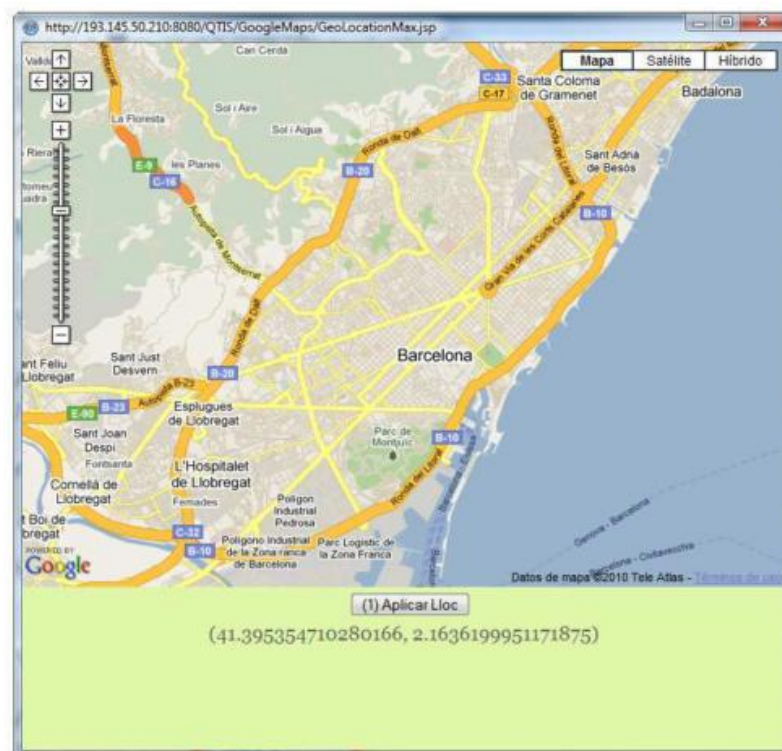


Figura 3.4.1.1.2.A Pantalla geolocalizar preguntas de QTIS

3.4.1.1.3 Clases que intervienen:

- **Creador de Test.java (IODA):** Clase de código Java que sirve para crear preguntas IMS QTI [13] del tipo *True / False*, a partir de plantillas que se encuentran en la base de datos "pfc".

- **CreadordeTestMC.java (IODA):** Clase de código Java que sirve para crear preguntas IMS QTI [13] del tipo *Multiple Choice*, a partir de plantillas que se encuentran en la base de datos "pfc".

- **CreadordeTestMR.java (IODA):** Clase de código Java que sirve para crear preguntas IMS QTI [13] del tipo *Multiple Response*, a partir de plantillas que se encuentran en la base de datos "pfc".

- **IOSQL.java (IODA):** Clase de código Java que sirve para crear un nuevo registro en la base de datos de la tabla test.

- **MyFileIO.java (IODA):** Clase de código Java tiene como objetivo la gestión de ficheros de entrada y de salida. Esta clase nos es útil para redactar los archivos “.xml” para cada pregunta de IMS QTI [13] que se crea en la aplicación.

- **TestNumber.java (IODA):** Clase de código Java que devuelve la cantidad de test que hay creados en QuesTInSitu.

- **TestQuery.java (IODA):** Clase de código Java que devuelve el contenido del archivo en la base de datos. Esta función es muy útil para cargar las plantillas de texto prefijo para confeccionar los *XML*.

- **GSetLatLong.java (Gmaps):** Clase de código Java que permite memorizar la latitud y longitud del nuevo cuestionario creado.

- **CrearTest.jsp:** Clase de código JSP que permite la representación de la pantalla para crear preguntas y la inclusión de los *Servlets* y *Scripts* necesarios.

- **GeoLocation.jsp:** Clase de código JSP que permite la representación del mapa pequeño para localizar la pregunta.

- **GeoLocationMax.jsp:** Clase de código JSP que permite la representación del mapa pequeño para localizar la pregunta de forma ampliada.

- **Javascript.js:** Clase de código JavaScript que permite el control del tráfico y gestión de la pantalla mediante funciones de *Scripts*.

- **Default.css:** Clase de tipo CSS que permite configurar estilos para la pantalla.

3.4.1.1.4 Trabajo futuro

Como se puede ver en la imagen, hay unas opciones para poder otorgar una categoría a las preguntas. Estas categorías clasifican las preguntas según su temática, como por ejemplo preguntas de ciencias, religión, deportes, etc.... Esta parte de código aún no está completamente operativa, aunque la opción se puede ver en la pantalla.

3.4.1.1.5 Novedades

En esta pantalla se incorpora un menú para poder asociar a una pregunta una semántica en concreto. Aunque esta funcionalidad como comentamos anteriormente no está plenamente incorporada a la herramienta, aunque si está preparada para ello. Esta más focalizada a cuando la herramienta tenga un volumen alto de preguntas y sea necesario filtrar por algún tipo de criterio todas ellas.

3.4.1.2 Pantalla: Responder Pregunta

Esta pantalla (ver imagen Figura 3.4.1.2.A Pantalla menú tipo interacción de QTIS) es la encargada de responder las preguntas geolocalizadas. En ella se puede escoger el tipo de pregunta que se quiere responder a partir de las siguientes posibilidades: responder una pregunta de forma aleatoria en el mapa, responder una pregunta de entre todas las que hay en la aplicación mediante un mapa, responder preguntas mediante la aplicación móvil.



Figura 3.4.1.2.A Pantalla menú tipo interacción de QTIS

Una vez seleccionada la modalidad de juego se muestra un mapa. Para las dos primeras opciones es necesario clicar en los punteros para que aparezca la pregunta. En el caso de la opción de móvil tendrá que desplazarse según el mapa con un dispositivo al punto que se indica con una estrella.

3.4.1.2.1 Acciones posibles:

- Escoger una pregunta de entre el mapa / aleatoria o responder en una yincana geolocalizada mediante la aplicación para móviles 3G.
- Ver las respuestas correctas e incorrectas y las que todavía están por contestar.
- Escoger preguntas por rutas.
- Ver el total de cuestionarios que hay en la herramienta.

A continuación se muestra el menú con los diferentes iconos que describen las acciones anteriormente enunciadas (ver imagen Figura 3.4.1.2.1.A Pantalla tipos de QTIS):

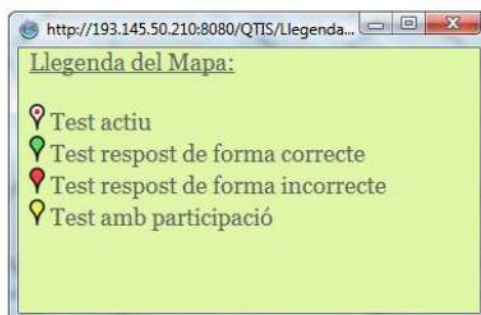


Figura 3.4.1.2.1.A Pantalla tipos de QTIS

Posibles interacciones con los marcadores: Disponemos de tres pestañas. En la pestaña 'Test' tenemos "ver enunciado" "pregunta y contestar" o ver el feedback si ésta ya está respuesta. En la pestaña 'Puntos' podemos ver los puntos de todos los usuarios en esta pregunta y en la pestaña 'Imágenes' poder visualizar las imágenes del usuario en esta pregunta o en todas las de la aplicación. Esto se muestra en la Figura 3.4.1.2.1.B (Pantalla con diferentes interacciones con preguntas en el mapa):



Figura 3.4.1.2.1.B Pantalla con diferentes interacciones con preguntas en el mapa

3.4.1.2.2 Clases que intervienen:

- **RunQTITest.java (apisRun):** Clase de código Java que crea un *Servlet*, el cual permite la conexión entre los Scripts de los mapas de *Google Maps* con el motor de cuestionarios *IMS QTI NewApis* [13] mediante los parámetros de entrada para el método *GET*: identidad de cuestionario, usuario e idioma.
- **IdNameUser.java (apisRun):** Traduce el *string* que recibe la clase de *RunQTITest* del alias activo en la aplicación a su número de Identidad correspondiente.
- **Monitor.java (Gmaps):** Clase de código Java que permite mostrar la puntuación de un marcador del mapa cuando se hace clic.
- **OwnQtiParser.java (apisRun):** Clase de Java que sirve para configurar la base de datos que tiene el motor de cuestionarios *IMS QTI NewApis* [13].
- **SaveQTIScore.java (apisRun):** Clase de código Java que guarda el usuario y la puntuación que recibe por parámetro en la base de datos "pfc".
- **BeanRute.java (coreservlets):** Clase de código Java que crea un *Servlet* que permite conocer cuál es la ruta que quiere visualizar al usuario en el mapa.
- **BeanRuteMap.java (coreservlets):** Clase de código Java que crea un *Servlet* que permite redireccionar a la pantalla del mapa con preguntas *IMS QTI* [13].

- **Mapa.jsp / MapaRandom.jsp:** Clase de código JSP que permite la representación de las dos pantallas para responder preguntas y la inclusión de los *Servlets* y *Scripts* necesarios.

- **Index.jsp / indexRandom.jsp:** Clase de código JSP, que permite incluir los *Scripts* de la API de *Google Maps* para mostrar el mapa.

- **Javascript.js:** Clase de código JavaScript que permite el control del tráfico y gestión de la pantalla mediante funciones de *Scripts*.

- **Default.css:** Clase de tipo CSS que permite configurar estilos para la pantalla.

* *La función encargada de cargar las imágenes es propia de la aplicación de móvil, ya que es ella quien se encarga de capturarlas.*

3.4.1.3 Pantalla: Monitor

Esta pantalla (ver imagen Figura 3.4.1.3.A Pantalla monitorización de QTIS) es la encargada de la monitorización de las actividades que se llevan a cabo con dispositivos móviles 3G. Permite controlar las interacciones que realizan los grupos en cada punto del recorrido y ver en qué punto de la ruta se encuentran los alumnos mediante los iconos azules que se ven en dicha figura.



Figura 3.4.1.3.A Pantalla monitorización de QTIS

3.4.1.3.1 Acciones posibles:

- Control in situ de la actividad mediante el mapa.
- Ver las respuestas correctas e incorrectas que hacen los alumnos durante la ruta y las que todavía están por contestar de la ruta.
- Escoger visualizar por rutas o todos los cuestionarios.
- Ver en qué punto del mapa se encuentra el grupo de alumnos haciendo la ruta.
- Consultar la leyenda del mapa.
- Consultar la puntuación obtenida en cada cuestionario de la ruta.
- Ver el total de cuestionarios que hay en la herramienta.

3.4.1.3.2 Clases que intervienen:

- **BeanRute.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual permite conocer cuál es la ruta que quiere visualizar al usuario en el mapa.
- **Monitor.java (Gmaps):** Clase de código Java que permite mostrar la puntuación de un marcador del mapa cuando se hace clic.
- **MapaMonitor.jsp:** Clase de código JSP que permite la representación de la pantalla para monitorizar actividades con dispositivos 3G como yincanas y la inclusión de los *Servlets* y *Scripts* necesarios.
- **IndexMonitor.jsp:** Clase de código JSP que permite incluir los *Scripts* de la API de *Google Maps* para mostrar el mapa.
- **Javascript.js:** Clase de código JavaScript que permite el control del tráfico y gestión de la pantalla mediante funciones de *Scripts*.
- **Default.css:** Clase de tipo CSS que permite configurar estilos para la pantalla.

3.4.1.3.3 Novedades

Esta pantalla presenta tanto en el motor de presentación de datos, como en la interfaz. La pantalla actualiza las posiciones de los usuarios y crea un icono para cada uno de ellos. Otro factor importante es que la localización de los usuarios es en tiempo real y precisa, no como pasaba con la versión anterior de la aplicación.

3.4.1.4 Pantalla: Gestor de Rutas

Esta pantalla (ver imagen Figura 3.4.1.4.A Pantalla menú gestor de rutas de QTIS) es la encargada del administrador de todas las rutas y los cuestionarios que se encuentran contenidos de toda la herramienta QuesTInSitu

Gestor de Rutes

En aquesta pantalla, podras crear o gestionar rutes per a la gimcana i gestionar els test de cada ruta

Si vols crear una ruta nova ruta:

[Editar Ruta des del Mapa](#)

Afegir a la Ruta

Carregar les rutes existents:

RUTES			
Test	Itineraris	Autor	Eliminar
Test1.xml		ADMIN	***
Test2.xml		ADMIN	***
Test3.xml		PSANTOS	***
Test4.xml	No	PSANTOS	***
Test5.xml		PSANTOS	***
Test6.xml	No	MCASAMITJANA	***
Test7.xml	No	MCOLLADO	***
Test8.xml		RVALLES	***
Test9.xml	No	PHEITMANN	***
Test10.xml	No	MBORDAS	***
Test11.xml	No	XRUIZ	***
Test12.xml	No	DVERGES	***

Figura 3.4.1.4.A Pantalla menú gestor de rutas de QTIS

3.4.1.4.1 Acciones posibles:

- Control y visualización de todas las rutas de la herramienta.
- Crear una ruta de la herramienta.
- Eliminar una ruta de la herramienta.
- Añadir un test a una ruta de la herramienta.
- Eliminar un test de una ruta de la herramienta.
- Ver el total de cuestionarios que hay en la herramienta (interfaz cartográfica).
- Realizar todas las acciones anteriores pero con una interfaz cartográfica (interfaz más intuitiva y localizada), mediante la opción: “Editar Ruta desde el Mapa”. En la siguiente página se muestra una captura de ejemplo (ver imagen Figura 3.4.1.4.1.A Pantalla editar preguntas sobre mapas de QTIS).

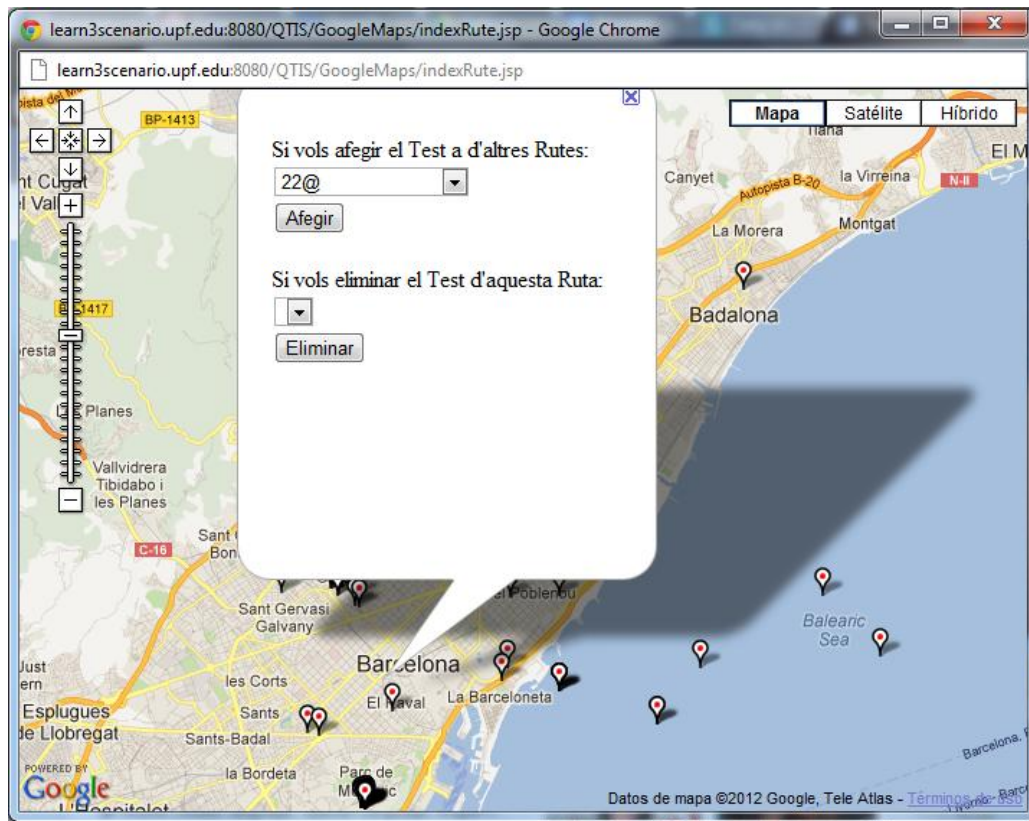


Figura 3.4.1.4.1.A Pantalla editar preguntas sobre mapas de QTIS

3.4.1.4.2 Clases que intervienen:

- **InsertTestRoutePopUp.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual permite insertar un cuestionario de una ruta del marcador del mapa seleccionado.
- **DropTestRoutePopUp.java (coreservlets):** Clase de código Java que crea un *Servlet* que permite eliminar un cuestionario de una ruta del marcador del mapa seleccionado.
- **DAO.java (DAO):** Clase de código Java que permite listar todas las rutas y cuestionarios en una tabla.
- **DAORute.java (DAO):** Clase de código Java que permite conectar con la base de datos de MySQL.
- **BeanInsertRuteMap.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual permite insertar un cuestionario a una ruta de la lista.
- **BeanDeleteRuteMap.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual permite eliminar un cuestionario a una ruta de la lista.

- **RuteMaker.jsp:** Clase de código JSP que permite la gestión de rutas mediante una presentación con un listado de todos los cuestionarios y con su ruta correspondiente. También permite la inclusión de los *Servlets* y *Scripts* necesarios.

- **RutePopUp.jsp:** Clase de código JSP que permite la gestión de rutas mediante una presentación con un mapa donde están los marcadores para cada uno de los cuestionarios y que permite la gestión de rutas. También permite la inclusión de los *Servlets* y *Scripts* necesarios.

- **IndexRute.jsp:** Clase de código JSP que permite incluir los *Scripts* de la API de *Google Maps* para mostrar el mapa (sólo interfaz cartográfica).

- **Javascript.js:** Clase de código JavaScript que permite el control del tráfico y gestión de las pantallas mediante funciones de *Scripts*.

- **Default.css:** Clase de tipo CSS que permite configurar estilos para la pantalla.

3.4.1.4.3 Novedades:

Esta pantalla presenta un mejor funcionamiento que la versión anterior. En la versión anterior a veces ocurrían fallos internos que provocaban resultados incorrectos de las acciones que se podían realizar.

3.4.1.5 Pantalla: Administrador

Esta pantalla (ver imagen Figura 3.4.1.5.A Botones administrador de QTIS) permite realizar operaciones delimitadas para que puedan realizarse sólo por un administrador. En ella el administrador puede encontrar botones con acciones que facilitan la generación de resultados o control de datos y que facilitan la tarea al Administrador, puesto que no tiene que hacer estas operaciones accediendo directamente al servidor y realizarlas de manera manual. Además, se aumenta la seguridad en el acceso del servidor, ya que hace que sólo sea necesario para operaciones realmente críticas y necesarias.

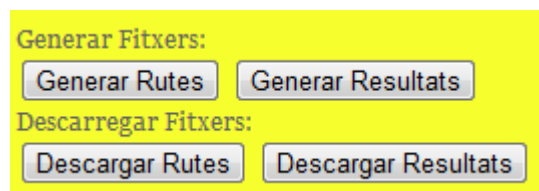


Figura 3.4.1.5.A Botones administrador de QTIS

3.4.1.5.1 Acciones posibles:

- Generar fichero con una lista de todas las rutas de la aplicación en XML.
- Generar fichero con un listado de todos los resultados de los usuarios de la aplicación en XML.
- Descargar fichero con una lista de todas las rutas de la aplicación en XML.
- Descargar fichero con un listado de todos los resultados de los usuarios de la aplicación en XML.

3.4.1.5.2 Clases que intervienen:

- **BeanAdminMenu.java (coreservlets):** Clase de código Java que crea un *Servlet* que permite rellenar el contenido del XML de todas las rutas existentes en la aplicación y las memoriza en el fichero Rute.txt.

- **BeanAdminResults.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual permite rellenar el contenido del XML de resultados de todos los usuarios mediante código y lo salva en un fichero llamado RuteResults.txt.

- **Admin.jsp:** Clase de código JSP que sirve para realizar la presentación del menú de botones por pantalla para que el usuario pueda interactuar y ver que botones puede escoger.

- **MostrarAux.jsp:** Clase de código JSP que permite escoger el directorio de destino para guardar los archivos XML.

3.4.1.5.3 Resultados obtenidos

Fichero Rute.txt

Este fichero se encuentra en el servidor en el directorio: "E:/apache-tomcat-6.0.24/webapps/QTIS/XMLRute/" (Ver *Figura 3.4.1.5.3.a Resultados fichero Rute.txt*)

```
<routes>
<number_routes> numero total de rutas </number_routes>
<route>
<route_name>nombre ruta </route_name>
<id_route> id ruta</id_routes>
<author_route> nombre autor </author_route>
<date_route> fecha creación ruta </date_route>
<questions>
<number_questions> numero de preguntas que contiene la ruta </number_questions>
<question>
<id_question> id question </id_question>
<author_question> nombre autor pregunta </author_question>
<date_question> fecha creación pregunta </date_question>
</question>
<question> ..... </question>
...
</questions>
</route>
<route>
... próxima información de la ruta
...
</route>
</routes>
```

Figura 3.4.1.5.3.A Resultados fichero Rute.txt

Fichero RuteResults.txt

Este fichero se encuentra en el servidor en el directorio: "E:/apache-tomcat-6.0.24/webapps/QTIS/XMLRute/" (Ver *Figura 3.4.1.5.3.B Resultados fichero RuteResults.txt*)

```
<questions>
<number_questions> numero total de preguntas realizadas </number_questions >
<question>
<id_question> id pregunta</id_routes>
<author_question> nombre autor </author_route>
<date_question> fecha creación pregunta </date_route>
</question>
<question> ..... </question>
... próxima información de resultados de otra pregunta
</questions>
```

Figura 3.4.1.5.3.B Resultados fichero RuteResults.txt

3.4.1.5.4 Objetivos XML

El objetivo de los archivos XML es permitir ayudar al profesor en la generación de listados y en la monitorización de la actividad orquestada. Esto quiere decir que, con los resultados que puede extraer de estos XML, puede obtener más datos para su evaluación y así dar más rendimiento a la evaluación geolocalizada. Para obtener estos informes se han creado unos botones de código crítico que sólo pueden ser accedidos y sólo pueden obtenerlos los usuarios con rol maestro, puesto que contienen datos confidenciales de los usuarios y hay que preservar su anonimato. Dado que era información que los profesores exigían y era útil para ellos se han creado estos botones, los cuales realizan operaciones que anteriormente solo podía realizar un administrador accediendo al servidor y haciendo peticiones de forma manual.

3.4.1.5.5 Trabajo futuro

En un futuro se tiene pensado incrementar la lista de botones con nuevas funcionalidades críticas que el administrador debe de hacer accediendo de forma manual en el servidor. En el campo de los ficheros XML, se pretende también poder memorizar la fecha de creación de las rutas y el momento en que los usuarios van obteniendo sus marcas.

3.4.1.5.6 Novedades

Esta pantalla es totalmente nueva, en ella se facilita al administrador una serie de botones para poder automatizar acciones de gestión de la herramienta (sobre todo de aspecto técnico), sin necesidad de tener que disponer de un técnico para que realice este tipo de tareas.

3.4.2 Pantallas más importantes de QTISM

En este apartado se detallaran las pantallas más importantes del software de QTISM, subdividiendo cada apartado en una pantalla en concreto.

3.4.2.1 Pantalla: Escoger Ruta

En esta pantalla (ver imagen Figura 3.4.2.1.A Menú escoger rutas de QTISM) se presenta un menú con las diferentes rutas activas en la aplicación. El usuario puede elegir la ruta que más le interese y escogerla para empezar la actividad con el recorrido deseado.



Nom	Autor	***
UPF-BP-2012	psantos	Escollir
PDiamant_GrupaB	psantos	Escollir
PDiamant_GrupaA	psantos	Escollir
VBotanic	admin	Escollir
Girona	psantos	Escollir
22@	admin	Escollir

QuestInSitu Mobile
Universitat Pompeu Fabra

Figura 3.4.2.1.A Menú escoger rutas de QTISM

3.4.2.1.1 Acciones posibles:

- Escoger una ruta en el menú de rutas disponibles.

3.4.2.1.2 Clases que intervienen:

- **ChooseRuteMapa.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual redirecciona al mapa de *Google Maps* con la ruta seleccionada.

- **QTISMRequest.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual interpreta el fichero .XML que envía QTI con la ruta que hemos seleccionado, para así conseguir su información y *parsear* los datos en el mapa de *Google Maps*.

3.4.2.1.3 Novedades

Esta pantalla es nueva en la aplicación móvil de la herramienta. Con esta nueva versión cada usuario puede escoger la ruta que quiera cargar des su dispositivo móvil y poder escoger realizar las rutas de forma ordenada o desordenada. Anteriormente tenía que ser un administrador que prefijara una ruta válida para todos los usuarios de la herramienta.

3.4.2.2 Pantalla: Mapa interactuar en el espacio

Esta pantalla (ver imagen Figura 3.4.2.2.A Menú escoger rutas de QTISM en la página siguiente) es la más importante de la aplicación móvil. Su funcionalidad es para desplazarse por el mapa, viendo como el usuario se mueve y se acerca a los puntos calientes (preguntas), con la finalidad de realizar la ruta y finalizar la actividad. En ella podemos observar varios elementos como:

- Mapa: Representación geográfica del entorno en que se encuentra.
- FPS: Este valor nos indica la velocidad de refresco de la posición.
- Usuario: Nos permite ver con que usuario nos hemos identificado.
- Ruta: Podemos ver que ruta se ha escogido.
- Score: Valor que indica la puntuación que se realiza durante la actividad. Este valor se inicializa a 0 y corresponde a un valor para cada ruta. Según las respuestas que responda el alumno en la ruta se irá acumulando puntos en este valor. Si el usuario responde de forma correcta sumará 1 punto, pero si lo hace de manera incorrecta se le restará -1/3 de puntos.

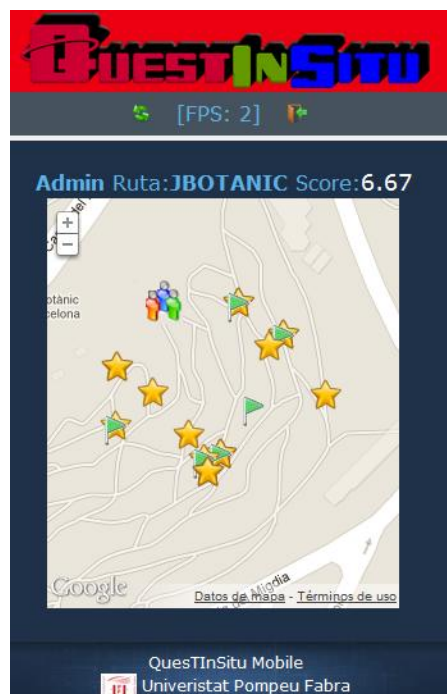


Figura 3.4.2.2.A Menú interactuar de QTISM

3.4.2.2.1 Acciones posibles:

- Interactuar con el mapa.
- Realizar una ruta entre las rutas disponibles.
- Ver puntos calientes en el mapa.
- Ver nuestra posición en la ruta.
- Actualizar nuestra posición en la ruta.
- Ver estado preguntas de la ruta.

3.4.2.2.2 Clases que intervienen:

- **anteriores.js (JavaScript):** Clase de código Java que crea un JavaScript el cual permite rellenar el contenido del XML de todas las rutas existentes en la aplicación y memorizarlas en el fichero Rute.txt.

- **calcruta.js (JavaScript):** Clase de código Java que crea un JavaScript el cual sitúa en el mapa de *Google Maps* las preguntas y el acceso a ellas.

- **cancelar.js (JavaScript):** Clase de código Java que crea un JavaScript el cual permite al usuario descartar una pregunta de la ruta, para que este pueda proseguir con la ruta, y marca la pregunta de color azul.

- **anteriores.js (JavaScript):** Clase de código Java que crea un JavaScript el cual comprueba el orden de la ruta. Esta funcionalidad es opcional, ya que en la última versión de la aplicación se pueden contestar las preguntas sin seguir el orden de la ruta.

- **contestadas.js (JavaScript):** Clase de código Java que crea un JavaScript el cual comprueba que preguntas de la ruta ya se han realizado para que no vuelvan a saltar cuando nos el usuario se acerque a ellas.

- **contador (JSP):** Clase de presentación en formato JSP que cuenta las preguntas que el usuario ha realizado y las que faltan para finalizar la ruta, así también el redireccionamiento a la pantalla de fin.

- **events (JSP):** Clase de presentación en formato JSP que comprueba la distancia entre el usuario y la pregunta. En caso de que el radio sea menor al valor definido (12 metros en este caso) la aplicación lanzará un evento.

- **ppal.jsp (JSP):** Clase de presentación en formato JSP que es el núcleo de toda la aplicación de QTISM. Desde ella se controla que ruta ha escogido el usuario, en que posición se encuentra, si tiene eventos cerca de su posición (configurado de forma predeterminada a 12 metros de radio de acción) y refrescar la posición de forma continua, entre otras acciones.

- **PrintPosition.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual memoriza en cada segundo la posición del usuario en un archivo .XML, para posteriormente *renderizar* su recorrido.

- **PrintPositionRefresh.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual memoriza en cada segundo la posición del usuario en un archivo .XML, para posteriormente *renderizar* su recorrido.

- **PrintPositionEvents.java (coreservlets):** Clase de código Java que crea un *Servlet* el cual memorizan en cada segundo la posición del usuario en un archivo .XML, para posteriormente *renderizar* su recorrido.

3.4.2.2.3 Novedades

Esta versión incluye un nuevo diseño de interacción para realizar actividades orquestadas en la aplicación móvil. Se han incorporado nuevos iconos como por ejemplo los diferentes tipos de banderas para cada tipo de interacción (bandera roja, bandera verde, bandera azul), o nueva información como la cantidad de puntos que lleva acumulada el usuario en la actividad. Aunque la mayor novedad en esta pantalla, es la de ofrecer al usuario la opción de poder elegir el no contestar una pregunta que le haya aparecido durante la actividad, ya sea porque no conoce su respuesta y no desea penalizar puntos o porque el usuario desea contestarla más tarde o ver más la zona para obtener más información y detalle, por ejemplo sea el caso de buscar una reseña o placa en un jardín sobre el artista que ha construido el monumento sobre el cual se le esta preguntando al alumno.

3.4.2.3 Detalle elementos del mapa

En la Figura 3.4.2.3.A (Iconos mapa de QTISM) se muestra una pequeña leyenda creada con un listado de los iconos que aparecen en el mapa. A continuación se explica que significan cada uno y cómo el usuario debe enfrentarse a ellos.



Figura 3.4.2.3.A Iconos mapa de QTISM

- **Figura 1:** Este icono aparecerá en la pantalla de interactuar con el espacio. Estará situado dentro del mapa e indicará la posición en que el usuario se encuentra dentro del espacio.
- **Figura 2:** Este icono aparecerá en la pantalla de interactuar con el espacio. Estará situado dentro del mapa e indicará a que dirección tiene que dirigirse el alumno para encontrar una pregunta a realizar.
- **Figura 3:** Este icono aparecerá en la pantalla de interactuar con el espacio. Estará situado dentro del mapa e indicará que se ha contestado una pregunta de forma positiva en dicha posición.
- **Figura 4:** Este icono aparecerá en la pantalla de interactuar con el espacio. Estará situado dentro del mapa e indicará que la pregunta de dicha posición ha sido accedida pero no contestada, dejando su resolución para otra ocasión.
- **Figura 5:** Este icono aparecerá en la pantalla de interactuar con el espacio. Estará situado dentro del mapa e indicará que se ha contestado una pregunta de forma negativa en dicha posición.
- **Figura 6:** Este icono aparecerá una vez identificado en la aplicación y servirá para que el dispositivo refresque la posición.
- **Figura 7:** Este icono aparecerá una vez identificado en la aplicación y servirá para retroceder a la página anterior.
- **Figura 8:** Este icono aparecerá una vez identificados en la aplicación y servirá para dejar de estar identificados en la aplicación y salir de ella.

3.4.2.3.1 Novedades

Como hemos dicho en el apartado 3.2.2.2.3 *Novedades*, esta versión incluye un nuevo diseño de interacción para realizar actividades orquestadas en la aplicación móvil. Y para ello se ha extendido la leyenda de iconos que se utilizan en la herramienta. Por ejemplo como hemos citado anteriormente, disponemos de una nueva gama de tipos de banderas para cada tipo de interacción (bandera roja, bandera verde, bandera azul).

3.4.2.4 Pantalla: Responder Pregunta

La Figura 3.4.2.4.A (Enunciado pregunta en QTISM) aparece cuando el alumno se encuentra a una distancia de 12 a 15 metros de radio respecto a una de las estrellas que aparece en el mapa. Una vez se despliega la pregunta se puede pulsar en la pantalla táctil del dispositivo que opción se desea responder y posteriormente el botón de “*submit*”.

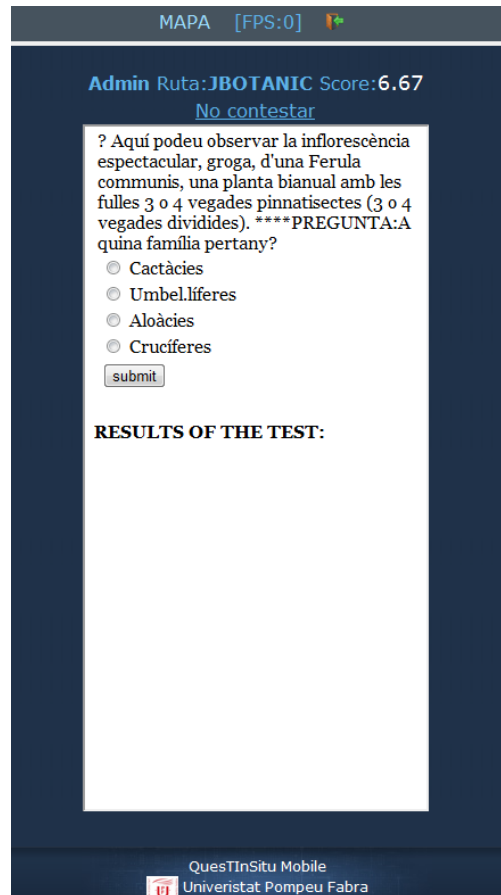


Figura 3.4.2.4.A Enunciado pregunta en QTISM

3.4.2.4.1 Novedades

Esta nueva versión como ya hemos citado en el apartado 3.4.2.2.3 *Novedades*, disponemos de la acción “No contestar” como se puede ver en la imagen Figura 3.4.2.4.A (Enunciado pregunta en QTISM), con la cual el usuario puede decir resolver la cuestión posteriormente, o simplemente obviarla.

3.4.2.5 Pantalla: Feedback Pregunta

La Figura 3.4.2.5.A (Enunciado pregunta feedback en QTISM) aparece una vez se ha contestado una pregunta. Esta se muestra la opción que se ha seleccionado en la pantalla anterior para resolver el test. Si la respuesta fue correcta, se mostrará la pantalla de color verde. En caso contrario, aparecerá de color rojo. También se muestran datos de la puntuación que se ha obtenido en esta respuesta y un feedback dinámico dependiendo de si ha contestado correctamente o incorrectamente, felicitando por el buen resultado o indicando cual era la opción correcta, para que el alumno pueda autocorregir los conceptos erróneos que pueda tener.

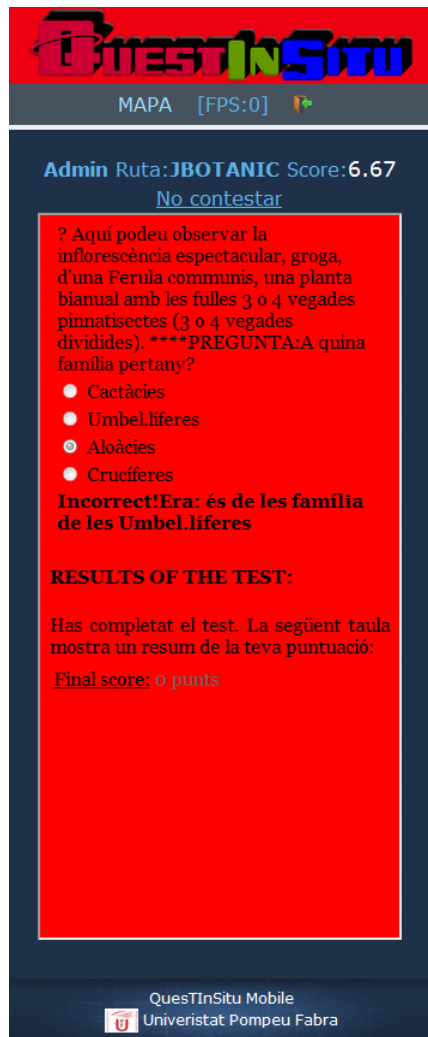


Figura 3.4.2.5.A Enunciado pregunta feeback en QTISM

3.4.2.5.1 Novedades

Esta nueva versión incorpora nuevas mejoras de diseño, por ejemplo permite el uso de nuevos caracteres como los acentos abiertos o incorporar imágenes, tanto en la formulación de preguntas, como en los resultados. Ver en la imagen Figura 3.4.2.5.A (Enunciado pregunta feeback en QTISM),

3.5 Actividades realizadas con QuesTInSitu en 2012

A continuación se explican dos actividades que se realizaron en dos escenarios distintos, por un lado una dentro de un escenario urbano como es la ciudad de Girona y, por otro, en un ambiente natural - aunque dentro de un núcleo urbano - como es el del Jardín Botánico de Barcelona.

Estas experiencias - tanto la de la ciudad de Girona como la del Jardín de Botánico de Barcelona - han sido diseñadas y lideradas por investigadores del grupo de investigación GTI de la Universidad Pompeu Fabra y se han realizado con QTIS dentro del marco del proyecto EEE: Espacios Educativos Especulares [12]. Este tipo de evaluación no entra dentro del trabajo del PFC, ya que no se incluyen ni formularios ni resultados que se utilizaron y, por tanto, la evaluación de estas actividades no se incluye en este trabajo final de carrera, sino que dentro del marco del proyecto se resumen para demostrar que QTIS se ha utilizado en contextos reales, como los que a continuación se presentan.

3.5.1 Actividad de Girona

El 23 de marzo de 2012, sesenta estudiantes de segundo curso de Bachillerato de la escuela Pía Santa Anna de Mataró, realizaron una ruta de Historia del Arte in situ guiados por teléfonos móviles utilizando la herramienta QuesTInSitu. El profesor de la asignatura diseñó una parrilla de setenta preguntas relacionadas con Historia del Arte de la ciudad de Girona y estas se introdujeron dentro del aplicativo QTIS para que este gestionara la parte de cuestionarios y se mostrara en la aplicación para dispositivos móviles QTISM.

Los alumnos de la escuela, como actividad de la asignatura de Historia del Arte, se dirigieron hacia la capital del Gironès (Provincia de Girona) para realizar la actividad diseñada. Allí se dividieron en grupos y recibieron instrucciones para poder configurar sus propios dispositivos móviles para utilizarlos durante la actividad. Una vez configurados, los teléfonos se convirtieron en su guía principal a lo largo de la yincana geolocalizada. En la siguiente figura (*Figura 3.5.1.A Alumnos en la actividad de Girona haciendo uso de la aplicación*) se pueden ver alumnos del instituto escuela Pía Santa Anna de Mataró haciendo uso de la aplicación.



Figura 3.5.1.A Alumnos en la actividad de Girona haciendo uso de la aplicación

Tras superar algunos pequeños problemas de conexión debido a la cobertura, se inició la actividad (ver imagen *Figura 3.5.1.B Inicio actividad*). Las cuestiones preparadas por el profesor de la escuela con antelación fueron apareciendo en la pantalla del móvil de los estudiantes cuando estos se encontraban en el lugar adecuado.

Una vez finalizada la visita, la valoración de la experiencia fue muy favorable, ya que, según la propia opinión de los estudiantes, "nos ha permitido conocer un poco mejor la ciudad de Girona, repasar la materia estudiada en clase, acercarnos a las nuevas tecnologías y, por supuesto, pasar un buen rato con los compañeros" [10]. Los alumnos también afirmaron que respondieron cuestiones "que tenían que ver con las curiosidades de la ciudad y, por supuesto, en relación con lo que ya habíamos estudiado en clase".



Figura 3.5.1.B Inicio actividad

Con esta actividad, también se puso a prueba el **nuevo sistema de monitorización**. Gracias a las nuevas mejoras implementadas, en esta nueva versión el sistema de monitorización es más intuitivo, puesto que sitúa a los usuarios en el mapa en posición en vivo con una mayor precisión que antiguamente. Durante la actividad el maestro dio libertad a los alumnos para que se movieran por la ciudad mientras él creó algo parecido a un puesto de mando donde podía ver las zonas por donde iban los alumnos y que puntuación iban sacando en las diferentes cuestiones (ver imagen *Figura 3.5.1.C Maestro monitorizando la actividad*).



Figura 3.5.1.C Maestro monitorizando la actividad

A través de la aplicación QuesTInSitu, los alumnos pudieron **ir contestando las preguntas geolocalizadas usando un teléfono móvil a medida que iban haciendo una ruta determinada**. Con ello, QuesTInSitu demuestra ser una aplicación educativa innovadora diseñada para apoyar la creación y la realización de rutas educativas interactivas para actividades de test in situ, con un gran potencial de aplicaciones.



Figura 3.5.1.D Reporteros TV3 realizando entrevistas

Durante el transcurso de la actividad, dos reporteros de la televisión pública catalana de TV3 (ver imagen *Figura 3.5.1.D Reporteros TV3 realizando entrevistas*) [11] estuvieron filmando las diferentes partes de la actividad anteriormente explicada y entrevistaron a los alumnos en su aventura, preguntándoles que es lo que hacían y de que manera. Posteriormente el documento gráfico se publicó como noticia de la ciudad en sus respectivos telediarios de la jornada.

Para finalizar la actividad se pasó un cuestionario, tanto al profesor como a los alumnos, para que calificaran su valoración de la actividad. Este cuestionario dio como resultado que la utilización de QTISM con las mejoras aportadas les parecía una mejor manera de hacer la excursión y ver el casco antiguo de la ciudad de Girona, tanto para ellos, como actividad y por la forma de evaluación.

3.5.1.1 Conclusiones de la actividad de Girona

Gracias a esta actividad se pudieron analizar y poner en práctica las nuevas funcionalidades de la aplicación, pero sobretodo se pudieron analizar tres de gran importancia, que son: el poder contestar preguntas de forma no secuencial, la persistencia de datos mediante XML y la monitorización con el profesor. Por lo que respecta a la primera, se pudo comprobar como los alumnos podían escoger a que pregunta dirigirse sin tener que seguir un orden preestablecido, lo que les beneficiaba puesto que iban a una zona que conocían o creían que podían sacar mejores resultados, dejándose las preguntas más complejas para el final de la actividad. Por lo que respecta a la persistencia de datos en XML, esto fue útil cuando el profesor, una vez finalizada la actividad, nos solicitaba los resultados que habían realizado los estudiantes para poder calificarlos en su asignatura. Por último también se evaluó la monitorización con el profesor, aunque no era la primera vez que se evaluaba esta parte del proyecto, dedicamos más atención esta vez acompañando al profesor durante más tiempo durante el transcurso de la actividad. De esta manera pudimos tomar dedicar más tiempo a evaluar esta nueva faceta y poder tomar notas que con la actividad de Girona no tuvimos tanto tiempo a poder hacer. Gracias a este método, se pudo generar un resumen persistente y facilitarle de mejor manera dichos datos al profesor.

También, como se indica en el resumen, se detectaron problemas de conectividad provocados por dos problemas que se detectaron durante el transcurso de esta. Por un lado se detectó una

primera causa, que fue la cantidad de usuarios que hacían que la actividad presentara problemas de concurrencia y de no poder satisfacer el servicio adecuado y por otro lado, las características y la estructura del casco antiguo de la ciudad de Girona hacia que la conexión disminuyera o se crearan puntos negros que imposibilitaban su uso.

3.5.2 Actividad Jardín Botánico de Barcelona

El 4 de mayo de 2012, dos meses más tarde de la primera actividad, se orquesta una segunda actividad in situ. Esta vez la actividad no se enmarca en el ámbito de una asignatura de Historia de bachillerato, sino de Botánica para los alumnos de primer curso de Biología de la Universidad Pompeu Fabra (ver Imagen *Figura 3.5.2.A Inicio actividad jardín botánico*) [6].



Figura 3.5.2.A Inicio actividad jardín botánico

Para esta ocasión, el tamaño del grupo también es semejante al de la actividad en Girona, con un total de sesenta estudiantes. Dados por los resultados de la anterior actividad, la actual se encaró de manera distinta y, con la ayuda de la maestra de la asignatura, se dividió el grupo en dos partes diferentes.

La actividad es de semejantes características que la actividad de Girona y se compone de tres fases:

- **Fase Inicial - Configuración:** Fase en que todos los alumnos se reúnen para configurar sus dispositivos móviles y estar listos para realizar la actividad. Para ello se les facilitó un manual para que ellos mismos pudieran realizar tal tarea sin necesidad de un monitor.
- **Fase de Trabajo - Buscar plantas:** Esta fase empieza una vez formados los distintos grupos de trabajo en grupos de 5 o 6 alumnos, donde cada persona realizaba un rol diferente con una tarea específica, aunque todos tuvieron que ayudarse entre ellos. Los diferentes roles se repartían entre el portador del dispositivo móvil, el portador de un mapa sin marcar de la zona del jardín, un fotógrafo para fotografiar las fotos para su posterior subida a la aplicación y una persona encargada de buscar información sobre este tipo de planta para poder contestar correctamente la pregunta, sea tomando como referencia sus apuntes de clase como en la web (ver Imagen *Figura 3.5.2.B Grupo estudiantes realizando actividad*).



Imagen Figura 3.5.2.B Grupo estudiantes realizando actividad

- **Fase de Evaluación:** Una vez los alumnos consiguen encontrar la mayoría o la totalidad de las plantas, se les pasa un formulario para evaluar el proyecto. En este formulario se les pregunta que les ha parecido la aplicación y que ventajas e inconvenientes han tenido. En este caso, la actividad obtuvo mejores cualificaciones que en la anterior de la ciudad de Girona.

3.5.2.1 Conclusiones de la actividad Jardín Botánico de Barcelona

En esta actividad, la cual en tema practico es muy similar a la realizada en la ciudad de Girona, se obtuvieron mejores resultados en la evaluación, como ya se indica en el apartado anterior. Esto fue así puesto que ya se preveían los fallos anteriormente explicados en las conclusiones de la actividad en la ciudad de Girona en el apartado 3.5.1.1.

En el anterior caso, los dos mayores problemas detectados fueron de cobertura y concurrencia de usuarios, los cuales también sucedieron en este caso, pero no tuvieron tanta repercusión. Esto fue así gracias a que se planificó la actividad de forma diferente, agrupando los usuarios en grupos para reducir el número de dispositivos operando simultáneamente, como también por la situación del jardín botánico, el cual se ubica dentro de la ciudad de Barcelona donde la conexión 3G es de mayor calidad.

4. Conclusiones

Como conclusión del trabajo en este apartado comentaré que ha significado para mí la realización de este proyecto y cuáles son las conclusiones, experiencias, resultados y beneficios obtenidos en la realización del trabajo.

Este proyecto, que he realizado durante la última etapa de mis estudios, me sirve como ejemplo y banco de pruebas para los futuros trabajos que pueda realizar como ingeniero una vez finalizada mi etapa de estudiante. Además, este es un gran paso para poder introducir al mundo de la enseñanza en el campo de las TIC y de la orquestación de actividades en entornos abiertos.

Durante la realización de este proyecto he aprendido lo difícil que puede llegar a ser trabajar a partir de un trabajo ya iniciado. A pesar que ya conocía el proyecto, el hecho de retomar partes de código para ampliarlas o retocarlas siempre requiere un gran coste de trabajo y documentación. Gracias a ello, ahora valoro más las convenciones que se determinan al trabajar en grupo, como también las metodologías usadas para la organización de proyectos. Gracias a ello se facilita que los equipos de trabajo puedan progresar con código que no es propio y evitar cometer los mismos errores que anteriores programadores hicieron, lo que se replica en un coste menor en la elaboración.

El objetivo del proyecto era ampliar y ofrecer nuevas funcionalidades a la herramienta QuesTInSitu, sobretodo en la parte que corresponde a su versión para dispositivos móviles.

Extensión de QTIS y QTISM a partir de proyectos anteriores, principalmente:

- Mejora de gestión y ejecución de rutas.
- Mejora de monitorización de información.
- Encapsulación de datos.
- Las nuevas funcionalidades han sido utilizadas en entornos educativos reales, permitiendo la creación de 2 rutas educativas (Girona y Jardín Botánico).

La nueva herramienta ofrece nuevas funcionalidades, sobretodo en la parte que corresponde a su versión para dispositivos móviles. Si miramos la herramienta desde el punto de vista técnico, la nueva herramienta se convierte en un punto de encuentro de elementos, la cual está suficientemente fraccionada en módulos, para así prácticamente no tener que modificar su estructura fundamental en futuros trabajos. De hecho, se ha conservado toda la independencia e invariabilidad de funciones, como también la estructura de todos los elementos para facilitar la incorporación de nuevos paquetes o readaptaciones a futuras nuevas versiones de herramientas o posibles cambios. Este factor ha sido fundamental para conseguir los objetivos planteados del proyecto, y ha sido el punto que me ha demostrado lo importante que es tener una buena base y escalabilidad para proseguir la tarea con futuros retos.

Creo conveniente destacar que este proyecto me ha servido para dar más importancia a la interfaz del usuario. En esta nueva versión se ha rediseñado como facilitar los datos al usuario, logrando una presentación más entendible y una operativa de la herramienta por partes de estos más fluida. También se ha mejorado en como interpretar y trasladar la información que nos da un GPS a un programa. Para todo ello me ha sido necesario entender como tener reflejar los

datos en un fichero para luego poder *renderizarlos* y, finalmente, mostrarlos y trabajar con ellos, ya sea en una monitorización o durante el transcurso de la actividad.

Con la creación de esta herramienta se pretende dar soporte al aprendizaje *in situ* para las diferentes asignaturas que se imparten en los centros docentes, dando una experiencia mejorada gracias a los nuevos menús y los sistemas de monitorización como también la herramienta móvil. La nueva versión ofrece la posibilidad de poder realizar actividades orquestadas a pequeña escala, como se pretendía con este proyecto.

Finalmente, con este proyecto se ha pretendido seguir el trabajo realizado en un proyecto anterior al cual se le han añadido nuevas metas. Con ello se ha pretendido acercar un poco más la tecnología al mundo educativo

5. Trabajo futuro

La creación de este trabajo abre futuras puertas para la realización de otros relacionados con este proyecto final de carrera, como también abre un nuevo mundo dentro de las nuevas tecnologías de la información, ya que se puede hablar de un nuevo concepto llamado enseñanza geolocalizada en tiempo real. Este proyecto es pionero en la realización de espacios educativos abiertos mediante la posición de los elementos que intervienen durante el método docente. La evaluación se realiza de forma interactiva y a tiempo real, ya que según su interacción y movimientos en el espacio se pueden realizar sistemas evaluadores, como la herramienta que defiende este proyecto.

En este apartado se menciona las diferentes posibilidades de trabajo que se abren y pueden ser razón de futuros proyectos a partir del trabajo ya realizado. Se han dividido estas posibilidades en dos casos: Un apartado de mejoras generales y posibilidades que se pueden incorporar a la herramienta y, por otro lado, mejoras técnicas y revisión de partes del proyecto ya realizado.

5.1 Futuras mejoras generales para la aplicación

Como se ha mencionado anteriormente, en este apartado se mostraran posibilidades y futuros proyectos que se podrían realizar a partir del trabajo ya realizado.

5.1.1 Organizar las preguntas por temáticas

Actualmente el editor de preguntas de la aplicación Web posee un campo para poder clasificar las preguntas según su temática, como por ejemplo: preguntas de historia, de ciencias naturales, etc. Sin embargo, este campo todavía no está operativo y sería muy positivo para los profesores ya que podrían filtrar preguntas de una zona y encontrar preguntas relacionadas con el ámbito de su asignatura y reutilizar en sus rutas o como soporte para confeccionar las suyas propias.

5.1.2 Nuevos modelos de preguntas

En la actualidad la herramienta ofrece tres tipos básicos de preguntas: *Multiple Choice*, *Multiple Response* y *True / False*. Las interacciones en estos tres tipos de preguntas quedan muy delimitadas, cosa que reduce las posibilidades de la herramienta. Una mejora sería la incorporación de nuevos modelos de evaluación como preguntas abiertas, uso de grabaciones, incorporación de archivos media en los enunciados, etc....

5.1.3 Mejoras de diseño y gestión de datos de las aplicaciones

El uso que realiza la aplicación de los datos se podría mejorar con la incorporación de más funcionalidades 2.0, generación de datos estadísticos, elaborar un diseño nuevo y superior a la plataforma actual. Una solución para ambos casos (la mejora del diseño y de los datos) sería la adaptación de los ficheros de los proyectos web a plataformas como CMS (*Content Management System*) y así actualizar las presentaciones a diseños más actuales en *WordPress*.

Además se aplicaría la gestión de datos de una manera más eficaz con paquetes preparados que ofrecen este tipo de plataformas.

5.1.4 Nuevas propuestas de monitorización

El sistema de monitorización que dispone el profesor para poder hacer el seguimiento del transcurso de la actividad podría complementarse con nuevas funcionalidades. Una de estas sería la opción de poder recibir avisos de sus alumnos in situ, mediante mensajes de alerta. Otra sería aprovechar las cámaras de los dispositivos móviles para poder hacer videoconferencia y poder hacer un mejor seguimiento, como la resolución de dudas que puedan surgir durante el evento.

5.1.5 Editor de preguntas en la aplicación móvil

Actualmente sólo se pueden crear preguntas mediante la aplicación web. Aunque esta es accesible mediante soporte 3G en dispositivos móviles, no está optimizada para su visualización en estos dispositivos. Sería una buena mejora poder añadir la funcionalidad de poder crear preguntas mediante la aplicación móvil, ya que durante la confección de los eventos pudimos ver que a veces es necesario realizar un trabajo previo en el lugar de las rutas para confeccionarlas. Nuestro software debería de poder dar soporte a esta necesidad incorporando funcionalidades en la aplicación móvil, como la toma de notas, la obtención del punto exacto de la pregunta, medir duración del recorrido a pie, etc....

5.1.6 Incorporación de nuevas tecnologías

Otro aspecto futuro, si nos focalizamos en el aspecto de añadir nuevas tecnologías, sería incorporar sistemas IPS (*Indoor Positioning System*) a nuestro sistema de geoposicionamiento, para mejorar el rendimiento de la aplicación en espacios cerrados como centros comerciales. Con este tipo de tecnología, los dispositivos móviles pueden nutrirse de soporte de red en espacios con poca cobertura (como pueden ser espacios cerrados o callejones) lo que solucionaría los problemas de cobertura actuales.

5.1.7 Incorporación de actividades de realidad aumentada

Un futuro trabajo a realizar referido a realidad aumentada podría ser jugar con la cámara del dispositivo móvil mostrando objetos o leyendas para complementar la información del entorno (ver Imagen *Figura 5.1.7.A Realidad aumentada con cámara de dispositivo móvil [23]*).



Figura 5.1.7.A Realidad aumentada con cámara de dispositivo móvil de Layar[23]

Otra propuesta sería combinar el espacio físico con espacios virtuales 3D de realidad aumentada. Mediante la posición e interacción de los estudiantes que están haciendo uso de la herramienta se tendría que generar un espacio 3D virtual, como por ejemplo un personaje virtual que apareciera en lugares concretos como guía de la ruta. Un ejemplo de personaje virtual didáctico podría ser el hombre del tiempo Sam desarrollado por la UPF (ver Imagen *Figura 5.1.7.B Personaje virtual Sam, proyecto de la UPF*).



Figura 5.1.7.B Personaje virtual Sam, proyecto de la UPF

5.1.8 Creación de aplicaciones nativas para su mejor distribución

Hace falta estudiar diferentes formas para poder hacer accesible la herramienta y que esta pueda llegar a los usuarios finales. Para ello sería bueno adaptarla para su distribución como software para su distribución en redes sociales o en congresos de ámbito escolar. Otra manera de mejorar su distribución será la posibilidad de poder encapsular nuestra aplicación en cada una de las diferentes aplicaciones nativas para dispositivos móviles que existen en la actualidad, como Android (Java) o Apple (Xcode). Esto nos permitiría poder incluir la herramienta en las diferentes plataformas de distribución de software que poseen estas compañías - como por ejemplo la tienda virtual *Apple Store* (<http://store.apple.com/es>) [14] - y así poder ofrecer la herramienta a un gran número de usuarios potenciales.

La herramienta QuesTInSitu está muy focalizada en el mundo educativa y, más concretamente, en la evaluación. Si nos fijamos en los elementos que utiliza observamos que podría usarse para otros motivos no educativos, como en el comercio. Un ejemplo de esta variante de uso, sería la de un posible cliente que al acercarse a un centro comercial le aparezcan interacciones relacionadas con las ofertas de los productos que allí se venden.

5.2 Futuras mejoras técnicas para la aplicación

Este apartado se centra en detallar problemas detectados de origen técnico:

- **Mal funcionamiento en ambientes con poca cobertura:** En sitios donde la intensidad de señal no es adecuada la aplicación resulta lenta y más propicia a tener errores de sincronización entre las diferentes clases de código. Este fenómeno se produce ya que una clase espera un valor. Para ello hay que evitar zonas alejadas o avenidas estrechas donde la señal normalmente es más débil.
- **Errores de concurrencia:** La aplicación no es capaz de dar servicio cuando la cantidad de usuarios que la están utilizando la aplicación es superior a 8.
- **Tiempo excesivo de carga de mapas aplicación web:** En la aplicación Web, los mapas dedicados a dar servicios para responder preguntas mediante la web tienen un tiempo de carga excesivo (mayor a 30 segundos) y que se incrementa a mayor número de preguntas contenga el mapa.
- **Depurar código y complejidad de cálculo de algoritmos:** Los algoritmos internos para mostrar los valores de mapas son muy independientes y desarrollados con muchos bucles que se recorren completamente. Se tendrían que buscar nuevas formas de cálculo para optimizar esta parte de código y cargar sólo las partes del mapa útiles, obviando aquellas que no interesan.
- **Aplicar métodos *synchronize*:** Aplicar con mayor rigor métodos *synchronize* que evitan que se ejecute código si antes no se ha esperado la respuesta de código ejecutado anteriormente. Con estos métodos se pueden paliar errores de concurrencia ya que se aplican semáforos para esperar de manera que el código siguiente tiene tiempo para rellenar los valores de sus variables y no quedar bloqueado.
- **Revisar rendimiento del servidor:** Buscar software de rendimiento para evaluar si las características que ofrece el servidor son suficientes para las expectativas y requerimientos que deseamos para el software y así poder ofrecer un servicio adecuado y no lento.
- **Crear sistemas baterías de usuarios para pruebas:** Crear un gran número de usuarios ficticios que ataquen nuestro software para poder evaluar el límite al que puede llegar nuestro software y que niveles de servicio ofrece en cada valor.
- **Creación de automatismos:** Actualmente gran parte de la operativa requiere de actualizar los datos mediante un administrador. A pesar que se ha mejorado este aspecto con botones de procesos automáticos, aún queda mucho camino para mejorar.

6. Bibliografía

- [1] Santos, P., Pérez-Sanagustín, M., Hernández-Leo, D., & Blat, J. (2011). QuesTInSitu: from tests to routes for assessment in situ activities. *Computers & Education* 57(4): 2517-2534, Encontrado 06/01, 2013, en http://www.dtic.upf.edu/~daviniah/santos_etal-QuesTInSitu-C&E_camera-ready.pdf
- [2] Garcia, F. (2012) Questinsitu mobile: Aplicació per a rutes de preguntes geolocalitzades. Projecte final de carrera, Universitat Pompeu Fabra. Encontrado 11/01, 2012, en <http://hdl.handle.net/2072/171828>.
- [3] Pérez Calle, D. (2010), QuesTInSitu: Aplicació per crear preguntes QTI geolocalitzades. Projecte final de carrera, Universitat Pompeu Fabra. Encontrado 06/01, 2013, en [http://catalag.upf.edu/search~\\$11*spl?/b1005949435/b1005949435/-3%2C-1%2C0%2CB/frameset&FF=b1005949435&1%2C1%2C](http://catalag.upf.edu/search~$11*spl?/b1005949435/b1005949435/-3%2C-1%2C0%2CB/frameset&FF=b1005949435&1%2C1%2C).
- [4] OpenProj, Project Management (2012). Encontrado 06/01, 2013, en <http://sourceforge.net/projects/openproj/>
- [5] Santos, P., Hernández-Leo, D., Blat, J., To be or not to be in situ outdoors, and other implications for the design and implementation, in geolocated mobile learning, *Pervasive and Mobile Computing*, (under review)
- [6] Santos, P., Pérez-Sanagustín, M., Hernández-Leo, D., & Blat, J. (2012). Routes of Geolocated Questions in Formal and Informal Learning Contexts. *Bulletin of the IEEE Technical Committee on Learning Technology*, 14(4), 21.
- [7] Traxler, J. (2007). Defining, Discussing and Evaluating Mobile Learning: The moving finger writes and having writ... *The International Review of Research in Open and Distance Learning*, 8(2). Encontrado 11/01, 2012, en <http://www.irrodl.org/index.php/irrodl/article/view/346/875%2Bdiscussing%26amp%3Bsa%3DX>
- [8] Manjón Fernández, B. (2006). Uso de estándares aplicados a TIC en Educación (Informe 16). *Centro Nacional de Información y Comunicación Educativa (CNICE-MEC), Ministerio de educación y ciencia*. Encontrado 06/01, 2013, en <http://ares.cnice.mec.es/informes/16/contenido/indice.htm>
- [9] IMS. (2006). *IMS question and test interoperability assessment test, section, and item information model*. Disponible en: http://www.imsglobal.org/question/qtiv2p1pd2/imsqti_infov2p1pd2.html. Encontrado 05/12, 2010.
- [10] Trenchs, M. (2012). Visita ciutat de Girona. Encontrado 11/01, 2012, en <http://storify.com/ismaelcabezudo/ruta-geolocalitzada-d-historia-de-l-art-a-girona>
- [11] TV3 (2012). Visita ciutat de Girona. Encontrado 11/01, 2012, minuto 16' 55" en <http://www.tv3.cat/videos/4015290/Telenoticies-Barcelona-23032012>
- [12] Learn3 (2011). Universidad Carlos III de Madrid, Universitat Pompeu Fabra de Barcelona. *Spanish Ministry of Science and Innovation, TIN2008-05163*. Acceso en este link; <http://gti.upf.edu/learn3/>.

- [13] IMS QTI, IMS Global Consortium (2012), *Wikipedia*. Encontrado 06/01, 2013, en http://en.wikipedia.org/wiki/IMS_Global.
- [14] Apple (2012). *Apple Store*. Encontrado 06/01, 2013, en <http://store.apple.com/es>
- [15] CAA (2012). *Computer Asisted Assessment Center*. Encontrado 06/01, 2013, en <http://www.caacentre.ac.uk/>
- [16] Life 2.0 project (2012). *EU funded project*. Encontrado 06/01, 2013, en <http://www.life2project.eu/>
- [17] Api Google Maps (2012). *Google Company*. Encontrado 06/01, 2013, en <http://developers.google.com/maps/?hl=es>
- [18] OGC (2012), *OGC foundation*. Encontrado 06/01, 2013, en <http://www.opengeospatial.org>
- [19] eduMac (2012). *eduMac Centre d'arts Digitals*. Encontrado 05/22, 2010, en <http://edumac.com.mx/blog/539/mapa-de-bits-vs-vectores>
- [20] Fernández, L., & Hernández, D. (2005). Plataforma para servicios de valor agregado basados en localización, en una red gsm, a partir de la medición de la intensidad de señal (Parte I). *Revista de la Facultad de Ingeniería Universidad Central de Venezuela*, 20(3), 1-15. Encontrado 11/01, 2012, en http://www.scielo.org.ve/scielo.php?pid=S0798-40652005000300001&script=sci_arttext
- [21] World Wide Web Consortium (W3C), Encontrado 06/01, 2013, en <http://www.w3c.es/>
- [22] IPS (2012). *Indoor Positioning System*. Encontrado 06/01, 2013, en <http://tecnologiadospuntocero.wordpress.com/2011/04/30/ips-indoor-positioning-system/>
- [23] Layar (2012). *Layar Foundation*. Encontrado 06/01, 2013, en <http://www.layar.com/>

NOTA: Todos los enlaces han sido revisados por última vez en Julio de 2013.



UNIVERSITAT
POMPEU FABRA

ESCOLA SUPERIOR POLITÈCNICA
ESTUDIS D'INFORMÀTICA

A
N
N
E
X

Projecte Fi de Carrera

QuesTInSitu: mejora del sistema de autoría y móvil para la creación, realización y monitorización de rutas abiertas de preguntas geolocalizadas

David Pérez Calle

Curs 2012-2013

ENGINYERIA
EN INFORMÀTICA

Director:
Patricia Santos Rodríguez

7. Anexos

7.1 Manual de Instalación

Para la instalación de todo el software de QTIS y QTISM, se ha realizado una guía con unos sencillos pasos a realizar. Describiendo los requisitos y pasos a realizar para su correcta instalación y posterior uso.

7.1.1 Manual de Instalación de QTIS

Pasos a realizar para instalar la aplicación QTIS.

7.1.2 Requisitos para la instalación de QTIS

A continuación se detallan un listado de requisitos que hay que tener para poder iniciar la instalación y lanzamiento de la aplicación QTIS.

- Tener un servidor Web (Recomendado una máquina virtual Microsoft Server)
- Tener un servidor para albergar apache TOMCAT instalado
- Tener un servidor MySQL
- Archivo QTIS.jar
- *Scripts* de carga base de datos NewApis
- *Scripts* de carga base de datos QTIS

7.1.3 Pasos a seguir para la instalación de QTIS

Para empezar con la instalación primero tendremos que crear instancias de las diferentes bases de datos en un servidor *MySQL*. Este servidor se encontrará dentro de nuestro servidor instalado. Para poder crear una instancia de la base de datos, se usa desde la consola de *MySQL* el comando *source* de la siguiente manera, siendo “*buil.sql*” el *script* que lanza toda la base de datos y el resto de *Scripts* de soporte de QTIS (ver imagen *Figura 7.1.3.A Línea de comandos MySQL*).

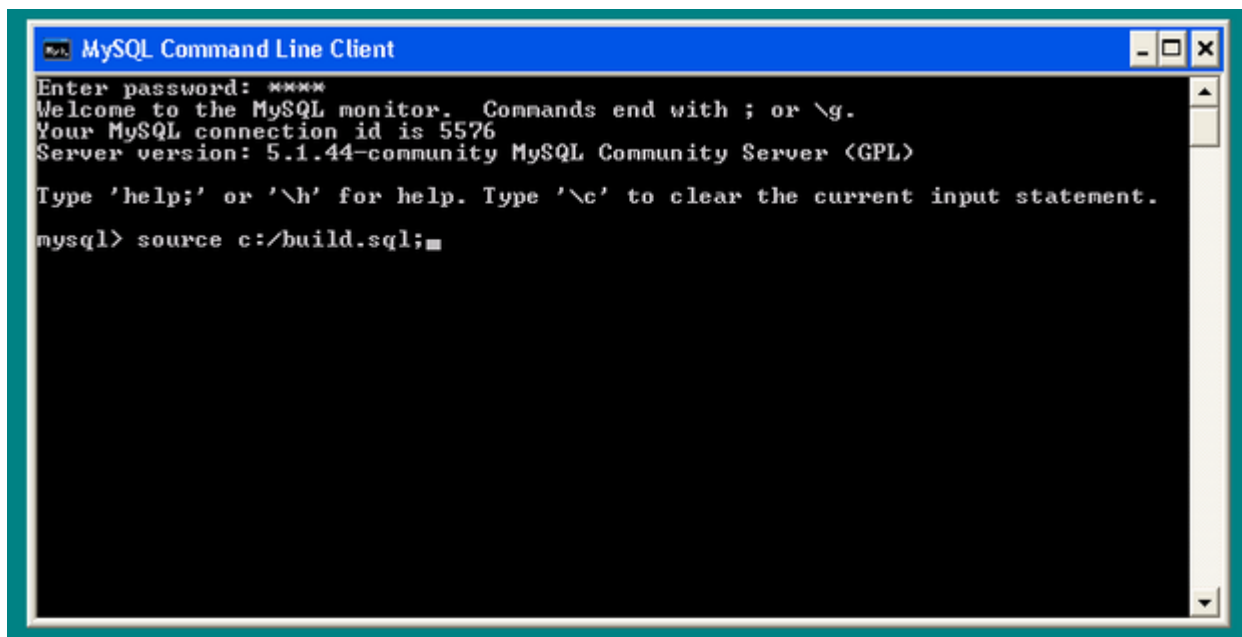


Figura 7.1.3.A Línea de comandos MySQL

Para realizar la instancia de la base de datos del motor de NewApis, se realiza la misma operación pero en este caso con el *script* “apis_db.sql”.

Posteriormente hay que añadir el proyecto web en nuestro servidor web. Para ello necesitaremos tener previamente instalado Apache Tomcat en nuestra maquina. Una vez tenemos este punto realizado, deberemos acceder al gestor de proyectos que nos facilita Apache Tomcat. Para ello, abriremos el navegador web del servidor e introduciremos esta URL: ip_servidor:puerto_servidor/manager/html (Por ejemplo, si la ip del servidor y el puerto 8080, escribiremos ip_servidor:8080/manager/html). Posteriormente introducir el usuario y contraseña para poder acceder al gestor de aplicaciones si es necesario (ver imagen *Figura 7.1.3.B Aplicación web gestión proyectos Apache Tomcat*).

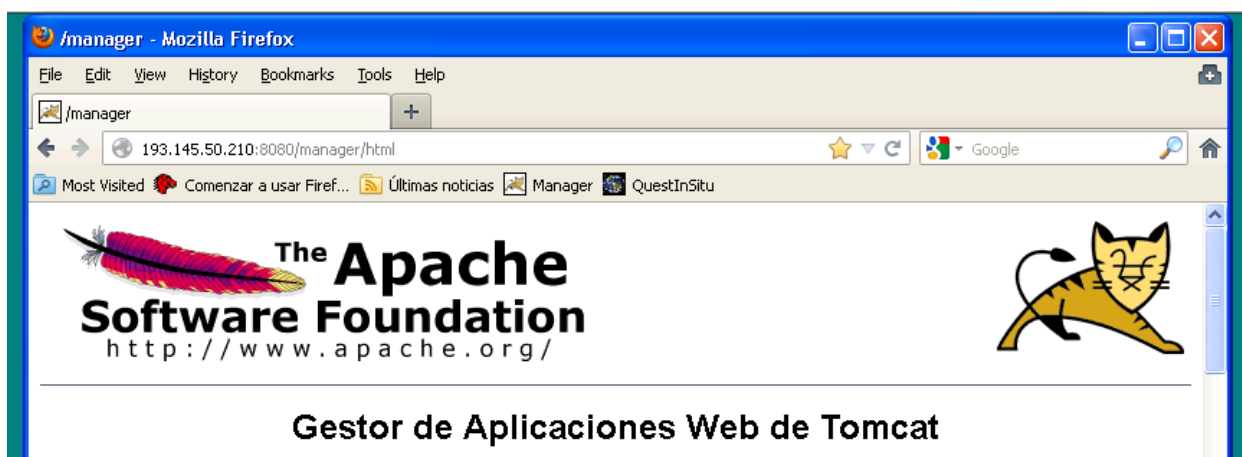


Figura 7.1.3.B Aplicación web gestión proyectos Apache Tomcat

Una vez en el gestor de aplicaciones de Tomcat, tendremos que buscar la opción para poder desplegar nuestra aplicación y así poder publicarla en la Web. Para ello, seleccionaremos la

opción de “Browse” y buscaremos nuestro archivo “QTIS.jar”. Posteriormente pulsaremos el botón de “Desplegar” (ver imagen *Figura 7.1.3.C Menú desplegar proyectos en Apache Tomcat*).

Desplegar	
Desplegar directorio o archivo WAR localizado en servidor	
Trayectoria de Contexto (opcional):	<input type="text"/>
URL de archivo de Configuración XML:	<input type="text"/>
URL de WAR o Directorio:	<input type="text"/>
	<input type="button" value="Desplegar"/>
Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="text"/> <input type="button" value="Browse..."/>
	<input type="button" value="Desplegar"/>

Figura 7.1.3.C Menú desplegar proyectos en Apache Tomcat

Una vez desplegado el proyecto QTIS, hay que tener en cuenta que dentro del proyecto ya está contenido también el motor de NewApis y funciones útiles para la aplicación QTISM. Para finalizar abrir el navegador con la dirección web.

7.2 Manual de Instalación de QTISM

Pasos a realizar para instalar la aplicación QTISM.

7.2.1 Requisitos para la instalación de QTISM

A continuación se detallan un listado de requisitos que hay que tener para poder iniciar la instalación y lanzamiento de la aplicación QTISM.

- Tener instalada la aplicación de QTIS y tener todos sus requisitos (Ver apartado 7.1.2 Requisitos para la instalación de QTIS y 7.1.3 Pasos a seguir para la instalación de QTIS)
- Archivo QTISM.jar

7.2.2 Pasos a seguir para la instalación de QTISM

Una vez instalada la aplicación QTIS, podremos instalar la aplicación de QTISM. Para ello necesitaremos tener previamente instalado Apache Tomcat en nuestra maquina como ya teníamos anteriormente en QTIS. Una vez tenemos este punto, deberemos de acceder al mismo servidor Apache Tomcat en el que tenemos instalado nuestra aplicación QTIS y allí crear la instancia la aplicación de QTISM (ver imagen *Figura 7.2.2.A Aplicación web gestión proyectos Apache Tomcat*).

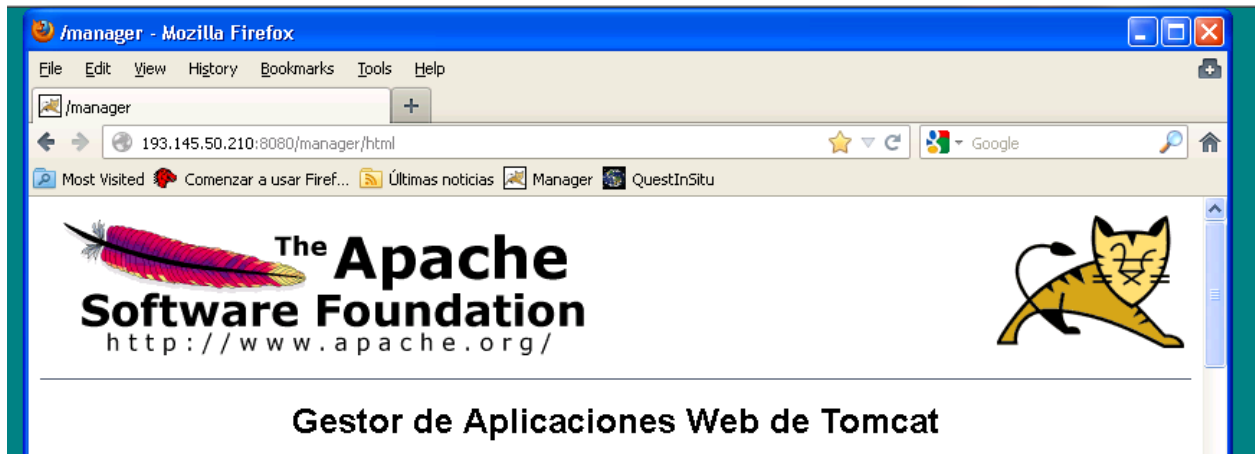


Figura 7.2.2.A Aplicación web gestión proyectos Apache Tomcat

Una vez en el gestor de aplicaciones de Tomcat, tendremos que buscar la opción para poder desplegar nuestra aplicación y así poder publicarla en la Web. Para ello, seleccionaremos la opción de “Browse” y buscaremos nuestro archivo “QTISM.jar” y pulsaremos posteriormente el botón de “Desplegar” (ver imagen *Figura 7.2.2.B Menú desplegar proyectos en Apache Tomcat*).

Desplegar	
Desplegar directorio o archivo WAR localizado en servidor	
Trayectoria de Contexto (opcional):	<input type="text"/>
URL de archivo de Configuración XML:	<input type="text"/>
URL de WAR o Directorio:	<input type="text"/>
	<input type="button" value="Desplegar"/>
Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="text"/> <input type="button" value="Browse..."/>
	<input type="button" value="Desplegar"/>

Figura 7.2.2.B Menú desplegar proyectos en Apache Tomcat

Hay que recordar que QTISM se enlaza a la base de datos de QTIS, por lo que no es necesario crear ninguna nueva instancia de base de datos ya que se conecta a la misma y obtiene los datos de la misma fuente. Para finalizar abrir el navegador con la dirección web desde un dispositivo móvil como *smartphone* o *tablet*.

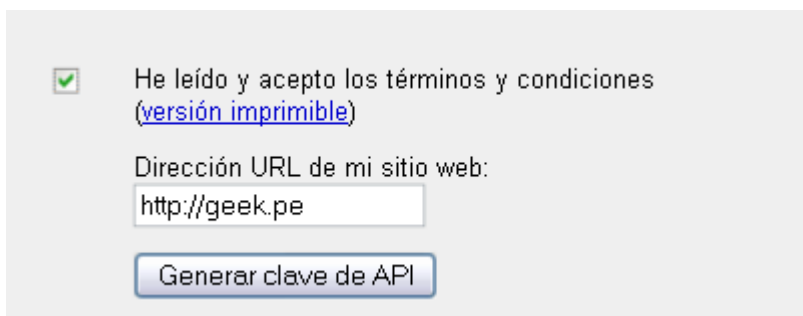
7.3 Manual para obtener una *Key* de *Google Maps*

Para obtener una *Key* de *Google Maps*, hay que registrarse en el siguiente enlace: “<https://developers.google.com/maps/documentation/javascript/?hl=es>”.

Antes de empezar la instalación, es conveniente revisar algunas de las condiciones que nos aplica Google para utilizar sus mapas, como por ejemplo:

- Hay un límite en el número de solicitudes de codificación geográfica por día, 15.000,
- No hay límite al número de páginas vistas se pueden generar por día.
- La publicidad (*AdSense* / *AdWords*) no está incluido en la API de *Google Maps*
- Se está obligado a ofrecer mapas de Google como un servicio gratuito para sus usuarios.

Una vez considerados los puntos anteriores, acepte los términos y condiciones. Ahora escriba el dominio real del sitio que utilizará *Google Maps* en el recuadro. Si dispone de más de un sitio, tendrá que obtener una clave de API para cada sitio. Haga clic en “Generar clave API” (ver imagen *Figura 7.3.A Botón generar claves para API Google Maps*).



He leído y acepto los términos y condiciones ([versión imprimible](#))

Dirección URL de mi sitio web:

Figura 7.3.A Botón generar claves para API Google Maps

La siguiente ventana se proporcionará una clave de API de Google. Esta clase puede ser utilizada para insertar mapas en sitios web, así como para una variedad de *Google Maps* relacionadas como puede ser nuestra aplicación. Recuerde que esta clave sólo es válida en el dominio que usted proporcionó (ver imagen *Figura 7.3.B Menú gestión claves de API Google Maps*).

Regístrate y obtén una clave de API

[Creación de un mapa del sitio KML](#)

API de Google Maps

[Página principal](#)

[Documentación](#)

API de Google Maps para Flash

[Página principal](#)

Registro en el API de Google Maps

Thank You for Signing Up for a Google Maps API Key!

Your key is:

ABQIAAAAWoMCiX9TavE6a9vp5qy0bBSkoT5jt5TTXycwXZWnEzwAUtGR

Note: for more information on the API key system, consult <http://code.google.com/apis/maps/faq.html#keysystem>.

How you use your key depends on what Maps API product or service

Figura 7.3.B Menú gestión claves de API Google Maps

Google también ofrece ejemplos de código para JavaScript y código Flash que muestran dónde insertar la clave de API para el mapa de Google. Para poder utilizar la *key* en nuestros mapas tendremos que introducir en nuestro *script* de la siguiente manera “key =”, como en el siguiente ejemplo substituyendo <Your_Key> por nuestra clave de acceso (ver imagen Figura 7.3.C Llama en código JavaScript de la key API Google Maps)

```
<script src="http://maps.google.com/maps?file=api&v=2&key=<Your_Key>&sensor=false" type="text/javascript"></script>
```

Figura 7.3.C Llamada en código JavaScript de la key API Google Maps