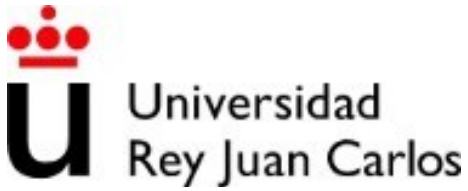

Escuela Técnica Superior de Ingeniería Informática

Ingeniería Técnica en Informática de Sistemas

Proyecto de Fin de Carrera

Aplicaciones Móviles para Herramientas Docentes Interactivas en el ámbito del Espacio Europeo de Educación Superior (EEES)

Autor: Alejandro López Gutiérrez

Tutora: Cristina Rodríguez Sánchez

Co-tutor: Ángel Torrado Carvajal

Curso Académico 2011/2012

Proyecto Fin de Carrera

APLICACIONES MÓVILES PARA HERRAMIENTAS DOCENTES INTERACTIVAS EN EL ÁMBITO DEL ESPACIO EUROPEO DE EDUCACIÓN SUPERIOR (EEES)

AUTOR

ALEJANDRO LÓPEZ GUTIÉRREZ

TUTORA

CRISTINA RODRÍGUEZ SÁNCHEZ

CO-TUTOR

ÁNGEL TORRADO CARVAJAL

La defensa del presente Proyecto Fin de Carrera se realizó el día de de , siendo calificada por el siguiente tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

y habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Móstoles, a de de .

Agradecimientos

Esta sección es especial, es el lugar elegido para acordarme de toda la gente que ha estado a mi lado durante el desarrollo de este proyecto sobre todo un enorme gracias a todos ellos.

Para empezar, como no a mi familia dándome caña, preguntándome día sí, día también como lo llevaba hasta el punto de agotarme pero aportándome la fuerza necesaria para avanzar sin pausa.

A mis tutores, Cristina y Ángel que me han sabido guiar desde un torpe e ignorante inicio hasta un "proyecto completísimo", palabras textuales de ambos.

A los compañeros del departamento, a los que invadía sus mesas de vez en cuando para que me echaran un cable y porque siempre que pudieron lo hicieron casi dejando a un lado su propio trabajo y aportando esa pizca de experiencia necesaria a mis ideas o aportando las suyas. En especial a Miguel Ángel, que fue mi contacto para conseguirme este proyecto cuando estaba perdido y el que me introdujo en este mundillo.

A los niños a los que entrené durante este periodo que perdonaron mis ausencias cuando este proyecto necesitaba más de 24 horas diarias y a mis compañeros entrenadores que me substituyeron.

A mis amigos que aguantaron mis malas caras cuando avanzaba y que supieron "raptarme" y apartarme de la pantalla del ordenador en la medida justa.

Agradecer incluso a los grupos de música que ocuparon mis oídos mientras mis dedos daban forma a este proyecto al fin acabado.

A todos ellos, un enorme y sincerísimo GRACIAS.

Resumen

Las Tecnologías de la Información y la Comunicación están en continuo crecimiento a lo largo de los últimos años, sobre todo la parte relacionada con los dispositivos móviles que han permitido la aparición de un gran número de aplicaciones y servicios interactivos dirigidos a ellos.

La oportunidad de aplicar las TICs (tecnologías de la información y la comunicación) a los *Smartphones* y el cambio producido por la introducción del Plan Bolonia en el espacio de la educación superior motiva este Proyecto Fin de Carrera.

Este cambio del que hablamos del paradigma de la Educación Superior en el ámbito Europeo causado por la aplicación del Plan Bolonia provoca la instauración de la evaluación continua como medida más relevante del Plan, cambiando la educación tal y como la conocemos hoy en día y generando un espacio en el que poder integrar las nuevas tecnologías.

Además, durante los últimos años en el Departamento de Tecnología Electrónica de la Universidad Rey Juan Carlos se ha establecido una linea de investigación sobre electrónica industrial y aplicaciones inalámbricas en el que se enfoca este proyecto.

En este Proyecto Fin de Carrera, se pretende mejorar los mecanismos para aplicar la evaluación continua por parte del profesor facilitando y flexibilizando la distribución y generación de exámenes.

Para conseguir evaluar a todos sus alumnos de forma más rápida y cómoda, automatizar correcciones y gestionar las notas de todos los alumnos en los test realizados se ha creado una plataforma interactiva de evaluación continua. Esta plataforma consiste en un software gratuito del que dispone el profesor con una interfaz fácil e intuitiva que cualquier profesor sin conocimientos informáticos pueda usar con la que poder gestionar exámenes y notas. Además, crearemos una app para dispositivos móviles con los que los alumnos podrán responder a los exámenes generados por sus profesores. Todo esto apoyado en una base de datos que evite la volatilidad de los datos generados. Con esto, conseguiremos que el profesor agilice la gestión de sus exámenes (creación, envío a los alumnos y corrección automática) y que los alumnos realicen dichos exámenes a través de sus propios dispositivo móviles.

Una de las grades ventajas de estos dispositivos es la posibilidad de hacer de forma inalámbrica, por medio de Wi-Fi en este caso y en una línea de trabajo futuro dotar a esta aplicación de ubicuidad total y poder realizar todas estas funciones desde cualquier lugar del mundo a través de una conexión a internet.

¿Por qué esta magnífica tecnología científica, que ahorra trabajo

y nos hace la vida mas fácil, nos aporta tan poca felicidad?

La repuesta es ésta, simplemente:

porque aún no hemos aprendido a usarla con tino.

Albert Einstein

Índice general

Agradecimientos....I
Resumen....III
Índice general....VII
Índice de figuras....IX
Índice de tablas....XI
1. Introducción....1
1.1. Situación Actual del Mercado de Smartphones....1
1.2. Espacio Europeo de Educación Superior (EEES)....4
1.3. Proyectos y Aplicaciones TIC para el EEES....6
1.4. Descripción del Problema....8
2. Objetivo....11
2.1 Objetivos específicos....11
3. Requisitos....13
3.1. Metodología....14
3.1.1 Estudio de Alternativas y Planificación....15
3.1.2. Recursos....18
4. Materiales y métodos....19
4.1. Entorno de Desarrollo....19
4.1.1. MySQL Workbench....19
4.1.2. Eclipse....20
4.1.3. SDK de Android....21
4.1.4. NetBeans....22
4.1.5. PHP....23
4.2. Dispositivos....24
5. Desarrollo....27
5.1. Diseño e implementación....27
5.1.1 Base de datos....27
5.1.2 Mesa Interactiva del Profesor....33
5.1.3 MobiTest....36
5.1.4 Aplicación web asistente de BBDD....40

6. Resultados.....43
6.1 Guía de uso 47
6.1.1. Mesa Interactiva del Profesor.....47
6.1.2. MobiTest.....53
6.1.3 Aplicación web para la creación y borrado de usuarios.....55
7. Conclusiones y Líneas de Trabajo.....59
7.1 Conclusiones.....59
7.2 Líneas de Trabajo Futuras.....60
8. Bibliografía.....63
9. Apéndices.....65
Anexo 1: Metas y declaraciones del Plan Bolonia.....65
Anexo 2: Dispositivos usados en la integración.....67
Anexo 3: Instalación del SDK de Android:.....69
Anexo 4: Protocolo de comunicaciones.....73
Anexo 5: Ejemplo .xml.....75

Índice de figuras

Ilustración 1: Gráfica de ventas de Smartphones....3
Ilustración 2: Mapa de países participantes en el Plan Bolonia....4
Ilustración 3: Diagrama de casos de uso UML....13
Ilustración 4: Diagrama de Gant....17
Ilustración 5: Centro de Software Ubuntu....20
Ilustración 6: Ejemplo ejecutable Eclipse....21
Ilustración 7: HTC Desire....24
Ilustración 8: Sony Xperia Play....25
Ilustración 9: Samsung Galaxy S....25
Ilustración 10: Arquitectura del sistema completo....27
Ilustración 11: Representación E/R de la Base de Datos....28
Ilustración 12: Ejemplo de pregunta (primera versión)....38
Ilustración 13: Pantalla inicial (primera versión)....38
Ilustración 14: Gráfica de dispositivos en Ciencias Medioambientales....43
Ilustración 15: Gráfica de dispositivos Smartphones en Ing. Telecomunicaciones....44
Ilustración 16: Tiempo sin uso de MIP y MobiTest....45
Ilustración 17: Tiempo empleado con MIP y MobiTest....46
Ilustración 18: Pantalla de login de la MIP....47
Ilustración 19: Menú principal MIP....48
Ilustración 20: Nuevo test....48
Ilustración 21: Pantalla donde rellenar la pregunta....49
Ilustración 22: Ver tests....50
Ilustración 23: Iniciar cuestionario....51
Ilustración 24: Ver resultados....51
Ilustración 25: Ver aprobados....52
Ilustración 26: Ver respuestas....52
Ilustración 27: Ver notas....53
Ilustración 28: Pantalla inicial MobiTest....54
Ilustración 29: Ajustes MobiTest....54
Ilustración 30: Selección IP....54

Ilustración 31: Error 1 MobiTest.....54
Ilustración 32: Error 2 MobiTest.....54
Ilustración 33: Error 3 MobiTest.....54
Ilustración 34: Pregunta MobiTest.....55
Ilustración 35: Final MobiTest.....55
Ilustración 36: Pantalla inicial aplicación web.....56
Ilustración 37: Desplegable aplicación web.....56
Ilustración 38: Crear alumno.....57
Ilustración 39: Borrar alumno.....57
Ilustración 40: Añadir asignatura a un alumno.....58
Ilustración 41: .bashrc.....70
Ilustración 42: Android manager.....71
Ilustración 43: descarga de archivos.....71
Ilustración 44: descarga exitosa.....72
Ilustración 45: Protocolo de comunicaciones.....73

Índice de tablas

Tabla 1: Evolución de las ventas.....2
Tabla 2: Recursos materiales.....18
Tabla 3: Código SQL generado por SQL Workbench.....33

1. Introducción

Los avances producidos en las TICs (Tecnologías de la Información y de las Comunicaciones) a lo largo de los últimos años, sobre todo los relacionados con los dispositivos móviles, han provocado la creación de nuevos paradigmas, aplicaciones y servicios interactivos. De esta manera, podemos comprobar como esta evolución tecnológica está influyendo en muchos campos.

El modo de entender la enseñanza superior también ha cambiado con la introducción de los nuevos planes de estudio del Espacio Europeo de Educación Superior (EEES – Plan Bolonia)[1]. En este entorno, las clases son mucho más interactivas y la participación del alumno es mayor. Además, el método de evaluación es el principal cambio; para poder llevar a cabo el paradigma de evaluación continua establecido en los propósitos del Plan Bolonia, es necesario realizar test prácticamente todas las semanas a los alumnos y realizar estos test con un sentido de proximidad alto entre alumno y profesor.

El cambio que protagonizan éstas nuevas tecnologías ha contribuido formando una parte indispensable de nuestro día a día, como la penetración de los *Smartphones* en nuestra sociedad, pudiendo ser una herramienta que ayude a que el cambio hacia el EEES sea menos traumático tanto para el profesorado como para el alumnado por medio de la creación de aplicaciones que permitan realizar las tareas necesarias de forma más fácil, rápida y cómoda.

1.1. Situación Actual del Mercado de Smartphones

La evolución tecnológica a la que ya hemos hecho referencia apunta a un dispositivo que es referencia en el mercado actual: el *Smartphone*. Éste es un dispositivo en claro auge y ofrece un gran mercado para el desarrollo de aplicaciones multimedia, interactivas y distribuidas. De entre los *Smartphone*, los que disponen de un sistema operativo Android ofrecen un mayor número de ventajas, ya que cuenta con código abierto y permite desarrollar

aplicaciones con gran capacidad de forma sencilla, perfecta para iniciarse como programador en estas plataformas.

Decir que tanto el *Smatrphone* como el desarrollo de aplicaciones para éstos están en auge se puede considerar una obviedad. Sin embargo, de forma objetiva, el estudio de la situación del mercado de *Smartphones* arroja datos muy reveladores:

- El número de unidades de *Smartphones* vendidas durante el último cuatrimestre de 2010 según los datos de IDC[1] (International Data Corporation) ya superaba al de PCs[2]. Las cifras hablan de 101 millones de *Smartphones* frente a 92 millones de ordenadores personales, lo que supone un incremento del 87% anual en los primeros y apenas de un 3% de los segundos.

Worldwide smart phone and client PC shipments
Shipments and growth rates by category, Q4 2011 and full year 2011

Category	Q4 2011 shipments (millions)	Growth Q4'11/Q4'10	Full year 2011 shipments (millions)	Growth 2011/2010
Smart phones	158.5	56.6%	487.7	62.7%
Total client PCs	120.2	16.3%	414.6	14.8%
- Pads	26.5	186.2%	63.2	274.2%
- Netbooks	6.7	-32.4%	29.4	-25.3%
- Notebooks	57.9	7.3%	209.6	7.5%
- Desktops	29.1	-3.6%	112.4	2.3%

Source: Canalys estimates © Canalys 2012

Tabla 1: Evolución de las ventas

- Un estudio de Analysis Mason[3] estima que en 2014 habrá 1.700 millones de *Smartphone* en el mundo[4]. También pronostica que Android seguirá a la cabeza en número de dispositivos con este sistema operativo.
- Actualmente Google cifra en 190 millones la cantidad de dispositivos con Android que se encuentran en funcionamiento[5].
- Las cifras que predijo la consultora IDC en Abril de 2011 sobre 2012[6] se muestran en esta gráfica:

1 IDC es el principal proveedor mundial de inteligencia de mercado, servicios de consultoría y eventos para los mercados de tecnologías de la información, de telecomunicaciones y de consumo.

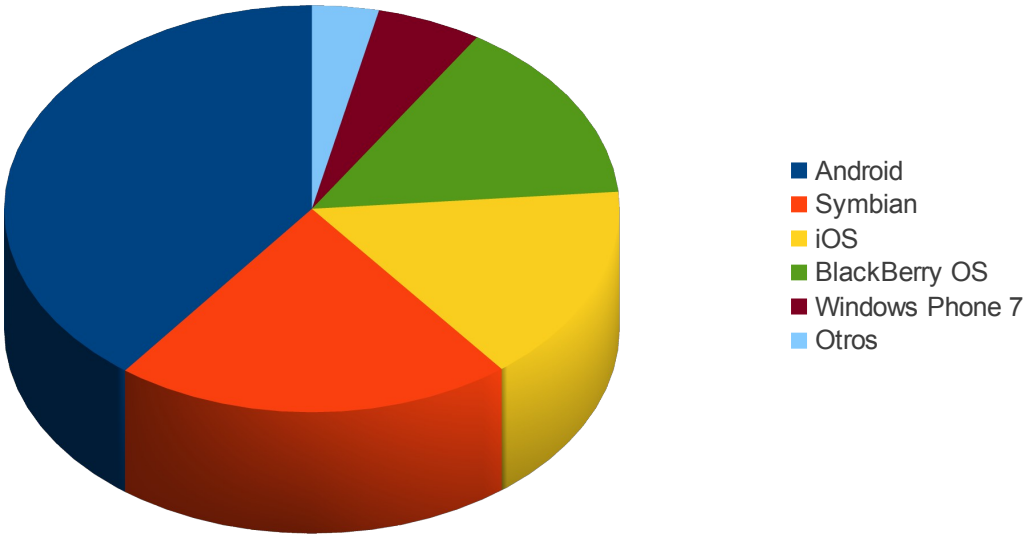

Ilustración 1: Gráfica de ventas de Smartphones

Además de estos datos, hemos realizado un estudio del tipo de conexión inalámbrica que incorporan los dispositivos móviles que se venden hoy en día [7]. Para ello, hemos estudiado los informes de ventas del ultimo año natural (junio 2010 – mayo 2011), del último año completado (2010) y podremos comparar las ventas del mes de mayo de 2010 con el mes de mayo de 2011.

- Durante el año 2010 se vendieron 4637 dispositivos móviles, de los cuales 3586 disponían de conectividad Bluetooth, lo que supone un 77,33 %. Solo 1222 tenían Wi-Fi (26,35%)
- En el mes de mayo de 2010 se vendieron 360 dispositivos, de los cuales 278 incorporaban Bluetooth (77,22%) y 88 Wi-Fi 88 (24,44%)
- En mayo de 2011 se dieron salida a 265 equipos; 210 de ellos disponían de Bluetooth (79,24%) y 143 de Wi-Fi (53,96%).

Las conclusiones que podemos sacar del estudio son claras:

- Aunque la mayoría de los móviles que se venden llevan Bluetooth no se experimenta un crecimiento importante en sus cifras.

- Aunque el porcentaje de móviles con Wi-Fi no es elevado, alcanza ya el 50%, habiéndose duplicado en el ultimo año, lo que hace pensar que si este aumento continúa (los datos parecen indicar que sí) alcanzará las cifras de Bluetooth en un futuro muy próximo.

1.2. Espacio Europeo de Educación Superior (EEES)

Proceso de Bolonia es el nombre que recibe el proceso iniciado a partir de la Declaración de Bolonia, acuerdo que en 1999 firmaron los ministros de educación de diversos países de Europa (tanto de la Unión Europea como de otros países como Rusia o Turquía), en la ciudad italiana de Bolonia. Se trató de una declaración conjunta (la UE no tiene competencias en materia de educación) que dio inicio a un proceso de convergencia que tenía como objetivos facilitar el intercambio de titulados y adaptar el contenido de los estudios universitarios a las demandas sociales mejorando su calidad y competitividad a través de una mayor transparencia y un aprendizaje basado en el estudiante cuantificado a través de los créditos ECTS (Sistema Europeo de Transferencia y Acumulación de Créditos).

Ilustración 2: Mapa de países participantes en el Plan Bolonia

El proceso de Bolonia, pese a no ser un tratado vinculante, condujo a la creación del EEES (Espacio Europeo de Educación Superior), un ámbito al que se incorporaron países y

que serviría de marco de referencia a las reformas educativas que muchos países habrían de iniciar en los primeros años del siglo XXI. Este acuerdo se enmarca dentro del Acuerdo General de Comercio de Servicios, firmado en 1995 y cuyo objetivo declarado es "liberalizar el comercio de servicios" a escala mundial (porque la OMC, Organización Mundial del Comercio, integra a 151 estados, incluyendo a toda la Unión Europea) para introducirlos en el mercado, ya que "la financiación pública es un elemento de distorsión de los mercados".

Las metas y mejoras que se atribuyen al Plan Bolonia[8] las podemos encontrar en el Anexo 1. Comentaremos los cambios que nos ayudarán a entender la motivación de este PFC.

La introducción del Plan Bolonia en España, con la intención de equiparar los títulos universitarios españoles con los del resto de Europa, ha cambiado la forma de entender la educación superior. El cambio principal es en la forma de evaluación que pasa a ser evaluación continua de forma obligada y en todas las asignaturas, pretendiendo que las notas se correspondan con el trabajo realizado a lo largo de toda la evaluación y no dependiendo de una única prueba. En este entorno, las clases deben volverse mucho más interactivas por parte del alumno, es decir, mucho más participativas.

La implantación del nuevo Plan Bolonia, hace que la educación, tal y como la conocíamos hasta el momento, cambie. La participación de los alumnos aumenta, las clases son más interactivas y el número de pruebas de evaluación que se realizan a los alumnos se incrementan (prácticamente todas las semanas se deben realizar exámenes).

Con esos cambios aumenta la dificultad hasta casi hacer imposible corregir ese gran número de tests. Otra de las repercusiones es que los profesores deben hacer las clases mas interactivas, aumentando la participación de los alumnos durante las clases, debe acercar las lecciones explicadas a los alumnos, el alumno aprenderá mas participando en la explicación que siendo un simple oyente.

1.3. Proyectos y Aplicaciones TIC para el EEES

Todo esto, unido a la intención del Gobierno de España de modernizar las técnicas tanto de enseñanza como en otros ámbitos por medio del Plan Avanza[9], crea un espacio en el que poder incluir las TICs para que faciliten la labor del profesor y del alumno.

A continuación describiremos diferentes soluciones disponibles en el mercado actual y, así, poder enmarcar el contexto para el desarrollo de nuestro proyecto.

Una de las herramientas más utilizadas en el ámbito de la educación superior que aprovecha las nuevas tecnologías y la colaboración activa de alumnos es el *Moodle*[10]. *Moodle* es un Ambiente Educativo Virtual, sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como LMS (*Learning Management System*).

Moodle fue creado por Martin Dougiamas, quien fue administrador de WebCT en la Universidad Tecnológica de Curtin. Basó su diseño en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo. Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer.

Otra herramienta utilizada es lo que se conoce como Campus Virtual[11]. Como campus virtual se entiende a una estructura creada a manera de comunidad virtual en la que se desarrollan las actividades académicas de una institución educativa en cualquiera de sus formas, desde un pequeño entorno de capacitación, hasta englobar una universidad completa.

Es un espacio exclusivo para los alumnos de los cursos y está orientado a facilitar su experiencia de capacitación a distancia. Ofrece información adicional, contacto interactivo de los alumnos con los docentes y entre los mismos alumnos para compartir sus experiencias, ofrece también acceso a informes, notas, artículos y libros escogidos por el Consejo Académico como material adicional al utilizado para el curso.

Habitualmente se utiliza como un espacio *online* para facilitar la comunicación entre

profesores y alumnos. Las asignaturas que utilizan el campus virtual ofrecen servicios como: publicación de material teórico de la asignatura, foros de contacto con alumnos, contacto con profesores, envío de prácticas y documentación al profesor, contacto con el profesor...

Además a través del campus virtual (o área virtual en algunas universidades) se ofrecen otros servicios administrativos útiles para los alumnos: matriculación, resultados de exámenes y notas de asignaturas, horarios de clases, consulta de expedientes académicos, etc.

En lo que respecta al mercado[12] podemos encontrar los siguientes productos relacionados con la introducción de las nuevas tecnologías en la enseñanza:

- Ecuaderno de la editorial Additio[13], una aplicación *online* que sirve de ayuda a los profesores en su día a día con la evaluación, la gestión de archivos, la agenda y los contactos.
- MimioClassroom de DYMO[14], que consta de una gama de dispositivos relacionados con el aula como MimioTeach (dispositivo que convierte cualquier pizarra blanca en una PDI, 595€), MimioPad (una tableta donde escribir y dibujar para que se muestre proyectado en la pizarra, 275€), MimioVote (un dispositivo con el que responder a las soluciones de forma automática, 1495€ con 24 mandos) y MimioCapture (gama de lapices y borrador digitales complementarios a MimioTech, 319€).
- LanSchool[15], un software para la gestión y monitorización de los distintos equipos de los alumnos. Precio: 2.450€
- Pizarra táctil interactiva UB-T880 de Panasonic[16]. Precio 2.121€ junto con proyector.
- SMART Board 800[17], familia de pizarras digitales de ultima generación que permite trabajar a varias personas interactuando con ella a la vez tanto con lapices digitales, dedos o cualquier objeto puntiagudo. También permite gestos con uno o dos dedos, pellizcar, ampliar-reducir... El precio orientativo en EE.UU. para el modelo estándar SMART Board 880 es 1.999 $ y3.899 $ para el sistema de pizarra digital interactiva 880i.
- Además de las tabletas que sirven incluso para la integración de alumnos con distintos tipos de discapacidades comunicativas. [18]

Del estudio del mercado actual podemos sacar algunas conclusiones para poder

establecer qué necesidades quedan sin satisfacer, de manera que encontremos el nicho en que amoldar nuestro proyecto, tratar de que solucione el mayor número de problemas y haga desaparecer el mayor número de necesidades para la realidad española.

Por una parte, las herramientas que convierten una simple pizarra blanca en un sistema táctil e interactivo, cuyo elevado coste es inviable para la educación española. De la misma manera, las herramientas de gestión de alumnos y notas están muy alejadas de las que permiten evaluar como el MimioClassroom; ésta aplicación se vende con unos mandos de votación con distintas opciones con alto coste y no compatibles con otros sistemas.

Idealmente, un producto adecuado pasaría por la creación de una aplicación que permita al profesor gestionar sus alumnos, añadiendo esa funcionalidad clave como crear exámenes y una herramienta que los corrija de forma automática, que sea fácil de integrar en el entorno educativo y cuyos costes sean lo más bajos posibles.

De esta manera podemos utilizar dispositivos de uso cotidiano para sustituir los caros aparatos con los que cuentan otras soluciones. Dispositivos como los *Smartphones* o las *tablets* tienen un reducido coste en comparación a su funcionalidad, o están reduciéndolo de forma importante. Además, la mayoría de alumnos ya cuentan con su propio *Smartphone*; creando una aplicación que puedan utilizar desde su propio teléfono hará mas interactivas las clases, disminuirá los costes, facilitara su integración en las actividades y aumentará la ubicuidad (allí donde lleven su teléfono, tendrán su aplicación).

1.4. Descripción del Problema

Procedemos a explicar varios de lo que creemos problemas derivados de la situación actual.

Uno de estos factores externos determinantes y muy a tener en cuenta en la descripción del problema y en la posterior elaboración de una solución es la crisis económica, mundial y sobretodo europea en la que vivimos instalados. Muchos de los recortes anunciados de los gobiernos europeos afectan a la educación por lo que encontrar una solución de bajo coste de creación, instalación y mantenimiento será clave.

El numero de test que se realizan ahora han aumentado hasta convertirse en exámenes

diarios o semanales ya que el nuevo Plan Bolonia sugiere un seguimiento mucho mas exhaustivo del progreso de los alumnos a lo largo de la evaluación. Estos test son mucho menos voluminosos en los que a materia se refiere, es decir, incluyen menos información pero deben ser muchos mas ágiles en su creación, resolución y corrección. Debemos crear por tanto un "centro de exámenes" de fácil manejo para profesores que ya están adaptándose a un nuevo método de enseñanza y a los que no podemos hacer invertir tiempo en aprender a usar difíciles nuevas aplicaciones. Este centro de exámenes tiene que permitir una gestión total sobre los mismos, nos debe permitir crearlos, poder servirlos a los alumnos y, también, ver cuales y cuantas preguntas contienen los test creados ya.

Relacionado con este centro de exámenes, debemos contar con un el servidor de exámenes que a su vez debe permitir a los alumnos contestar el examen seleccionado por el profesor, creado con anterioridad. Una aplicación móvil permitirá a los alumnos contestar a los test planteados por los profesores en tiempo real.

Con respecto a esta aplicación móvil, podemos decir que los alumnos deben poder identificarse en esta aplicación antes de realizar los exámenes. Esta identificación apoyada en una base de datos servirá para que no se produzcan identificaciones falsas y otros problemas de seguridad.

La base de datos en la cual se apoyan las anteriores aplicaciones debe contener usuarios de los dos perfiles existentes, profesores y alumnos, con la información suficiente para que no se produzcan perfiles falsos y con información sobre las cosas que los interrelacionan, asignaturas, cursadas por unos e impartidas por otros, y exámenes, pertenecientes a asignaturas, creados por profesores y realizados por alumnos. Además, esta base de datos nos dará la no volatilidad en los datos necesaria.

Los dispositivos móviles usados serán *Smartphones*, dispositivos de bajo coste con capacidad de conexión inalámbrica e instaurados en la vida diaria de un gran porcentaje de alumnos ya que la mayoría de ellos disponen del propio y lo utilizan a diario.

Por los motivos descritos anteriormente, se ha decidido crear una solución que trate de ahorrar tiempo de corrección al profesor, acercando a los alumnos a las clases y aumentando la realimentación sobre su propio progreso en la asignatura.

2. Objetivo

En este capítulo explicaremos la línea de trabajo a seguir durante el desarrollo del proyecto. Describiremos el problema que se quiere abordar, los requisitos que deben cumplir las soluciones que se adopten y qué métodos de trabajo se han seguido. Para todo esto, contamos con la ayuda del estudio realizado en el apartado anterior.

Como hemos visto, el EEES requiere de cambios conceptuales en la metodología de evaluación. Por otra parte, los *Smatphone* incorporan una funcionalidad excepcional que puede dar respuesta a las necesidades del Plan Bolonia.

En el Departamento de Tecnología Electrónica de la Universidad Rey Juan Carlos (URJC) se investiga en este ámbito para resolver necesidades propias y ajenas. Un ejemplo de esto es la necesidad de recoger la opinión de los asistentes a las Jornadas de Electrónica de la URJC organizadas por el departamento de forma informatizada; así como también poder examinar a alumnos de los profesores adscritos al departamento en sus distintas asignaturas.

De esta manera, con la automatización de los procesos, se abriría un nuevo camino a la educación personalizada mediante el conocimiento de las capacidades en tiempo real. Además, mediante la organización de los datos, se podrían conocer estadísticas deseadas.

2.1 Objetivos específicos

Entonces, con la información de la que disponemos podemos establecer que los objetivos específicos que este proyecto debe cumplir son:

- Crear un centro de creación de exámenes sencillo y ágil.

  Debemos crear una aplicación para el profesor donde poder crear exámenes, ver dichos exámenes ya creados, ver las notas y sacar algunos datos de los exámenes ya realizados.

- Crear un sistema automático de corrección de exámenes.

- Crear una aplicación móvil para resolver los exámenes propuestos.

  La forma de acercar al alumno estos test y por tanto las clases será utilizando dispositivos con los que ellos están familiarizados, el *Smartphone.*

- Base de datos de evaluaciones.

  Debemos crear una base de datos donde guardar los datos de los alumnos, profesores y los resultados obtenidos por cada alumno en cada examen porque ya hemos dicho que nos interesa que el profesor tenga una herramienta completa de gestión.

Con estos requisitos el profesor ahorra tiempo en la creación y corrección de los test y el alumno se siente participe de las clases ya que, con su propio dispositivo, podrá responder a las preguntas planteadas por el profesor.

El problema principal está orientado a la educación superior, pero no podemos renunciar a otros posibles ámbitos. Será posible utilizar estas aplicaciones para que niños en un colegio respondan a sus exámenes por medio de éstas, los alumnos responderán de forma más ágil y rápida, incluso captando la atención de los alumnos de edades mas tempranas con mayor facilidad al introducir la tecnología en sus clases.

También es posible que un colegio que trate con niños discapacitados (o algún otro tipo de colectivo especial), acostumbrados al uso de este tipo de dispositivos para su integración, estuviese interesado en el uso de la solución creada, por tanto tendremos bien abiertos los ojos ante la posible aplicación de este tipo de solución a problemas similares.

Expondremos en este Proyecto Fin de Carrera la solución adoptada para resolver todo este tipo de problemas con una serie de funcionalidades principales: crear cuestionarios, corregirlos y gestionar los resultados desde el lado del profesor (o encuestador) y resolver cuestionarios desde el lado del alumno (o encuestado). Podremos resolver los cuestionarios desde dispositivos móviles (*Smartphones*) por medio de conexión inalámbrica (Wi-Fi).

3. Requisitos

Ya hemos analizado el problema, ahora debemos definir bien que requisitos debe cumplir nuestra solución, más tarde veremos que alternativas tenemos para cumplir estos requisitos que ofrecen solución al problema.

Nos ayudaremos de un diagrama de casos de uso en UML para definir bien estos requisitos:

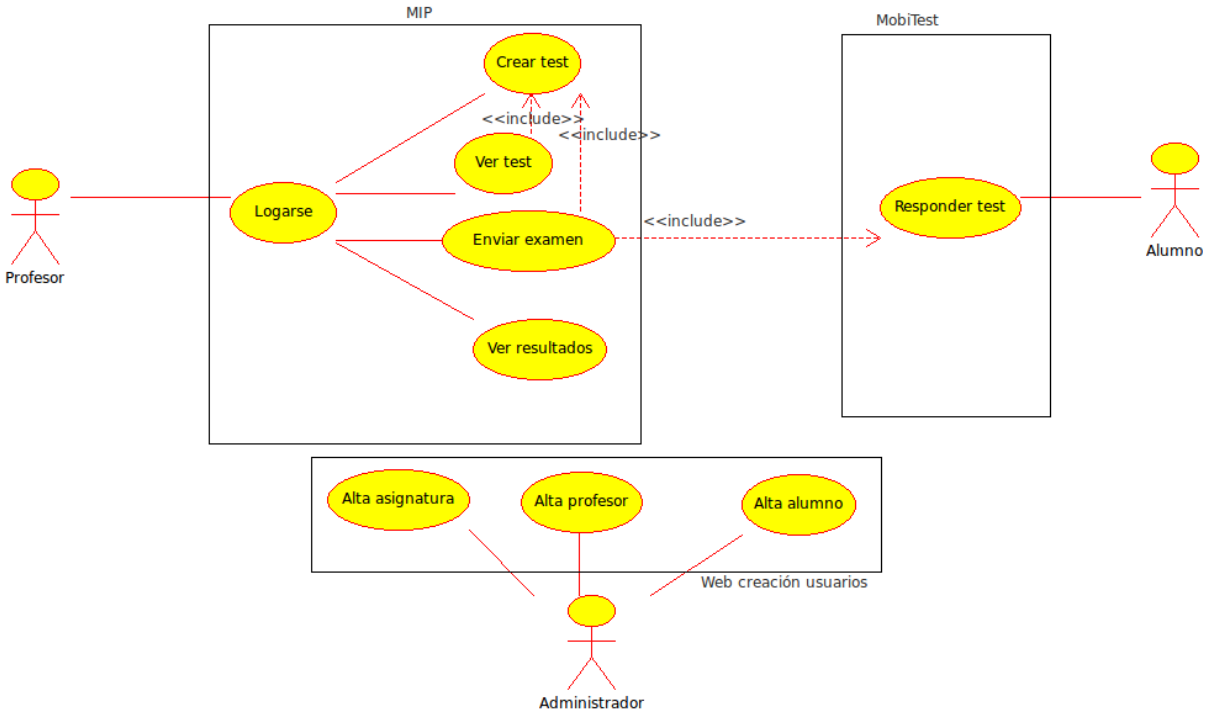

Ilustración 3: Diagrama de casos de uso UML

El proyecto realizado se plantea con una serie de requisitos mínimos que debe cumplir su implementación:

- Hemos definido tres tipos de usuarios, aunque el administrador es un usuario eventual, es decir, aparece y desaparece para realizar alguna función determinada pero no participa de forma continua en el sistema. Los tres tipos de usuario son: administrador, profesor y alumno.
- El administrador se encargará de la gestión de la base de datos en la que el resto de usuarios deben estar dados de alta. Crearemos una pequeña web para la gestión de la

base de datos encapsulando las operaciones necesarias para que el administrador no tenga que tener conocimientos sobre bases de datos.

- La base de datos que debe permitir almacenar alumnos, profesores y asignaturas, que se sepa que profesor imparte cada asignatura y que alumnos las cursan. También debe almacenar los exámenes que crean los profesores indicando a que asignatura corresponden y el patrón de respuestas correctas de cada uno de ellos. Tiene que permitir almacenar las notas de los alumnos en los exámenes realizados por cada uno de ellos. Las demás aplicaciones deben comprobar en esta base de datos la identidad de dichos usuarios ya que ningún alumno debe acceder a la aplicación del profesor y ningún profesor a la aplicación de los alumnos
- Para el perfil de alumno solo se debe satisfacer un requisito o necesidad, debe permitir le responder a los test creados por el profesor y que se ofrecen en cada momento al "iniciar un cuestionario". Esto se realizará a través de una aplicación para dispositivos móviles.
- El perfil del profesor es que el que tiene más necesidades que satisfacer, nuestro problema se centraba en la posibilidad de ahorrar tiempo debido al aumento en numero de test con el cambio en la educación. Por tanto, este perfil profesor debe poder crear nuevos test, ver los que ya hay creados, iniciar un cuestionario con alguno de estos test ya creados para que los alumnos los respondan y comprobar todo aquello con los resultados obtenidos. La información que se ofrecerá en este apartado serán las notas de los alumnos en cada uno de sus test y alguna también relacionada con las opciones más escogidas por parte de los alumnos en cada una de las preguntas de los test. Esto será posible a la comunicación también de esta aplicación que crearemos para el profesor con la base de datos ya que le servirá para guardar las notas de los alumnos entre otras muchas más cosas.

3.1. Metodología

Todos los módulos que forman el proyecto han sido desarrollados de forma simultanea ya que unos bebían de otros. El modelo en espiral ha sido el elegido para desarrollar cada uno

de ellos.

El modelo en espiral es un modelo de ciclo de vida del software definido por primera vez en 1988 por Barry Boehm[19], y utilizado en la Ingeniería del Software. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. En cada vuelta o iteración hay que tener en cuenta los objetivos y las alternativas. La idea principal es decidir que problema vamos a resolver, estudiar todas las alternativas, escoger una, desarrollarla y comprobar su buen funcionamiento antes de afrontar la siguiente etapa (vuelta de la espiral).

Una de las grandes ventajas de este modelo es que después de cada una de las vueltas a la espiral disponemos de un producto acabado que poder mostrar al cliente. De este modo, dicho cliente va evaluando la evolución de producto y puede atajar un posible fallo o alguna opción que no es de su agrado justo después de implementarla sin que afecte a posteriores requisitos.[20]

3.1.1 Estudio de Alternativas y Planificación

Ya hemos visto cual es el principal problema, ahora debemos estudiar cuales son las posibles soluciones. Una idea muy básica para nuestra solución es realizar algo similar al videojuego Buzz, que esta basado en un concurso de televisión con varias rondas de preguntas que los participantes (máximo de 4) deben ir contestando para conseguir puntos[21]. Evidentemente hay diferencias, en nuestra solución no interesa que los participantes compitan entre sí; tampoco disponemos de los pulsadores, pero sí podemos simularlos con un *Smartphone*.

Una de las soluciones pensadas para hacer llegar los exámenes a los alumnos sería a través de diapositivas y un proyector pero esto haría que la proximidad del alumno con el profesor y su participación se viese afectada. Además, la aplicación para alumnos debe ir conectada a la aplicación del profesor para enviarle los datos, conectémosla también para recibir los exámenes en su propio dispositivo móvil.

Durante el desarrollo de las aplicaciones irán surgiendo problemas que solucionar pero de momento tenemos importantes rasgos de nuestra aplicación.

En resumen, la aplicación del profesor que llamaremos Mesa Interactiva del Profesor

(MIP) debe ser capaz de: gestionar los cuestionarios (crear nuevos de forma rápida y ágil y ver los ya creados), enviarlos a los dispositivos de los alumnos para que puedan responderlos, recoger las respuestas de los alumnos y ver los resultados obtenidos por los alumnos para poder analizarlos.

La aplicación del dispositivo móvil, que como concluimos en el estado del arte será para Android y que llamaremos MobiTest, debe ser capaz recibir el examen desde el ordenador del profesor, mostrarle al alumno cada una de las preguntas, que el alumno las responda y enviar de vuelta a la MIP las respuestas de todos los alumnos.

También vamos a crear una aplicación web donde de forma cómoda podamos ir añadiendo usuarios a la base de datos (alumnos y profesores) y la información que sea necesaria. Más tarde, en la parte de desarrollo veremos como lo primero en diseñarse va a ser la base de datos donde luego van apoyadas las demás aplicaciones.

Conocidos los distintos objetivos y requisitos de nuestro proyecto, podemos estipular que la duración del mismo será de aproximadamente 150 días, lo que viene a ser 30 semanas. Suponiendo que cada semana se realizará un trabajo de aproximadamente 25 horas, en el siguiente diagrama podemos ver una estimación de la distribución del trabajo a realizar.

Para empezar el proyecto debemos realizar el estudio y la planificación del mismo, el primer paso fue realizar un estudio de requisitos, estudiando cual era el problema y cuales las necesidades que debía satisfacer (definidas anteriormente en el apartado de objetivos específicos). Definimos el plan de acción, que dispositivos tecnológicos iban a ser necesarios y como íbamos a organizar el trabajo. Tras determinar esto pudimos empezar con el diseño, como queríamos que fuesen nuestras aplicaciones, los perfiles de los usuarios, bocetos, etc.

Una vez acabado esto, dedicamos un tiempo a formarnos en el desarrollo y afrontamos la implementación de las aplicaciones. Solo queda probar y comprobar los resultados.

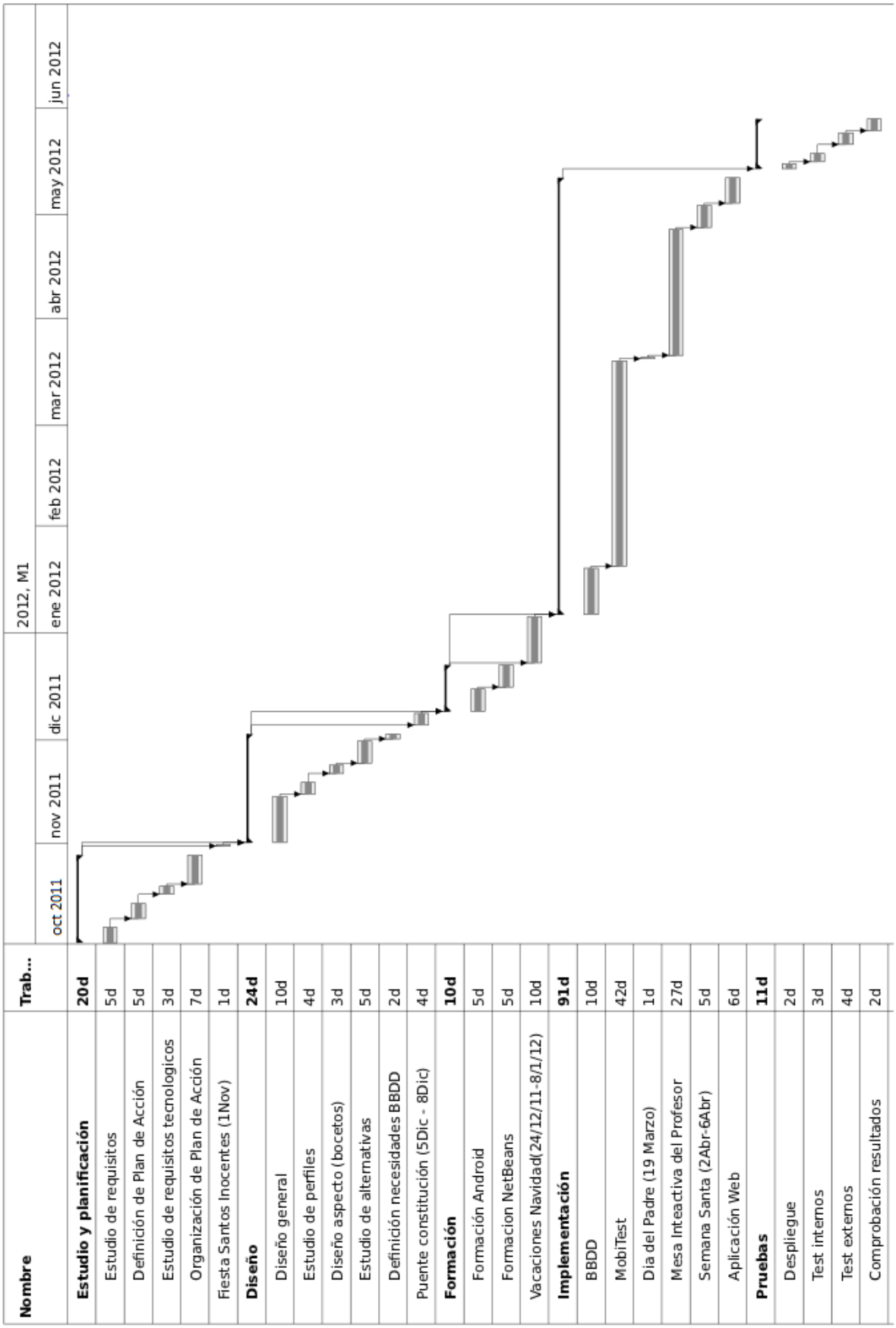

Ilustración 4: Diagrama de Gant

3.1.2. Recursos

A continuación pasaremos a formalizar los recursos necesarios en el desarrollo de nuestro proyecto, incluyendo por una parte los recursos materiales y por otra los recursos humanos, acorde con nuestra planificación.

Recursos materiales:

Dispositivo	Cantidad	Precio
TRENDNET Router WiFi N 300 Mbp/s TEW-652BRP	1	32,90 €
HTC Desire C	1	369 €
Sony Ericsson Experia PLAY	1	269 €
Samsung Galaxy S	1	308,99 €
Ordenador Portátil Samsung R55	1	850€

Tabla 2: Recursos materiales

Los Recursos humanos se distribuyen en jornadas de trabajo de 5 horas, de lunes a viernes y contando las vacaciones del calendario laboral de la Comunidad de Madrid, como se muestra en el Diagrama de Gant (Ilustración 4). La realización del proyecto comienza el 3 de Octubre de 2012 y finaliza el 1 de junio y abarca 150 jornadas de trabajo de una única persona.

4. Materiales y métodos

En este capítulo nos centraremos en explicar las principales herramientas con las cuales vamos a trabajar para desarrollar nuestro proyecto, ya que, una vez conocidas estas, podremos centrarnos en los aspectos importantes a aplicar.

Para ello, vamos a explicar brevemente las distintas herramientas y tecnologías que hemos empleado para la realización del proyecto completo y que definirían parte de las necesidades y limitaciones a la hora de desarrollar el mismo.

4.1. Entorno de Desarrollo

Lo primero, señalar que el sistema operativo usado ha sido Ubuntu 11.04, la única desventaja que podía existir es que algún programa de los necesarios no estuviese disponible para este SO (Sistema Operativo) pero una vez descartado eso Ubuntu ofrece muchísimas ventajas. Para empezar es *freeware* (tipo de software que se distribuye sin costo, disponible para su uso y por tiempo ilimitado), instalar los programas necesarios es fácil y rápido gracias a su "Centro de Software". Otra de las ventajas que ofrece Ubuntu es su sistema de escritorios múltiples, que facilita el trabajo de desarrollo al ofrecer la posibilidad de tener un escritorio donde hacer consultas y otro donde poder programar, también su potente *Shell*, además de otras más importantes ventajas.

La instalación de Ubuntu es extremadamente sencilla y se puede hacer desde su web[22] . Sobre este sistema operativo se asentarán una serie de herramientas necesarias para implementar las distintas partes de nuestro proyecto y que pasamos a explicar a continuación.

4.1.1. MySQL Workbench

Para realizar la base de datos hemos empleado "MySQL Workbench". MySQL Workbench es un software creado por la empresa Sun Microsystems, esta herramienta permite

modelar diagramas de Entidad-Relación para bases de datos MySQL.

Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para exportare e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador.

MySQL Workbench puede generar también el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x.

La instalación la realicé buscando en el "Centro de Software" de Ubuntu y pulsando instalar. Aparece entonces en el menú de Aplicaciones, en el apartado de Programación.

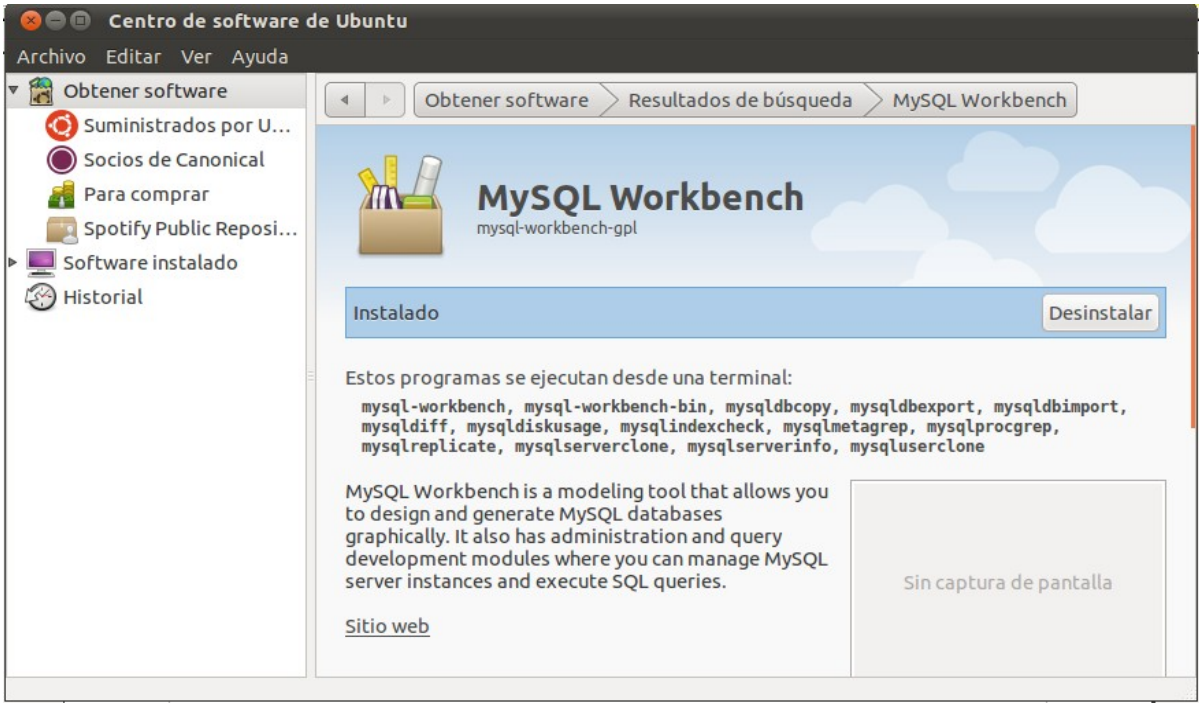

Ilustración 5: Centro de Software Ubuntu

4.1.2. Eclipse

Eclipse [23]es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar, entre otras cosas, Entornos de Desarrollo Integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se

entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

En nuestro caso hemos empleado la versión “Indigo”, al que hemos incorporado el SDK (Software Development Kit) de Android para realizar la aplicación Android.

Eclipse no necesita de instalación, solamente descargar el paquete desde su web[23], extraerlo dónde creamos mas conveniente y ejecutarlo.

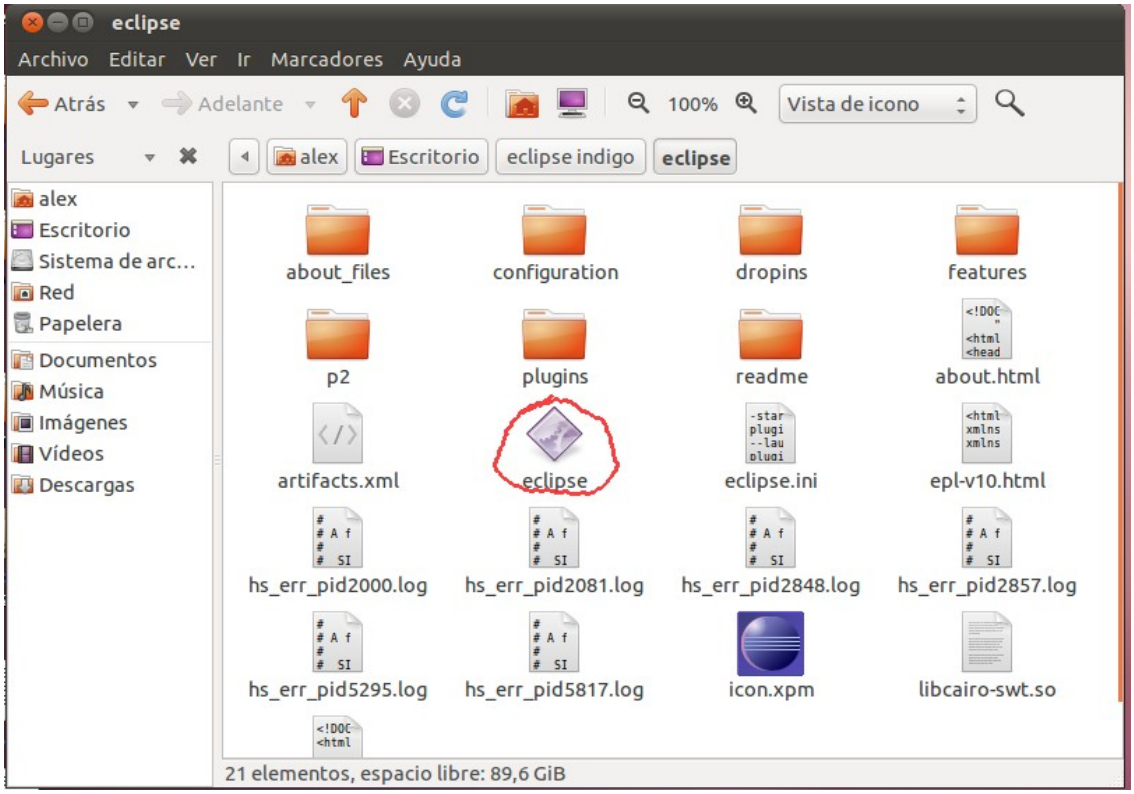

Ilustración 6: Ejemplo ejecutable Eclipse

Eclipse es un gran entrono de programación ya que completa muchas de las instrucciones que escribes, ayuda mucho en la depuración tanto en tiempo de compilación como en tiempo de ejecución. Fue muy sencillo instalar los *plugins* necesarios y el SDK para Android lo contamos en el apartado siguiente.

4.1.3. SDK de Android

Android **SDK** es un kit de desarrollo con el que podremos desde desarrollar aplicaciones hasta ejecutar un emulador del sistema Android en la versión que queramos,

incluye un conjunto de herramientas de desarrollo[24]: un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen Linux (cualquier distribución moderna), MacOS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento ADT (Android Development Tools plugin), aunque también puede utilizarse un editor de texto para escribir ficheros Java y Xml y utilizar comandos en un terminal (se necesitan los paquetes JDK, Java Development Kit y Apache Ant) para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados (e.g. reiniciarlos, instalar aplicaciones en remoto).

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo Android (este directorio necesita permisos de superusuario , root, por razones de seguridad). Un paquete APK incluye ficheros .dex (ejecutables Dalvik, un código intermedio compilado), recursos, etc.

En el anexo 3 podemos ver cual es el proceso de instalación.

4.1.4. NetBeans

Eclipse es un entorno de desarrollo muy rico, sin embargo, el desarrollo de GUIs (Graphical User Interfaces) es mucho más sencillo si empleamos NetBeans. Una gran parte de nuestro proyecto esta realizada con NetBeans, que nos proporciona un nivel de abstracción suficiente para realizar la interfaz amigable necesaria, y una vez más el “Centro de Software” de Ubuntu nos ayudó en la tarea de instalación. Tan sencillo como buscarlo y pulsar instalar. NetBeans no ayuda en la programación como Eclipse auto-completando sentencias o sugiriendo como continuar, es algo más sobrio tanto en su aspecto como en su

comportamiento pero la facilidad con la que se crean interfaces gráficos con NetBeans bien lo vale. Un principiante en esos temas ha sido capaz de diseñar una aplicación completa empezando desde 0 y con conocimientos muy básicos, mejorando y aprendiendo a medida que desarrollaba la MIP. Otras herramientas no hubieran permitido ir avanzando y conociéndola con la facilidad que lo permite NetBeans. Además, viene con una buena cantidad de *plugins* instalados en la versión mas básica, al contrario que Eclipse.

Para poder luego gestionar la base de datos utilizamos MySQL Administrator, que es un sistema gestor de bases de datos y me permitió hacer pruebas con la base de datos montada sobre mi propio PC, esta BBDD debe ir montada en un servidor cuando el funcionamiento de MobiTest sea completo pero esta herramienta esta muy bien, es libre y se instala con un sencillo comando en la consola.

4.1.5. PHP

PHP es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Para usar PHP es necesario instalar un servidor que le de soporte, Apache es el idóneo y con dos instrucciones en un Terminal de nuestro Ubuntu será suficiente:

sudo apt-get install php5-mysql

sudo apt-get install php5-common libapache2-mod-php5 php5-cli

4.2. Dispositivos

Como hemos dicho anteriormente vamos a usar distintos dispositivos móviles, además de un ordenador donde irá instalada la aplicación del profesor. Posteriormente veremos una ilustración que nos dejara más claro cuales son los dispositivos usados. En este apartado veremos cuales fueron necesarios para el desarrollo y algunas pruebas con sus caracteristicas principales.

Empezamos por el ordenador con el que se han realizado la totalidad del proyecto, un portátil Samsung R55 de unos 4 años de antigüedad, que cuenta con un procesador Core 2 DUO T5600 a 1,83 GHz, 2 GB de memoria y 120 GB de disco duro como características principales.

Los dispositivos móviles usados son el Sony Ericsson Xperia PLAY, con sistema operativo Android v2.3, conexión inalámbrica Wi-Fi y bluetooth y procesador Qualcomm MSM8255 Snapdragon 1 Ghz como datos más relevantes. También cuenta con 380 MB de RAM disponibles para el usuario.

El HTC Desire dispone de un procesador Qualcomm Snapdragon QSD8250 1 Ghz, una memoria de 576 MB RAM, sistema operativo Android v2.1 y conexión inalámbrica Wi-Fi y Bluetooth.

El último dispositivo utilizado en el desarrollo del proyecto es el Samsung Galaxy S, que dispone de un procesador ARM Cortex A8 Hummingbird 1GHz, 512 MB de RAM y sistema operativo Android v2.1 Eclair.

Ilustración 7: HTC Desire

Ilustración 8: Sony Xperia Play

Ilustración 9: Samsung Galaxy S

5. Desarrollo

Una vez realizado el análisis del estado del arte, de haber realizado el estudio de las alternativas para solucionar el problema planteado, de las necesidades de nuestro proyecto y de los materiales y métodos que vamos a emplear, podemos proceder a desarrollar nuestra aplicación.

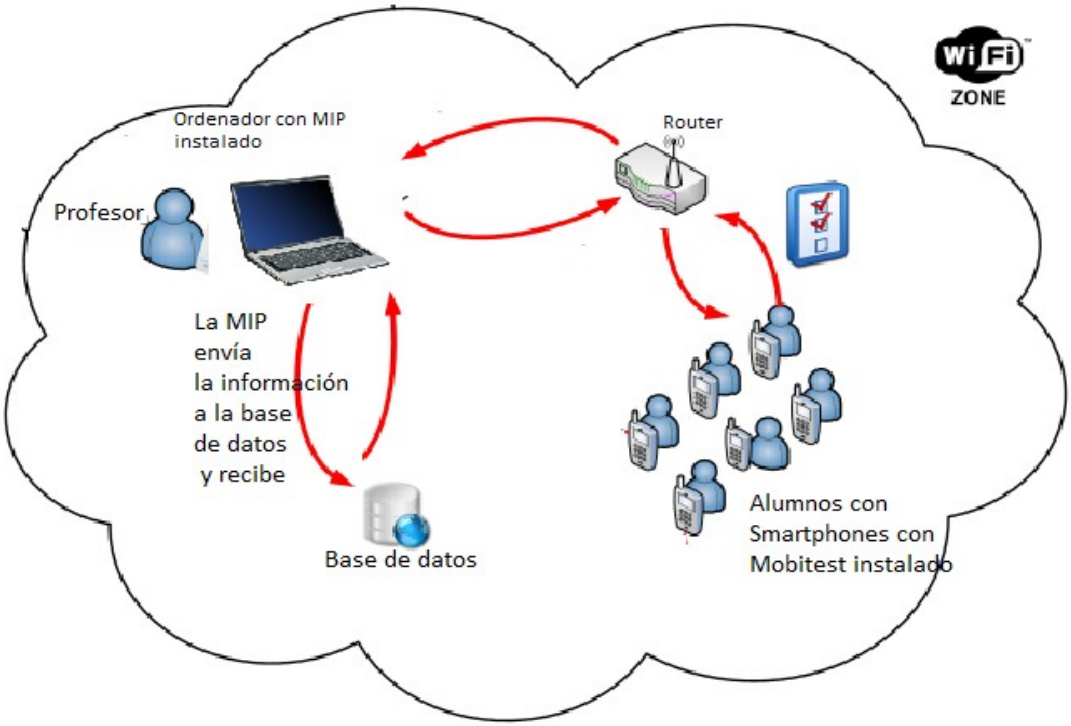

Ilustración 10: Arquitectura del sistema completo

5.1. Diseño e implementación

5.1.1 Base de datos

Debemos empezar implementando la base de datos ya que el resto de aplicaciones se apoyan en ella. Veamos que características son las más relevantes y las que tenemos que reflejar en nuestra base de datos. Hemos analizado los tipos de perfiles que deben estar en la

base de datos, los cuales son alumnos y profesores, cada uno con su información básica (nombre, apellidos, DNI, password...). Los profesores imparten asignaturas que son cursadas por alumnos, luego las asignaturas son el nexo de unión entre ellos. También tenemos que registrar exámenes que son elaborados por los profesores, que pertenecen a una asignatura y que son realizados por los alumnos. Añadimos los datos necesarios de cada una de las dos últimas tablas, el código de examen, el código de asignatura, la cadena de respuestas correctas y ya tenemos la base de datos concluida.

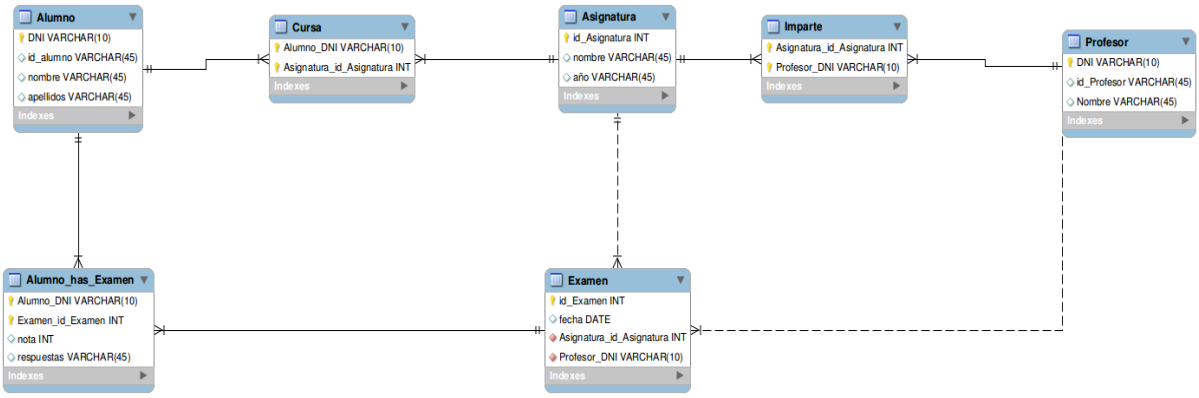

Ilustración 11: Representación E/R de la Base de Datos

Esta base de datos la modelamos en nuestro MySQL Workbench y nos genera el código en mySQL:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `dbmovitest` DEFAULT CHARACTER SET
latin1 COLLATE latin1_swedish_ci ;
USE `dbmovitest` ;

-- -----------------------------------------------------
-- Table `dbmovitest`.`Alumno`
-- -----------------------------------------------------
```

```
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Alumno` (
  `DNI` VARCHAR(10) NOT NULL ,
  `id_alumno` VARCHAR(45) NULL ,
  `nombre` VARCHAR(45) NULL ,
  `apellidos` VARCHAR(45) NULL ,
  `password` VARCHAR(15) NULL ,
  PRIMARY KEY (`DNI`) )
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `dbmovitest`.`Asignatura`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Asignatura` (
  `id_Asignatura` INT NOT NULL ,
  `nombre` VARCHAR(45) NULL ,
  `año` VARCHAR(45) NULL ,
  PRIMARY KEY (`id_Asignatura`) )
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `dbmovitest`.`Profesor`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Profesor` (
  `DNI` VARCHAR(10) NOT NULL ,
  `id_Profesor` VARCHAR(45) NULL ,
  `Nombre` VARCHAR(45) NULL ,
  `Apellidos`VARCHAR(45) NULL,
  `password` VARCHAR(15) NULL ,
  PRIMARY KEY (`DNI`) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `dbmovitest`.`Examen`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Examen` (
  `id_Examen` VARCHAR(10) NOT NULL ,
  `fecha` DATE NULL ,
  `Asignatura_id_Asignatura` INT NOT NULL ,
  `Profesor_DNI` VARCHAR(10) NOT NULL ,
  `respuestas` VARCHAR(45) NULL ,
  PRIMARY KEY (`id_Examen`) ,
  INDEX `fk_Examen_Asignatura` (`Asignatura_id_Asignatura` ASC) ,
  INDEX `fk_Examen_Profesor1` (`Profesor_DNI` ASC) ,
  CONSTRAINT `fk_Examen_Asignatura`
    FOREIGN KEY (`Asignatura_id_Asignatura` )
    REFERENCES `dbmovitest`.`Asignatura` (`id_Asignatura` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Examen_Profesor1`
    FOREIGN KEY (`Profesor_DNI` )
    REFERENCES `dbmovitest`.`Profesor` (`DNI` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `dbmovitest`.`Cursa`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Cursa` (
  `Alumno_DNI` VARCHAR(10) NOT NULL ,
  `Asignatura_id_Asignatura` INT NOT NULL ,
  PRIMARY KEY (`Alumno_DNI`, `Asignatura_id_Asignatura`) ,
```

```
                INDEX        `fk_Alumno_has_Asignatura_Asignatura1`
(`Asignatura_id_Asignatura` ASC) ,
  INDEX `fk_Alumno_has_Asignatura_Alumno1` (`Alumno_DNI` ASC) ,
  CONSTRAINT `fk_Alumno_has_Asignatura_Alumno1`
    FOREIGN KEY (`Alumno_DNI` )
    REFERENCES `dbmovitest`.`Alumno` (`DNI` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Alumno_has_Asignatura_Asignatura1`
    FOREIGN KEY (`Asignatura_id_Asignatura` )
    REFERENCES `dbmovitest`.`Asignatura` (`id_Asignatura` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `dbmovitest`.`Imparte`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Imparte` (
  `Asignatura_id_Asignatura` INT NOT NULL ,
  `Profesor_DNI` VARCHAR(10) NOT NULL ,
  PRIMARY KEY (`Asignatura_id_Asignatura`, `Profesor_DNI`) ,
    INDEX  `fk_Asignatura_has_Profesor_Profesor1`  (`Profesor_DNI`
ASC) ,
               INDEX       `fk_Asignatura_has_Profesor_Asignatura1`
(`Asignatura_id_Asignatura` ASC) ,
  CONSTRAINT `fk_Asignatura_has_Profesor_Asignatura1`
    FOREIGN KEY (`Asignatura_id_Asignatura` )
    REFERENCES `dbmovitest`.`Asignatura` (`id_Asignatura` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Asignatura_has_Profesor_Profesor1`
```

```
    FOREIGN KEY (`Profesor_DNI` )
    REFERENCES `dbmovitest`.`Profesor` (`DNI` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `dbmovitest`.`Alumno_has_Examen`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `dbmovitest`.`Alumno_has_Examen` (
  `Alumno_DNI` VARCHAR(10) NOT NULL ,
  `Examen_id_Examen` VARCHAR(10) NOT NULL ,
  `nota` FLOAT NULL ,
  `respuestas` VARCHAR(45) NULL ,
  `fecha` DATE NULL ,
  PRIMARY KEY (`Alumno_DNI`, `Examen_id_Examen`) ,
  INDEX `fk_Alumno_has_Examen_Examen1` (`Examen_id_Examen` ASC) ,
  INDEX `fk_Alumno_has_Examen_Alumno1` (`Alumno_DNI` ASC) ,
  CONSTRAINT `fk_Alumno_has_Examen_Alumno1`
    FOREIGN KEY (`Alumno_DNI` )
    REFERENCES `dbmovitest`.`Alumno` (`DNI` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Alumno_has_Examen_Examen1`
    FOREIGN KEY (`Examen_id_Examen` )
    REFERENCES `dbmovitest`.`Examen` (`id_Examen` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Tabla 3: Código SQL generado por SQL Workbench

Este código lo insertamos a través de una terminal en el servidor de mySQL y ya tenemos las tablas creadas. Sabemos que la MIP creará un registro en la tabla "Exámen" cuando un profesor genere un test; cada alumno creará un registro en la tabla "Alumno_has_Examen" cuando termine su examen con su app Mobitest; a través de la aplicación web podemos insertar registros en las tablas "Alumno", "Profesor" y "Asignatura". A medida que vayamos avanzando en la exposición de como ha sido desarrollado el proyecto veremos como hace cada una de esas inserciones.

5.1.2 Mesa Interactiva del Profesor

Aunque el desarrollo de las aplicaciones se ha ido solapando explicamos de forma concreta el desarrollo de la MIP utilizando el desarrollo en espiral que ya hemos explicado en que consistía antes. En la primera etapa afrontaremos el desarrollo de una primera versión de la MIP que solo traiga la opción de "iniciar cuestionario" que es la única de las opciones de la MIP que interactúa con la app MobiTest y así pondremos en funcionamiento parte del protocolo de comunicación (excepto la parte que afecta a la BBDD).

Comencemos con la primera etapa:

- Objetivos: crear una primera versión de la Mesa Interactiva del Profesor que permita al menos que se se conecte la app móvil, que se intercambien información. Será necesario que la MIP le envíe el examen y que el la app móvil le responda con la cadena de respuestas y con la información necesaria.
- Evaluación de riesgos y alternativas: el protocolo de información ya esta creado, ahora debemos respetarlo y decidir de que forma enviaremos el examen desde la MIP.
- Desarrollo y prueba: el inicio con la MIP es familiarizarme con el uso de NetBeans y crear una interfaz gráfica con dos botones, uno que active el "servidor de exámenes" y

otro que lo detenga. El botón de “iniciar” crea un ServerSocket al que luego se podrán atar los distintos móviles. En el momento que uno se ata, iniciamos el envío del archivo .xml en el que se encuentran almacenadas las preguntas del examen en lineas (leo una linea del fichero y la envío a través de un buffer). Luego espera que el alumno finalice su test y la aplicación reenvíe sus respuestas, las pasa por un corrector y obtiene la nota. Cuando el usuario pulsa detener, se cierra el socket.Planificación de la siguiente etapa: completar la MIP con las demás funciones.

La segunda etapa (otra vuelta a la espiral) añadirá funcionalidad a la MIP:

- Objetivos: creación de otras funciones necesarias en la Mesa Interactiva del Profesor. Estas funcionalidades son “Crear cuestionario”, “Ver test” y “Ver Resultados”. Además, debemos agrupar la funcionalidad ya creada en “Iniciar Cuestionario”.
- Evaluación de riesgos y alternativas: debemos crear una pantalla de menú para ofrecer acceso a todas las funcionalidades, dentro de cada una de ellas los riesgos son pocos y las alternativas son resueltas en cada una de ellas. “Crear cuestionario” debe permitir guardar las preguntas y posibles respuestas en formato .xml desde la aplicación. “Ver test” debe permitir mostrar las preguntas y respuestas que contiene un examen seleccionado por el usuario. Por último, “Ver Resultados” que debe mostrar gráficas y notas sobre las respuestas de los alumnos.
- Desarrollo y prueba: creamos un jFrame que contenga botones para acceder a cada una de las funcionalidades (cada una con su imagen), hemos añadido un fondo, por medio de un jPanel. Para “crear cuestionario” capturamos los textos introducidos en cuadros de texto (jTextField) para cada una de las respuestas y el enunciado y tras pulsar el botón de finalizar test o de siguiente pregunta volcamos los textos en un archivo .xml añadiendo las etiquetas. En “Ver Test” ofrecemos en un desplegable los nombres de los archivos a través de su código de examen desde la carpeta en la que estamos guardando, el usuario selecciona y cuando pulsa el botón OK se abre otra pantalla que muestra las preguntas y respuestas del examen. “Ver Resultados” muestra varias gráficas sobre cuales son las respuestas más elegidas y el porcentaje de aprobados con ayuda de la librería jFreeChart, también las notas, tanto de un alumno en todos sus exámenes como los de todos los alumnos en un examen.
- Planificación de la siguiente etapa: en la siguiente etapa hay que incorporar la base de datos. La MIP se comunicará con ella.

Veamos como fue la etapa siguiente:

- Objetivos: incorporar la base de datos al sistema.
- Evaluación de riesgos y alternativas: crearemos una clase que tenga funciones para comunicarse con la BBDD. Las funciones deben dar solución a los siguientes problemas:
    - Si existe un alumno con un DNI y password determinado en la base de datos.
    - Si existe un profesor con un DNI y password determinado en la BBDD.
    - Introducir un examen en la BBDD con su código, su cadena de respuestas correctas, el profesor que lo ha creado, etc
    - Insertar en la tabla "Alumno_has_Examen" cuando un alumno acaba su examen los datos (nota, respuestas, alumno y examen).
    - Dar el número de aprobados que hay en un determinado examen.
    - Averiguar cuales son los exámenes creados por un determinado profesor.
    - Ofrecer cuales son las asignaturas que imparte un profesor, de las cuales puede generar exámenes.
    - Saber cuales son las respuestas correctas de un examen concreto.
    - Saber cuales son las notas de un alumno en todos sus exámenes o saber las notas de todos los alumnos para un determinado examen.
    - Saber cuales son las respuestas dadas por todos los alumnos para un determinado examen.
    - Sin tener que consultar a la BBDD también guardaremos la MAC del dispositivo móvil para evitar que desde un dispositivo se realice varias veces el mismo test. Debemos responder de forma negativa a la petición de examen si esto sucediera
- Desarrollo y prueba: creamos una función para cada una de las necesidades reflejadas anteriormente, cada función ejecuta una instrucción SQL (query), gracias a la librería que ofrece java (java.sql). Esta librería permite crear querys para BBDD (SELECT, INSERT...), rellenar posteriormente los campos que dejes incompletos (hay que

dejarlos con “?”) y ejecutarlas. Además de tratar posteriormente el resultado de ejecutar estas querys en un ResultSet.

- Planificación de la siguiente etapa: en la siguiente etapa debemos resolver la forma de la cual la app móvil debe saber cual es la IP a la que debe atarse, es decir, en que IP y puerto se encuentra la MIP y atarse ya que hasta ahora lo estaba haciendo a una fija.

A continuación, explicaremos como fue la ultima etapa en la que intervienen la MIP y MobiTest, ya que después solo quedará realizar la web que facilitará las inserciones en la BBDD:

- Objetivos: Decidir de que forma MobiTest encontrara la IP y el puerto al que debe atarse, donde estará la MIP e implementarla.
- Evaluación de riesgos y alternativas: como ya evaluamos en la parte de la aplicación MobiTest como resolver esta disyuntiva aquí solo debemos ofrecer la posibilidad de saber en todo momento la IP en la que se encuentra la MIP para que el profesor se la pueda transmitir a los alumnos.
- Desarrollo y prueba: creamos un botón “Configuración” que muestre la dirección IP de esta máquina, el puerto es fijo (5000) por lo que con la IP es suficiente.
- Planificación de la siguiente etapa: lo próximo que debemos hacer es crear la aplicación MobiTest de la que ya tenemos definido el protocolo de comunicaciones con anterioridad pero además debe coincidir con lo creado en la MIP.. Lo comentaremos en el próximo apartado aunque es el siguiente paso natural del desarrollo.

5.1.3 MobiTest

En este apartado contaremos como ha sido el desarrollo de la aplicación móvil en Android. Para iniciarnos en el desarrollo en Android se realizaron actividades formativas como cursos, talleres y estudio de la documentación. Una vez familiarizados e introducidos en Android podíamos afrontar con garantías el desarrollo de la app.

Contaremos cómo fue el desarrollo de MobiTest que también utiliza el método en espiral. Ésta próxima vuelta de espiral, la primera que afrontaba una parte del desarrollo de la app MobiTest se definió del siguiente método:

- Objetivos: crear una versión "mini" de MobiTest, que lea el examen de un archivo que esta guardado en la memoria del teléfono (en la versión final es enviado desde la MIP) y genere el mensaje que se debe enviar a la MIP, con mi identificación y la cadena de respuestas notas (en esta versión no la envía porque todavía no hay conexión con la MIP).
- Evaluación de riesgos y alternativas: hay que definir el protocolo de comunicación (ver anexo 3) de la app con la MIP (los mensajes que se van a enviar) y en que tipo de archivo irá guardado el examen. Nos decidimos por formato .xml ya que existían parseadores que nos podían ayudar a volcar el examen de un dispositivo a otro.
- Desarrollo y prueba: lo primero que hicimos en esta etapa fue determinar el formato del archivo .xml en el que se guardará el examen, es decir, debemos definir las etiquetas y los atributos que llevará (ejemplo en el anexo 4). Una vez definimos el protocolo de comunicación (como podemos ver en el anexo 3) y el formato del .xml comenzamos a parsear el archivo .xml. Creamos un hilo para parsear el archivo y con las instrucciones que ofrece la librería "XmlPullParser". Esta librería tiene instrucciones para obtener los contenidos, atributos y los nombres de las etiquetas por lo que ha sido de gran ayuda.

Una vez parseado el texto de las preguntas (enunciado y posibles respuestas), las guardamos en una instancia de la clase DBPreguntas que es un ArrayList de Question, la clase Question.java es una clase creada por mí que consta de enunciado y las 4 posibles respuestas (enum, resp1, resp2, resp3, resp4) almacenadas en String. De este ArrayList vamos volcando las preguntas a actividades donde mostrarlas y el alumno podrá responder una de las cuatro opciones, la actividad que contiene la pregunta y las posibles respuestas la llamamos Pregunta.java. Lo único que tenemos que hacer es ir recogiendo el DNI de la primera actividad e ir rellenando un mensaje final con una cadena de respuestas con la opción escogida por el alumno. Cuando el alumno responde a la ultima pregunta ya tenemos formado el mensaje final que debe ser enviado para corregir sus respuestas (la corrección la realizará la MIP).

Las actividades se van enviando información de unas a otras, además del mensaje final

que se va llenando así:

- Actividad inicial pasa a actividad pregunta 1 el DNI.
- Pregunta 1 pasa a pregunta 2: DNI;resp1, (donde resp1 será a, b, c o d).

- Pregunta n-1 pasara a pregunta n: DNI;resp1,resp2...respN-1Pregunta n a final (llamado Form.java): DNI;resp1,resp2...respN-1,respN. Y Form.java se encargará del envío.

Otra información es necesaria enviar de actividad a actividad, como el numero de orden de pregunta (para saber cuando llego a la última y lanzar Form.java).

Al final de esta primera versión este es el aspecto de la app. Ilustración 12 y Ilustración 13

Ilustración 13: Pantalla inicial (primera versión)

Ilustración 12: Ejemplo de pregunta (primera versión)

- Planificación de la siguiente etapa: añadiremos la conexión de esta app con la MIP, que descargue el examen desde la MIP y que interactúe con ella de forma concreta.

Comenzamos la siguiente etapa:

- Objetivos: que la app MobiTest interactúe con la MIP, que esta en sus primeras versiones pero ya permite que la app se conecte a ella y el envío de exámenes.
- Evaluación de riesgos y alternativas: hay que ver como nos atamos a la MIP, en estas

primeras versiones la IP y el puerto es fijo por lo que no hay confusión, más tarde evaluaremos como saber donde esta la MIP. También veremos como responde la app a parsear un archivo que esta en la memoria del teléfono.

- Desarrollo y prueba: la app se ata a una IP y puerto donde la MIP ha creado un ServerSocket, recibe el asentimiento y los paquetes que forman el archivo .xml donde se guarda el examen. Tras almacenar estos paquetes ya puede parsear el archivo, hemos de cambiar la ruta del archivo a parsear, ahora es donde lo hayamos guardado. Lo probamos y lo resuelve sin problemas.
- Planificación de la próxima etapa: la MIP habrá incorporado la base de datos por lo que ya debemos enviarle nuestro usuario y contraseña para que las compruebe.

Siguiente etapa:

- Objetivos: debemos aumentar la seguridad exigiendo un login antes de enviar el examen.
- Evaluación de riesgos y alternativas: ese aumento de seguridad lo realizaremos añadiendo nuevos campos, que son DNI y password. Sólo enviará ese examen si nuestros datos son correctos, si no es así, debemos controlar las otras posibles respuestas. También debimos registrar la MAC del móvil para que un mismo dispositivo no pueda realizar dos veces el mismo test.
- Desarrollo y pruebas: Añadimos el campo de password que no estaba y al pulsar el botón de “Enviar” debemos comprobar si están los campos rellenos de forma adecuada y si es así, atarnos a la IP y puerto correctos y enviarle esa información. Hay que conseguir tener contempladas todas las respuestas posibles, si el usuario o password son incorrectos lo debemos controlar y avisar y si todo es correcto debemos estar preparados para recibir el examen y parsearlo como hacíamos con anterioridad. En ese momento todo fluirá como antes de este paso. También la MIP nos puede avisar de que ese dispositivo ya ha realizado este test, en ese caso no debemos dejarle continuar,
- Planificación de la siguiente etapa: debemos resolver el problema de la IP fija y debemos mejorar el aspecto de la app.

Última etapa:

- Objetivos: ante la posibilidad de que la MIP no esté siempre montada en la misma IP

debemos ser capaces de al menos cambiar la IP a la que la app móvil debe atarse sin tener que volver a instalarla. También debemos darle un mejor aspecto.

- Evaluación de riesgos y alternativas: una de las posibles soluciones era hacer un barrido a la red en la que nos encontrábamos, es decir, como la MIP se iba a encontrar en la misma red local que los dispositivos móviles manteniendo las tres primeras cifras de nuestra dirección IP hacer un barrido en la última (X.X.X.0,...,255) pero la solución era compleja y lenta. La otra opción es crear un botón de ajustes y ahí poder añadir una IP distinta si fuese necesario. Con respecto al cambio de aspecto, adoptamos un tema relacionado con los exámenes por lo que un fondo que asemeje a una hoja de papel será adecuado.
- Desarrollo y pruebas: habilitamos el botón menú del que saldrá un desplegable con el botón de ajustes. Ese botón de ajustes nos manada a otra actividad (ScreenSettings) y ahí pedimos una nueva IP, guardando la anterior. También hemos añadido una imagen de una hoja de papel como fondo en todas las actividades. También hemos redistribuido los botones agrupándolos en la parte baja de la actividad para mejorar la legibilidad de las preguntas.
- No es necesario planificar más etapas ya que hemos concluido.

5.1.4 Aplicación web asistente de BBDD

A continuación, mostraremos como fue esta etapa que también resulta ser la última de todas ya que el funcionamiento de todos los demás módulos del proyecto funcionan perfectamente y su interrelación es completa:

- Objetivos: crear las paginas web que faciliten la inserción de registros en la BBDD.
- Evaluación de riesgos y alternativas: lo único que debemos elegir es con que lenguaje hacemos las webs, lo haremos con HTTP (Hypertext Transfer Protocol) y PHP. Estos lenguajes permiten la interacción con BBDD. Debe permitirnos introducir alumnos, profesores y asignaturas en la BBDD..
- Desarrollo y prueba: creamos una pagina con un desplegable en la que podemos elegir que es lo que deseamos hacer. Luego haremos una pagina para cada opción pero en el

fondo todas hacen lo mismo, ofrecen un formulario, recogen los datos que el usuario introduce y al pulsar el botón de “Enviar Formulario” ejecuta un código php que se conecta a la base de datos y ejecuta la query respectiva (INSERT, DELETE...). La instrucciones usadas son “mysql_connect”, “mysql_db_query” y “mysql_query”.

- Planificación de la siguiente etapa: no es necesario planificar más etapas ya que hemos concluido. En el apartado de conclusiones y lineas de futuro comentaremos de que modo se podría avanzar con estas aplicaciones.

6. Resultados

Es un proyecto de muy bajo coste, tanto en recursos humanos como en recursos materiales. La instalación es muy sencilla y no es necesario soporte externo para ninguna de las tareas. Una de las claves en su bajo coste es la no necesidad de añadir nuevos dispositivos, los alumnos con sus propios dispositivos *Smartphones* pueden responder a los test, es decir, no es necesario desplegar una costosa infraestructura cada vez que se necesite usar. Además en estas gráficas vemos cual es la cantidad de *Smartphones* de alumnos en carreras relacionadas de algún modo con estos dispositivos (Ingeniería de Telecomunicaciones) como en otras que no lo están (Grado en Ciencias Medioambientales).

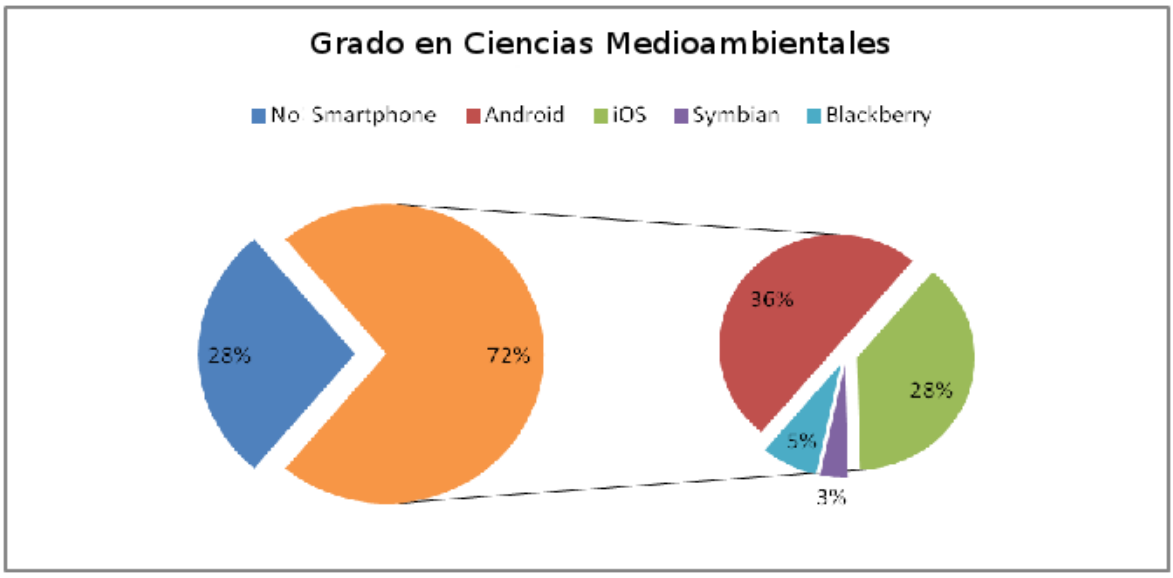

Ilustración 14: Gráfica de dispositivos en Ciencias Medioambientales

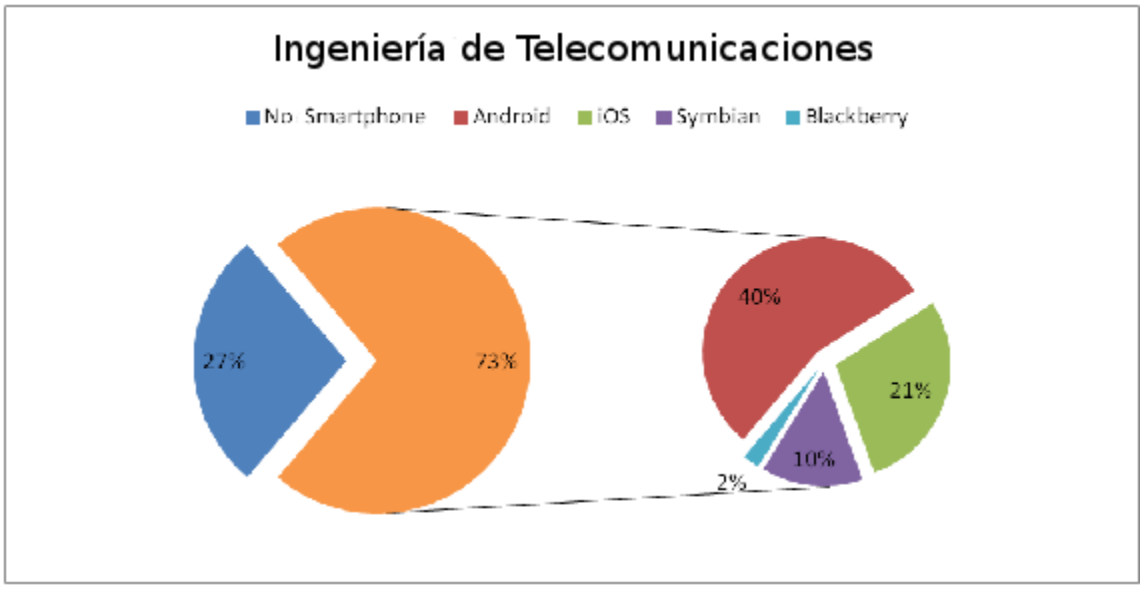

Ilustración 15: Gráfica de dispositivos Smartphones en Ing. Telecomunicaciones

Este sistema permite el incremento de la motivación de los alumnos al posibilitar combinar la enseñanza tradicional con las nuevas tecnologías, representadas en los teléfonos inteligentes usados.

El sistema de test interactivos es muy sencillo de usar y su corrección automática permite a los profesores aumentar su cifra de exámenes hasta hacerlos de forma semanal o incluso diaria. También permite alcanzar nuevos niveles de "*feedback*" o retroalimentación entre profesores y alumnos, por ejemplo, permite conocer la cantidad de alumnos que han comprendido un difícil tema que se acaba de explicar justo después de hacerlo, planteando un rápido test y observando los resultados y si fuese necesario volviendo a explicar el tema.

En cuanto a la motivación de los alumnos, que test prácticamente diarios se reflejen en sus notas finales les ayudará a estudiar con mas constancia o conocer los resultados de los test y las respuestas correctas de los mismos les ayudará al aprendizaje conociendo cuales son sus errores.

Una versión previa de estas aplicaciones fueron probadas experimentalmente en varios contextos de la Universidad Rey Juan Carlos (URJC). Fue validado en sujetos de varias titulaciones, algunas de ellas muy relacionadas con las TIC (telecomunicaciones y distintos grados relacionados con la electrónica y la informática) pero otras titulaciones en las que las TIC son sólo herramientas y en las que su conocimiento no es un requisito indispensable para el uso de estas (Ingeniería Química, Ciencias Medioambientales). Estos resultados nos hacen

pensar que este sistema funcionará muy bien en colegios e institutos donde la penetración de los dispositivos móviles es mayor debido a la juventud de los sujetos.

El sistema fue probado también en las "II Jornadas de Electrónica, Optoelectrónica y Electromagnetismo" que tuvo lugar durante los días 1 y 2 de Junio de 2011. En esos días se probó el sistema con 72 alumnos de forma simultanea y se descubrió que el sistema no perdía tiempo en la recogida de preguntas y respuestas.

En esos test los participantes en las jornadas respondían con su propio dispositivo preguntas sobre las charlas previas a las preguntas que eran mostradas en un proyector. En la nueva versión ya hemos visto que las preguntas viajan también al dispositivo. Las respuestas eran recibidas en la MIP y tratadas para mostrar gráficas sobre las respuestas.

La segunda vez que se probó este sistema en un Grado en Ingeniería de Telecomunicaciones 42 de 68 alumnos disponían de *Smartphone.* Otro test académico se realizó en la carrera de Ciencias Medioambientales, en la que 52 de 72 alumnos disponían de *smartphones*. Comprobamos que test que tardarían mas de una hora en realizarse del método tradicional y sin tener en cuenta la corrección, estaban concluidos en 10 minutos y sus resultados almacenados en bases de datos.

Las experiencias adquiridas por profesores del departamento tras el uso de la MIP y este sistema de evaluación con MobiTest nos demuestran que solo la realización de un test que necesite de 15 minutos para resolverse se invierte alrededor de una hora y media sin corregirlo ni recogerlo que se reparten del siguiente modo:

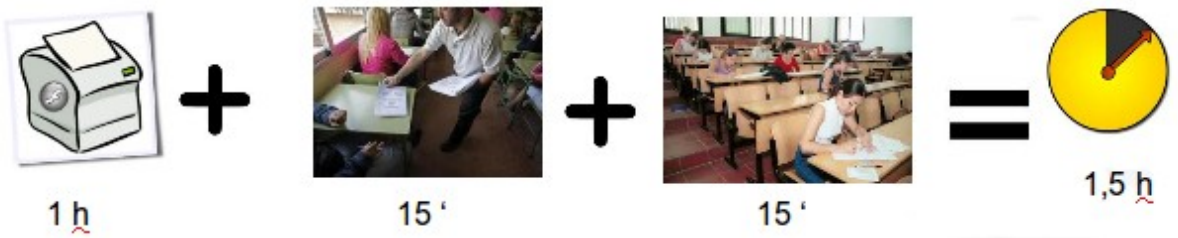

Ilustración 16: Tiempo sin uso de MIP y MobiTest

Sin embargo, con el uso de MobiTest y la MIP no es necesario imprimir test, ni repartirlos. Por tanto, con MobiTest y la MIP el tiempo e reduce considerablemente y, además, no es necesario recoger los test ni corregirlos por lo que el ahorro de tiempo se multiplica.

Ilustración 17: Tiempo empleado con MIP y MobiTest

Si además, contamos el tiempo estimado en corregir cada uno de estos exámenes, unido al tiempo de publicación el método tradicional puede tardar varios días o incluso semanas mientras que en este nuevo método la corrección y la publicación son automáticas.

Este sistema previo registró problemas de seguridad, en cuanto a accesos a los test de sujetos no registrados o sin permiso de los profesores, identificaciones falsas o copias de varios alumnos. Estos problemas han sido resueltos en el proyecto que aquí se presenta.

El orden en que se muestran las preguntas es aleatorio para cada uno de los dispositivos que reciben el examen para así evitar las copias. El dispositivo queda registrado para no poder realizar un mismo alumno su examen y el de un compañero. Y por último el alumno debe estar registrado de forma previa en el sistema con usuario (DNI) y contraseña, así evitaremos que entren a realizar el test usuarios no registrados o usuarios falsos.

El sistema nuevo mejora al previo en aspectos de seguridad principalmente, añade nuevas funcionalidades en la app móvil(como que las preguntas de los test aparezcan en el dispositivo) y añade toda la funcionalidad a la MIP que antes solo podía recoger los resultados y ahora puede configurar nuevos test y gestionar estos nuevos test junto con las notas de los alumnos. Además, al añadir la base de datos evita la volatilidad de los datos recogidos.

Y todas estas mejoras sin perder nada de las virtudes de sistemas anteriores, principalmente que funciona perfectamente en tiempo real.

La experiencia no solo ha sido satisfactoria para los alumnos si no también para profesores. Dos profesores del departamento probaron la MIP y quedaron encantados con el interfaz fácil e intuitivo del que consta la aplicación. Además contamos con una pequeña guía de uso que nos será de utilidad en los primeros pasos hasta que nos familiaricemos con la aplicación.

6.1 Guía de uso

A continuación, explicaremos el funcionamiento de las aplicaciones de las que consta el proyecto, comencemos por la Mesa Interactiva.

6.1.1. Mesa Interactiva del Profesor

Cuando iniciamos la aplicación nos aparece una pantalla de login.

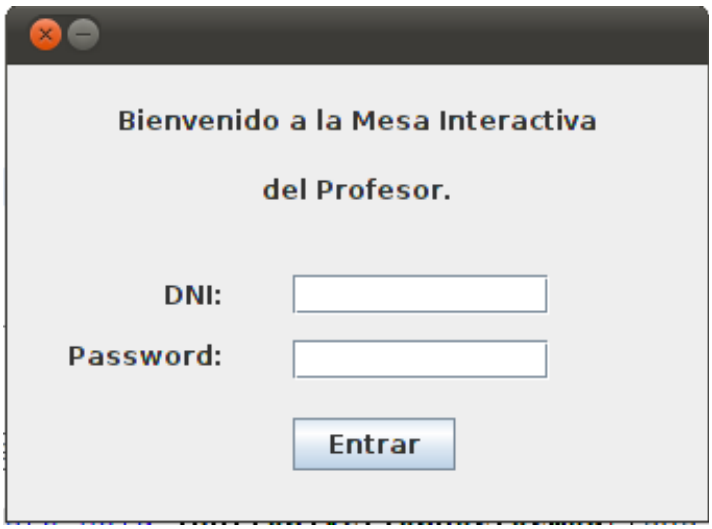

Ilustración 18: Pantalla de login de la MIP

Deberemos estar dados de alta en el sistema como profesor, introducimos el DNI y la contraseña y entramos al menú principal (Ilustración 19). Este menú tiene cinco opciones: “Generar nuevo test”, “Ver Test”, “Iniciar Cuestionario”, “Ver resultados” y “Salir”. Además contamos con un botón de configuración donde podremos ver la IP en la que se encuentra alojada la Mesa Interactiva. Es lo primero que debemos ver si queremos realizar un nuevo test, ya que debemos facilitarle esta dirección IP a los alumnos que vayan a realizar el test.

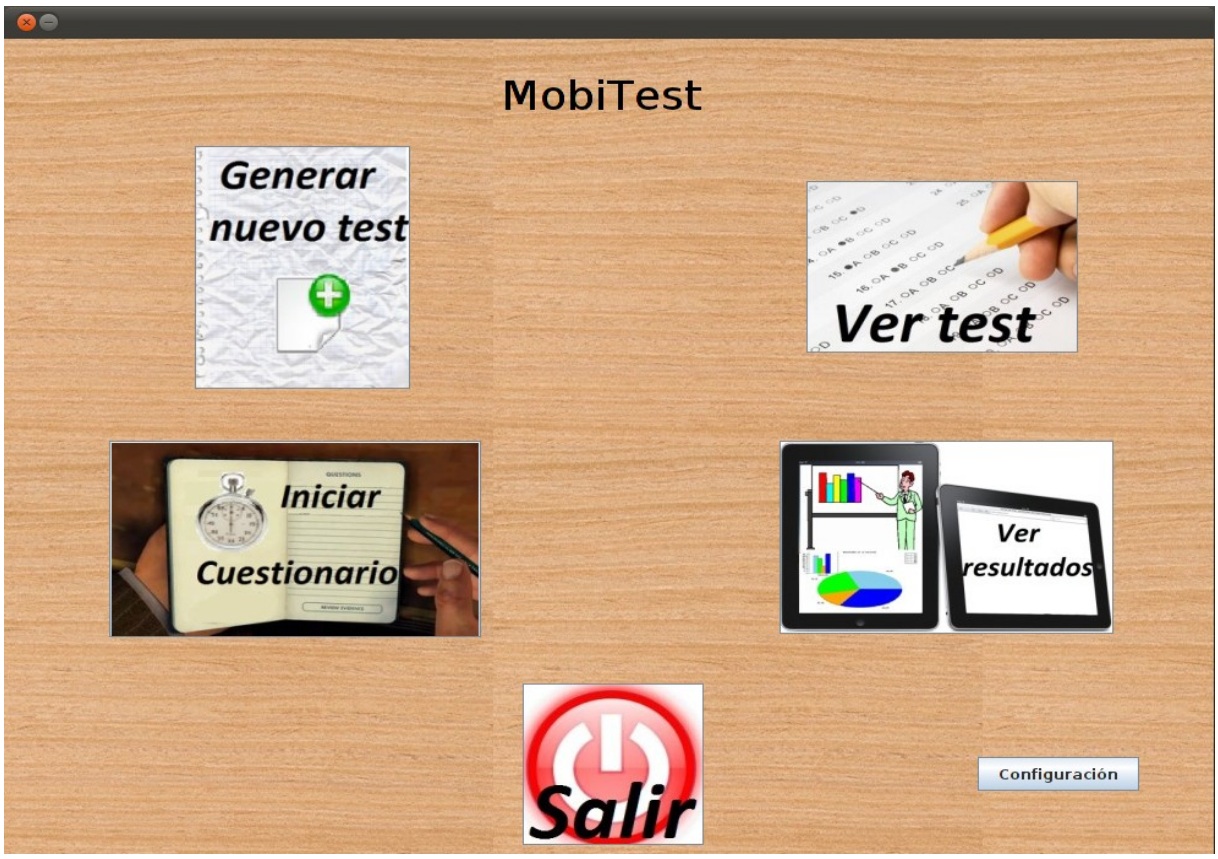

Ilustración 19: Menú principal MIP

Ahora iremos explicando uno a uno para que sirve cada botón:

- Generar nuevo test: En esta parte del programa podremos crear un nuevo test, lo primero que debemos hacer es elegir el código del test y elegir a que asignatura pertenece completando los campos que aparecen en la Ilustración 20.

Ilustración 20: Nuevo test

Una vez rellenados los campos, pulsamos aceptar y se abrirá una pantalla en la que escribiremos el enunciado, las posibles respuestas y a la única respuesta correcta la seleccionaremos con un tick. Si la pregunta que estamos redactando es la última pulsaremos "Finalizar Test", si queremos escribir más pulsaremos "Siguiente pregunta". (Ilustración 21).

Ilustración 21: Pantalla donde rellenar la pregunta

- Ver test: Aparecerá un desplegable con todos los exámenes que han sido elaborados en el equipo. Tan solo elegiremos uno pulsaremos "Ver Examen" para ver las preguntas y respuestas.(Ilustración 22).

Ilustración 22: Ver tests

- Iniciar Cuestionario: esta parte es la que va relacionada directamente con la aplicación móvil. Como hemos avisado antes, debemos consultar la IP donde esta alojada la Mesa Interactiva en el botón de "Configuración" en la pantalla inicial. Tras esto, debemos seleccionar el test que deseamos que los alumnos contesten y pulsar "iniciar". En ese momento, cuando un alumno entre en su aplicación del móvil el examen que se enviará a su móvil será el elegido por el profesor. Desde esta pantalla también podemos ver el test como en la opción anterior. Solo podemos ver test cuando el test no este en marcha, una vez se inicie las opciones se deshabilitan, aparece un mensaje indicando a la hora que comenzó el test. Cuando creamos que ha pasado el tiempo suficiente para la realización del examen pulsaremos "Detener". (El mensaje de inicio puede cerrarse cuando se desee). Ilustración 23.

Importante: La aplicación móvil fallara mientras el test no este iniciado.

Ilustración 23: Iniciar cuestionario

- Ver Resultados: en este amplio apartado podemos ver una gráfica con el porcentaje de aprobados, otra que refleja las respuestas elegidas por los alumnos en las distintas respuestas de cada pregunta y las notas, tanto las de un alumno en los diferentes exámenes como los de todos los alumnos para un determinado examen. (Ilustración 24).

Ilustración 24: Ver resultados

Vamos con ello: tras elegir “ver aprobados”, elegiremos un test en el desplegable y al pulsar “Ver” se mostrará la gráfica (Ilustración 25).

Ilustración 25: Ver aprobados

“Ver respuestas” nos permite elegir un determinado examen, tras pulsar “ok” con el examen elegido podemos elegir sobre que pregunta queremos ver la gráfica, entonces pulsaremos “ver” y la gráfica se mostrará. (Ilustración 26).

Ilustración 26: Ver respuestas

En el apartado “Ver notas” veremos las notas de todos los alumnos que han realizado el examen escogiendo la primera opción o podremos ver las notas de un alumno en todos los exámenes introduciendo su DNI y pulsando el botón “Ver” inferior. (Ilustración 27).

Ilustración 27: Ver notas

6.1.2. MobiTest

A continuación vamos a explicar el modo de empleo de la app MobiTest. La app se iniciará en una pantalla inicial donde nos pedirá nuestro DNI y una contraseña. En esa pantalla podemos pulsar el botón MENÚ, pulsar ajustes e introducir la IP que de forma previa nos ha facilitado el profesor, la aplicación se conectará a esta IP que es donde se encuentra emplazada la Mesa Interactiva del profesor.

Ilustración 28: Pantalla inicial MobiTest

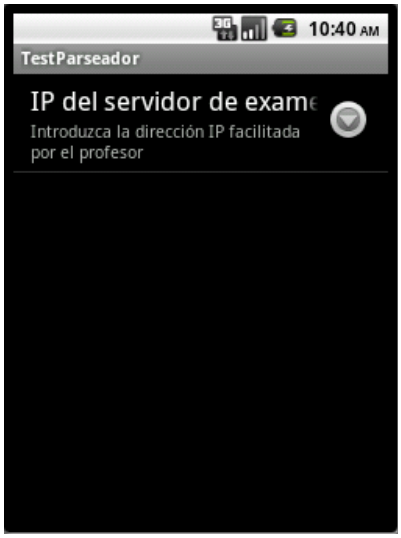

Ilustración 29: Ajustes MobiTest

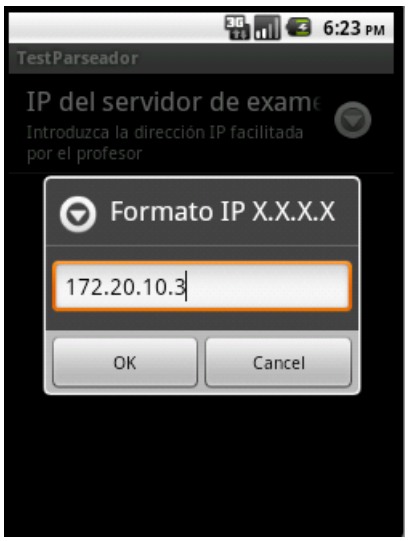

Ilustración 30: Selección IP

Tras comprobar que la IP es la correcta o cambiarla a la adecuada podemos introducir nuestro DNI y nuestra contraseña, es importante que el profesor haya pulsado ya la opción "Iniciar" dentro de "Iniciar examen", si no es así aparecerá un mensaje de error y se cerrara la aplicación. Otros errores se pueden producir derivados de la verificación del usuario, como por ejemplo que el usuario no exista con esa contraseña, que ya haya realizado el examen o que ese dispositivo ya haya sido utilizado para realizar ese examen.

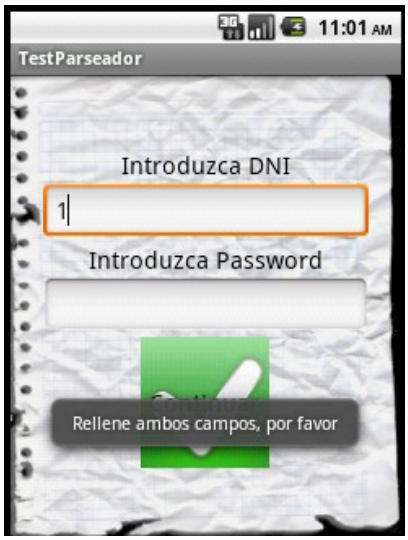

Ilustración 31: Error 1 MobiTest

Ilustración 32: Error 2 MobiTest

Ilustración 33: Error 3 MobiTest

Si todo es correcto, aparecerá la primera pregunta, tras pulsar uno de los botones de respuesta de la parte inferior van apareciendo las sucesivas hasta llegar a la pantalla en la que nos indica que el test ha finalizado. Pulsamos salir y se guardan nuestros resultados.

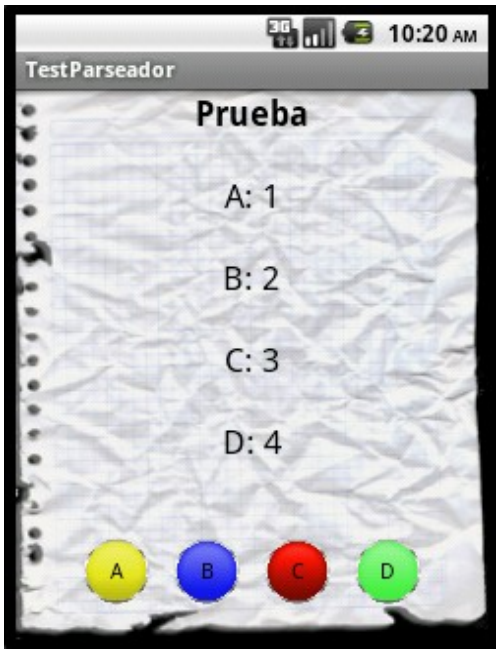

Ilustración 34: Pregunta MobiTest

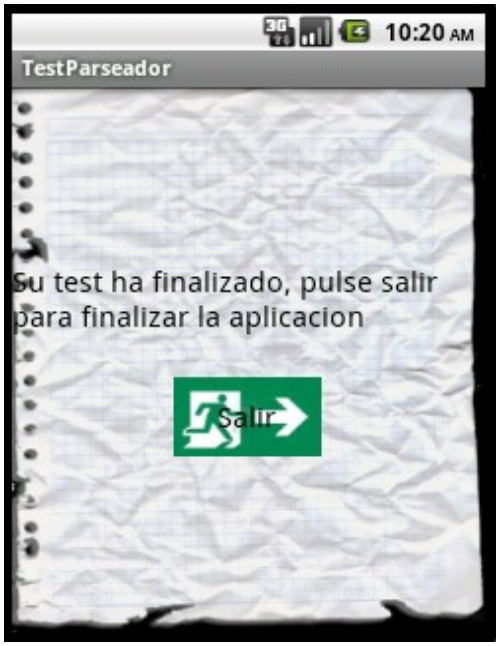

Ilustración 35: Final MobiTest

Todas estas aplicaciones dependen de una base de datos en las que hay que estar dado de alta. Como profesor para utilizar la mesa interactiva y como alumno para MobiTest. Estos registros los llevará a cabo un superusuario que contará con la ayuda de una aplicación web para introducir y borrar registros (profesores, alumnos y asignaturas) en la base de datos.

6.1.3 Aplicación web para la creación y borrado de usuarios

La finalidad de esta web es facilitar lo que serian sentencias de acceso a bases de datos, se utilizará para dar de alta y baja alumnos, profesores y asignaturas.

En la pantalla inicial elegiremos de un desplegable cual es la operación que deseamos realizar, y pulsaremos “Entrar”.

Ilustración 36: Pantalla inicial aplicación web

Ilustración 37: Desplegable aplicación web

Para crear un profesor que pueda utilizar la mesa interactiva es tan sencillo como rellenar los campos y pulsar "Enviar formulario". En la opción borrar profesor será suficiente con que pongamos su DNI.

El mecanismo es igual para alumnos y asignaturas.

Ilustración 38: Crear alumno

Ilustración 39: Borrar alumno

Una opción un poco mas especial será "Añadir Asignatura Cursando" que será utilizada cuando a un alumno que ya esta dado de alta en el sistema queremos añadirle una asignatura que cursa. Rellenaremos los campos de DNI, ID Alumno y Código de Asignatura. El código de asignatura será asignado al crearla y luego podemos comprobar cual es el código de cada asignatura desde la mesa interactiva en el apartado "Generar Nuevo Test".

Ilustración 40: Añadir asignatura a un alumno

7. Conclusiones y Líneas de Trabajo

Por último, dedicamos este capítulo a desarrollar las conclusiones finales del trabajo realizado y efectuamos un estudio exhaustivo sobre las posibles líneas futuras pensadas para mejorar la aplicación y para añadir nuevas funcionalidades al sistema.

7.1 Conclusiones

En apartados anteriores vimos cómo la introducción del Plan Bolonia con su aplicación de la evaluación continua genera un problema sobre todo de tiempo, tanto a profesores como a alumnos que hemos tratado de solventar por medio de las TICs. Hemos pensado y desarrollado una solución de bajo coste que reduce considerablemente los tiempos invertidos por unos y por otros en la evaluación continua. Usando la plataforma propuesta para la evaluación continua dispondremos de unos tests que permitirán calificar los conocimientos de los alumnos en diferentes aspectos de las asignaturas. Además, esta solución ayuda en la generación de la información derivada de estos test y facilita compartirla mejorando la comunicación entre profesores y alumnos mejorando la enseñanza y el aprendizaje.

Una vez finalizado el proyecto y observando los resultados obtenidos podemos concluir que hemos ofrecido una gran solución al problema. Hemos utilizado dispositivos Android del cual disponen un gran porcentaje de alumnos lo cual les ayudará en la integración y participación en las clases. Otra de las ventajas que no debemos olvidar es que al ser dispositivos tan populares no es necesario ningún despliegue adicional para su uso en las aulas. En cuanto al uso de la MIP (Mesa Interactiva del Profesor), es un software gratuito y multiplataforma que podrá estar instalado en el ordenador personal del profesor o en el perteneciente al aula.

Esta solución será adoptada por algunos de los profesores del departamento para su uso en las clases que imparten. Todo aquel profesor que use el sistema verá el tiempo invertido en crear, ofrecer al alumno, gestionar y corregir sus test reducido. También

comprobaremos que sus alumnos se sentirán con un mayor grado de integración y participación en las clases lo que será positivo para ambos.

En cuanto al desarrollo, la programación para Android esta basado en java por lo que es muy parecido, sólo que tiene sus propias funciones para determinadas cosas. La realización del proyecto, me va enseñando cosas poco a poco sobre Android pero finalmente me permite realizar una aplicación muy completa y haber descubierto la infinidad de posibilidades que permite Android.

Este proyecto también me ha permitido introducirme en el mundo de las interfaces gráficas de las cuales no había aprendido nada durante todos mis estudios. Al igual que en lo que respecta a Android los entornos de programación (Eclipse y NetBeans) ayudan al aprendizaje por la gran cantidad de facilidades que dan al desarrollador. Desde luego, personalmente he acabado encantado con los dos entornos.

También he aprendido a manejar código php, realmente este proyecto me ha aportado gran cantidad de conocimientos sobre la programación, sobre todo a nivel practico. Entendiendo que sin la base teórica adquirida en las distintas asignaturas el aprendizaje no hubiera sido tan rápido y grande se podría decir que durante este proyecto he aprendido tanto como durante los años previos.

A tenor de los resultados obtenidos podemos decir que los objetivos marcados se han cumplido creando una solución elegante al problema de la gran cantidad de exámenes que corregir e incorporando a los alumnos de una forma ágil a la participación en las clases. La solución adoptada incorpora el *smartphone*, que prácticamente se ha convertido en una prolongación de nuestro cuerpo, a las clases y así el alumno se siente más familiarizado con las herramientas utilizadas en las clases.

En resumen, este proyecto además de enseñar desde el punto de vista académico aporta cosas a nivel personal, desde un acercamiento al mundo laboral desarrollando un proyecto algo mas profesional hasta la toma de decisiones ante problemas reales.

7.2 Líneas de Trabajo Futuras

En cuanto a más funcionalidades las líneas de trabajo de futuro son extremadamente

abiertas, cualquier cosa que se pueda añadir, suponga solucionar un problema más y encaje en la línea en la que ha evolucionado el proyecto será adecuado, como por ejemplo acceder desde MobiTest a las notas del alumno, cosa que actualmente sólo se puede hacer desde la MIP. También se podría crear una versión móvil de la MIP con la mayor parte de las opciones de la original.

Otra de las lineas de trabajo futuras y posiblemente la mas obvia y la primera que se debería afrontar es conseguir convertirla en una aplicación multiplataforma, es decir, conseguir que MobiTest funcione para todas los dispositivos de forma simultanea (Android, iOS, Windows Phone, Blackberry, Symbian, etc.). No serán necesarios cambios en la MIP si se respeta el protocolo de comunicación, tratará a todos los dispositivos por igual.

Una de las mejoras más positivas será extrapolar las aplicaciones desde la red local a internet y permitir realizar exámenes desde cualquier lugar con una conexión a internet. Si el profesor pudiese abrir la MIP en un servidor fijo y la aplicación se conectara a ese servidor desde cualquier lugar se añadiría ubicuidad, unas de las grandes ventajas de los dispositivos móviles y aprovechada solo en parte en este proyecto. Si se consiguiese este objetivo en el futuro la ubicuidad sería total y la app ganaría consistencia de forma exponencial.

Nuestras líneas de futuro exploran otras vías, por ejemplo, introducir niveles de dificultad tanto en exámenes como en preguntas, esto podría derivar en generación automática de exámenes indicando el nivel o niveles que queremos que tengan las preguntas. Esta introducción de niveles de dificultad podría también ayudarnos en la seguridad, evitando las copias ya que podríamos generar distintos test para distintos alumnos a la vez que tuviesen distintas preguntas con la misma dificultad. Añadir esta calificación por niveles de dificultad aportaría una mayor variedad en las funciones de nuestro proyecto.

8. Bibliografía

[1] http://www.educacion.es/dctm/boloniaeees/documentos/02que/declaracion-bolonia.pdf?documentId=0901e72b8004aa6a

[2] Datos recogidos de la web Fortune.com. Web activa el 4 de abril de 2012.

[3] Analysis Mason Consultora especializada en telecomunicaciones, tecnología y medios de comunicación.

[4] Información obtenida de un artículo aparecido en la web http://www.tendencias21.net/. Web activa el 4 de abril de 2012

[5] Fuente: Google (cifras publicadas en Octubre de 2011). Web activa el 4 de abril de 2012

[6] http://www.tendenciadigital.com.ar/sociedad/noticias/las-cifras-que-definiran-la-guerra-de-sistemas-operativos-para-smartphones.html

[7] Los datos del estudio son las ventas reales de la Tienda Teabla, distribuidor oficial de Movistar, situada en el CC. Islazul.

[8] http://www.queesbolonia.es

[9] http://www.planavanza.es

[10] http://moodle.org/

[11] www.campusvirtual.urjc.es/

[12] Productos ofrecidos en el número 2 (primavera 2011) de la revista “Educación 3.0”.

[13] Caracteristicas del eCuaderno de Additio en: http://www.ecuadernoprofesorado.com/

[14] Web de todos los productos de la familia MimioClassroom : http://www.mimio.dymo.com/es-EM/Products/MimioClassroom-Family-of-Products.aspx

[15] Web de LanSchool : http://www.lanschool.com/

[16] Web de la pizarra tactil de Panasonic: http://www.panasonic.es/

[17] Web del producto SmartTech: http://smarttech.com/

[18] Ejemplo de ayuda a la integración de alumnos con discapacidades. Fuente : El Mundo (Ver anexo 1)

[19] Barry Boehm: http://es.wikipedia.org/wiki/Barry_Boehm

[20] Parte de la información sobre el modelo en espiral ha sido obtenido de: http://www.slideshare.net/SadotHernandez1/modelo-en-espiral

[21] http://www.buzzthegame.com/es-es/

[22] Podemos descargar Ubuntu desde su web e instalarlo junto a nuestro SO o sustituirlo. www.ubuntu.com

[23] Web de descarga de eclipse: http://www.eclipse.org/downloads/

[24] http://developer.android.com/guide/developing/tools/index.html

9. Apéndices

Anexo 1: Metas y declaraciones del Plan Bolonia.

La Declaración de Bolonia plantea las siguientes metas:

- Reestructurar el sistema de enseñanza de acuerdo a tres niveles: un primer nivel, con el que se obtendría un título de grado que capacita para el acceso al mercado laboral, un segundo nivel, dirigido a la obtención del título de máster y un tercer nivel dirigido a la obtención del título de doctorado, donde el segundo y tercer nivel garantizaría una formación con mayor grado de especialización.
- Establecer un sistema común de créditos que permita garantizar que para la obtención de un título todos los estudiantes realizan el mismo esfuerzo.
- Implantar un Suplemento Europeo al Título donde se describa con precisión las capacidades adquiridas por el alumno durante sus estudios y permita promover la adopción de un sistema homologable y comparable de titulaciones superiores, con el fin de facilitar las mismas oportunidades de trabajo para todos los egresados.
- Fomentar la movilidad de estudiantes y profesores dentro del EEES.
- Promover la cooperación europea para garantizar la calidad de los estudios superiores de acuerdo a criterios equiparables.

En 2001, con la Declaración de Praga, se introducen nuevas líneas que destacan la importancia de fomentar el aprendizaje a lo largo de la vida y el papel activo de las universidades en la sociedad.

Las mejoras que se atribuyen al Plan Bolonia son:

- Las titulaciones han sido verificadas por la Agencia Nacional para la Evaluación de la Calidad y la Acreditación (en adelante ANECA), que como organismo público dependiente del ministerio vela porque se cumplan los requisitos para impartir los nuevos títulos) y por ello será entendida en el Espacio Europeo de Educación Superior (en adelante EEES).

- Facilitar la incorporación al mercado laboral al terminar la titulación.
- Todas las titulaciones se revisarán por la ANECA cada seis años para mantener su acreditación.
- La carga de trabajo (horas de clase magistral, preparación de trabajos/informes/prácticas, estudio por tu cuenta, etc.) que se empleen para superar cada materia se medirá en créditos del Sistema Europeo de Transferencia y Acumulación de Créditos (En adelante créditos ECTS o ECTS).
- Gracias a los ECTS se facilita tu movilidad dado que en todos los países del EEES (actualmente son 46 estados) se reconocerá y transferirá créditos ECTS que son transparentes y fácilmente entendibles por todos los sistemas universitarios europeos.
- En el caso de que se desee moverse a otra titulación no se perderá ni un solo crédito. Se reconocerán automáticamente todos aquellos que pertenezcan a esa rama de conocimiento; se reconocerán todos aquellos que sean coincidentes y el resto se transferirán.
- Se reconocerá el trabajo en forma de créditos por participación en actividades universitarias culturales, deportivas, de representación estudiantil, solidarias y de cooperación.
- Hay acceso al sistema de becas del Ministerio, tu Comunidad Autónoma, etc., que deberán ser portables (es decir, que si se va a otro país del EEES a estudiar se mantendrá esa beca), además de las becas Erasmus y otras de la propia universidad.
- Cualquier alumno tendrá el valor añadido de haber adquirido diversas habilidades transversales que serán valoradas muy positivamente por la sociedad (hablar en público, conocimiento en idiomas, trabajar en equipo, etc.)
- Al finalizar se tendrá un Suplemento Europeo al Título que explicará sus competencias para que cualquier empleador o universidad europea pueda entenderlo.
- Podrá acceder a titulaciones de Máster y posteriormente a Doctorado de cualquier universidad del EEES.

Anexo 2: Dispositivos usados en la integración

INNOVADORES | Tecnología

iPad para convivir con el autismo

Uno de los usuarios del programa informático visualiza las herramientas 'e-mintza'. | El Mundo

- Pictogramas y sonidos facilitan la comunicación a los niños con problemas

Patxi Arostegi | Bilbao

Actualizado **lunes 10/10/2011 09:43 horas**

Las nuevas tecnologías pueden convertirse en los mejores aliados de personas autistas. Un ejemplo son los **dispositivos electrónicos iPad,** que dotados de pantallas táctiles y dispositivos atractivos, pueden ayudar a que personas con problemas de comunicación desarrollen sus habilidades sociales de una manera más interactiva y entretenida.

La Fundación vasca Dr. Carlos Elósegui en colaboración con la Fundación Orange han ideado un programa informático para iPad y tablets que permite que personas autistas y con dificultad de habla puedan **"disponer de una herramienta que les sirve de voz".** "Vimos que herramientas de este estilo estaban muy consolidadas en el Reino Unido con niños con problemas de comunicación y decidimos que había un amplio terreno para la innovación", indica Joaquín Fuentes, jefe del servicio de psiquiatra infantil en la Policlínica Gipuzkoa y responsable del proyecto.

Bajo el nombre de 'e–mintza' la aplicación ha sido fruto del trabajo de la empresa tecnológica donostiarra Nesplora, que ha contado con el apoyo de la asociación guipuzcoana de autismo Gautena para la validación de este producto innovador. "Es un sistema absolutamente flexible, personalizable, fácil de utilizar que puede dar voz a personas que o no pueden o no saben hablar", indica.

En la práctica el software ideado por este consorcio de empresas puede descargarse fácilmente en cualquier tableta o iPad del mercado. Una vez se accede al programa el usuario puede emplear la herramienta 'on line' que dispone de **un servicio personalizable en función de la lengua**, textos, imágenes, vídeos y sonidos, programados en base a las "necesidades del usuario". Este podrá interactuar a través de la pantalla táctil en un iPad o "bien a través de un ratón en el caso de que lo utilice en un ordenador convencional", detalla.

"La herramienta le sirve para programar y para ello basta con saber enviar correos electrónicos y adjuntar un archivo. La tableta se convierte así en un panel comunicativo personal para los usuarios". Entre sus principales servicio figura un archivo de 400 pictogramas basados en tecnología Arassac y dispone, asimismo, de una **capacidad que alberga un total de 9000 pictogramas.** El programa posibilita sustitución y modificación de imágenes y fotos, también cuenta con una aplicación bilingüe. "Tiene voces reales en euskara y castellano, bilingüe, obtenidas de personas con sexo y edades diferentes".

Junto a estas prestaciones la aplicación esta programada para que funcione en diversas plataformas existentes en el mercado: Windows, MacOS y Linux. A esta facilidad une una agenda en la que el usuario puede **"ubicar hasta seis actividades** para planificar su día en un plan temporal secuenciado".

La herramienta, inicialmente pensada para facilitar la comunicación de personas con autismo, puede ser también empleada por otras personas por su "alta capacidad de personalización". "El 40% de los niños con autismo tiene importantes problemas de comunicación, pero les gusta repetir y **tienen buena memoria visual.** Por ello, hemos diseñado el sistema con su participación y su ayuda a sido clave para la aplicación", concluye.

Anexo 3: Instalación del SDK de Android:

Desde Mozilla Firefox accederemos a la URL:

http://developer.android.com/sdk/index.html

- Pulsaremos en la descarga "android-sdk_r10-linux_x86.tgz" correspondiente a la instalación de Android SDK para Linux.
- A continuación marcaremos "Guardar archivo" y pulsaremos "Aceptar" para iniciar la descarga de Android SDK.
- Accederemos a la carpeta donde hayamos descargado el fichero *android-sdk_r10-linux_x86.tgz*, pulsaremos con el botón derecho sobre él y seleccionaremos "Extraer aquí" para descomprimirlo.
- A continuación cortaremos la carpeta descomprimida "android-sdk-linux_x86":
- Y la pegaremos en la que será su ubicación definitiva, puede ser en cualquier carpeta, por ejemplo en */home/nombre_usuario/Android:*
- Ahora añadiremos la subcarpeta "tools" del SDK de Android:

  /home/nombre_usuario/Android/android-sdk-linux_x86/tools

- A la variable de entorno PATH del sistema, para que podamos hacer uso de los ejecutables del SDK de Android desde cualquier parte del sistema. Para ello editaremos el fichero /home/nombre_usuario/.bashrc o bien con GEdit en modo gráfico o bien ejecutando el siguiente comando linux desde una ventana de terminal:

  sudo nano /home/nombre_usuario/.bashrc

- Añadiremos la siguiente línea al final del fichero *.bashrc*:

  export PATH=${PATH}:/home/nombre_usuario/Android/android-sdk-linux_x86/tools

- Guardaremos los cambios pulsando Control + O y cerraremos nano pulsando Control + X:

```
ajpdsoft@pcubuntu: ~
Archivo  Editar  Ver  Terminal  Ayuda
  GNU nano 2.2.2        Archivo: /home/ajpdsoft/.bashrc            Modificado

# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi


export PATH=${PATH}:/home/ajpdsoft/Android/android-sdk-linux_x86/tools


^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y RePág.     ^K Cortar Tex ^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt   ^T Ortografía
```

Ilustración 41: .bashrc

- A continuación cerraremos todas las ventanas de terminal que tengamos abiertas para que podamos abrirlas nuevamente y se aplique la nueva variable de entorno PATH. Abriremos una ventana de terminal y ejecutaremos el fichero android introduciendo:
  - *android*
- Y pulsando INTRO, si hemos establecido correctamente la variable PATH anterior deberá aparecernos el programa Android SDK and AVD Manager, desde esta utilidad podremos actualizar el SDK de Android, crear dispositivos virtuales, etc. Desde esta utilidad, en la parte izquierda, pulsaremos en "Available packages", en la parte derecha desmarcaremos el check "Display updas only" y pulsaremos en "Refresh" (requiere de conexión a Internet):

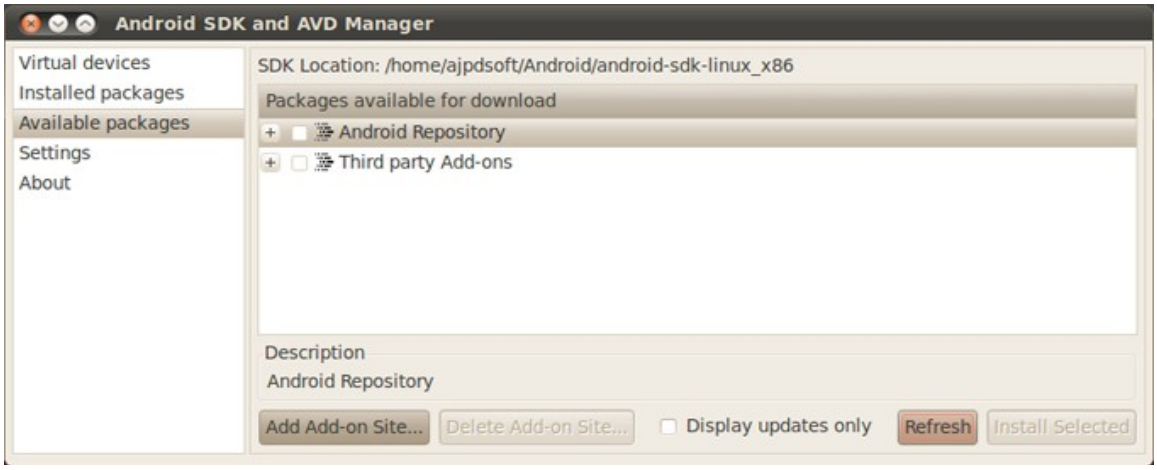

Ilustración 42: Android manager

Desde "Available packages" marcaremos todos los paquetes a instalar, en nuestro caso los seleccionaremos todos, aunque algunos no son necesarios por corresponder a versiones obsoletas o a emuladores que no necesitaremos. Si no queremos instalar todos los paquetes pulsaremos en el "+", desplegaremos el árbol y desmarcaremos los paquetes que no queramos instalar. Tras seleccionar los paquetes a instalar pulsaremos en "Install Selected".

- Leeremos los términos de licencia, si estamos de acuerdo marcaremos "Accept All" y pulsaremos "Install" (requiere de conexión a Internet).
- Se iniciará la descarga de los paquetes seleccionados, este proceso puede durar unos minutos, en función de la conexión a Internet y los paquetes seleccionados:

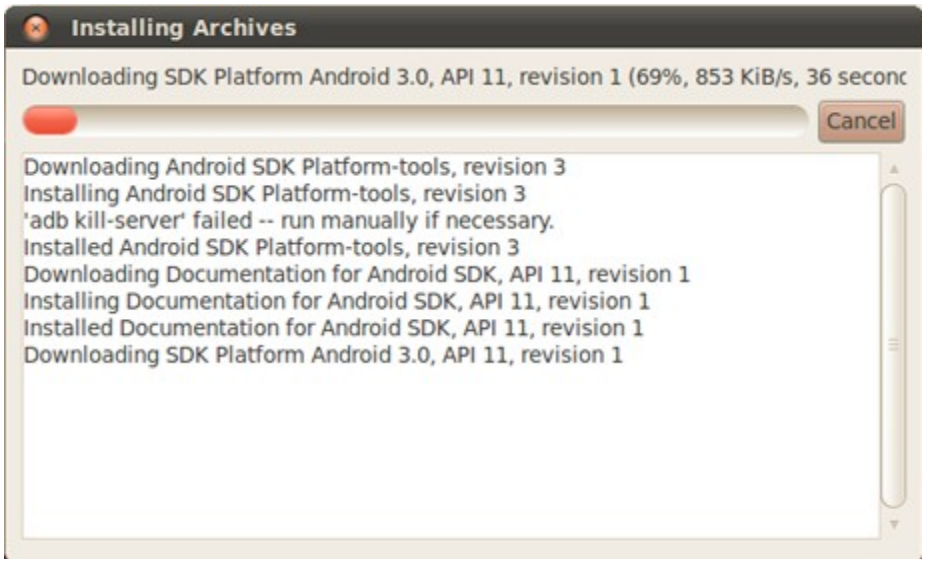

Ilustración 43: descarga de archivos

- Tras la instalación nos mostrará un mensaje indicando que ha concluido, pulsaremos "OK" y cerraremos Android SKD and AVD Manager:

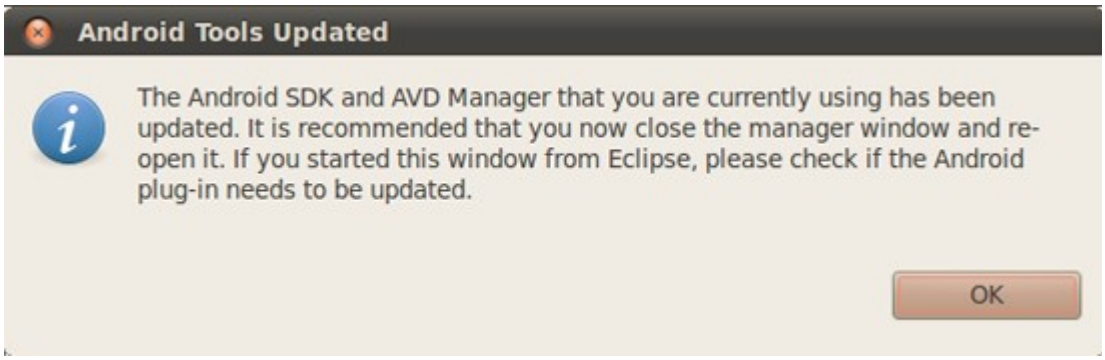

Ilustración 44: descarga exitosa

Anexo 4: Protocolo de comunicaciones.

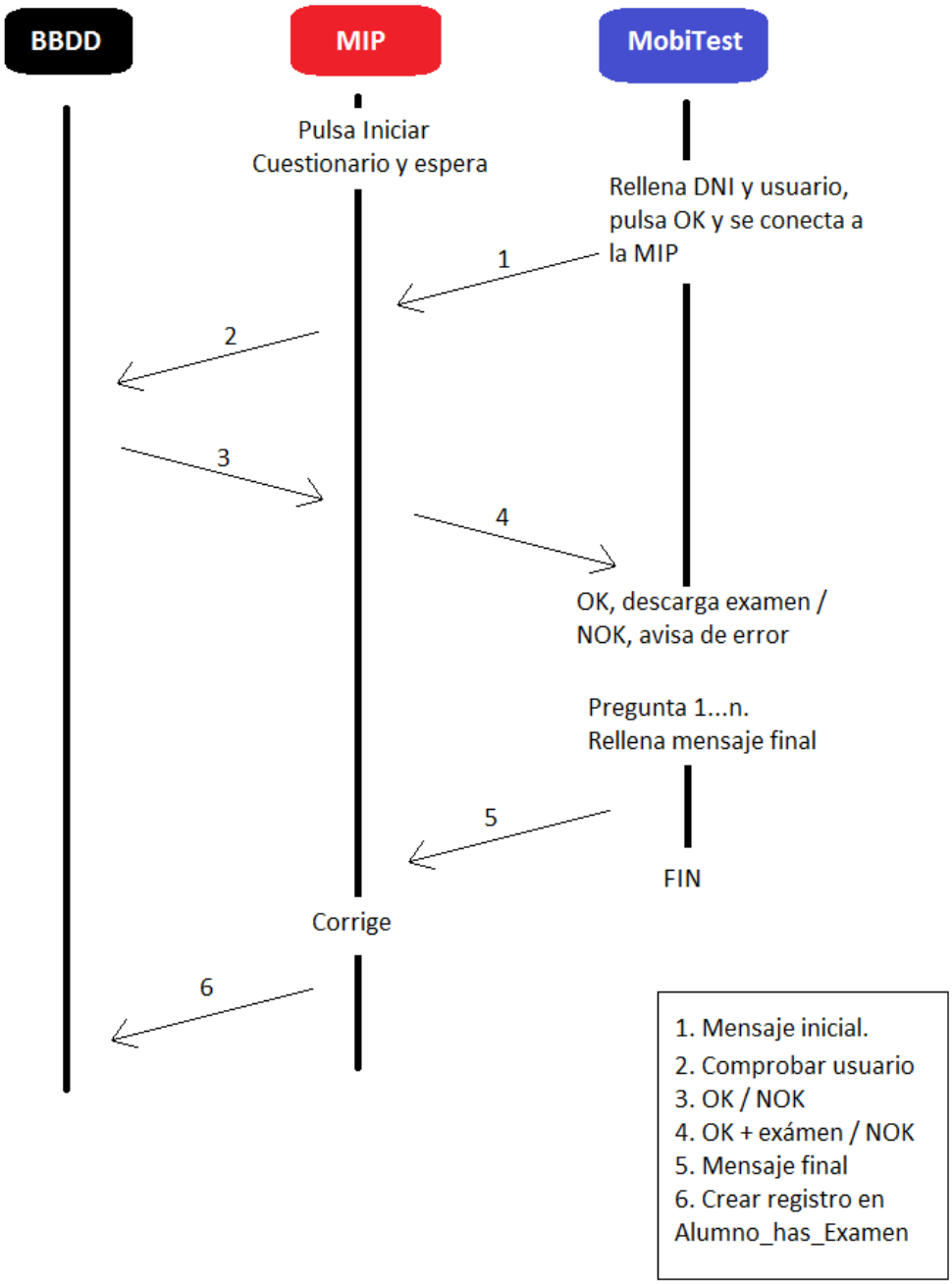

Ilustración 45: Protocolo de comunicaciones

- 1. Mensaje inicial: “INIT;DNI;password;mac\n”
- 2. Comprobar usuario: función existsAlumno de la clase AccesoDB.java
- 3. Devuelve TRUE (OK) la función anterior si el usuario y la contraseña existian. FALSE (NOK) de lo contrario.
- 4. Si es OK, enviamos OK además del exámen en formato .xml, la app lo guarda en la memoria del teléfono. Si es NOK generamos el mensaje de error para mostrarlo al usuario.
- 5. Mensaje final: “END;DNI;cadena_de_respuestas\n”
- 6. Crear registro: función insertNota de la clase AccesoDB.java

Anexo 5: Ejemplo .xml

Ejemplo de archivo .xml donde van guardados los exámenes.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<quiz codigo="1000">
	<question>
		<statement>Enunciado 1</statement>
		<op1 correct="True">Respuesta 1.1</op1>
		<op2>Respuesta 1.2</op2>
		<op3>Respuesta 1.3</op3>
		<op4>Respuesta 1.4</op4>
	</question>

	<question>
		<statement>Enunciado 2</statement>
		<op1 correct="True">Respuesta 2.1</op1>
		<op2>Respuesta 2.2</op2>
		<op3>Respuesta 2.3</op3>
		<op4>Respuesta 2.4</op4>
	</question>
</quiz>
```