



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE GRADO

Título
Tele-repaso android

Autor/es
Raul Lázaro López
Director/es
Eduardo Sáenz de Cabezón Irigaray
Facultad
Facultad de Ciencias, Estudios Agroalimentarios e Informática
Titulación
Grado en Ingeniería Informática
Departamento
Curso Académico
2012-2013



Tele-repaso android, trabajo fin de grado
de Raul Lázaro López, dirigido por Eduardo Sáenz de Cabezón Irigaray (publicado por la
Universidad de La Rioja), se difunde bajo una Licencia
Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.
Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.

ÍNDICE DE CONTENIDOS

1. DOCUMENTO DE OBJETIVOS DEL PROYECTO.....	1
1.1 Introducción y descripción del proyecto	1
1.2 Ámbito.....	1
1.3 Objetivos del proyecto.....	2
1.4 Alcance del proyecto	2
1.5 Planificación temporal.....	2
1.5.1 Entregables	2
1.5.2 Estructura de descomposición de tareas	3
1.5.3 Calendario de trabajo y método de trabajo.....	4
1.5.4 Esquema del diagrama de Gantt	5
1.6 Glosario de términos.....	6
2. ANÁLISIS DEL SISTEMA DE INFORMACIÓN	7
2.1 Definición del sistema	7
2.1.1 Determinación del alcance del sistema.....	7
2.1.1.1 Gestión de usuarios.....	7
2.1.1.2 Realizar tests.....	7
2.1.1.3 Consultar estadísticas	7
2.1.1.4 Cambiar tus datos	7
2.1.2 Identificación del entorno tecnológico	8
2.1.2.1 Hardware	8
2.1.2.2 Software.....	8
2.2 Establecimiento de requisitos	8
2.2.1 Obtención de requisitos	8
2.2.1.1 Diagrama de casos de uso.....	9
2.2.2 Especificación de casos de uso	10
2.2.2.1 Diagrama de actividad ‘Hacer test’	10
2.3 Elaboración del modelo de datos.....	10
2.3.1 Elaboración del modelo conceptual de datos	10
2.3.2 Elaboración del modelo lógico de datos.....	12
2.3.3 Especificación de necesidades de migración de datos y carga inicial.....	13
2.4 Definición de interfaces de usuario	13
2.4.1 Identificación de perfiles y diálogos.....	13
2.5 Aprobación del análisis del sistema de información	14
3. DISEÑO DEL SISTEMA DE INFORMACIÓN	15
3.1 Definición de la arquitectura del sistema	15
3.1.1 Definición de la arquitectura	15
3.1.2 Identificación de requisitos.....	16
3.2 Diseño de la base de datos.....	17
3.3 Diseño de la migración y carga inicial de datos	20
3.4 Especificación técnica del plan de pruebas	20
3.5 Aprobación del diseño del sistema de información.....	22
4 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN	23
4.1 Preparación del entorno de generación y construcción	23
4.1.1 Preparación del entorno de construcción.....	23
4.1.2 Implantación de la base de datos física	23
4.2 Generación del código de los componentes y procedimientos.....	26
4.2.1 Generación del código de componentes	26

4.2.2 Código destacable	27
4.2.2.1 Pantalla principal y de identificación	27
4.2.2.2 Acceso a base de datos	29
4.3 Ejecución de las pruebas unitarias	31
4.3.1 Preparación, realización y evaluación de las pruebas unitarias	31
4.4 Ejecución de las pruebas de integración	31
4.4.1 Preparación, realización y evaluación de las pruebas de integración	31
4.5 Ejecución de las pruebas del sistema	31
4.5.1 Preparación, realización y evaluación de las pruebas del sistema	31
4.6 Interfaces finales de usuario	32
4.6.1 Pantalla principal de identificación	32
4.6.2 Pantalla de entrada del usuario profesor	33
4.6.3 Pantalla de entrada del usuario alumno	34
4.6.4 Pantalla de cambiar tus datos	35
4.6.5 Pantalla de hacer test	36
4.6.6 Pantalla de consultar estadísticas	39
4.7 Aprobación del sistema de información	41
4.7.1 Presentación y aprobación del sistema de información	41
5 IMPLANTACIÓN, PRUEBAS Y ACEPTACIÓN DEL SISTEMA	42
5.1 Establecimiento del plan de implantación	42
5.1.1 Definición del plan de implantación	42
5.1.2 Especificación del equipo de implantación	44
5.2 Formación necesaria para la implantación	44
5.2.1 Preparación y formación del equipo de implantación	44
5.3 Incorporación del sistema al entorno de operación	44
5.3.1 Realización de la instalación y de la implantación	44
5.4 Carga de datos al entorno de operación	44
5.4.1 Migración y carga inicial de datos	44
5.5 Pruebas de implantación del sistema	44
5.5.1 Realización de las pruebas de implantación	44
5.5.2 Evaluación del resultado de las pruebas de implantación	44
5.6 Pruebas de aceptación del sistema	45
5.6.1 Realización y evaluación de las pruebas de aceptación	45
5.7 Presentación y aprobación de la implantación del sistema	45
5.7.1 Convocatoria de la presentación del sistema	45
5.7.2 Aprobación del sistema	45
5.8 Paso a producción	45
6 CONCLUSIONES	46
6.1 Concordancia entre los resultados y los objetivos	46
6.2 Lecciones aprendidas	46
7 BIBLIOGRAFÍA	47
ANEXOS	48
ANEXO 1 CONTENIDO DEL CD ADJUNTO	48
A.1 Contenido del CD adjunto	48

ÍNDICE DE FIGURAS

<i>Figura 1: Estructura de descomposición del proyecto</i>	<i>4</i>
<i>Figura 2: Diagrama de Gantt</i>	<i>5</i>
<i>Figura 3: Diagrama general de casos de uso</i>	<i>9</i>
<i>Figura 4: Diagrama de actividad de hacer test</i>	<i>10</i>
<i>Figura 5: Modelo conceptual de datos</i>	<i>11</i>
<i>Figura 6: Diagrama de arquitectura de aplicación móvil.....</i>	<i>15</i>
<i>Figura 7: Diagrama de arquitectura MVC</i>	<i>15</i>
<i>Figura 8: Base de datos Tele-repaso Android en SQL</i>	<i>26</i>
<i>Figura 9: Pantalla principal de identificación.....</i>	<i>32</i>
<i>Figura 10: Pantalla entrada del usuario profesor.....</i>	<i>33</i>
<i>Figura 11: Pantalla entrada del usuario alumno</i>	<i>34</i>
<i>Figura 12: Pantalla cambiar tus datos de profesores o alumnos</i>	<i>35</i>
<i>Figura 13: Pantalla elegir parámetros test.....</i>	<i>36</i>
<i>Figura 14: Pantalla realizar test.....</i>	<i>37</i>
<i>Figura 15: Pantalla corrección test</i>	<i>38</i>
<i>Figura 16: Pantalla consultar estadísticas de alumno</i>	<i>39</i>
<i>Figura 17: Pantalla consultar estadísticas1 de profesores</i>	<i>40</i>
<i>Figura 18: Pantalla consultar estadísticas2 de profesor.....</i>	<i>41</i>
<i>Figura 19: Pantalla datos servidor Hostinger</i>	<i>42</i>
<i>Figura 20: Pantalla ftp de servidor Hostinger.....</i>	<i>43</i>
<i>Figura 21: Pantalla base de datos del servidor Hostinger</i>	<i>43</i>

RESUMEN

Como proyecto fin de grado he escogido el desarrollar una aplicación móvil con Android para el repaso en modo online de los alumnos en cualquier momento y de cualquier asignatura que cursan y que los profesores puedan consultar las estadísticas de las notas de los alumnos que cursan sus asignaturas. Deberá estar finalizado en Junio del curso 2012-2013, para depositarlo y defenderlo en ese mismo mes.

ABSTRACT

As a final degree project I have chosen to develop a mobile application for Android review online mode students at any time and in any subject that present and that teachers can consult the statistics of the notes of the students attending their subjects. To be finalized in June year 2012-2013, to deposit it and defend it in the same month.

1. DOCUMENTO DE OBJETIVOS DEL PROYECTO

1.1 Introducción y descripción del proyecto

Como desarrollo del proyecto fin de grado (PFG), he escogido seguir con mi anterior proyecto fin de carrera (PFC), que consiste en una pequeña reducción del mismo pero desarrollado en Android. Se realizará una aplicación móvil, en la que los alumnos podrán realizar test, tanto como repaso o como exámenes que les manda el profesor, consultar estadísticas, cambiar sus datos, etc.

Del PFC ya se creaban los usuarios y la contraseña de cada profesor y alumno, así que ya contamos con esa parte desarrollada.

El proyecto será realizado en Java para Android, usando a la vez MYSQL. Se realizará una aplicación móvil, para que los alumnos puedan usarla sin tener que estar delante de un ordenador, es decir, desde cualquier dispositivo móvil Android en cualquier lugar donde se disponga de internet.

El dominio en el que va a funcionar es en Internet, para el repaso de los alumnos en las asignaturas que les interese y para que los profesores y alumnos puedan consultar estadísticas, además de poder cambiar sus datos. La aplicación se desarrolla para poner en práctica lo aprendido en el grado de informática, coger experiencia, agilidad, conseguir más conocimientos y conseguir el título. Este proyecto es la continuación de un proyecto realizado anteriormente, en el cuál hay que usar otra tecnología de desarrollo y el alcance es un poco menor.

1.2 Ámbito

El proyecto se implantará al final en un servidor gratuito del proveedor 'Hostinger' que crearemos nosotros, para poder acceder a Tele-repaso Android desde los móviles con ese sistema operativo en cualquier momento.

1.3 Objetivos del proyecto

- La aplicación deberá servir de repaso para los alumnos o incluso de evaluación de los profesores a los alumnos (poniendo un tiempo de finalización del test, realizándolo en clase).
- Se establecerá un diseño básico por temas, asignatura y curso.
- El sistema no deberá necesitar mantenimiento.
- Los alumnos deberán poder acceder a preguntas de ese curso y anteriores si es que tiene alguna asignatura pendiente de cursos anteriores.
- La aplicación deberá tener dos niveles de usuario: alumno y profesor.
- El sistema deberá tener la seguridad suficiente para que no puedan entrar otros usuarios ajenos al sistema, validando el usuario y la contraseña de alumnos y profesores.
- El sistema deberá tener unas estadísticas finales.
- El sistema podrá utilizar imágenes en el test, por lo que se deberá utilizar el mínimo de datos, para que se produzca la mínima transferencia posible en el móvil.
- Se deberá depositar el proyecto como muy tarde a mediados de junio de 2013.
- Se terminarán todos los entregables antes de la entrega de la memoria.
- Se acabará y entregará la memoria antes de la defensa.
- Se defenderá el proyecto a finales de junio de 2013.

1.4 Alcance del proyecto

La tecnología empleada en el desarrollo del proyecto será Java para Android y MYSQL.

1.5 Planificación temporal

1.5.1 Entregables

Las entregas que tendrán lugar a lo largo del proyecto serán.

- Análisis de requisitos (Documento de objetivos del proyecto): se entregarán los requisitos analizados al tutor, para que sea validado, y una vez que dé el visto bueno, pues seguir adelante con el proyecto.
- Análisis del sistema: se entregará el análisis del sistema al tutor para que lo valide, y seguir con el proyecto. Al final también será entregado al tribunal de evaluación, junto con la memoria del mismo.
- Diseño del sistema: una vez que se ha validado el análisis se procederá al diseño del sistema y se entregará cuando se tenga hecho al tutor para que lo valide y seguir con el proyecto. Al final también será entregado al tribunal de evaluación, junto con la memoria del mismo.
- Implementación y pruebas: una vez que se ha validado el diseño se procederá a la implementación del sistema y las pruebas. Se entregará al tutor ya la aplicación final, con el código y el listado de las pruebas realizadas, además se mostrará la aplicación.

- Implantación y pruebas: una vez que se ha validado la implementación y las pruebas se procederá a la implantación del sistema y las pruebas correspondientes. Al final se entregará al tutor la documentación de la implantación y se mostrará la aplicación finalizada por completo.
- Memoria: documento final del proyecto, que se entregará al tribunal para la evaluación del mismo.
- Presentación del Proyecto: presentación que se mostrará en la defensa del proyecto.

1.5.2 Estructura de descomposición de tareas

La Estructura de Descomposición del Proyecto se compone de:

	Nombre de tarea	Duración	Comienzo	Fin	Prede
1	<input type="checkbox"/> Proyecto Fin de Grado	300 horas	lun 21/01/13	vie 07/06/13	
2	<input type="checkbox"/> Análisis de Requisitos	37 horas	lun 21/01/13	mié 06/02/13	
3	Recopilación de información	15 horas	lun 21/01/13	vie 25/01/13	
4	Documento de objetivos	22 horas	lun 28/01/13	mié 06/02/13	3
5	<input type="checkbox"/> Análisis del Sistema de Información	33 horas	mié 06/02/13	jue 21/02/13	4
6	<input type="checkbox"/> Definición del Sistema	3 horas	mié 06/02/13	jue 07/02/13	
7	Determinación del Alcance del Sistema	2 horas	mié 06/02/13	mié 06/02/13	
8	Identificación del entorno tecnológico	1 hora	jue 07/02/13	jue 07/02/13	7
9	<input type="checkbox"/> Establecimiento de requisitos	12 horas	jue 07/02/13	mié 13/02/13	8
10	Obtención de requisitos	6 horas	jue 07/02/13	lun 11/02/13	
11	Especificación de casos de uso	4 horas	lun 11/02/13	mar 12/02/13	10
12	Validación de requisitos	2 horas	mar 12/02/13	mié 13/02/13	11
13	<input type="checkbox"/> Elaboración del modelo de datos	14 horas	mié 13/02/13	mar 19/02/13	12
14	Elaboración del modelo conceptual de datos	5 horas	mié 13/02/13	jue 14/02/13	
15	Elaboración del modelo lógico de datos	6 horas	vie 15/02/13	lun 18/02/13	14
16	Normalización del modelo lógico de datos	2 horas	mar 19/02/13	mar 19/02/13	15
17	Especificación de necesidades de migración de datos y carga inicial	1 hora	mar 19/02/13	mar 19/02/13	16
18	<input type="checkbox"/> Definición de Interfaces de usuario	2 horas	mié 20/02/13	mié 20/02/13	17
19	Identificación de perfiles y diálogos	2 horas	mié 20/02/13	mié 20/02/13	
20	Aprobación del Análisis del Sistema de Información	2 horas	mié 20/02/13	jue 21/02/13	19
21	<input type="checkbox"/> Diseño del Sistema de Información	20 horas	jue 21/02/13	vie 01/03/13	20
22	<input type="checkbox"/> Definición de la arquitectura del sistema	6 horas	jue 21/02/13	lun 25/02/13	
23	Definición de la arquitectura del sistema	4 horas	jue 21/02/13	vie 22/02/13	
24	Identificación de requisitos	2 horas	vie 22/02/13	lun 25/02/13	23
25	Diseño de la Base de Datos	7 horas	lun 25/02/13	mié 27/02/13	24
26	Diseño de la migración y carga inicial de datos	1 hora	mié 27/02/13	mié 27/02/13	25
27	Especificación Técnica del Plan de Pruebas	4 horas	jue 28/02/13	vie 01/03/13	26
28	Aprobación del Diseño del Sistema de Información	2 horas	vie 01/03/13	vie 01/03/13	27
29	<input type="checkbox"/> Construcción del Sistema de Información	149 horas	lun 04/03/13	vie 10/05/13	28
30	<input type="checkbox"/> Preparación del entorno de generación y construcción	7 horas	lun 04/03/13	mié 06/03/13	
31	Preparación del entorno de operación	1 hora	lun 04/03/13	lun 04/03/13	
32	Implantación de la base de datos física	6 horas	lun 04/03/13	mié 06/03/13	31
33	<input type="checkbox"/> Generación del código de los componentes y procedimientos	117 horas	mié 06/03/13	mar 30/04/13	32
34	Generación del código de componentes	112 horas	mié 06/03/13	vie 26/04/13	
35	Código destacable	5 horas	vie 26/04/13	mar 30/04/13	34
36	<input type="checkbox"/> Ejecución de las pruebas unitarias	2 horas	mar 30/04/13	mar 30/04/13	35
37	Preparación, realización y evaluación de las pruebas unitarias	2 horas	mar 30/04/13	mar 30/04/13	
38	<input type="checkbox"/> Ejecución de las pruebas de integración	2 horas	mié 01/05/13	mié 01/05/13	37

38	[-] Ejecución de las pruebas de integración	2 horas	mié 01/05/13	mié 01/05/13	37
39	Preparación, realización y evaluación de las pruebas de integración	2 horas	mié 01/05/13	mié 01/05/13	
40	[-] Ejecución de las pruebas del sistema	6 horas	mié 01/05/13	vie 03/05/13	39
41	Preparación, realización y evaluación de las pruebas del sistema	6 horas	mié 01/05/13	vie 03/05/13	
42	Interfaces finales de usuario	15 horas	vie 03/05/13	vie 10/05/13	41
43	Aprobación del sistema de información	2 horas	vie 10/05/13	lun 13/05/13	42
44	[-] Implantación del Sistema de Información	12 horas	lun 13/05/13	vie 17/05/13	43
45	[-] Establecimiento del plan de implantación	3 horas	lun 13/05/13	mar 14/05/13	
46	Definición del Plan de Implantación	2 horas	lun 13/05/13	lun 13/05/13	
47	Especificación del Equipo de Implantación	1 hora	mar 14/05/13	mar 14/05/13	46
48	[-] Formación necesaria para la implantación	1 hora	mar 14/05/13	mar 14/05/13	47
49	Preparación y formación del equipo de implantación	1 hora	mar 14/05/13	mar 14/05/13	
50	[-] Incorporación del sistema al entorno de operación	3 horas	mar 14/05/13	mié 15/05/13	49
51	realización de la Instalación y de la implantación	3 horas	mar 14/05/13	mié 15/05/13	
52	[-] Carga de datos al entorno de operación	2 horas	mié 15/05/13	jue 16/05/13	51
53	Migración y carga inicial de datos	2 horas	mié 15/05/13	jue 16/05/13	
54	[-] pruebas de implantación del sistema	2 horas	jue 16/05/13	jue 16/05/13	53
55	realización de las pruebas de implantación	1 hora	jue 16/05/13	jue 16/05/13	
56	Evaluación del resultado de las pruebas de implantación	1 hora	jue 16/05/13	jue 16/05/13	55
57	[-] Pruebas de aceptación del sistema	1 hora	vie 17/05/13	vie 17/05/13	56
58	Realización y evaluación de las pruebas de aceptación	1 hora	vie 17/05/13	vie 17/05/13	
59	Presentación y aprobación de la Implantación del Sistema	2 horas	lun 13/05/13	lun 13/05/13	
60	Paso a producción	2 horas	mar 14/05/13	mar 14/05/13	59
61	Conclusiones	2 horas	mar 14/05/13	mié 15/05/13	60
62	Bibliografía	2 horas	mié 15/05/13	mié 15/05/13	61
63	[-] Anexos	1 hora	jue 16/05/13	jue 16/05/13	62
64	Anexo 1 Contenido del CD	1 hora	jue 16/05/13	jue 16/05/13	
65	Preparar documentación	20 horas	jue 16/05/13	vie 24/05/13	64
66	Preparar defensa	30 horas	lun 27/05/13	vie 07/06/13	65

Figura 1: Estructura de descomposición del proyecto

1.5.3 Calendario de trabajo y método de trabajo

Eventualmente cada dos semanas aproximadamente se entregará al tutor el resumen de lo que hemos estado haciendo y nos reuniremos si es necesario.

Se comunicará con el tutor mediante correo, teléfono, o incluso en persona, pidiendo una cita anteriormente.

Se trabajará en el proyecto 3 horas diarias todos los días entre semana, salvo que surja algún imprevisto, en el cual se intentará hacer más horas algún otro día o avanzar en el fin de semana para recuperar el tiempo previsto. El fin de semana está planificado en un principio para descansar, salvo que haya que recuperar horas de realización del proyecto, por no haberlas cumplido entre semana.

La comunicación con el tribunal de evaluación será el día de la defensa, en la cual tendremos 20 minutos para exponerles lo más relevante del proyecto.

Se irá siguiendo la planificación prevista, si se retrasa se intentara volver a seguir el transcurso metiendo más horas al proyecto.

La metodología empleada para la realización del proyecto, es lo más parecido a metrica3, pues ya se tiene conocimiento de ella y está el alcance del proyecto muy definido, sin intención de cambiarlo.

1.5.4 Esquema del diagrama de Gantt

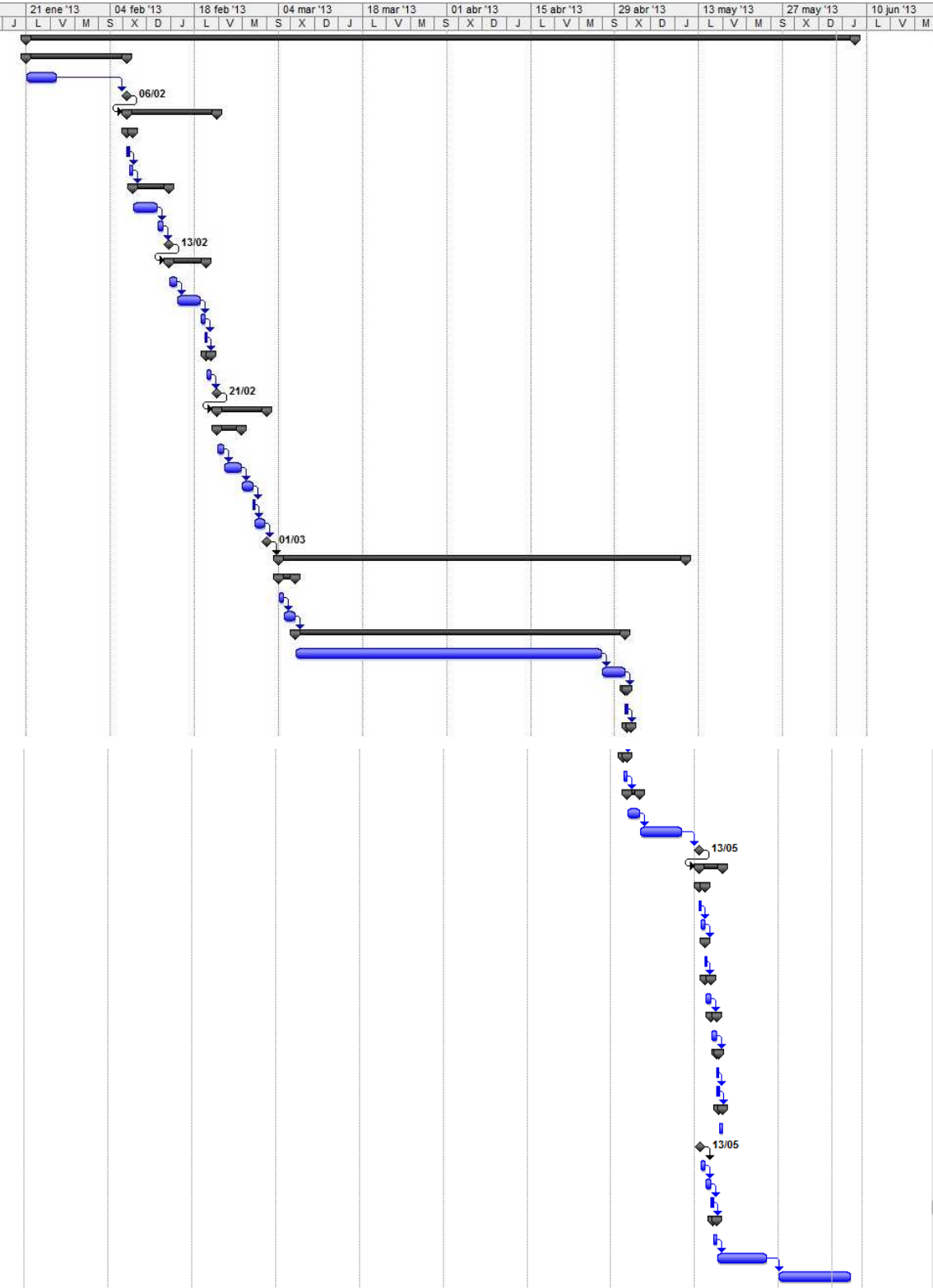


Figura 2: Diagrama de Gantt

1.6 Glosario de términos

- *Android*: es una plataforma de código abierto. Android es un Sistema Operativo además de una plataforma de Software basada en el núcleo de Linux. Diseñada en un principio para dispositivos móviles, Android permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java.
- *JSON*: acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
- *MySQL*: es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.
- *Eclipse IDE for Java EE Developers*: Entorno de desarrollo para implementar aplicaciones que se ejecutan en dispositivos Android.
- *Android SDK*: es el kit de desarrollo necesario para programar e implementar todo tipo de aplicaciones para Android, el sistema operativo para teléfonos móviles propuesto por Google. Este paquete o kit de desarrollo incluye las APIs y herramientas necesarias para desarrollar aplicaciones utilizando JAVA como lenguaje de programación.

2. ANÁLISIS DEL SISTEMA DE INFORMACIÓN

2.1 Definición del sistema

2.1.1 Determinación del alcance del sistema

2.1.1.1 Gestión de usuarios

Los usuarios que accederán a Tele-repaso Android serán, los profesores y los alumnos, pero no dispondrán de los mismos permisos. Por lo tanto habrá que gestionar los dos tipos de usuarios.

Debido a que cada usuario podrá realizar un subconjunto de las acciones de Tele-repaso Android, estos estarán agrupados en distintos roles donde cada uno de ellos podrá realizar las acciones que le corresponden.

Habrán dos roles, que será uno por cada tipo de usuario:

- *Profesor*: Usuarios que podrán consultar estadísticas de las materias que imparten, consultar estadísticas de los test realizados por asignaturas, cursos y temas y cambiar sus datos (nombre, apellidos, clave, contraseña). Sin embargo, añadir, modificar o borrar preguntas y respuestas de sus asignaturas no se podrá hacer, pues es muy complicado y dificultoso realizarlo desde el móvil, así que se realizará solo a través de la aplicación web realizada anteriormente en PFC.
- *Alumno*: Usuarios que podrán realizar tests de las asignaturas que cursan, consultar estadísticas de las notas sacadas en los test realizados de sus asignaturas y cambiar sus datos (nombre, apellidos, clave, contraseña).

2.1.1.2 Realizar tests

Los alumnos serán los encargados de realizar los test ya sean para repasar o para ser evaluados por los profesores. Cada test estará compuesto de 5 preguntas aleatorias, de todas las introducidas al sistema, de las cuales tendrán unas respuestas y solo una será la correcta. De las preguntas y respuestas no nos ocuparemos en este proyecto, sino que ya estaba implementado en el proyecto fin de carrera, y desde el móvil no se podrá insertar preguntas por parte del profesor.

2.1.1.3 Consultar estadísticas

El sistema debe tener unas estadísticas finales, que dependiendo de si es un alumno se mostrará de cada asignatura que cursa el número de test realizados y la media de las notas sacadas en esos test o si es un profesor, se mostrarán las notas de los test por cada alumno según los parámetros elegidos de asignatura, curso y tema.

2.1.1.4 Cambiar tus datos

Cada usuario del sistema, una vez entrado al sistema, podrá cambiar sus datos: nombre, apellidos, clave y contraseña.

2.1.2 Identificación del entorno tecnológico

Se desarrollará una aplicación móvil, con información dinámica, ya que la información estará almacenada en una base de datos del servidor, y la aplicación accederá a un script PHP que se conectará a la base de datos y así se conseguirán los datos mínimos necesarios.

Como solución tecnológica para satisfacer las necesidades del sistema se ha optado por lo siguiente:

2.1.2.1 Hardware

El Hardware necesario para el desarrollo de esta aplicación es un ordenador personal (se utiliza el del alumno).

Y para la utilización de la aplicación será necesario:

- Un servidor con IP fija gratuito que esté funcionando las 24 horas, donde se almacenará la base de datos MySQL y archivos PHP necesarios.
- Cualquier móvil Android conectado a Internet es válido para utilizar la aplicación.

2.1.2.2 Software

El Software necesario para la creación de la aplicación es:

- Microsoft Office Word 2010 (Licencia de la Universidad de la Rioja) para la realización del texto necesario para la memoria.
- Microsoft Office Project 2010 (Licencia de la Universidad de la Rioja) para la realización de los diagramas de Gantt necesarios para la memoria.
- Microsoft Office Visio 2010 (Licencia de la Universidad de la Rioja) para la realización de los diferentes diagramas de la memoria como diagramas de casos de uso, diseño de interfaces de usuario,...
- Prototyper Free 2.1.0 (Software libre) para la realización de los diferentes prototipos.
- XAMPP 1.4.15 (Software libre) que posee PHP, Apache, MySQL y phpMyAdmin, el cual hará de servidor Web y de servidor de base de datos de la aplicación, localmente. PhpMyAdmin nos facilitará el trabajo a la hora de crear tablas en la base de datos. Posteriormente se usará el servidor gratuito de "Hostinger", donde colocaremos la base de datos y los archivos PHP.
- Eclipse IDE for Java EE Developers (ADT Eclipse) y Android SDK (Software libre) que nos ayudarán a desarrollar el código necesario para la programación de la aplicación Tele-repaso Android.

Lo necesario para la utilización de la aplicación es cualquier móvil Android con internet.

2.2 Establecimiento de requisitos

2.2.1 Obtención de requisitos

Los requisitos detallados son:

- Se adaptará la aplicación Tele-repaso a Android, para dispositivos móviles.

- La aplicación debe servir de repaso para los alumnos de las asignaturas que cursan.
- Los alumnos deben poder realizar test de un tema y asignatura que elijan. El sistema deberá sacar cinco preguntas aleatorias, de las que haya introducidas en la base de datos, cuando accedan los alumnos al test de ese curso, asignatura y tema. Cada pregunta tendrá un número de respuestas determinadas por el profesor que la ha introducido, de las cuales solo una será la correcta, aunque ello no tiene importancia para este proyecto.
- El sistema no debe necesitar mantenimiento.
- Los alumnos deben poder hacer test de las materias que cursan, incluido asignaturas que sean pendientes de otros cursos anteriores, si es que tiene.
- La aplicación debe tener 2 niveles de usuario: alumno y profesor.
- Se debe establecer un diseño básico de temas por cada una de las asignaturas y cursos.
- El sistema debe tener la seguridad suficiente para que no puedan entrar otros usuarios ajenos al sistema, validando el usuario y la contraseña de los usuarios.
- El sistema debe tener unas estadísticas finales, en las que se muestren la media de las calificaciones obtenidas de los alumnos por curso y asignatura.
- El sistema debe poder utilizar imágenes en el test, por lo que se deberá utilizar el mínimo de datos para que se produzca la mínima transferencia posible en el móvil, de forma que se transfieran solo los datos de las preguntas, sus respuestas y las imágenes que se necesitan en el test a realizar.
- El proyecto debe estar finalizado como muy tarde para mediados de junio de 2013.

Las tres tareas que pueden realizar los alumnos son:

- **Modificar datos de usuario:** Cualquier usuario del sistema puede cambiar sus datos personales, como son nombre, apellidos, clave y contraseña.
- **Hacer test:** Los alumnos pueden realizar test para repasar los temas de las asignaturas que cursan.
- **Consultar estadísticas:** Cualquier usuario puede ver las estadísticas de los test que se van realizando a lo largo del curso.

2.2.1.1 Diagrama de casos de uso

En los diagramas de casos de uso, el objetivo es conocer las interacciones del usuario con el sistema.

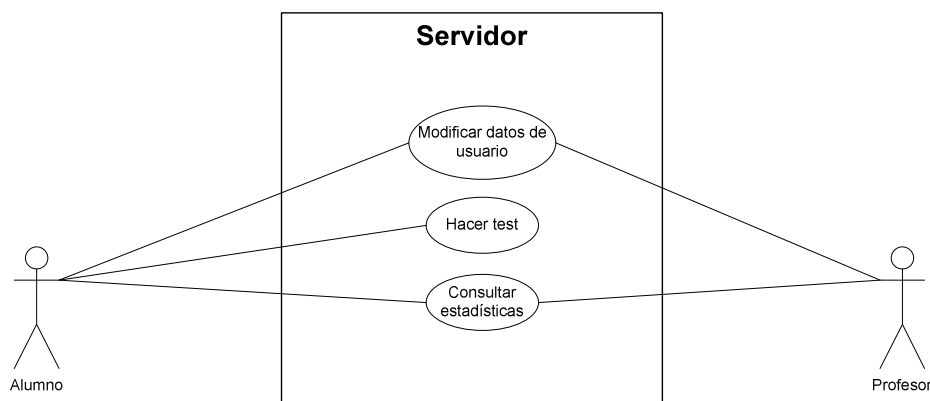


Figura 3: Diagrama general de casos de uso

2.2.2 Especificación de casos de uso

2.2.2.1 Diagrama de actividad 'Hacer test'

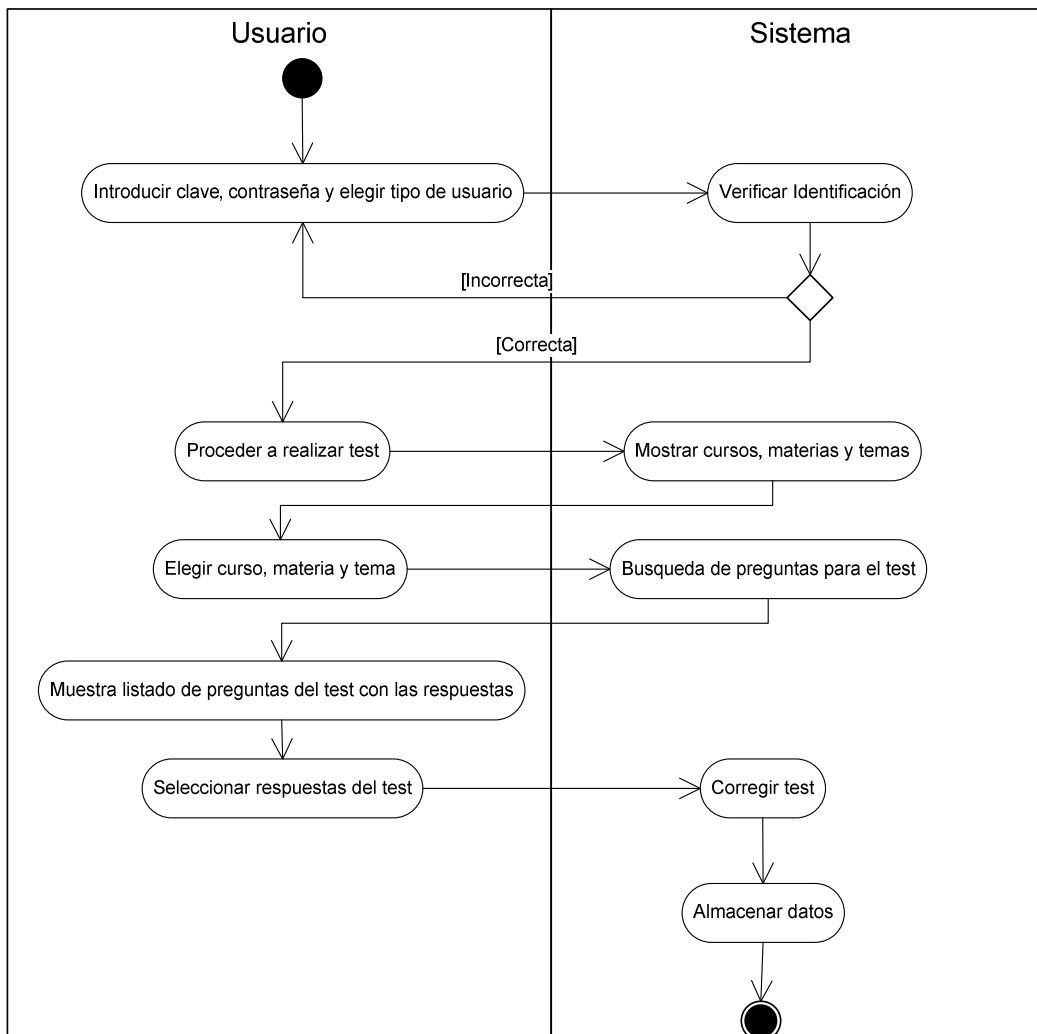


Figura 4: Diagrama de actividad de hacer test

2.3 Elaboración del modelo de datos

2.3.1 Elaboración del modelo conceptual de datos

Para ello he utilizado diagramas UML (Unified Modeling Language) puesto que es una técnica estudiada en el grado, es un lenguaje más sencillo, está más estandarizado y es más útil que los diagramas E/R (Entidad Relación). La única diferencia con respecto al anterior proyecto, son dos campos que se han añadido en la tabla preguntas, para poder asociarlo a un archivo que ocupe menos espacio o a una url donde se encuentre el archivo. Esto es para que el consumo de datos en el móvil sea lo mínimo posible.

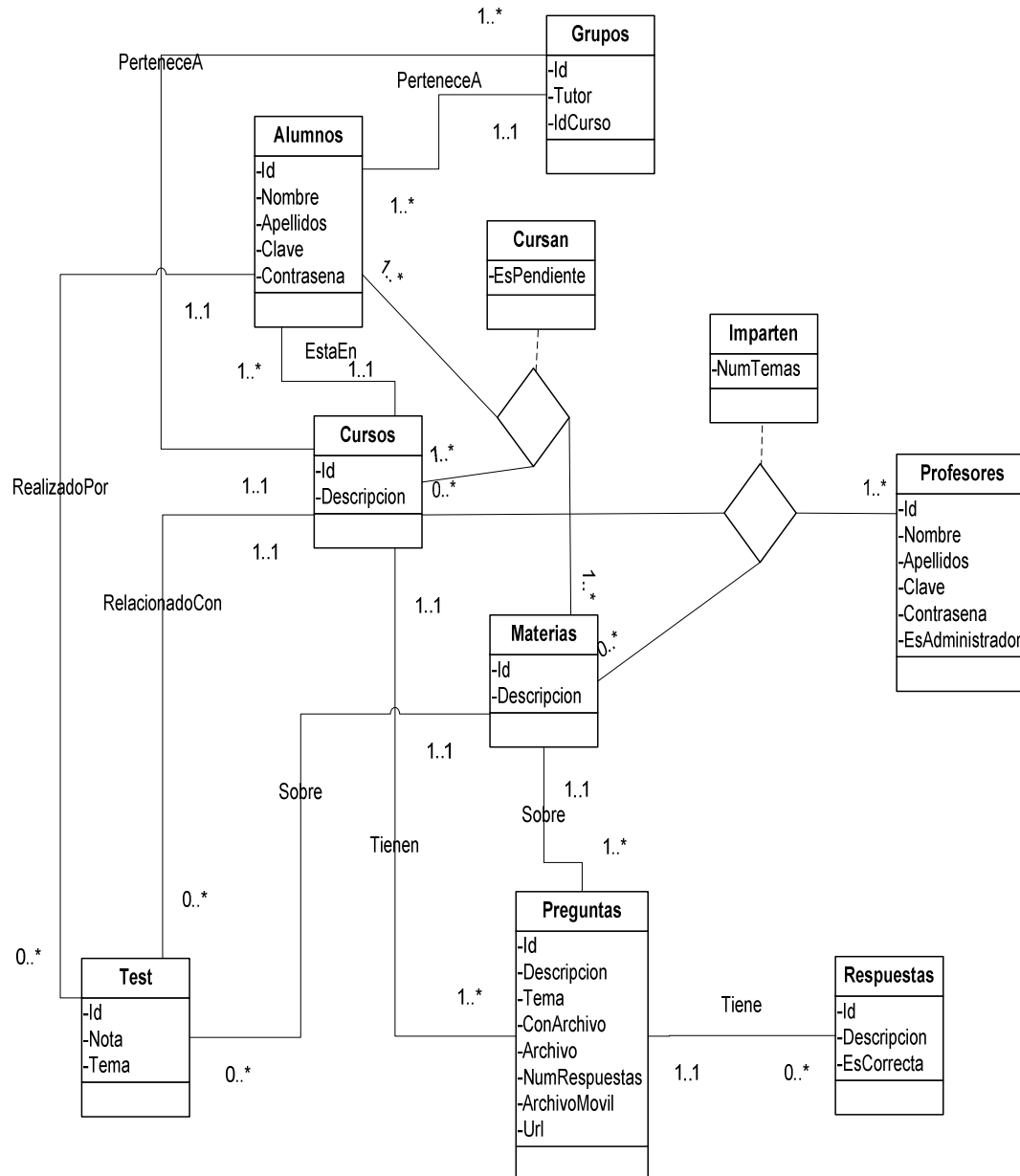


Figura 5: Modelo conceptual de datos

Del diagrama UML vemos que se forman 10 entidades que son:

- Cursos: representa a todos los cursos que hay en el centro en el curso actual. Sus atributos son el id y la descripción del curso.
- Materias: representa a todas las asignaturas que se dan en el centro. Sus atributos son el id y la descripción de la materia.
- Grupos: representa a todos los grupos que hay en el centro en el curso actual. Sus atributos son el id, el tutor del grupo y el curso al que hace referencia.
- Alumnos: representa a todos los alumnos que están matriculados en el centro en el curso actual. Sus atributos necesarios son el id, el nombre, los apellidos, la clave, la contraseña y el curso y el grupo al que pertenecen.
- Profesores: representa a todos los profesores del instituto y al administrador. Un atributo será el que haga diferenciarlos, el de si es o no administrador. Se ha optado por esta opción, por no crear otra entidad con un solo registro, el del administrador, y además tiene los mismos atributos

que los profesores. La función del administrador no se usa para este proyecto, actúa como un profesor más.

Sus atributos son el id, el nombre, los apellidos, la clave, la contraseña y el de si es o no administrador.

- Preguntas: representa todas las preguntas que aparecerán en los test, según la asignatura, el curso y el tema elegido. Sus atributos son el id, la descripción de la pregunta, el tema, la materia y el curso al que pertenecen, si tiene archivo asociado o no, el nombre del archivo asociado si es que tiene alguno y el número de respuestas que tiene la pregunta.

- Respuestas: representa a todas las respuestas de las preguntas. Sus atributos son el id, la descripción de la respuesta, si es la correcta o no, y el id de la pregunta a la que hace referencia.

- Test: representa a los test realizados por los alumnos en el curso actual. Sus atributos son el id, el alumno que realiza el test, el curso, la materia, el tema y la nota sacada al que pertenece el test.

- Cursan: representa a las asignaturas en las cuales están matriculados los alumnos, es decir a las asignaturas ya sean pendientes o no, que cursan los alumnos en los distintos cursos del centro en el curso actual. Sus atributos son el alumno, el curso, la asignatura, y si es pendiente o no.

- Imparten: representa a las asignaturas que imparten los profesores en los distintos cursos del centro en el curso actual. Sus atributos son el profesor, el curso, la asignatura y el número de temas que tiene esa asignatura en ese curso.

2.3.2 Elaboración del modelo lógico de datos

CURSOS

<u>Id</u>	Descripcion
-----------	-------------

En esta tabla se almacenarán todos los cursos que haya en el centro.

GRUPOS

<u>Id</u>	Tutor	IdCurso
-----------	-------	---------

En esta tabla se almacenarán todos los grupos de que dispondrá cada curso en el centro.

MATERIAS

<u>Id</u>	Descripcion
-----------	-------------

En esta tabla se almacenarán todas las materias que se imparten en el centro.

ALUMNOS

<u>Id</u>	Nombre	Apellidos	Clave	Contrasena	IdCurso	IdGrupo
-----------	--------	-----------	-------	------------	---------	---------

En esta tabla se almacenarán los datos de todos los alumnos que están matriculados en el centro, incluso se guardan los datos de acceso al sistema.

TEST

<u>Id</u>	IdAlumno	IdCurso	IdMateria	Nota	Tema
-----------	----------	---------	-----------	------	------

En esta tabla se guardarán los datos referentes a los test que realizan los usuarios del centro en el curso actual. Estos datos sirven para consultar estadísticas de las notas de los usuarios en cada materia, curso y tema.

PREGUNTAS

<u>Id</u>	Descripcion	IdMateria	IdCurso	Tema	ConArchivo	Archivo	NumRespuestas	ArchivoMovil
-----------	-------------	-----------	---------	------	------------	---------	---------------	--------------

<u>Url</u>

En esta tabla se almacenan las preguntas que hacen los profesores del centro para el repaso de los alumnos de los temas de cada asignatura en el curso actual. Para el uso de los archivos en el móvil, se pueden usar dos campos más, que son para que ocupe menos espacio, el archivoMovil o poner la url que le asocie con el recurso de un archivo y que sea el asociado con la pregunta.

RESPUESTAS

<u>Id</u>	Descripcion	EsCorrecta	IdPregunta
-----------	-------------	------------	------------

En esta tabla se almacenan las respuestas a las preguntas que formulan los profesores del centro para el repaso de los alumnos, poniendo solo una de las posibles respuestas como correcta.

PROFESORES

<u>Id</u>	Nombre	Apellidos	Clave	Contrasena	EsAdministrador
-----------	--------	-----------	-------	------------	-----------------

En esta tabla se guardan los datos de todos los profesores que están trabajando en el centro en el curso actual. Incluso se guardan los datos de acceso al sistema.

CURSAN

<u>IdAlumno</u>	<u>IdMateria</u>	<u>IdCurso</u>	EsPendiente
-----------------	------------------	----------------	-------------

En esta tabla se almacenan los datos referentes a las asignaturas en las que está matriculado cada alumno del centro en el curso actual, incluso las materias que tiene pendientes de otros años.

IMPARTEN

<u>IdMateria</u>	<u>IdCurso</u>	<u>IdProfesor</u>	NumTemas
------------------	----------------	-------------------	----------

En esta tabla se guardan los datos referentes a las asignaturas que imparte cada profesor.

Ya se encuentra normalizado el modelo lógico de datos.

2.3.3 Especificación de necesidades de migración de datos y carga inicial

No será prácticamente necesaria una migración de datos, pues servirán los mismos datos que se usaron en el anterior sistema desarrollado, pero con unas pequeñas variaciones.

Al no tener acceso al servidor del instituto Duques de Nájera, como cuando se desarrolló el PFC, se implantará tanto localmente como en el servidor gratuito “Hostinger” prácticamente la misma estructura de datos, salvo los nuevos campos para el móvil que se añadirán.

También es necesaria para la puesta en marcha del sistema una carga inicial de datos en todas las tablas de la base de datos para la realización de pruebas y el funcionamiento del sistema.

2.4 Definición de interfaces de usuario

2.4.1 Identificación de perfiles y diálogos.

Los distintos perfiles del sistema son profesor y alumno.

El rol de profesor lo adquirirán todos los profesores que estén dando clases en el centro.

El rol de alumno lo tendrán todos los alumnos que estén matriculados en el centro.

Cada perfil tiene un rol distinto, por lo tanto desempeñan unas funciones distintas unos de otros respectivamente.

Descomposición funcional en diálogos:

- Diálogos con los que interactúa profesor
 - Cambiar tus datos de usuario
 - Consultar estadísticas

- Diálogos con los que interactúa alumno:
 - Cambiar tus datos de usuario
 - Hacer test
 - Consultar estadísticas

2.5 Aprobación del análisis del sistema de información

Se entrega el análisis del sistema de información al tutor, para la aprobación del mismo.

3. DISEÑO DEL SISTEMA DE INFORMACIÓN

3.1 Definición de la arquitectura del sistema

3.1.1 Definición de la arquitectura

Tele-repaso Android, es una aplicación con el patrón de arquitectura MVC (Model - View - Controller), con separación por capas.

La información está online, en una base de datos MySQL, la aplicación móvil solicita unos datos al servidor mediante unos archivos PHP que se conectan con la base de datos y devuelven los datos pedidos a la aplicación móvil a través de JSON y así se descarga la información mínima necesaria, incluido los archivos de imágenes, si es que hay. También se usa el mismo proceso para actualizar datos en base de datos.

Esquema de Comunicación

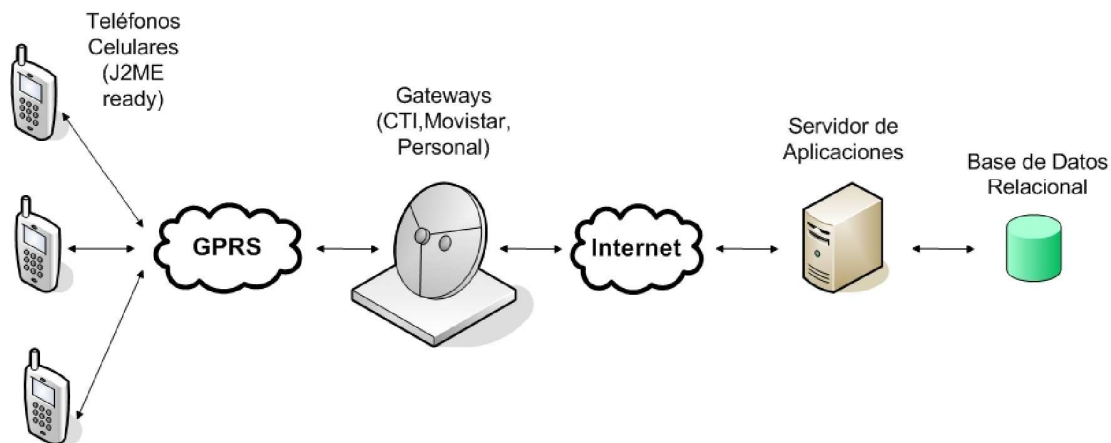


Figura 6: Diagrama de arquitectura de aplicación móvil

El principal objetivo de una arquitectura de desarrollo bajo el patrón MVC es:

Separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres capas diferentes:

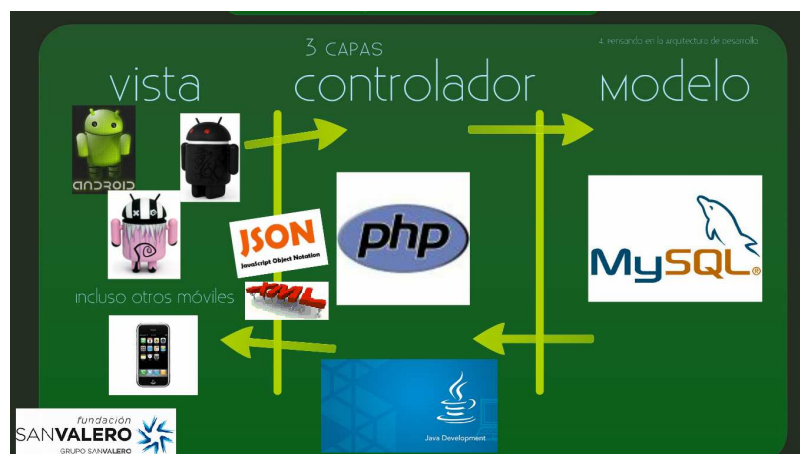


Figura 7: Diagrama de arquitectura MVC

Las ventajas del MVC son:

- **Facilita la distribución de trabajo en equipos:**
Cada grupo se encarga de desarrollar una capa de forma independiente (pequeños componentes de software separados).
- **Reutilización de los componentes:**
Una vez desarrollada la aplicación en Android MVC, podríamos realizar un cliente Web aprovechando las capas “*Controlador y Modelo*”. Sólo nos haría falta construir la “*Vista*” y se reduciría el tiempo de desarrollo y coste.
- **Mantenimiento más sencillo:**
Es más fácil mantener un desarrollo, a nivel empresarial, y resolver sus incidencias siguiendo una arquitectura MVC. Las capas son pequeños componentes de software reutilizables y fáciles de trazar para detectar errores.
- **Simplicidad en el desarrollo.**
- **Facilidad para la realización de pruebas.**
- **Escalabilidad del proyecto.**

La información está almacenada en una base de datos y la aplicación accede a un script PHP que se conecta a la base de datos y retorna los datos necesarios a la aplicación.

El usuario interactúa con el sistema a través del móvil, donde se encuentra parte de la aplicación. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja otra parte de la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al móvil que la procesa y presenta al usuario.

3.1.2 Identificación de requisitos

Los requisitos de diseño y construcción son:

- El sistema deberá acceder a una base de datos MySQL, que es en la que está realizada la base de datos de tele-repaso Android.
- El desarrollo del sistema se realizará en Java con SDK Android, e instalaré XAMPP5 para poder modificar la base de datos y usar nuestra aplicación para ir probándola, como si estuviera en el servidor.
- La información del sistema se guardará en tablas de MySQL, en un servidor gratuito ‘Hostinger’.
- En el sistema no hay tiempos exigidos de respuesta, pero el sistema debe tener unos tiempos de respuesta normales, que no sean desesperantes.

Los requisitos de operación y seguridad son:

- El acceso al sistema se restringirá mediante el uso de logins tipo clave/contraseña, que tendrá cada usuario.
- Cada cierto tiempo (recomendablemente cada semana) se hará una copia de seguridad de la base de datos del sistema, para la recuperación ante posibles fallos inesperados que se produzcan.
- Habrá confidencialidad de los datos de los usuarios.

Los requisitos de implantación son:

- El administrador, es decir, el autor del proyecto, será el encargado de realizar la implantación.
- Será indispensable tener conexión a Internet para realizar la instalación del sistema.
- Se necesitará un nombre de dominio gratuito y la contraseña del servicio, para crear la base de datos en el servidor e introducir datos para el funcionamiento y las pruebas.

3.2 Diseño de la base de datos

Las tablas y campos de que se compone la base de datos de Tele-repaso Android es:

CURSOS

<u>Id</u>	Descripcion
-----------	-------------

En esta tabla se almacenarán todos los cursos que habrá en el centro.

Campos:

- Id: código del curso, que es texto de tipo varchar(5), es clave primaria.
- Descripcion: texto del código del curso de tipo varchar(50).

GRUPOS

<u>Id</u>	Tutor	IdCurso
-----------	-------	---------

En esta tabla se almacenarán todos los grupos de que dispondrá cada curso. Aunque esta tabla no se utilizará en la aplicación en Android.

Campos:

- Id: código del grupo, que es texto de tipo varchar(5), es clave primaria.
- Tutor: texto del código del profesor que es tutor de ese grupo, es de tipo varchar(10).
- IdCurso: código del curso al que pertenece este grupo, es texto de tipo varchar(5).

MATERIAS

<u>Id</u>	Descripcion
-----------	-------------

En esta tabla se almacenarán todas las materias que se imparten.

Campos:

- Id: código de la materia, que es texto de tipo varchar(10), es clave primaria.
- Descripcion: texto del código de la materia, es de tipo varchar(10).

ALUMNOS

<u>Id</u>	Nombre	Apellidos	Clave	Contraseña	IdCurso	IdGrupo
-----------	--------	-----------	-------	------------	---------	---------

En esta tabla se almacenarán los datos de todos los alumnos que están matriculados, incluso se guardan los datos de acceso al sistema.

Campos:

- Id: código del alumno, que es numérico de tipo int(10), es clave primaria, y es un campo que se va incrementando automáticamente en número cada nuevo alumno.
- Nombre: texto del nombre del alumno, es de tipo varchar(20).
- Apellidos: texto del apellido del alumno, es de tipo varchar(30).
- Clave: texto de la clave de acceso al sistema de ese alumno, es de tipo varchar(30).
- Contraseña: texto de la contraseña de acceso al sistema de ese alumno, es de tipo varchar(30).
- IdCurso: código del curso al que pertenece este alumno, que es texto de tipo varchar(5).
- IdGrupo: código del grupo al que pertenece este alumno, que es texto de tipo varchar(5).

TEST

<u>Id</u>	IdUsuario	IdCurso	IdMateria	Nota	Tema
-----------	-----------	---------	-----------	------	------

En esta tabla se guardarán los datos referentes a los test que realizan los alumnos. Estos datos sirven para consultar estadísticas de las notas de los alumnos por materia, curso y tema.

Campos:

- Id: código del test, que es numérico de tipo int(10), es clave primaria, y es un campo que se va incrementando automáticamente en número cada nuevo test.
- IdUsuario: código del usuario que ha realizado el test, es de tipo varchar(10).
- IdCurso: texto del curso para el que realiza el test, es de tipo varchar(5).
- IdMateria: texto del código de la materia sobre la que es el test, es de tipo varchar(10).
- Nota: nota sacada en el test, es de tipo int(11).
- Tema: número del tema de la materia que realiza el test.

PREGUNTAS

<u>Id</u>	Descripcion	IdMateria	IdCurso	Tema	ConArchivo	Archivo	NumRespuestas
-----------	-------------	-----------	---------	------	------------	---------	---------------

ArchivoMovil	Url
--------------	-----

En esta tabla se almacenan las preguntas que hacen los profesores para el repaso de los alumnos de los temas de cada asignatura.

Campos:

- Id: código de la pregunta, que es numérico de tipo int(10), es clave primaria, y es un campo que se va incrementando automáticamente en número cada nueva pregunta.
- Descripcion: texto de la pregunta, es de tipo text.
- IdMateria: texto del código de la materia de la que es la pregunta, es de tipo varchar(10).
- IdCurso: código del curso a la que pertenece la pregunta, es de tipo varchar(5).
- Tema: número del tema al que pertenece la pregunta, es de tipo int(11), con valor predeterminado de 1.
- ConArchivo: campo que nos dice si la pregunta tiene archivo o no, es de tipo tinyint(1), con valor predeterminado 0.
- Archivo: nombre del archivo de la pregunta, es de tipo text.
- NumRespuestas: campo que nos dice el número de respuestas que tiene la pregunta, es de tipo tinyint(3).
- ArchivoMovil: archivo de menor tamaño para el uso en el móvil, de tipo MediumBlob.
- Url: enlace al recurso del archivo, ya sea que este en el servidor o en internet y que este asociado a la pregunta, es de tipo varchar(250).

RESPUESTAS

<u>Id</u>	Descripcion	EsCorrecta	IdPregunta
-----------	-------------	------------	------------

En esta tabla se almacenan las respuestas a las preguntas que formulan los profesores para el repaso de los alumnos, poniendo solo una de las posibles respuestas como correcta.

Campos:

- Id: código de la respuesta, que es numérico de tipo int(10), es clave primaria, y es un campo que se va incrementando automáticamente en número cada nueva respuesta.
- Descripcion: texto de la respuesta a esa pregunta, es de tipo text.
- EsCorrecta: este campo nos dice si es correcta o no la respuesta, es de tipo tinyint(1), con valor predeterminado de 0.
- IdPregunta: código de la pregunta a la que pertenece la respuesta, es de tipo int(10).

PROFESORES

<u>Id</u>	Nombre	Apellidos	Clave	Contrasena	EsAdministrador
-----------	--------	-----------	-------	------------	-----------------

En esta tabla se guardan los datos de todos los profesores que están trabajando, incluso se guardan los datos de acceso al sistema.

Campos:

- Id: código del profesor, que es de tipo varchar(10), es clave primaria.
- Nombre: texto del nombre del profesor, es de tipo varchar(20).
- Apellidos: texto del apellido del profesor, es de tipo varchar(30).
- Clave: texto de la clave de acceso al sistema de ese profesor, es de tipo varchar(30).
- Contraseña: texto de la contraseña de acceso al sistema de ese profesor, es de tipo varchar(30).
- EsAdministrador: campo que nos dice si es o no administrador, es de tipo tinyint(1), con valor predeterminado de 0.

CURSAN

<u>IdAlumno</u>	<u>IdMateria</u>	<u>IdCurso</u>	EsPendiente
-----------------	------------------	----------------	-------------

En esta tabla se almacenan los datos referentes a las asignaturas en las que está matriculado cada alumno, incluso las materias que tiene pendientes de años anteriores.

Campos:

- IdAlumno: código del alumno que ha realizado el test, es de tipo int(10).
- IdMateria: texto del código de la materia de la que es la pregunta, es de tipo varchar(10).
- IdCurso: código del curso a la que pertenece la pregunta, es de tipo varchar(5).
- EsPendiente: campo que nos dice si es una asignatura pendiente o no, es de tipo tinyint(1), con valor predeterminado de 0.

El conjunto de campos IdAlumno, IdMateria, IdCurso son clave primaria.

IMPARTEN

<u>IdMateria</u>	<u>IdCurso</u>	<u>IdProfesor</u>	NumTemas
------------------	----------------	-------------------	----------

En esta tabla se guardan los datos referentes a las asignaturas que dan cada profesor.

Campos:

- IdMateria: texto del código de la materia de la que imparte clase el profesor, es de tipo varchar(10).
- IdCurso: código del grupo en el que imparte clase ese profesor en esa materia, que es texto de tipo varchar(5).
- IdProfesor: código del profesor que imparte clase, que es de tipo varchar(10).
- NumTemas: número de temas que tiene esa materia, es de tipo tinyint(3), con valor predeterminado de 25.

El conjunto de campos IdMateria, IdCurso, IdProfesor son clave primaria.

3.3 Diseño de la migración y carga inicial de datos

El servidor gratuito contendrá la base de datos de Tele-repaso en MySQL. Por lo tanto la migración de datos existentes se realizaría con los pasos de la aplicación web que se realizó anteriormente, y al principio de cada nuevo curso, para tener los datos actualizados de alumnos, profesores, grupos, cursos, materias, etc., pero como no tenemos acceso al Servidor ‘Strato’, que es donde está implantado la aplicación web, se usará el servidor gratuito “Hostinger”, y se creará la base de datos con los cambios incorporados. El traspaso de datos será automático con la aplicación que se desarrolló en PHP en el anterior PFC y pasará los datos que necesito a Tele-repaso Android, pero lo hacemos de forma independiente.

Los registros de las tablas cursos, grupos, materias, test, profesores, alumnos, cursan e imparten serán borrados automáticamente por la aplicación, justo antes de pasar los datos.

Los registros de las tablas de preguntas y respuestas se mantendrán todos los años.

No será prácticamente necesaria una migración de datos, pues servirán los mismos datos que se usaron en el anterior sistema desarrollado, pero con unas pequeñas variaciones.

Al no tener acceso al servidor del instituto Duques de Nájera, se implantará prácticamente la misma estructura de datos, salvo los nuevos campos para el móvil que se añadirán.

También es necesaria para la puesta en marcha del sistema una carga inicial de datos en todas las tablas de la base de datos para la realización de pruebas y el funcionamiento del sistema.

3.4 Especificación técnica del plan de pruebas

Los Niveles de Pruebas son:

- **Pruebas unitarias:** Se trata de las pruebas formales que permiten declarar que un módulo está listo y terminado. Puede abarcar desde un módulo hasta un grupo de módulos. Comprenden las verificaciones asociadas a cada componente del Sistema de Información. Se desarrollarán pruebas tanto de caja negra (sabiendo cómo se ha implementado el sistema para que entre en cada uno de los bucles y cumpla o no cada una de las condiciones) como de caja blanca (sin saber cómo está construido el sistema). Estas pruebas se realizarán seguidamente después de desarrollar cada uno de los componentes, verificando que no se producen fallos y en caso de que se produzcan, proceder al correspondiente estudio y corrección del error.
- **Pruebas de integración:** Implican una progresión ordenada de pruebas que van desde los componentes o módulos y que culminan en el Sistema completo. Se utilizará el modelo de integración no incremental, es decir, se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.
- **Pruebas del Sistema:** son pruebas de integración del sistema de información completo. Se comprueba :
 - El cumplimiento de todos los requisitos funcionales, considerando el producto software final al completo en un entorno de Sistema.
 - El funcionamiento y rendimiento en las interfaces.
 - La seguridad que hay en el sistema, para que accedan solo usuarios válidos y sólo a las partes de la aplicación que tienen permiso.
- **Prueba de implantación:** El software totalmente ensamblado se prueba como un todo para comprobar si cumple los requisitos funcionales. Verificamos que el sistema funciona correctamente poniéndolo en el equipo de desarrollo del sistema como servidor, y se hacen pruebas sobre él, con otros equipos. La base de datos estará en vez de en el servidor ‘Strato’, en el servidor gratuito, al igual que los archivos

necesarios para la aplicación, para comprobar la implantación del sistema. Una vez implantado Tele-repaso Android se le harán las mismas pruebas que la del sistema.

- **Prueba de aceptación:** Es la prueba planificada y organizada formalmente para determinar si se cumplen los requisitos de aceptación marcados por el cliente. Está enfocada hacia la prueba de los requisitos de usuario especificados. Está considerada como la fase final del proceso para crear una confianza en que el producto es el apropiado para su uso en explotación. La aplicación ya la podrán usar alumnos y profesores, esta es la mejor manera de probarlo, ya que accederán muchos usuarios a la vez, tanto alumnos, como profesores y es cuando mejor se apreciará cual será el rendimiento en condiciones de límite y de sobrecarga en verdadera ejecución real. Esta prueba se realizará cuando se implante en un centro la aplicación móvil.

Los requisitos básicos serán:

- Las pruebas unitarias, de integración y del sistema, se realizarán en el equipo en el que se está desarrollando el proyecto. Estas pruebas serán realizadas por el autor del proyecto, verificando que todos los requisitos establecidos se cumplan.
- Para la prueba de implantación y de aceptación, la base de datos estará ya en el servidor gratuito “Hostinger”, y se necesitará que se tenga conexión a Internet. La prueba de implantación del sistema se realizará después de haber realizado las pruebas del sistema. La prueba de aceptación del sistema se llevará a cabo una vez que el sistema esté construido totalmente y se hayan ejecutado y evaluado las pruebas anteriores, habiendo dado los resultados esperados, es decir, sin errores. Las pruebas de implantación serán realizadas por el autor del proyecto, mientras que las pruebas de aceptación serán realizadas por los usuarios de la aplicación (alumnos y profesores), verificando que todos los requisitos establecidos se cumplen, cuando algún centro quiera usar la aplicación móvil.
- Será necesario un servidor que contendrá la base de datos (ya sea localmente mediante apache o el servidor gratuito) y un móvil con Android e internet que hará de cliente.
- Todas las pruebas se realizarán con el sistema operativo Windows, XAMPP5 y Java con SDK Android.

Una vez que estén comprobados los datos de la base de datos, se pasarán a hacer las pruebas de cada interfaz, haciendo al menos una prueba de cada acción que se pueda hacer, moviéndonos por todos los sitios posibles y buscando errores para resolverlos cuanto antes.

Las pruebas que se realizarán en cada interfaz serán:

- En la pantalla de identificación:
 - se intentará acceder al sistema con los campos vacíos de clave y contraseña en cualquier tipo de usuario.
 - se intentará acceder al sistema a cada uno de los usuarios con datos incorrectos de clave y contraseña.
 - se intentará acceder al sistema a cada uno de los usuarios con datos correctos de clave y contraseña.
- Pantalla del usuario alumno:
 - se intentará acceder a hacer test, cambiar tus datos, consultar estadísticas y volver.
- Modificar tus datos del usuario:
 - se modificarán los datos y se pulsará enviar para comprobar si se guardan los cambios o no se guardan si es que la contraseña actual no se corresponde con la que hay.

- se dejará algún campo vacío para comprobar si nos muestra un aviso de campo vacío.
- Realización del test:
 - Comprobaremos que no nos deja hacer test, si es que no hay 5 o más preguntas de la asignatura, curso y tema seleccionado
 - seleccionamos las respuestas a las preguntas y pulsamos enviar respuestas.
- Corrección del test:
 - comprobaremos si cuenta bien las respuestas correctas.
 - intentaremos volver y hacer otro test del mismo tema.
- Consultar estadísticas alumno:
 - nos debe mostrar las estadísticas de los test realizados de las asignaturas que cursa el alumno.
 - intentaremos volver.

Para el usuario profesor serán las mismas pruebas con la diferencia de que no puede realizar test, y que al consultar estadísticas selecciona los parámetros de asignatura, curso y tema, para que nos muestre las estadísticas de la media de los test realizado por los alumnos de los parámetros seleccionados.

3.5 Aprobación del diseño del sistema de información

En esta tarea se entrega el diseño del sistema de información al tutor, para la aprobación final del mismo.

4 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

4.1 Preparación del entorno de generación y construcción

4.1.1 Preparación del entorno de construcción

Para empezar deberemos instalar la aplicación XAMP, que será la que se utilice, en el equipo que usemos para el desarrollo del proyecto en local. Con XAMPP tendremos PHP, Apache, MySQL y phpMyAdmin, que hará sobre todo de servidor de base de datos de la aplicación. También instalaremos Eclipse y la SDK Android, que nos ayudará a generar el código necesario para la programación de la aplicación tele-repaso Android.

4.1.2 Implantación de la base de datos física

Trasformamos la base de datos del sistema a SQL, y queda:

```
--
-- Estructura de tabla para la tabla `alumnos`
--

DROP TABLE IF EXISTS `alumnos`;
CREATE TABLE `alumnos` (
  `id` int(10) NOT NULL auto_increment,
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default "",
  `apellidos` varchar(30) collate latin1_general_ci NOT NULL default "",
  `clave` varchar(30) collate latin1_general_ci NOT NULL default "",
  `contrasena` varchar(30) collate latin1_general_ci NOT NULL default "",
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default "",
  `idgrupo` varchar(5) collate latin1_general_ci NOT NULL default "",
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

-----

--
-- Estructura de tabla para la tabla `cursan`
--

DROP TABLE IF EXISTS `cursan`;
CREATE TABLE `cursan` (
  `idalumno` int(10) NOT NULL default '0',
  `idmateria` varchar(10) collate latin1_general_ci NOT NULL default "",
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default "",
  `espendiente` tinyint(1) NOT NULL default '0',
  PRIMARY KEY (`idalumno`,`idmateria`,`idcurso`)
) ENGINE=InnoDB;

-----

--
```

```
-- Estructura de tabla para la tabla `cursos`  
--
```

```
DROP TABLE IF EXISTS `cursos`;  
CREATE TABLE `cursos` (  
  `id` varchar(5) collate latin1_general_ci NOT NULL default "",  
  `Descripcion` varchar(50) collate latin1_general_ci NOT NULL default "",  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `grupos`  
--
```

```
DROP TABLE IF EXISTS `grupos`;  
CREATE TABLE `grupos` (  
  `id` varchar(5) collate latin1_general_ci NOT NULL default "",  
  `tutor` varchar(10) collate latin1_general_ci NOT NULL default "",  
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default "",  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `imparten`  
--
```

```
DROP TABLE IF EXISTS `imparten`;  
CREATE TABLE `imparten` (  
  `idmateria` varchar(10) collate latin1_general_ci NOT NULL default "",  
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default "",  
  `idprofesor` varchar(10) collate latin1_general_ci NOT NULL default "",  
  `numtemas` tinyint(3) NOT NULL default '25',  
  PRIMARY KEY (`idmateria`,`idcurso`,`idprofesor`)  
) ENGINE=InnoDB;
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `materias`  
--
```

```
DROP TABLE IF EXISTS `materias`;  
CREATE TABLE `materias` (  
  `id` varchar(10) collate latin1_general_ci NOT NULL default "",  
  `descripcion` varchar(50) collate latin1_general_ci NOT NULL default "",  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
-----
```

```
--
-- Estructura de tabla para la tabla `preguntas`
--

DROP TABLE IF EXISTS `preguntas`;
CREATE TABLE `preguntas` (
  `id` int(10) NOT NULL auto_increment,
  `descripcion` text collate latin1_general_ci NOT NULL,
  `idmateria` varchar(10) collate latin1_general_ci NOT NULL default '',
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default '',
  `tema` int(10) NOT NULL default '1',
  `conarchivo` tinyint(1) NOT NULL default '0',
  `archivo` text collate latin1_general_ci NOT NULL,
  `numrespuestas` int(10) NOT NULL default '4',
  `archivomovil` mediumblob,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
```

```
-----

--
-- Estructura de tabla para la tabla `profesores`
--

DROP TABLE IF EXISTS `profesores`;
CREATE TABLE `profesores` (
  `id` varchar(10) collate latin1_general_ci NOT NULL default '',
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',
  `apellidos` varchar(30) collate latin1_general_ci NOT NULL default '',
  `clave` varchar(30) collate latin1_general_ci NOT NULL default '',
  `contrasena` varchar(30) collate latin1_general_ci NOT NULL default '',
  `esadministrador` tinyint(1) NOT NULL default '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
```

```
-----

--
-- Estructura de tabla para la tabla `respuestas`
--

DROP TABLE IF EXISTS `respuestas`;
CREATE TABLE `respuestas` (
  `id` int(10) NOT NULL auto_increment,
  `descripcion` text collate latin1_general_ci NOT NULL,
  `escorrecta` tinyint(1) NOT NULL default '0',
  `idpregunta` int(10) NOT NULL default '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
```

```
-----

--
-- Estructura de tabla para la tabla `test`
--
```

```
DROP TABLE IF EXISTS `test`;  
CREATE TABLE `test` (  
  `id` int(10) NOT NULL auto_increment,  
  `idalumno` int(10) NOT NULL default '0',  
  `idcurso` varchar(5) collate latin1_general_ci NOT NULL default '',  
  `idmateria` varchar(10) collate latin1_general_ci NOT NULL default '',  
  `tema` int(10) NOT NULL default '1',  
  `nota` int(11) NOT NULL default '5',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB ;
```

Figura 8: Base de datos Tele-repaso Android en SQL

InnoDB es una tecnología de almacenamiento de datos de fuente abierta para MySQL, incluido como formato de tabla estándar en todas las distribuciones de MySQL a partir de las versiones 4.0. Su característica principal es que soporta transacciones de tipo ACID y bloqueo de registros e integridad referencial. *InnoDB* ofrece una fiabilidad y consistencia muy superior a *MyISAM* la anterior tecnología de tablas de MySQL, si bien el mejor rendimiento de uno u otro formato dependerá de la aplicación específica.

Para implantar la estructura de la base de datos, ejecutaremos estas sentencias SQL en phpMyAdmin, para que cree las tablas necesarias del sistema, donde se insertarán los datos de alumnos, profesores, preguntas, respuestas, cursos, materias, test, grupos, imparten y cursan. Para ello entraremos a phpMyAdmin (desde el navegador poniendo: <http://localhost/phpmyadmin/>), seleccionamos en Language “spanish (es-utf-8)”, en cotejamiento de las conexiones MySQL “utf8_spanish2_ci”, ponemos como nombre de la base de datos telerepaso, seleccionamos alado de ello “utf8_spanish2_ci” y pulsamos crear. Tendremos la base de datos ya creada, pero sin la estructura que queremos. Ahora entraremos a la base de datos telerepaso, y pinchando en SQL, pegaremos el texto SQL de la estructura, o pulsaremos examinar y seleccionaremos el archivo que contiene la estructura de las tablas. Ya solo nos queda pulsar continúe, y ya tenemos la base de datos creada.

Ahora solo nos quedaría insertar los datos que queramos en la base de datos, para hacer pruebas después de que hayamos implementado el sistema.

4.2 Generación del código de los componentes y procedimientos

4.2.1 Generación del código de componentes

Con Eclipse y la SDK Android generamos todo el código del sistema.

Para visualizar lo que vamos haciendo, podemos ejecutarlo desde eclipse o conectando el móvil, pero teniendo almacenado los archivos PHP del proyecto dentro de la carpeta htdocs del programa XAMPP.

En cada archivo del sistema, hay diferente código generado, que realiza una acción completa, o parte de las acciones posibles por los usuarios. El proyecto se compone básicamente de archivos XML, actividades y Javas, además de alguna imagen.

No voy a colocar aquí todo el código de todos los archivos generado, debido a que serían muchísimas hojas, sino que solo una parte de todo ello, donde se ve el funcionamiento general.

4.2.2 Código destacable

4.2.2.1 Pantalla principal y de identificación

El contenido por partes del código de la actividad “TelerepasoAndroid.Java” y sus respectivos comentarios son:

```
public class TelerepasoAndroid extends Activity {

    private Button botonEnviar;
    private EditText clave;
    private EditText contrasena;
    private Spinner tipoUsuario;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_telerepaso_android);
```

Con ello salvamos lo que hay en la instancia para cuando se vuelva a ella tener los datos, y asignamos que la vista que tiene que cargar es la del xml que le decimos.

```
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
```

Con ello permitimos que el móvil tenga políticas de poder realizar cualquier acción, como conectarse a internet, etc.

```
botonEnviar = (Button) findViewById(R.id.botonEnviar);
clave=(EditText) findViewById(R.id.editTextClave);
contrasena=(EditText) findViewById(R.id.editTextContrasena);
tipoUsuario=(Spinner) findViewById(R.id.spinnerTipoUsuario);

botonEnviar.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        boolean existeusuario=false;
        Intent intent=new Intent(TelerepasoAndroid.this,EntradaActivity.class);
        if(tipoUsuario.getSelectedItem().toString().equals("Alumnos") ){
            Alumno alumno=new Alumno();
            alumno=AccesoBaseDatos.getAlumno(clave.getText().toString(),
            contrasena.getText().toString(), tipoUsuario.getSelectedItem().toString());
            if(alumno!=null){
                Bundle bundle = new Bundle();
                bundle.putString("tipousuario", tipoUsuario.getSelectedItem().toString());
                bundle.putString("nombre", alumno.getNombre());
                bundle.putString("apellidos", alumno.getApellidos());
                bundle.putString("curso", alumno.getCurso());
                bundle.putString("grupo", alumno.getGrupo());
                bundle.putInt("idalumno", alumno.getId());
                bundle.putString("clave",clave.getText().toString());
                bundle.putString("contrasena",contrasena.getText().toString());
                existeusuario=true;
            }
        }
    }
});
```

```

                                intent.putExtras(bundle);
                                startActivity(intent);
                            }
                        }else{
                            Profesor profesor=new Profesor();
                            profesor=AccesoBaseDatos.getProfesor(clave.getText().toString(),
                                contrasena.getText().toString(), tipoUsuario.getSelectedItem().toString());
                            if(profesor!=null){
                                Bundle bundle = new Bundle();
                                bundle.putString("tipousuario", tipoUsuario.getSelectedItem().toString());
                                bundle.putString("nombre", profesor.getNombre());
                                bundle.putString("apellidos", profesor.getApellidos());
                                bundle.putString("idprofesor", profesor.getId());
                                bundle.putString("clave",clave.getText().toString());
                                bundle.putString("contrasena",contrasena.getText().toString());
                                existeusuario=true;
                                intent.putExtras(bundle);
                                startActivity(intent);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Asociamos a las variables de la clase algunos elementos de la interfaz gráfica (XML), como botones, textos, etc. Luego creamos el evento de al pulsar el botón de enviar, para que llame a una de las funciones y busque en base de datos al usuario ya sea alumno o profesor, con “getAlumno” o “getProfesor” respectivamente, y si existe devuelve los datos correspondientes. Luego creamos un “intent” para poder pasar de esa actividad a la de entrada del usuario en el caso de que exista el usuario y se haya validado correctamente. Dependiendo de si es un tipo de usuario de alumno o de profesor, se pasan los parámetros de los datos del usuario a la actividad a la que se va, para usarlos. También hemos creado una variable para que controle si existe el usuario o no.

```

if(clave.getText().length()==0)
    Toast.makeText(getApplicationContext(), "La clave es obligatoria",
        Toast.LENGTH_SHORT).show();
else if(contrasena.getText().length()==0)
    Toast.makeText(getApplicationContext(), "La contraseña es obligatoria",
        Toast.LENGTH_SHORT).show();
else if( ((String)(tipoUsuario.getSelectedItem())).length()==0)
    Toast.makeText(getApplicationContext(), "El tipo de usuario es obligatorio",
        Toast.LENGTH_SHORT).show();
else{
    if(!existeusuario){
        Toast.makeText(getApplicationContext(), "La clave, contraseña y tipo de usuario no es
            correcto", Toast.LENGTH_SHORT).show();
    }
}
}
});
}

```

Muestra un mensaje en pantalla si hemos dejado algún dato vacío en la clave y contraseña, o si no nos validamos correctamente.

```

    public boolean onKeyDown(int keyCode, android.view.KeyEvent event) {
        if(keyCode == KeyEvent.KEYCODE_BACK)
            finish();
        return true;
    };

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        finish();
    }
}

```

Para acabar en este último trozo de código de la clase, lo que hace es que si pulsamos el botón de atrás del móvil, finaliza la actividad.

4.2.2.2 Acceso a base de datos

El archivo que se ocupa del acceso a la base de datos es “AccesoBaseDatos.Java”, junto a los archivos PHP que se encuentran en el servidor “Hostinger” o en local, como es por ejemplo “getAlumno.php”.

Ahora vemos parte del contenido de “AccesoBaseDatos.Java” y lo explico:

```

public class AccesoBaseDatos {

    private static final String IP_SERVER = "telerepasoandroid.zz.mu";
    private static InputStream is;

    public static String getIpServer() {
        return IP_SERVER;
    }
}

```

Creamos la clase “AccesoBaseDatos”, y dos variables privadas y estáticas, una es la “IP_SERVER” que es una constante que contiene la dirección IP del servidor, donde están alojados los archivos PHP y la base de datos.

```

    public static Alumno getAlumno(String clave,String contrasena){
        boolean esta = false;
        String result = "";
        Alumno alumno=null;
        ArrayList<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>();
        nameValuePairs.add(new BasicNameValuePair("clave",clave));
        nameValuePairs.add(new BasicNameValuePair("contrasena",contrasena));

        try{
            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new HttpPost("http://" + IP_SERVER +
"/telerepaso/getAlumno.php");
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();

```

```

        is = entity.getContent();
    } catch (Exception e) {
        Log.e("log_tag", "Error in http connection "+e.toString());
    }
    try {
        BufferedReader reader = new BufferedReader(new
InputStreamReader(is,"iso-8859-1"),8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();

        result=sb.toString();
    } catch (Exception e) {
        Log.e("log_tag", "Error converting result "+e.toString());
    }
    try {
        if(result!=null){
            JSONArray jArray = new JSONArray(result);
            for(int i=0;i<jArray.length();i++){
                JSONObject json_data = jArray.getJSONObject(i);
                String clav = json_data.getString("clave");
                String contrasen = json_data.getString("contrasena");
                String nombre = json_data.getString("nombre");
                String apellidos = json_data.getString("apellidos");
                if(clav.equalsIgnoreCase(clav) &&
contrasena.equalsIgnoreCase(contrasen)){
                    alumno=new Alumno(json_data.getInt("id"),
json_data.getString("nombre"), json_data.getString("apellidos"),json_data.getString("idcurso"),
json_data.getString("idgrupo"), json_data.getString("clave"),json_data.getString("contrasena"));
                    esta = true;
                }
            }
        }
    } catch (JSONException e) {
        Log.e("log_tag", "Error parsing data "+e.toString());
    }
    return alumno;
}

```

La función “getAlumno”, pasa como parámetros la clave y contraseña al archivo “getAlumno.php” del servidor, y devuelve los datos del alumno en su clase correspondiente, si es que existe, sino devuelve “null”.

```

<?php
mysql_connect("mysql.hostinger.es","u191754633_teler","123456raul");
mysql_select_db("u191754633_telerepa");
$query=mysql_query("SELECT * FROM alumnos WHERE clave='".$$_REQUEST['clave']."'
and contrasena='".$$_REQUEST['contrasena']."'");
while($e=mysql_fetch_array($query)){
    $output[]=$e;
}
print(json_encode($output));

```

```
return $output;  
mysql_close();  
?>
```

Este código corresponde al archivo “getAlumno.php” que se encuentra en el servidor. Los parámetros de “mysql.hostinger.es”, “u191754633_teler” y “123456raul” son host, usuario y contraseña respectivamente. Si queremos conectarnos localmente sería “localhost” y “root”. Luego se conecta a la base de datos “u191754633_telerepa” para el servidor de hostinger, pero si queremos conectarnos localmente sería a la base de datos de “telerepaso”. Todo esto es para todos los ficheros PHP del sistema. Lo que hace este fichero es devolver los datos del alumno que tiene esa clave y contraseña.

4.3 Ejecución de las pruebas unitarias

4.3.1 Preparación, realización y evaluación de las pruebas unitarias

No necesitaremos preparación para realizar estas pruebas unitarias. Probaremos cada una de las interfaces que vamos creando, y comprobaremos que funcionan correctamente independientemente del resto. Son pruebas que afectan a una parte de código concreto.

Después de realizar las pruebas de todas las interfaces independientemente, su evaluación es positiva, ya que ha funcionado correctamente el sistema y no se han producido fallos.

4.4 Ejecución de las pruebas de integración

4.4.1 Preparación, realización y evaluación de las pruebas de integración

No necesitaremos preparación para realizar estas pruebas de integración. Usaremos el modelo de integración no incremental, de manera que iremos probando la interacción entre componentes.

Se prueba que funciona correctamente todo en el conjunto de la aplicación, por ejemplo entrar a unas pantallas y a otras, en distinto orden.

Después de realizar las pruebas de interacción entre las interfaces, su evaluación es positiva, ya que ha funcionado correctamente el sistema.

4.5 Ejecución de las pruebas del sistema

4.5.1 Preparación, realización y evaluación de las pruebas del sistema

Se evalúa todo el sistema completamente, tanto individualmente cada pantalla como en su conjunto, probando la funcionalidad entre ellas.

Se han realizado pruebas de identificación de usuario, mensajes que tienen que salir, pantallas a las que va, acciones que realiza, cambiar datos de usuario, hacer test, consultar estadísticas, etc. tanto para usuario alumno como profesor.

Los resultados obtenidos en las pruebas, son los esperados, se cumplen todos los requisitos funcionales, el rendimiento es correcto, y la seguridad del sistema es suficiente.

4.6 Interfaces finales de usuario

4.6.1 Pantalla principal de identificación

La pantalla principal de identificación de Tele-repaso Android tendrá esta interfaz:



Figura 9: Pantalla principal de identificación

- El usuario debe rellenar los dos campos del formulario y elegir el tipo de usuario que es.
 - Si no rellena alguno de ellos se mostrará la siguiente advertencia en pantalla: “La clave es obligatoria”, “La contraseña es obligatoria” o “El tipo de usuario es obligatorio”.
 - Si la clave de usuario, la contraseña o el tipo de usuario son incorrectos, se mostrará la siguiente advertencia en pantalla: “La clave, contraseña y tipo de usuario no es correcto”.

Desde esta pantalla se podrá acceder a la pantalla principal de entrada del usuario, una vez autenticado.

4.6.2 Pantalla de entrada del usuario profesor

La pantalla de entrada del usuario profesor cuando ha accedido ya al sistema, tendrá esta interfaz:



Figura 10: Pantalla entrada del usuario profesor

Esta interfaz del profesor no tiene comportamiento dinámico, ya que solo es la pantalla donde se eligen las operaciones a realizar por éste usuario.

Las acciones que podrá realizar el profesor en esta interfaz son:

1. Consultar estadísticas.
2. Cambiar tus datos.
3. Volver.

4.6.3 Pantalla de entrada del usuario alumno

La pantalla del usuario alumno cuando ha accedido ya al sistema, tendrá esta interfaz:



Figura 11: Pantalla entrada del usuario alumno

Esta interfaz del alumno no tiene comportamiento dinámico, ya que solo es la pantalla donde se eligen las operaciones a realizar por éste usuario.

Las acciones que podrá realizar el alumno en esta interfaz son:

1. Hacer test.
2. Consultar estadísticas.
3. Cambiar tus datos.
4. Volver.

4.6.4 Pantalla de cambiar tus datos

Si el usuario es un profesor o alumno, cambiar tus datos tendrá esta interfaz:



Figura 12: Pantalla cambiar tus datos de profesores o alumnos

La única acción que se puede realizar en esta interfaz es volver a la pantalla principal del usuario o enviar información, para almacenar los datos.

Si tenemos algún dato sin rellenar al pulsar el botón “enviar información”, se nos mostrará un mensaje de advertencia de campo vacío, como éste: “No puede estar vacío alguno de los datos a cambiar”. Si la contraseña actual no coincide con la introducida en su campo, se nos mostrará “La contraseña anterior no coincide con la actual”.

Al pulsar el botón “Enviar Información” se actualizarán los datos, y si se han cambiado los datos correctamente nos mostrará el mensaje de “Se han actualizado los datos”.

4.6.5 Pantalla de hacer test

La primera interfaz para hacer test los alumnos, será esta:



Figura 13: Pantalla elegir parámetros test

Las dos únicas acciones que se podrán realizar en esta interfaz son volver a la pantalla principal del usuario o “Hacer Test”.

Al pulsar el botón “Hacer test”, si no hay 5 o más preguntas de la asignatura, curso y tema seleccionado se nos mostrará un mensaje de advertencia de “No hay preguntas para ese curso, asignatura y tema, seleccione otro”.

Después de pulsar el botón “Hacer test” si hay 5 o más preguntas pasaremos a realizar el test y se mostrará lo siguiente:



Figura 14: Pantalla realizar test

La única acción que se puede realizar en esta interfaz es “Enviar respuestas” con lo que pasaremos a la pantalla de corrección del test. Si al pulsar el botón tenemos alguna respuesta sin seleccionar nos avisa diciendo “Debes seleccionar alguna respuesta de cada pregunta”. Después de haber terminado de realizar el test pasaremos a la siguiente pantalla de corrección:

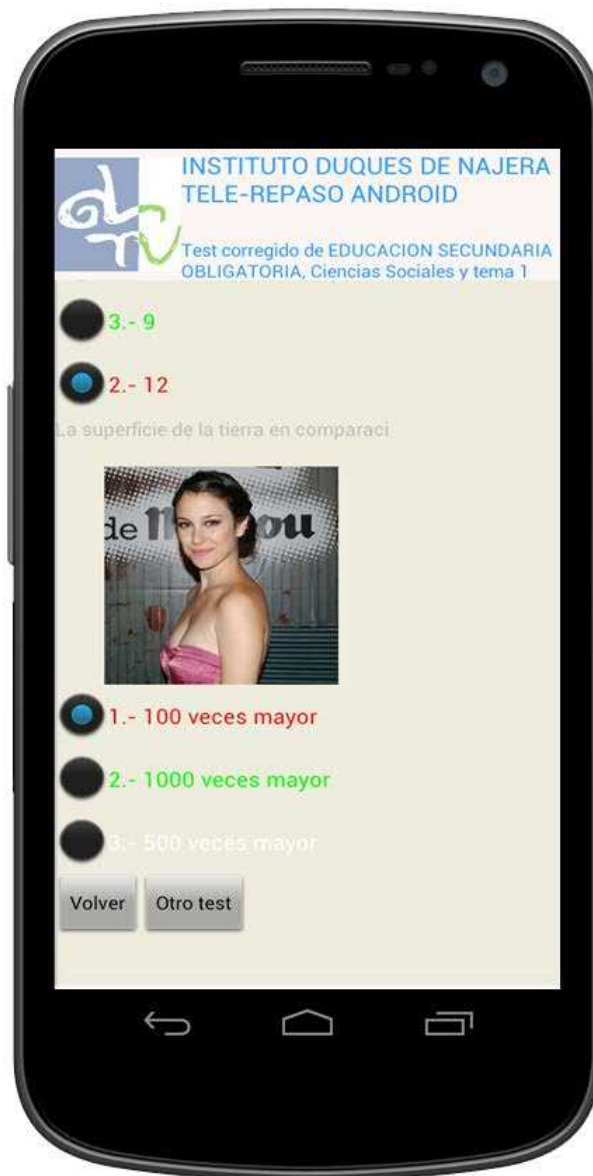


Figura 15: Pantalla corrección test

Primeramente nos muestra en la pantalla un mensaje diciéndonos la nota obtenida. Esta interfaz no tiene comportamiento dinámico. Las únicas acciones que se pueden realizar en esta interfaz son volver a la pantalla principal del usuario u “Otro test” con lo que pasaremos a la pantalla de realizar el test, pero con otras preguntas aleatorias del mismo curso, asignatura y tema.

4.6.6 Pantalla de consultar estadísticas

La interfaz de consultar estadísticas para los alumnos será la siguiente:



The image shows a smartphone screen displaying the 'INSTITUTO DUQUES DE NAJERA TELE-REPASO ANDROID' app. The screen shows a table of student statistics for 'Estadísticas de 1ESO, Ciencias Sociales y tema 1'. The table has three columns: 'Alumno', 'Test realizados', and 'Nota media'. The data is as follows:

Alumno	Test realizados	Nota media
ADINA ANDREEA ANGHELACHE	0	
ADRIAN CASTROVIEJO PARRAGA	0	
ADRIAN FLA	0	
AITOR OLIVAN FERNANDEZ	0	
ALBA BLANCO HIJAZO	0	
ALBA MOYA HERCE	13	2.9231
ALBERTO VIDARTE SAN PEDRO	0	
ALEJANDRA MONTERO SAIZ-AJA	0	
ALEJANDRO CRUZ SANTA	0	
ALEJANDRO NICOLAS EIDEL	0	

At the bottom left of the table, there is a 'Volver' button. The smartphone's navigation bar at the bottom shows the back, home, and recent apps icons.

Figura 16: Pantalla consultar estadísticas de alumno

Esta interfaz no tiene comportamiento dinámico, ya que solo es una pantalla donde se pueden ver las estadísticas.

La única acción que se puede realizar en esta pantalla es volver a la pantalla principal del usuario.

La primera pantalla de consultar estadísticas para profesores es la siguiente:



Figura 17: Pantalla consultar estadísticas1 de profesores

Elegiremos la asignatura, curso y tema, y las únicas acciones que se pueden realizar en esta interfaz son volver a la pantalla principal del usuario, y pulsar el botón “Mostrar Estadísticas” con lo que pasaremos a ver las estadísticas de los alumnos de la asignatura, curso y tema seleccionado. El profesor puede elegir cualquiera de las asignaturas que imparte.

La segunda y última pantalla de consultar estadísticas para profesores es la siguiente:



INSTITUTO DUQUES DE NAJERA TELE-REPASO ANDROID		
Estadísticas de 1ESO, Ciencias Sociales y tema 1		
Alumno	Test realizados	Nota media
ADINA ANDREEA ANGHELACHE	0	
ADRIAN CASTROVIEJO PARRAGA	0	
ADRIAN FLA	0	
AITOR OLIVAN FERNANDEZ	0	
ALBA BLANCO HIJAZO	0	
ALBA MOYA HERCE	13	2.9231
ALBERTO VIDARTE SAN PEDRO	0	
ALEJANDRA MONTERO SAIZ-AJA	0	
ALEJANDRO CRUZ SANTA	0	
ALEJANDRO NICOLAS EIBEL	0	

Volver

Figura 18: Pantalla consultar estadísticas2 de profesor

Esta interfaz no tiene comportamiento dinámico, ya que solo es una pantalla donde se pueden ver las estadísticas.

Desde esta pantalla se sólo se puede volver a la pantalla principal del usuario.

4.7 Aprobación del sistema de información

4.7.1 Presentación y aprobación del sistema de información

Se muestra el funcionamiento del sistema al tutor, y tendrá que dar el visto bueno para continuar.

5 IMPLANTACIÓN, PRUEBAS Y ACEPTACIÓN DEL SISTEMA

5.1 Establecimiento del plan de implantación

5.1.1 Definición del plan de implantación

No se necesita formación para realizar la implantación, vale con sólo seguir una serie de pasos muy sencillos. También se deberá hacer copias de seguridad de la base de datos cada cierto tiempo por si se borran datos del servidor.

El primer paso que tenemos que hacer es crear el servidor gratuito “Hostinger” y luego entrar mediante el email y la contraseña, que son rules166@gmail.com y “123456raul”, respectivamente.

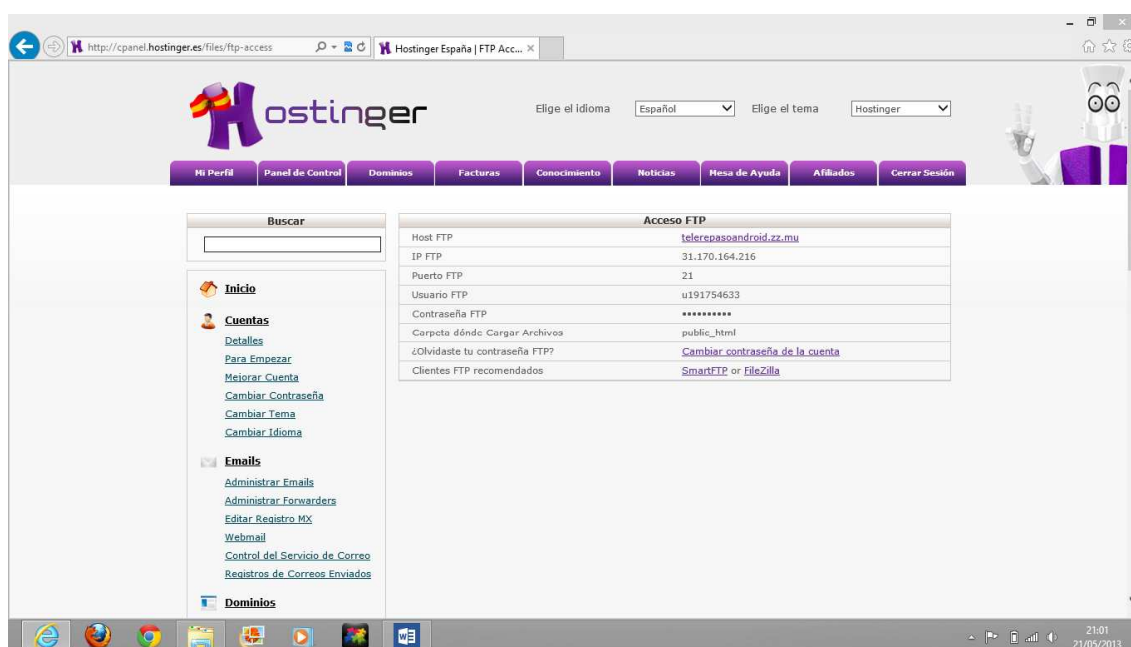


Figura 19: Pantalla datos servidor Hostinger

Los dos siguientes pasos no importa el orden, así que proponemos acceder al ftp, para copiar todos los archivos PHP que teníamos en local, que los usábamos para acceder a la base de datos de “Hostinger”, y lo que hacen es devolver la consulta solicitada al móvil, para que lo recoja la aplicación móvil y los use mostrando en pantalla los datos o lo que sea necesario. Los archivos los colocamos dentro de la carpeta telerepaso, que lo crearemos dentro del directorio public, que se encuentra en la raíz del ftp. Los datos de los archivos PHP de acceso a la base de datos cambian dependiendo se estén usando en local o en “Hostinger”.

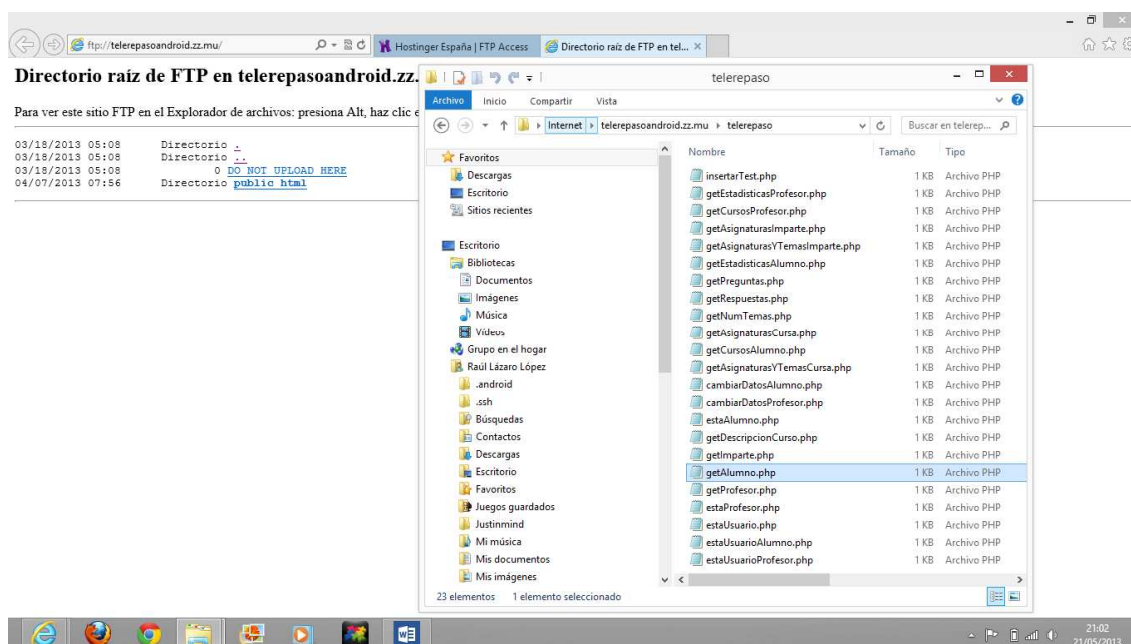


Figura 20: Pantalla ftp de servidor Hostinger

El tercer paso es acceder a phpMyAdmin del servidor “Hostinger” usando la contraseña “123456raul” y lanzar los scripts de la base de datos, para tener la estructura y datos que se necesita en la aplicación móvil para poder leer, sobrescribir, insertar o incluso borrar datos desde la aplicación móvil de Tele-repaso Android.

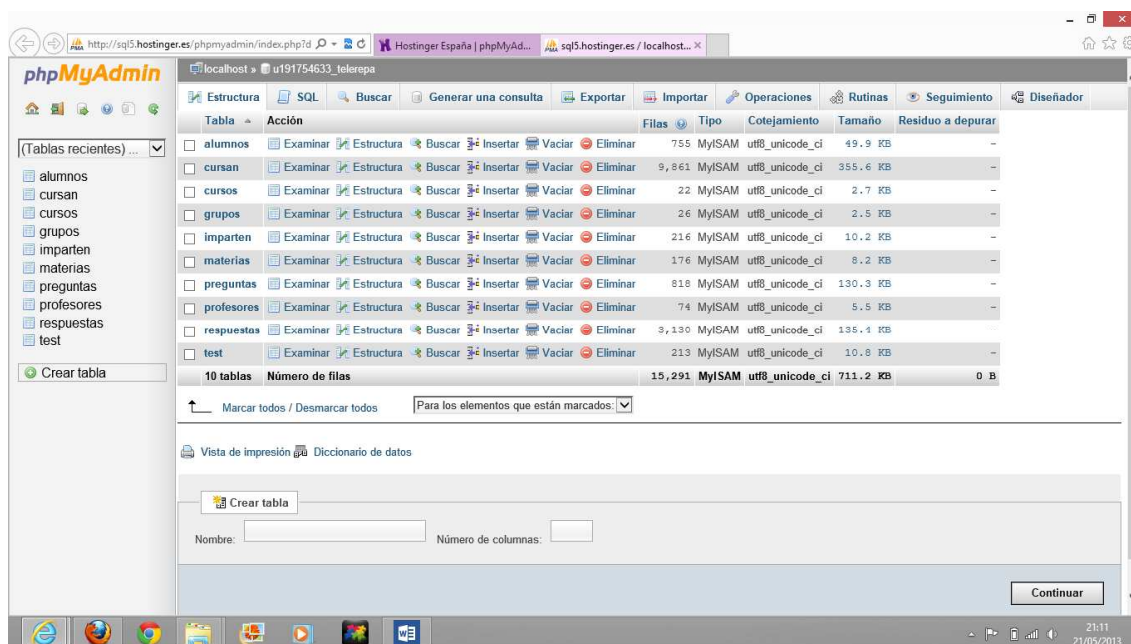


Figura 21: Pantalla base de datos del servidor Hostinger

Con sólo estos pasos sencillos, ya se puede usar la aplicación en cualquier móvil android, a falta de instalarse la aplicación en el móvil que se quiera. Para ello sólo hay que instalar la apk que se encuentra en el cd del PFG cada usuario que quiera usarla.

La base de datos es la que se usa en la aplicación web de Tele-repaso del PFC, salvo con algún añadido, como el archivo móvil por ejemplo, así que cada curso se usarían esos datos.

Y por último se realizarán las pruebas de implantación y de aceptación del sistema.

La aplicación no necesita mantenimiento.

5.1.2 Especificación del equipo de implantación

El usuario implicado en realizar toda la implantación del sistema es el autor del proyecto. El usuario implicado en las pruebas de implantación será en un principio solo el autor del proyecto, que será el encargado de llevar acabo las mismas pruebas del sistema realizadas anteriormente en modo local, pero esta vez se harán estando Tele-repaso Android implantado en el servidor “Hostinger”, luego una vez comprobado que las pruebas han sido positivas, se dejara que lo pruebe el tutor.

Una vez que las pruebas de implantación han sido positivas, se pasará a las pruebas de aceptación.

5.2 Formación necesaria para la implantación

5.2.1 Preparación y formación del equipo de implantación

No se necesita preparación, ni formación del equipo de implantación.

5.3 Incorporación del sistema al entorno de operación

5.3.1 Realización de la instalación y de la implantación

Los pasos a seguir para realizar la instalación del sistema al servidor “Hostinger” están explicados anteriormente en el punto 5.1.1 definición del plan de implantación.

5.4 Carga de datos al entorno de operación

5.4.1 Migración y carga inicial de datos

Se realiza al importar los scripts de la base de datos de Tele-repaso. Después de realizar todos los pasos, descritos anteriormente, se comprueba el sistema que funciona bien, y que se han pasado los datos correctamente. Ya tenemos la base de datos correctamente, con datos introducidos para su uso.

5.5 Pruebas de implantación del sistema.

5.5.1 Realización de las pruebas de implantación

El autor del proyecto hace las mismas pruebas que las realizadas en las pruebas del sistema, solo que ahora con el sistema implantado en donde va a estar realmente durante su uso. Las pruebas han sido también correctas, así que se pasa al siguiente paso.

5.5.2 Evaluación del resultado de las pruebas de implantación

Después de haber probado la aplicación por completo se comprueba que Tele-repaso Android funciona correctamente y que no se ha encontrado ningún fallo.

5.6 Pruebas de aceptación del sistema

5.6.1 Realización y evaluación de las pruebas de aceptación

Estas pruebas no se realizarán hasta que no se implante realmente en un centro escolar, y serán los alumnos y profesores quienes las realicen y evalúen los resultados de las pruebas.

Esta es la mejor manera de probarlo, ya que accederán muchos usuarios a la vez, tanto alumnos, como profesores, y es cuando mejor se apreciará cual será el rendimiento en condiciones de límite y de sobrecarga en verdadera ejecución real. Cada tipo de usuario usará las acciones que puede hacer, para así probarlo por completo.

5.7 Presentación y aprobación de la implantación del sistema

5.7.1 Convocatoria de la presentación del sistema

Presento al tutor la aplicación Tele-repaso Android.

Recopilo la información del sistema a entregar al Comité de Dirección. Cuando me den la confirmación por parte del Comité de Dirección, prepararé la presentación del sistema a la universidad.

5.7.2 Aprobación del sistema

Presento al Comité de Dirección el proyecto realizado, según el plan previsto.

5.8 Paso a producción

El sistema ya se encuentra en producción, pudiéndolo usar todos los usuarios (profesores y alumnos). El sistema está perfectamente probado e implantado.

6 CONCLUSIONES

Este proyecto ha sido un reto importante, por ser un proyecto desarrollado para móvil android y útil, sobre todo para los alumnos, quienes pueden repasar sus lecciones y comprobar sus conocimientos, también para los profesores que pueden evaluar a sus alumnos y ver cómo van entendiendo los alumnos los temas de sus asignaturas.

El proyecto se ha desarrollado según lo planificado, cumpliendo la condición además de que no supere las 300 horas, y el esfuerzo ha merecido la pena. En definitiva estoy bastante satisfecho con el resultado, Android SDK, PHP y MySQL son tres tecnologías muy utilizadas en el mundo laboral, ya que al ser gratuitas cuentan con más apoyo, por lo que pueden ser tecnologías que tenga que usar en la empresa donde trabaje en el futuro.

Las aplicaciones móviles están en auge, y cada vez se utilizan mucho más, por lo que me ha servido el proyecto para aprender a desarrollar para móviles android, ya que no tenía ningún conocimiento, aunque de Java sí que tenía.

6.1 Concordancia entre los resultados y los objetivos

Este proyecto es de carácter tecnológico, me refiero con esto, que se trataba en una parte muy importante del proyecto de investigar; no sólo desarrollar de forma automática.

En un principio no se sabía ni siquiera si se iban a llegar a cumplir todos los objetivos puestos, por no tener conocimiento de la tecnología para el desarrollo de aplicaciones para móviles android y por no estar seguro de que se iba a poder desarrollar y documentar en aproximadamente 300 horas.

Para el desarrollo de la aplicación, hubo varias herramientas a elegir, que, en un principio, nos garantizaban la realización del desarrollo, pero finalmente, se tuvieron que desechar unas cuantas, antes de conseguir lo buscado, y me decante por Android SDK, ya que era lo más estandarizado, con muchos tutoriales y ejemplos en internet.

Aún con todos los problemas que iban surgiendo en el desarrollo, estoy satisfecho del resultado del proyecto.

6.2 Lecciones aprendidas

Una de las más importantes lecciones aprendidas, es el desarrollo de un proyecto, desde el comienzo con el análisis de requisitos, pasando por el análisis, diseño, construcción, pruebas e implantación del sistema. He aprendido mejor a analizar y diseñar una aplicación con tiempo limitado, pensando, antes de escribir una línea de código, acción que es muy importante, pues ponerte a escribir código sin tener las cosas claras es un error, que te costará tiempo y dinero.

Hablando de tecnología, he aprendido mejor a programar en PHP, usar MySQL, además de la nueva tecnología para mí de Android SDK.

En definitiva una lección a buscarse uno la vida para conseguir lo que te piden.

7 BIBLIOGRAFÍA

- Metrica3: <http://www.csi.map.es/csi/metrica3/>
- Pruebas: http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas_d.php
- Manual de Android y Api: <http://developer.android.com/intl/es/training/index.html>
- Curso de Android: <http://www.maestrosdelweb.com/editorial/curso-android/>
- Filas dinámicas en TableLayout de Android: <http://blog.abcdroid.pe/2012/09/filas-dinamicas-en-tablelayout-android.html>
- Manual de PHP: <http://www.desarrolloweb.com/manuales/12/>
- Manual PHP y SQL:
http://www.gui.uva.es/~laertes/articulos/php/GuiaCursoPHP_SQL.pdf
- Programando en PHP:
<http://www.monografias.com/trabajos38/programacionphp/programacion-php.shtml>
- Arquitectura: <http://fsvelectronicainformatica.blogspot.com.es/2012/12/arquitectura-mvc-android-android-json-o.html>
- Entender Arquitectura: <http://gdroid.com.mx/blog/2011/06/07/entendiendo-la-arquitectura-de-una-aplicacion-android/>
- Arquitecturas móvil: <http://www.danielnavarroymas.com/aplicaciones-moviles/arquitectura-de-una-aplicacion-movil/>
- Arquitectura JSON: <http://thinkcoderepeat.blogspot.com.es/2006/08/arquitectura-cliente-servidor-con-ajax.html>
- J2ME – Aplicaciones móviles:
http://www.logicnets.com.ar/index.php?option=com_content&view=article&id=1465&Itemid=98

ANEXOS

ANEXO 1 CONTENIDO DEL CD ADJUNTO

A.1 Contenido del CD adjunto

- Memoria en formato pdf.
- Archivos necesarios de la aplicación (SQL de la base de datos, PHP de los archivos en servidor hostinger, etc.).
- El código fuente de la aplicación está dentro de la carpeta TeleRepasoAndroid.
- Aplicación Móvil de TelerepasoAndroid (APK).