



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**QuickTest Android**  
**Aplicación para el Acceso a**  
**Plataformas de ELearning desde**  
**Dispositivos Móviles**  
**Documentación Técnica**



Presentado por Daniel Puente Gabarri  
en Universidad de Burgos — 28 de junio de 2017  
Tutor: María Belén Vaquerizo García y Bruno  
Baruque Zanón

---

# Índice general

---

Índice general	I
Índice de figuras	III
Índice de tablas	VI
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	9
<b>Apéndice B Especificación de Requisitos</b>	<b>15</b>
B.1. Introducción . . . . .	15
B.2. Product Backlog . . . . .	15
B.3. Diagrama de casos de uso . . . . .	20
<b>Apéndice C Especificación de diseño</b>	<b>27</b>
C.1. Introducción . . . . .	27
C.2. Diseño de datos . . . . .	27
C.3. Diagrama de paquetes . . . . .	33
C.4. Diseño procedimental . . . . .	35
C.5. Diseño arquitectónico . . . . .	38
<b>Apéndice D Documentación técnica de programación</b>	<b>40</b>
D.1. Introducción . . . . .	40
D.2. Estructura de directorios . . . . .	40
D.3. Manual del programador . . . . .	41
D.4. Pruebas del sistema . . . . .	61
<b>Apéndice E Documentación de usuario</b>	<b>67</b>

<i>ÍNDICE GENERAL</i>	II
E.1. Introducción . . . . .	67
E.2. Requisitos de usuarios . . . . .	67
E.3. Instalación . . . . .	67
E.4. Manual del usuario . . . . .	71
<b>Bibliografía</b>	<b>78</b>

---

# Índice de figuras

---

A.1. Gráfico Burndown Sprint 1 - Inicio del proyecto. . . . .	3
A.2. Gráfico Burndown Sprint 2 - Estudio de refactorización del servi- dor. . . . .	4
A.3. Gráfico Burndown Sprint 3 - Diseño del BackEnd. . . . .	5
A.4. Gráfico Burndown Sprint 8 - Ampliación del Front End. . . . .	8
A.5. Gráfico Burndown Sprint 9 - . Finalizar FrontEnd y test. . . . .	9
B.1. Diagrama de casos de uso del alumno. . . . .	20
B.2. Diagrama de casos de uso del profesor. . . . .	21
B.3. Diagrama de casos de uso de CU1, CU2, CU3 y CU4. . . . .	23
B.4. Diagrama de casos de uso de CU5, CU6 y CU7. . . . .	25
B.5. Diagrama de casos de uso de CU8 y CU9. . . . .	26
C.1. Clase ControlAccesoProfesor . . . . .	28
C.2. Clase GestionCuestionario . . . . .	29
C.3. Clase SolucionCuestionario . . . . .	30
C.4. Clase ObtenerCuestionario . . . . .	31
C.5. Diagrama de clases de la aplicación. . . . .	32
C.6. Diagrama de paquetes del APIREST . . . . .	33
C.7. Diagrama de paquetes de la aplicación Android . . . . .	34
C.8. Diagrama de secuencias caso de uso 1. . . . .	35
C.9. Diagrama de secuencias caso de uso 2. . . . .	35
C.10. Diagrama de secuencias caso de uso 3. . . . .	36
C.11. Diagrama de secuencias caso de uso 4. . . . .	36
C.12. Diagrama de secuencias caso de uso 5 desde el rol de profesor. . . .	37
C.13. Diagrama de secuencias caso de uso 5 desde el rol de alumno. . . .	37
C.14. Diagrama de secuencias caso de uso 6. . . . .	37
C.15. Diagrama de secuencias caso de uso 7. . . . .	38
C.16. Diagrama de secuencias caso de uso 8. . . . .	38
C.17. Diagrama de secuencias caso de uso 9. . . . .	38

C.18.Diagrama de despliegue del proyecto. . . . .	39
D.1. Instalador de XAMMP . . . . .	42
D.2. Alerta del firewall de Apache . . . . .	43
D.3. Panel de XAMPP . . . . .	43
D.4. Alerta del firewall de MySQL . . . . .	44
D.5. Carpeta comprimida de Moodle . . . . .	45
D.6. url de Moodle . . . . .	45
D.7. Ubicación de donde instalar los directorios . . . . .	46
D.8. Selección de la base de datos . . . . .	46
D.9. Configuración de Moodle . . . . .	47
D.10.url de MySql . . . . .	47
D.11.Cambiamos el formato de la base de datos . . . . .	48
D.12.Creación de la tabla Moodle con el formato deseado . . . . .	48
D.13.Configuramos los datos del administrador de Moodle . . . . .	49
D.14.Configuramos el sitio . . . . .	49
D.15.Panel de configuración del Web Service de Moodle . . . . .	50
D.16.Obtención del token del administador . . . . .	51
D.17.Url de Mysql . . . . .	52
D.18.Añadimos el fichero de la base de datos . . . . .	52
D.19.Añadimos la carpeta al htdocs . . . . .	53
D.20.Abrimos la página de inicio de QuickTest . . . . .	53
D.21.Entramos en QuickTest . . . . .	54
D.22.Panel de primeros pasos . . . . .	55
D.23.Ventana emergente de publicar un cuestionario . . . . .	55
D.24.Inicio de sesión de Moodle . . . . .	56
D.25.Creamos un curso . . . . .	56
D.26.Rellenamos los campos del nuevo curso . . . . .	57
D.27.Creamos los nuevos usuarios . . . . .	57
D.28.Rellenamos los campos del primer usuario . . . . .	58
D.29.Rellenamos los campos del segundo usuario . . . . .	58
D.30.Agregamos una nueva herramienta externa . . . . .	59
D.31.Configuramos la nueva herramienta externa 1 . . . . .	60
D.32.Configuramos la nueva herramienta externa 2 . . . . .	60
D.33.Campos de la privacidad . . . . .	60
D.34.Contenido de la carpeta test_apiRest . . . . .	61
D.35.Conjunto de pruebas unitarias del FrontEnd. . . . .	62
D.36.Ejecución y resultado de las pruebas unitarias sobre el FrontEnd desde Android Studio. . . . .	63
D.37.Conjunto de interfaz de usuario del FrontEnd. . . . .	63
D.38.Habilitamos las pruebas de cobertura. . . . .	64
D.39.Ejecución de las pruebas de cobertura sobre el paquete models. . . . .	65
D.40.Añadimos la dependencia de sonarQube. . . . .	65
D.41.Configuración de SonarQube. . . . .	65

D.42.Ejecución de SonarQube . . . . .	66
E.1. Archivo config.php de Moodle. . . . .	68
E.2. Modificando la URL de inicio de las herramientas externas. . . . .	68
E.3. Archivos a modificar dentro del paquete api. . . . .	69
E.4. Modificamos las variables de la clase APIMoodle . . . . .	69
E.5. Modificamos la variable de la clase APIRest. . . . .	70
E.6. Instalación de la aplicación en un dispositivo físico. . . . .	71
E.7. Iniciar sesión en la aplicación. . . . .	72
E.8. Obtener la calificación de los alumnos matriculados en un curso. . . . .	73
E.9. Obtenemos el cuestionario al que se han enfrentado los alumnos. . . . .	74
E.10.Resolvemos un cuestionario. . . . .	75
E.11.Revisar un cuestionario. . . . .	76
E.12.Cerrar sesión desde ambos roles. . . . .	77

---

# Índice de tablas

---

A.1. Coste Hardware . . . . .	10
A.2. Coste Software 1 . . . . .	11
A.3. Coste Software 2 . . . . .	11
A.4. Coste de aprendizaje . . . . .	12
A.5. Otros costes . . . . .	12
A.6. Costes totales . . . . .	13
B.1. Historia de usuario 1 . . . . .	15
B.2. Historia de usuario 2 . . . . .	16
B.3. Historia de usuario 3 . . . . .	16
B.4. Historia de usuario 4 . . . . .	16
B.5. Historia de usuario 5 . . . . .	17
B.6. Historia de usuario 6 . . . . .	17
B.7. Historia de usuario 7 . . . . .	18
B.8. Historia de usuario 8 . . . . .	18
B.9. Historia de usuario 9 . . . . .	19
B.10. Historia de usuario 10 . . . . .	19
B.11. Historia de usuario 11 . . . . .	20
B.12. Historia de usuario 12 . . . . .	20
B.13. Caso de uso 1: Iniciar sesión . . . . .	21
B.14. Caso de uso 2: Olvidar campos . . . . .	21
B.15. Caso de uso 3: Recordar campos . . . . .	22
B.16. Caso de uso 4: Mostrar cuestionarios . . . . .	22
B.17. Caso de uso 5: Mostrar cuestionarios . . . . .	23
B.18. Caso de uso 6: Resolver cuestionario . . . . .	24
B.19. Caso de uso 7: Mostrar cuestionarios resueltos . . . . .	24
B.20. Caso de uso 8: Mostrar calificación de los alumnos . . . . .	25
B.21. Caso de uso 9: Ver cuestionarios . . . . .	26

## *Apéndice A*

---

# Plan de Proyecto Software

---

### **A.1. Introducción**

A continuación, en esta sección se encuentra toda aquella información relacionada con la gestión del proyecto. Esta información, no solamente va a estar relacionada con el desarrollo del producto pedido, sino que englobará otros aspectos relevantes como la definición de los riesgos o costes del plan proyecto. Como se ha comentado anteriormente en la sección 4.3.3, la metodología de trabajo utilizada para este proyecto es SCRUM. Para llevar cabo el desarrollo del proyecto se ha dividido el desarrollo del mismo en Sprints o Iteraciones de dos semanas. Cabe destacar que algún Sprint varía en esta duración establecida debido a la disponibilidad del equipo para poder llevar a cabo la reunión al finalizar cada iteración.

### **A.2. Planificación temporal**

Como se comentaba, se ha decidido dividir el desarrollo del proyecto en Sprints de dos semanas de duración. Gracias a esta duración se ha podido compaginar de una manera más eficiente el desarrollo del proyecto de final de grado junto con el resto de asignaturas de este cuatrimestre. Además, se ha estipulado que el número de horas de dedicación por iteración sea de alrededor de cuarenta, este es el principal motivo por el cual se ha decidido que la duración de cada iteración sea de quince días para poder compaginar el desarrollo del proyecto junto con el resto de asignaturas.

#### **Sprint 1. Inicio del proyecto (7 Feb 2017 a 16 Feb 2017)**

Este día tuvo lugar la primera reunión para la explicación sobre el producto a desarrollar. Una vez explicado el mismo, se decidió aceptar este proyecto y se comenzó con la definición de los principales pilares a superar para construir



el producto. Una vez definidas estas bases se decidió realizar en este Sprint las siguientes tareas.

- **Cargar código fuente:** este primer commit fue realizado por los tutores para incluir el código fuente del proyecto anterior en el repositorio de control de versiones empleado para el desarrollo actual.
- **Objetivos del proyecto:** esta tarea ha sido realizada por el alumno sobre la documentación a entregar. En dicho apartado del documento se definirán las bases principales del proyecto junto con sus objetivos más destacados.
- **Instalar proyecto de partida:** en tarea fue realizada por el alumno, para ello se ha realizado una lectura de la documentación del proyecto de partida y su posterior instalación para conocer al máximo la lógica a implementar.
- **Generar el Product Backlog:** esta tarea fue realizada por el alumno en GitHub y posteriormente en la memoria. En esta sección se llevó a cabo la definición de los principales requisitos funcionales las cuales derivarán en las historias de usuario a realizar en los diferentes Sprint.
- **Aprendizaje PHP:** esta tarea fue la más costosa de esta iteración ya que se desconocía cualquier conocimiento acorde a este lenguaje. Para ello se realizó una lectura de la documentación proporcionada en la propia web de PHP [1] junto con otras fuentes online como tutoriales sobre PHP desde cero.

A continuación, en la siguiente Ilustración se incluye un gráfico Burndown resumiendo el Sprint, cabe destacar que en esta ocasión solamente se aprecia que todas las tareas fueron finalizadas el mismo día, aunque esto no fue así. Este error nace fruto de un desconocimiento previo de la herramienta, al no incluir en la configuración de la creación de dicho gráfico las tareas que se encuentran dentro del tablero en la sección Done. Posteriormente, una vez localizado y subsanado este error en las siguientes iteraciones este fallo no se produce.



Figura A.1: Gráfico Burndown Sprint 1 - Inicio del proyecto.

## Sprint 2. Estudio de refactorización del servidor (16 Feb 2017 a 2 Mar 2017)

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 1. En esta iteración se decidió realizar las siguientes tareas:

- **Estudio de inclusión de un Framework:** esta tarea fue realizada por el alumno y se realizó un estudio de diferentes alternativas como: CakePHP [2], Laravel [3], Symfony [4], Slim [5]. No obstante, la decisión ha sido no utilizar ningún Framework al tratarse de un proyecto pequeño en dimensiones y al añadir mayor complejidad a la resolución del mismo.
- **Listar los métodos a implementar en el API:** esta tarea ha sido realizada por el alumno y en ella se listan las principales lógicas a implementar sobre la base de datos.
- **Estudio de diferencias entre REST y SOAP:** esta tarea también fue realizada por el alumno, para ello se realizó una correcta lectura y posterior documentación en el apartado Soap y Rest. Fruto de este estudio se ha decidido utilizar REST como arquitectura para facilitar el uso y tratamiento de las peticiones desde el programa.
- **Documentación: Aspectos Teóricos / Técnicas y Herramientas:** esta tarea consiste en llevar a cabo una documentación en la memoria sobre estos aspectos en la sección Conceptos teóricos y Técnicas y herramientas.

En la siguiente Ilustración podemos observar el gráfico Burndown correspondiente a esta iteración.



Figura A.2: Gráfico Burndown Sprint 2 - Estudio de refactorización del servidor.

### Sprint 3. Diseño del BackEnd (2 Mar 2017 a 15 Mar 2017)

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 2. En esta iteración se decidió realizar las siguientes tareas:

- **Implementación de los métodos del API:** esta tarea consiste en realizar la implementación del APIREST, para poder comunicarnos con el controlador del proyecto de partida.
- **Diseño arquitectura del BackEnd:** esta tarea consiste en construir los diagramas sobre el BackEnd.
- **Anexo 1 – Planificación del proyecto:** esta tarea consiste en añadir a la documentación este anexo.

En la siguiente Ilustración podemos observar el gráfico Burndown correspondiente a esta iteración.



Figura A.3: Gráfico Burndown Sprint 3 - Diseño del BackEnd.

#### Sprint 4. Interconexión FrontEnd-BackEnd (15 Mar 2017 a 29 Mar 2017)

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 3. En esta iteración se decidió realizar las siguientes tareas:

- **Test unitarios sobre el BackEnd:** en esta tarea se realizó una documentación sobre las pruebas unitarias en PHP. Posteriormente, se realizaron pruebas unitarias sobre las nuevas clases creadas en el APIREST.
- **Leer documentación LTI:** en esta tarea se realizó una lectura sobre el funcionamiento de LTI para su posterior utilización en la aplicación.
- **Prueba conexión Android - Moodle:** esta tarea fue la más complicada del Sprint y posiblemente una de las más costosas del proyecto. Después de mucha lectura se decidió utilizar los Web service de Moodle para obtener la información de la plataforma desde Android.
- **Documentación, Anexo Diseño:** en esta tarea se incluyó en la memoria los diagramas de clases y paquetes del APIREST.
- **Definir ProductBackLog:** en esta tarea se definió el conjunto de historias de usuario a desarrollar.

Destacar que desde la iteración 4 hasta la 7 no se puede incluir el gráfico Burndown del Sprint, ya que debido a un bug en la última actualización de ZenHub este no se encuentra disponible.

### **Sprint 5. Recuperación de datos desde el BackEnd (29 Mar 2017 a 20 Abr 2017)**

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 4. En esta iteración se decidió realizar las siguientes tareas:

- **Implementar conexión LTI desde Android:** en esta tarea se realizó la obtención del token del usuario que ha iniciado sesión en la aplicación.
- **MockUp del la App Android:** en esta tarea se realizo una construcción genérica de la app.

### **Sprint 6. Recuperación de Datos desde BackEnd (II) (20 Abr 2017 a 4 May 2017)**

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 5. En esta iteración se decidió realizar las siguientes tareas:

- **Diseño de la arquitectura del FrontEnd:** en esta tarea se incluyó en la memoria el diagrama de clases y paquetes de la aplicación Android.
- **Obtención datos, herramienta externa:** en esta tarea se realizó un estudio sobre los diferentes métodos del Web service de Moodle necesarios para la obtención de los cuestionarios de QuickTest.
- **HU1 - Iniciar sesión:** en esta tarea se proporcionó un diseño al login más cercano al producto final.
- **HU2 - Cuestionarios a resolver:** en esta tarea se obtuvieron los cuestionarios de QuickTest asociados al usuario logeado.
- **Añadir aspectos relevantes hasta el momento en el manual del programador:** en esta tarea se completo esta sección en la memoria.
- **Casos de Uso:** en esta tarea se completo esta sección en la memoria.
- **Obtener datos desde QuickTest:** en esta se incluyó en el APIREST la funcionalidad de obtener un cuestionario por identificador, ya que esta funcionalidad no se encontraba desarrollada en el controlador,
- **Incluir nuevas bibliotecas en Manual Programador:** en esta tarea se añadieron una explicación sobre las librerías Gson y Volley.

**Sprint 7. Grabado de datos desde el BackEnd (4 May 2017 a 18 May 2017)**

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 6. En esta iteración se decidió realizar las siguientes tareas:

- **HU3 - Resolver un cuestionario:** en esta tarea realizó el desarrollo de esta historia de usuario. Destacar que solamente se permitió que el alumno resolviera un cuestionario sin la posibilidad de los comodines.
- **HU4 - Finalizar un cuestionario:** en esta tarea se permitió que el alumno pueda finalizar un cuestionario desde la aplicación.
- **HU7 - Podrá cerrar sesión:** en esta tarea se realizó el desarrollo de esta historia de usuario en la aplicación.
- **Completar Aspectos Relevantes (Memoria):** en esta tarea se completo esta sección en la memoria.
- **Envío de respuestas desde la app a Quicktest:** en esta tarea se permitió que al finalizar un cuestionario, estos datos sean enviados al APIREST para su almacenamiento.

**Sprint 8. Ampliación del Front End. (18 May 2017 a 8 Jun 2017)**

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 7. En esta iteración se decidió realizar las siguientes tareas:

- **HU5 - Calificación obtenida:** en esta tarea se incluyó en la aplicación una sección donde el usuario pudiera ver la calificación obtenida en un cuestionario ya resuelto.
- **HU6 - Revisar un cuestionario:** en esta tarea se incluyó en la aplicación una sección donde el usuario pudiera ver una tabla con información sobre el cuestionario.
- **Implementación de Funcionalidad de Comodines:** en esta tarea se completó la HU3 al permitir al alumno utilizar comodines.

En la siguiente Ilustración podemos observar el gráfico Burndown correspondiente a esta iteración.

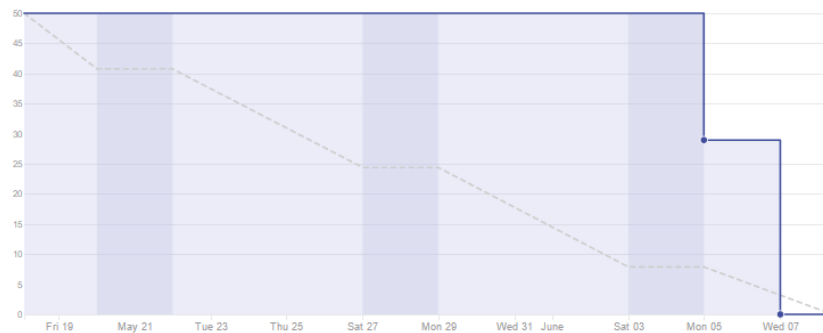


Figura A.4: Gráfico Burndown Sprint 8 - Ampliación del Front End.

### Sprint 9. Finalizar FrontEnd y test. (8 Jun 2017 a 19 Jun 2017)

Al comienzo de esta iteración se realizó la revisión de las tareas a realizar en el Sprint anterior y una vez revisadas se dio por finalizado el Sprint 8. En esta iteración se decidió realizar las siguientes tareas:

- **Pruebas en el BackEnd:** en esta tarea se realizó las pruebas unitarias sobre el BackEnd del proyecto.
- **Estudiar como realizar pruebas en Android:** en esta tarea se realizó una documentación sobre los tipos de pruebas disponibles y su funcionamiento.
- **Pruebas en el FrontEnd:** en esta tarea se puso en práctica todos los nuevos conceptos sobre pruebas adquiridos en el punto anterior.
- **Comentar, limpiar y solucionar bugs:** en esta tarea se corrigieron ciertos fallos.
- **Profesor podrá al entrar en un test pueda ver las notas de los alumnos:** en esta tarea se modificó la aplicación para que permita ambos roles.
- **El profesor podrá ver todos los tests:** en esta tarea se incluyó en la aplicación la funcionalidad dada.
- **Se podrá iniciar sesión en la app como profesor y alumno:** en esta tarea se incluyó dentro del propio login, que permita la inclusión en la aplicación de profesores.
- **El profesor podrá ver los cuestionarios como si fuera un alumno:** en esta tarea se incluyó la funcionalidad pedida.

- **Completar aspectos de la memoria:** en esta tarea se incluyeron ciertos aspectos a la memoria.
- **La aplicación incluirá un Splash:** en esta tarea se introdujo un splash que gestione el inicio de sesión en la aplicación.
- **Completar sección de ayuda en la app:** en esta tarea se introdujo dentro de la vista de los alumnos una sección de ayuda.

En la siguiente Ilustración podemos observar el gráfico Burndown correspondiente a esta iteración.



Figura A.5: Gráfico Burndown Sprint 9 - . Finalizar FrontEnd y test.

### A.3. Estudio de viabilidad

Uno de los objetivos principales de cualquier proyecto por no clasificarlo como el más importante es conocer si este proyecto es viable. Para ello se debe realizar un estudio sobre la viabilidad del mismo. Es por esto, que se deberá realizar este estudio sobre dos enfoques: *económico y legal*. Y gracias a estos sabremos si el proyecto es rentable y viable.

#### Viabilidad económica

Relacionado con lo anteriormente expuesto en esta sección se va realizar un estudio sobre la viabilidad del proyecto en el enfoque económico. Para ello se va realizar un análisis sobre los diferentes costes que entran en juego y son necesarios para el desarrollo del mismo junto con el mantenimiento una vez desarrollado.



### Coste Hardware

Para el desarrollo del proyecto y posterior corrección de ciertos problemas que puedan surgir será necesario:

- **Un ordenador potente en prestaciones**, ya que el IDE de Android Studio es bastante pesado en tareas principalmente de emulación.
- **Un dispositivo móvil**, es decir un Smartphone con S.O Android.

En este caso se utilizarán las herramientas propias del alumno encargado de realizar el proyecto, que son las siguientes:

- **Asus GL552VW-DM142T**: coste actual 1199 € sin S.O.
- **OnePlus 2**: coste actual 319 €.

Ambos componentes son bastante potentes y por lo tanto no ocasionarán ninguna problemática de que el hardware se quede anticuado en prestaciones en posteriores años. Para redondear los posteriores cálculos estimaremos que el tiempo de vida de ambos componentes será de 5 años, es decir 60 meses. Aunque seguramente su tiempo de vida podría alargarse aún más. Además, al realizarse el proyecto en un cuatrimestre la duración de la utilización de dichos componentes será de 4 meses.

Tiempo de amortización	60 meses
Coste de los componentes	1199,00 € + 319,00 € = 1518,00 €
Coste de amortización/mes	1518,00 € / 60 meses = 25,3 €/mes
Coste final	25,3 €/mes * 4 meses = 101,2 €

Tabla A.1: Coste Hardware

### Coste Software

Para la realización del proyecto y su posterior mantenimiento será necesario lo siguiente:

Software	Licencia	Coste
Microsoft Windows 10	Home	135,00 €
Android Studio	Licencia Apache 2.0 [6]	0,00 €
XAMPP	GNU [7]	0,00 €
GitHub	-	0,00 €
TortoiseGit	GNU General Public License [8]	0,00 €
Moodle	GNU GPL [9]	0,00 €
PhpStorm	Estudiante	0,00 €
SonarQube	LGPL	0,00 €
Advanced REST Client	-	0,00 €
Microsoft Office	Hogar	99,00 €
		234,00 €

Tabla A.2: Coste Software 1

En este caso la licencia de Microsoft Office es para un único año de duración, pero la licencia del S.O no tiene caducidad por lo que solamente tendremos en cuenta una duración anual.

Tiempo de amortización	12 meses
Coste de los componentes	135,00 € + 99,00 € = 234,00 €
Coste de amortización/mes	234,00 € / 12 meses = 19,5 €/mes
Coste final	19,5 €/mes * 4 meses = 78,00 €

Tabla A.3: Coste Software 2

### Coste de instalación

Este coste será nulo y no se tendrá en cuenta ya que la instalación de los distintos componentes será realizada por el propio alumno.

### Coste de aprendizaje

Este proyecto será desarrollado por el alumno por lo que este coste de realización será un sueldo mensual a dicho alumno por llevar a cabo el diseño, implementación y testeo del mismo. Como se comentaba anteriormente se ha estipulado que el número de horas dedicadas al desarrollo del mismo será de 40 horas cada dos semanas.

Duración	4 meses * 4 semanas = 16 semanas
Salario/hora	10,00 €/hora
Nº horas/semana	20 horas/semana
Coste final	10,00 €/hora * 20 horas/semana * 16 semanas = 3.200,00 €

Tabla A.4: Coste de aprendizaje

### Coste de soporte

Este coste sería realizado por un hipotético técnico que se ocuparía de resolver cualquier incidencia en la utilización del producto desarrollado. Al ser una aplicación que, si resulta viable, será publicada en el Play Store no existirá ningún técnico encargado de subsanar directamente los errores a los usuarios, sino que serán los propios operarios de dicha página los que se remitirán cualquier incidencia. Es por esto que coste de soporte será nulo y por tanto no se tendrá en cuenta.

### Coste de mantenimiento

En este caso será el propio alumno el encargado de realizar el mantenimiento de la aplicación por lo que esta tarea se encuentra remunerada dentro de su salario. No obstante, cabe destacar que la distribución desde esta App no incluye costes de almacenamiento de la información resultantes, es decir, las propias instituciones que compren el producto tendrán que realizar un mantenimiento propio de sus servidores.

### Otros costes

Además, también hay que tener en cuenta otros costes como los derivados del *material mobiliario, de oficina, documentación, electricidad, Internet*.

Tipos de coste	Coste	Total
Infraestructura	10,00 €/mes	10,00 €/mes * 4 meses = 40,00 €
Documentación	36,10 € <a href="#">[10]</a>	36,10 €
Electricidad	35,00 €	35 €/mes * 4 = 140 €
Internet	37,90 €	37,90 €/mes * 4 = 151,60 €

Tabla A.5: Otros costes

**Total**

Tipos de coste	Total
Coste Hardware	101,2 €
Coste Software	78,00 €
Coste de instalación	0,00 €
Coste de aprendizaje	3.200,00 €
Coste de soporte	0,00 €
Coste de mantenimiento	367,70 €
Otros coste	367,70 €
	3.746,90 €

Tabla A.6: Costes totales

**Análisis coste-beneficio**

Como ya se ha explicado, este proyecto nace de la base fundamental de otro proyecto. Este proyecto fue desarrollado con una base y un fin muy bien definido y fijado, la herramienta QuickTest es compatible con cualquier plataforma que cumpla el estándar LTI. Fruto de esta idea, se desarrolló este proyecto como un complemento de resolución de cuestionarios para Moodle. Es por esto, que este análisis de coste-beneficio se va realizar para Moodle.

Moodle cuenta actualmente con más de 100 millones de usuarios en todo el mundo, lo que la convierte en la plataforma digital más utilizada en todo mundo por la comunidad educativa [11] [12]. Además, es utilizada en 234 países, en la que se encuentra España en el segundo puesto con 7.152 inscripciones. Como el principal objetivo de esta aplicación es la utilización de la misma para un entorno universitario vamos a reducir estos cálculos de inscripciones al número de universidades en España.

Actualmente, el número de universidades tanto públicas como privadas es de 82. Al ser el proyecto compatible con Moodle y al ser este, la plataforma digital más influyente en este sector, cabe destacar que no deberíamos de tener ningún inconveniente en vender licencias de este software a las universidades. Sería una suposición muy idealizada que todas estas comprarán nuestro producto, por lo que realizaremos los cálculos estimando que solamente 40 universidades deciden adquirir la licencia.

Ante este estudio preliminar, se decide vender cada licencia por 100,00 €. Este precio, solamente incluirá el producto junto con la documentación necesaria para su correcta utilización, es decir, no se incluye un soporte de almacenamiento de la información generada por su utilización. Además, esta distribución de licencia se hará con carácter comercial [13]. Finalmente,

cabe destacar que valoraremos este análisis como factible ya que únicamente distribuyendo estas 40 licencias recuperaríamos la inversión realizada.

### **Viabilidad legal**

Al utilizar herramientas que se encuentran bajo licencias de software libre la distribución de esta herramienta no ocasionará ninguna problemática legal. No obstante, la herramienta fruto de este proyecto se distribuirá con la licencia Android License Verification Library [14] propia de PlayStore y utilizada por cualquier aplicación que resida en la misma para evitar posibles pirateos. Como ya se comentaba en la sección anterior este producto se distribuirá como licencia comercial, ya que todos los activos utilizados para el desarrollo del mismo poseen licencia GNU.

## Apéndice B

---

# Especificación de Requisitos

---

### B.1. Introducción

Como ya se ha comentado anteriormente, se va utilizar la metodología ágil SCRUM. Es por esto que no vamos a tratar las tareas a realizar como requisitos, sino como historias de usuario. Todas estas historias de usuario formarán parte del Product Backlog y serán asignadas a determinadas iteraciones durante el desarrollo del proyecto. A continuación, se incluye un listado sobre las historias de usuarios y sus correspondientes diagramas de casos de uso.

### B.2. Product Backlog

En esta sección se incluyen todas aquellas historias de usuarios necesarias para los alumnos y los profesores.

#### Historias de usuario de ambos roles

##### Historia de usuario 1: Iniciar sesión

HU1	
Título	Como un usuario autenticado, podrá iniciar sesión.
Rol	Alumno o profesor autenticado.
Descripción	El usuario introducirá sus credenciales y se le mostrará aquellos cursos en los que se encuentre matriculado.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.1: Historia de usuario 1

**Historia de usuario 2: Cerrar sesión**

HU2	
Título	Como un usuario autenticado, podrá cerrar sesión.
Rol	Alumno o profesor autenticado.
Descripción	Siempre y cuando el usuario se encuentre logeado, podrá cerrar sesión.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.2: Historia de usuario 2

**Historia de usuario 3: Recordar campos**

HU3	
Título	Como un usuario autenticado, podrá recordar los campos que se emplean para realizar el login y no tener que introducirlos otra vez.
Rol	Alumno o profesor autenticado.
Descripción	El sistema permitirá recordar los campos con los que el usuario inicia sesión.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.3: Historia de usuario 3

**Historia de usuario 4: Cursos**

HU4	
Título	Como un usuario autenticado, podrá ver los cursos en los que se encuentra matriculado en sus estudios, obtenidos desde la aplicación Moodle de su institución.
Rol	Alumno o profesor autenticado.
Descripción	El sistema mostrará los cursos en los que se encuentra matriculado.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.4: Historia de usuario 4

**Historias de usuario de los alumnos****Historia de usuario 5: Obtener cuestionarios**

HU5	
Título	Como un usuario autenticado, el sistema notificará sobre los cuestionarios a resolver.
Rol	Alumno autenticado.
Descripción	Una vez el usuario haya iniciado sesión y seleccionado un curso, obtendrá todos aquellos cuestionarios en los que se encuentre asignado.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> <li>- Existan cuestionarios en aquellos cursos en los que se encuentra matriculado el alumno.</li> </ul>

Tabla B.5: Historia de usuario 5

**Historia de usuario 6: Resolver cuestionario**

HU6	
Título	Como un usuario autenticado, podrá resolver un cuestionario.
Rol	Alumno autenticado.
Descripción	Una vez el usuario haya iniciado sesión y seleccionado un curso, podrá resolver aquellos cuestionarios en los que se encuentren asignado y no estén resueltos.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> <li>- Existan cuestionarios en aquellos cursos en los que se encuentra matriculado el alumno.</li> </ul>

Tabla B.6: Historia de usuario 6



**Historia de usuario 7: Finalizar cuestionario**

HU7	
Título	Como un usuario autenticado, podrá finalizar un cuestionario.
Rol	Alumno autenticado.
Descripción	El usuario podrá finalizar aquel cuestionario que este resolviendo.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> <li>- Existan cuestionarios en aquellos cursos en los que se encuentra matriculado el alumno.</li> <li>- Haber iniciado la resolución de un cuestionario.</li> </ul>

Tabla B.7: Historia de usuario 7

**Historia de usuario 8: Calificación obtenida**

HU8	
Título	Como un usuario autenticado, se podrá consultar al sistema sobre la calificación obtenida.
Rol	Alumno autenticado.
Descripción	Una vez el usuario finalice el cuestionario, el sistema proporcionará una retroalimentación sobre el cuestionario resuelto.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> <li>- Existan cuestionarios en aquellos cursos en los que se encuentra matriculado el alumno.</li> <li>- Haber iniciado la resolución de un cuestionario.</li> <li>- Haber finalizado un cuestionario.</li> </ul>

Tabla B.8: Historia de usuario 8

**Historia de usuario 9: Revisar cuestionario resuelto**

HU9	
Título	Como un usuario autenticado, podrá revisar un cuestionario resuelto.
Rol	Alumno autenticado.
Descripción	El usuario podrá revisar la retroalimentación de un cuestionario que ya esté finalizado.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> <li>- Existan cuestionarios en aquellos cursos en los que se encuentra matriculado el alumno.</li> <li>- Haber realizado un cuestionario.</li> </ul>

Tabla B.9: Historia de usuario 9

**Historias de usuario de los profesores****Historia de usuario 10: Obtener cuestionarios**

HU10	
Título	Como un usuario autenticado, podrá ver los cuestionarios de un curso que tienen disponibles para completar sus alumnos.
Rol	Profesor autenticado.
Descripción	El sistema mostrará al profesor los cuestionarios a los que se enfrentan sus alumnos.
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado.</li> <li>- Haber iniciado sesión.</li> </ul>

Tabla B.10: Historia de usuario 10

**Historia de usuario 11: Calificaciones**

HU11	
Título	Como un usuario autenticado, podrá ver las calificaciones de los alumnos en un cuestionario.
Rol	Profesor autenticado.
Descripción	El sistema mostrará al profesor la calificación de cada alumno en cada cuestionario.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.11: Historia de usuario 11

**Historia de usuario 12: Ver cuestionarios**

HU12	
Título	Como un usuario autenticado, podrá ver los cuestionarios.
Rol	Profesor autenticado.
Descripción	El sistema mostrará al profesor los cuestionarios a los que se enfrentan sus alumnos.
Precondiciones	- El usuario debe estar previamente registrado. - Haber iniciado sesión.

Tabla B.12: Historia de usuario 12

**B.3. Diagrama de casos de uso**

En la siguiente ilustración podemos ver el diagrama de los casos de uso de nuestro sistema:

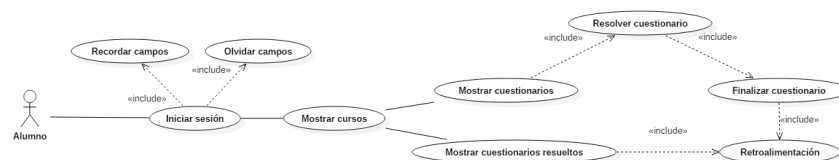


Figura B.1: Diagrama de casos de uso del alumno.

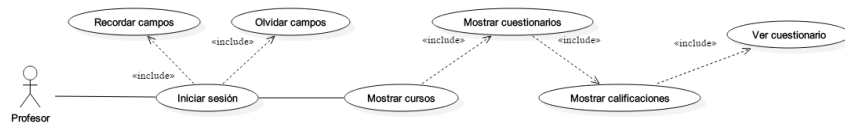


Figura B.2: Diagrama de casos de uso del profesor.

### Caso de uso 1: Iniciar sesión

CU1	
Título	Iniciar sesión.
Descripción	El sistema permitirá al usuario autenticado iniciar sesión para acceder al sistema.
Secuencia	<ol style="list-style-type: none"> <li>1. El usuario introducirá los campos de nombre y contraseña.</li> <li>2. El sistema comprobará si los campos son correctos: <ol style="list-style-type: none"> <li>2.1. Si son correctos, el usuario pasa a estar logeado teniendo pleno acceso a las funcionalidades del sistema.</li> <li>2.2. Si no son correctos, el sistema notificará que los campos son incorrectos.</li> </ol> </li> </ol>
Precondiciones	- El usuario debe estar previamente registrado en Moodle.
Comentarios	Este caso de uso, CU1, corresponde a la HU1. <a href="#">B.2</a>

Tabla B.13: Caso de uso 1: Iniciar sesión

### Caso de uso 2: Olvidar campos

CU2	
Título	Recordar campos.
Descripción	El sistema permitirá al usuario olvidar los campos con las que el usuario inicia sesión.
Secuencia	<ol style="list-style-type: none"> <li>1. El usuario seleccionara en el sistema cerrar sesión.</li> <li>2. El sistema olvidará las credenciales del usuario.</li> </ol>
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario debe estar previamente registrado en Moodle.</li> <li>- El usuario debe haber iniciado sesión y haber activado la opción de recordar campos.</li> </ul>
Comentarios	Este caso de uso, CU2, corresponde a la historia de usuario 2. <a href="#">B.2</a>

Tabla B.14: Caso de uso 2: Olvidar campos

**Caso de uso 3: Recordar campos**

CU3	
Título	Olvidar campos.
Descripción	El sistema permitirá al usuario recordar los campos con las que el usuario inicia sesión.
Secuencia	1. El sistema recordará las credenciales del usuario.
Precondiciones	- El usuario debe estar previamente registrado en Moodle.
Comentarios	Este caso de uso, CU3, corresponde a la historia de usuario 3. <a href="#">B.2</a>

Tabla B.15: Caso de uso 3: Recordar campos

**Caso de uso 4: Mostrar cursos**

CU4	
Título	Mostrar cursos.
Descripción	El sistema permitirá al usuario ver los cursos en los que se encuentra matriculado.
Secuencia	1. El sistema mostrará los cursos.
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe estar logeado en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado.
Comentarios	Este caso de uso, CU4, corresponde a la historia de usuario 4. <a href="#">B.2</a>

Tabla B.16: Caso de uso 4: Mostrar cuestionarios

En esta ilustración podemos ver el diagrama de casos de uso de los casos de los anteriores casos de uso expuestos.

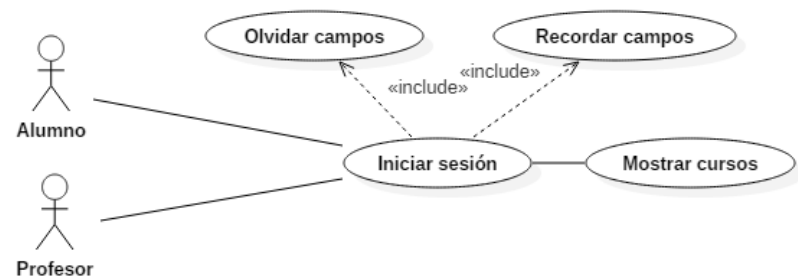


Figura B.3: Diagrama de casos de uso de CU1, CU2, CU3 y CU4.

Caso de uso 5: Mostrar cuestionarios

CU5	
Título	Mostrar cuestionarios.
Descripción	El sistema permitirá al usuario ver los cuestionarios de un curso.
Secuencia	1. El sistema mostrará los cuestionarios.
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe haber iniciado sesión en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado.
Comentarios	Este caso de uso, CU5, corresponde a la historia de usuario 5. <a href="#">B.2</a>

Tabla B.17: Caso de uso 5: Mostrar cuestionarios

**Caso de uso 6: Resolver cuestionario**

CU6	
Título	Resolver cuestionario.
Descripción	El sistema permitirá al usuario resolver un cuestionario.
Secuencia	1. El usuario seleccionará un cuestionario a resolver y comenzará a resolverlo. 2. Una vez finalizado el cuestionario, el sistema enviará el cuestionario resuelto HU7. <a href="#">B.2</a> 3. El sistema mostrará la retroalimentación del cuestionario resuelto HU8, HU9. <a href="#">B.2</a>
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe estar logeado en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado y que hayan sido resueltos.
Comentarios	Este caso de uso, CU6, corresponde a la historia de usuario 6, 7, 8, 9. <a href="#">B.2</a>

Tabla B.18: Caso de uso 6: Resolver cuestionario

**Caso de uso 7: Mostrar cuestionarios resueltos**

CU7	
Título	Mostrar cuestionarios resueltos.
Descripción	El sistema permitirá al usuario ver los cuestionarios resueltos.
Secuencia	1. El sistema mostrará los cuestionarios resueltos. 1.1. Si existe algún cuestionario resuelto el usuario podrá seleccionar un cuestionario y obtener la retroalimentación del mismo HU8, HU9.
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe haber iniciado sesión en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado y que hayan sido resueltos.
Comentarios	Este caso de uso, CU7, corresponde a la historia de usuario 6. <a href="#">B.2</a>

Tabla B.19: Caso de uso 7: Mostrar cuestionarios resueltos

En esta ilustración podemos ver el diagrama de casos de uso de los casos de los anteriores casos de uso expuestos.



Figura B.4: Diagrama de casos de uso de CU5, CU6 y CU7.

### Caso de uso 8: Mostrar calificación de los alumnos

CU8	
Título	Mostrar calificación de los alumnos.
Descripción	El sistema permitirá al usuario ver la calificación de los alumnos en un cuestionario.
Secuencia	1. El usuario seleccionará un cuestionario. H10 <a href="#">B.2</a> 2. Obtendrá la calificación de cada alumno en dicho cuestionario. HU11 <a href="#">B.2</a>
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe haber iniciado sesión en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado.
Comentarios	Este caso de uso, CU8, corresponde a la historia de usuario. 10 y 11 <a href="#">B.2</a>

Tabla B.20: Caso de uso 8: Mostrar calificación de los alumnos



**Caso de uso 9: Ver cuestionarios**

CU9	
Título	Ver cuestionarios.
Descripción	El sistema permitirá al usuario ver los cuestionarios, es decir, las preguntas y posibles respuestas de un cuestionario.
Secuencia	1. El sistema mostrará los cuestionarios. 2. El usuario seleccionara ver cuestionario.
Precondiciones	- El usuario debe estar previamente registrado en Moodle. - El usuario debe haber iniciado sesión en la aplicación para poder acceder a esta funcionalidad. - Deben existir cuestionarios de QuickTest en cursos en los que el usuario se encuentre matriculado.
Comentarios	Este caso de uso, CU9, corresponde a la historia de usuario 12. <a href="#">B.2</a>

Tabla B.21: Caso de uso 9: Ver cuestionarios

En esta ilustración podemos ver el diagrama de casos de uso de los casos de los anteriores casos de uso expuestos.

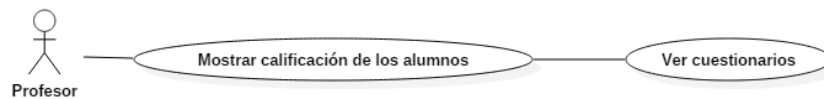


Figura B.5: Diagrama de casos de uso de CU8 y CU9.

---

## Especificación de diseño

---

### C.1. Introducción

A continuación, se incluye una explicación/descripción del diseño de la aplicación Android. Para ello, se dividirán los diagramas en dos enfoques: el APIREST utilizado para comunicarse con el proyecto de partida y la propia app Android que se comunicará con dicho API para la obtención y volcado de los datos.

### C.2. Diseño de datos

En este apartado se incluirán los diagramas de clases y paquetes de ambos enfoques.

#### Diagrama de clases

En estos diagramas se ha decidido utilizar un criterio de color para diferenciar aquellas clases que forman parte del APIREST, en color azul y en amarillo aquellos que forman parte del proyecto de partida.

#### APIREST

Este paquete será el encargado de comunicarse con el Controlador del proyecto de partida. A continuación, se mostrarán de manera fraccionada cada una de las clases que lo forman y con qué clases del controlador del proyecto de partida se comunican.

**Clase ControlAccesoProfesor** Esta clase será la encargada de realizar el control de acceso por parte de los profesores a la aplicación web. Esta web

provee a los profesores de una clave privada dentro de la aplicación para poder utilizarla en la sincronización de sus cuestionarios con Moodle. Esta clase permite registrar a un nuevo profesor o comprobar si puede iniciar sesión. Ambas lógicas se controlan mediante una petición post. Para llevar a cabo este cometido se comunican con el controlador llamando al método registrarNuevoUsuario o login respectivamente.

- **Post:** esta petición será la encargada de manejar la lógica de toda la clase.

- *Registrarse en QuickTest:* para ello es necesario incluir dentro del propio cuerpo de la petición el email y contraseña del usuario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/controlAccesoProfesor/registro](http://localhost/_QuickTest_TFG/app/apiREST/controlAccesoProfesor/registro).

*Iniciar sesión en QuickTest:* para ello es necesario incluir en el cuerpo de la petición el email y la contraseña de usuario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/controlAccesoProfesor/login](http://localhost/_QuickTest_TFG/app/apiREST/controlAccesoProfesor/login).

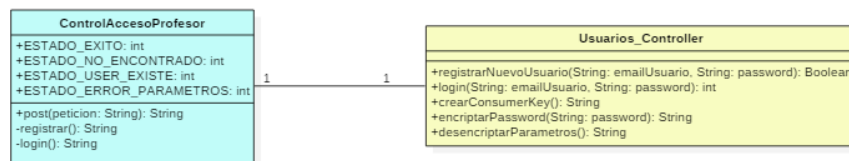


Figura C.1: Clase ControlAccesoProfesor

**Clase GestionCuestionario** La lógica de esta clase será toda aquella que se encuentre relacionada con la gestión sobre los Cuestionarios: *insertar, obtener, duplicar o borrar*. Cabe destacar, que en el controlador del proyecto de partida se ha incluido un método para poder duplicar un cuestionario existente, de esta manera evitamos comunicarnos con el modelo. Todas estas acciones se realizan mediante diferentes peticiones:

- **Get:** esta petición será la encargada de mostrar aquellos cuestionarios dado un identificador de asignatura. [http://localhost/\\_QuickTest\\_TFG/app/apiREST/gestionCuestionario/id](http://localhost/_QuickTest_TFG/app/apiREST/gestionCuestionario/id)
- **Post:** esta petición será la encargada de manejar dos lógicas.
  - *Duplicar un cuestionario:* para ello es necesario indicar el identificador del cuestionario. La petición es la siguiente: <http://localhost/>

[\\_QuickTest\\_TFG/app/apiREST/gestionCuestionario/duplicar/id](#).

*Insertar/Editar un cuestionario:* para ello es necesario incluir en el propio cuerpo de la petición si la petición es de insertar o de duplicar. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/gestionCuestionario/insertar](#).

- **Delete:** esta petición se encarga de eliminar un determinado cuestionario dado un identificador. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/gestionCuestionario/id](#).

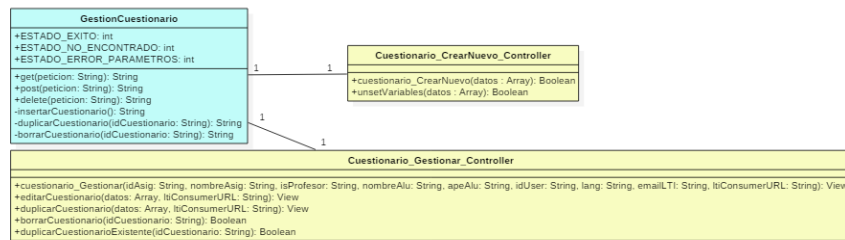


Figura C.2: Clase GestionCuestionario

Cabe destacar, que estas dos clases explicadas anteriormente no han sido utilizadas finalmente en la aplicación, al no ser necesarias para su correcto funcionamiento. Estas clases fueron desarrolladas durante las primeras fases del proyecto. Estas primeras fases, se dedicaron a la creación del APIREST que posteriormente se utilizaría en la aplicación Android.

**Clase SolucionCuestionario** Esta clase se encarga de la lógica a realizar durante la resolución de un cuestionario: *finalizar o mostrar resultado*. Todas estas acciones se realizan mediante las siguientes peticiones:

- **Post:** esta petición será la encargada de manejar la lógica a seguir al finalizar un cuestionario. Para ello, obtendrá el contenido del cuestionario resuelto de la petición, almacenará las respuestas del alumno y la nota del mismo. [http://localhost/\\_QuickTest\\_TFG/app/apiREST/solucionCuestionario/finalizar](#).
- **Get:** esta petición será la encargada de obtener la calificación de un alumno que haya resuelto el cuestionario desde la aplicación Android. Para ello, será necesario incluir en la propia petición el identificador del alumno y del cuestionario respectivamente. [http://localhost/\\_QuickTest\\_TFG/app/apiREST/solucionCuestionario/obtenerNota/idAlumno/idCuestionario](#)

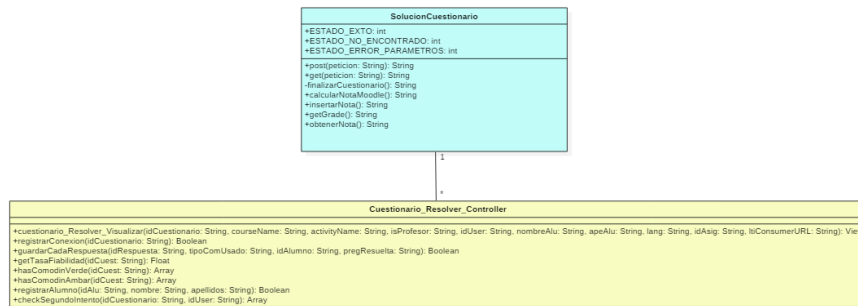


Figura C.3: Clase SolucionCuestionario

**Clase ObtenerCuestionario** Esta clase se encarga de la obtención de las preguntas y respuestas que forman un cuestionario, así como que preguntas tienen o no comodín y al finalizar un cuestionario, obtener información acerca de como ha sido resuelto el mismo. Destacar, que en el controlador no existe ningún método que permita obtener un cuestionario por identificador y el estado del mismo. Es por esto que esta funcionalidad ha sido incluida dentro del propio API. Además, ha sido necesario incluir dentro de este la funcionalidad de obtener los comodines verdes, ámbar y la información sobre como ha sido resuelto el cuestionario, ya que los métodos que se encuentran en el controlador corrompen la respuesta al incluir volcados del contenido.

- **Get:** esta petición será la encargada de manejar toda la lógica de la clase.
  - *Obtener cuestionario:* para ello sera necesario incluir en la propia petición el identificador del cuestionario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/obtenerCuestionario/obtener/idCuestionario](http://localhost/_QuickTest_TFG/app/apiREST/obtenerCuestionario/obtener/idCuestionario).
  - *Obtener el estado de un cuestionario:* es decir, si el cuestionario se encuentra resuelto o no por dicho alumno. Para ello, será necesarios incluir en la propia petición el identificador del alumno y el identificador de cuestionario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/obtenerCuestionario/estado/idAlumno/idCuestionario](http://localhost/_QuickTest_TFG/app/apiREST/obtenerCuestionario/estado/idAlumno/idCuestionario).
  - *Obtener comodín verde:* es decir, permite obtener aquellas preguntas que tienen comodín verde, dado un cuestionario. Para ello es necesario incluir en la propia petición el identificador del cuestionario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/obtenerCuestionario/obtenerComodin/verde/idCuestionario](http://localhost/_QuickTest_TFG/app/apiREST/obtenerCuestionario/obtenerComodin/verde/idCuestionario).
  - *Obtener comodín ámbar:* es decir, permite obtener aquellas preguntas que tienen comodín ámbar, dado un cuestionario. Para ello es

necesario incluir en la propia petición el identificador del cuestionario. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/obtenerCuestionario/obtenerComodin/ambar/idCuestionario](http://localhost/_QuickTest_TFG/app/apiREST/obtenerCuestionario/obtenerComodin/ambar/idCuestionario).

- *Obtener información sobre un cuestionario resuelto*: nos permite obtener información sobre como ha sido resuelto un cuestionario, es decir, en que grupo se encuentra, cuantos comodines a utilizado, etc. Para ello es necesario incluir en la propia petición el identificador de cuestionario y alumno. La petición es la siguiente: [http://localhost/\\_QuickTest\\_TFG/app/apiREST/obtenerCuestionario/feedback/idCuestionario/idAlumno](http://localhost/_QuickTest_TFG/app/apiREST/obtenerCuestionario/feedback/idCuestionario/idAlumno).

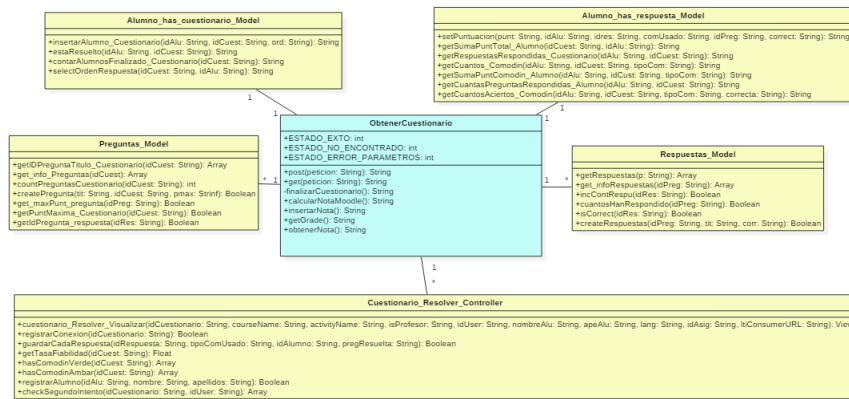


Figura C.4: Clase ObtenerCuestionario

## Android

En la siguiente ilustración podemos ver el diagrama de clases completo de la aplicación Android. Para ello, se ha decidido utilizar el color verde para representar dichas clases.

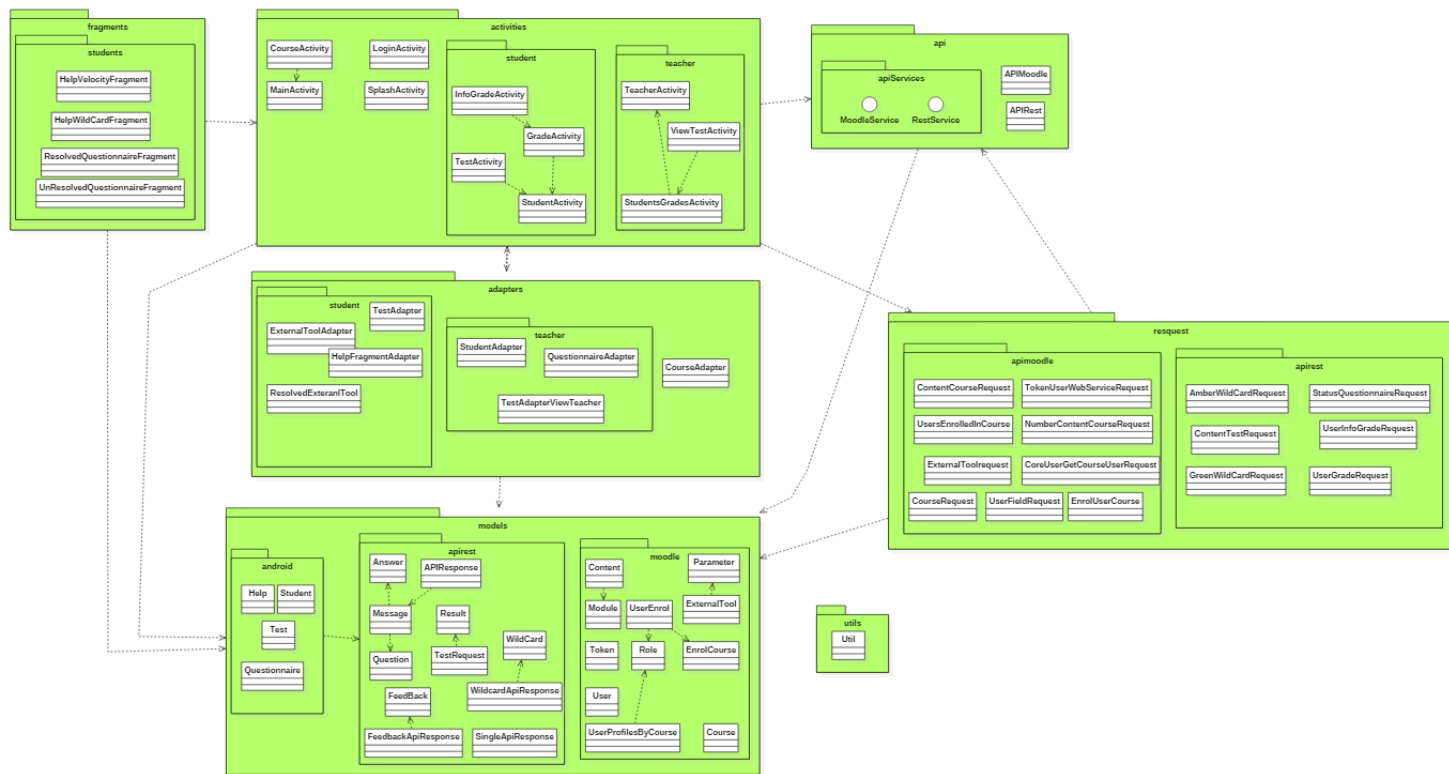


Figura C.5: Diagrama de clases de la aplicación.

### C.3. Diagrama de paquetes

Para estos diagramas se ha decidido utilizar el mismo criterio de color empleando anteriormente en el diagrama de clases.

#### APIREST

Este diagrama muestra un desglose de todos los componentes que entran en contacto en el API, como ya se ha comentado en color amarillo corresponde a la lógica dada en el proyecto de partida y en azul el APIREST desarrollado para interactuar con el controlador. Destacar que en color azul representa la estructura que forma el BackEnd de la aplicación. Toda esta estructura se encuentra en `_QuickTest.TFG_app.apiREST` y esta formada por:

- **modelos:** en este paquete se encuentran todas aquellas clases que se comunican con el controlador del proyecto de partida.
- **test\_apiRest:** en este paquete se encuentran las pruebas unitarias creadas para probar comprobar el correcto funcionamiento del APIREST.
- **utilidades:** en este paquete se encuentran ciertas clases y ficheros de utilidad.
- **vista:** en este paquete se encuentran aquellas clases encargadas de gestionar la lógica de la vista de los datos.

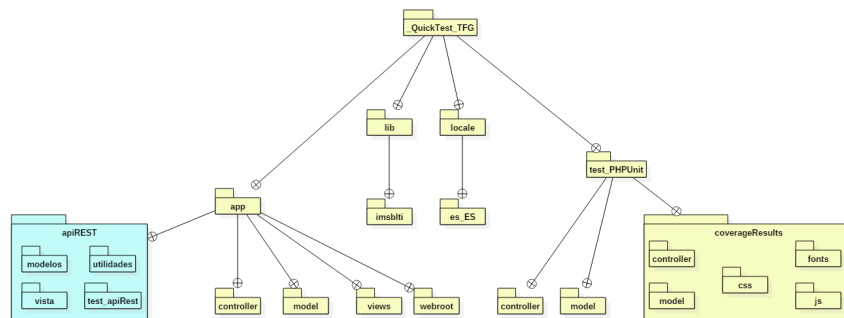


Figura C.6: Diagrama de paquetes del APIREST

#### Aplicación Android

En este diagrama se muestra el desglose de todos los componentes que forman la arquitectura del FrontEnd de la aplicación. Toda esta estructura se encuentra dentro de `QuickTest.Android.app.src` y esta formada por:



- **main.activities:** en este paquete se encuentran todas las actividades necesarias para la construcción de la aplicación.
- **main.adapters:** en este paquete se encuentran todos los adaptadores creados para modificar el layout de las diferentes actividades y fragmentos.
- **main.api:** en esta paquete se encuentran todas clases encargadas de gestionar la correcta conexión de la aplicación con el APIREST y con el web services de Moodle.
- **main.fragments.students:** en este paquete se encuentran todos los fragmentos utilizados para la construcción de la aplicación.
- **main.models:** en este paquete se encuentran todas las clases necesarias para el tratamiento de los datos proporcionados por las diferentes peticiones, tanto al APIREST como al web services de Moodle. Es decir, estas clases son las encargadas de realizar la conversión de los datos en formato JSON a objetos Java o viceversa.
- **main.request:** en este paquete se encuentran todas aquellas clases encargadas de realizar una petición síncrona al APIREST o al web service de Moodle.
- **main.utils:** en este paquete se encuentra una clases encargada de la gestión de las SharedPreferences utilizadas para el correcto funcionamiento del login.
- **test:** en este paquete se encuentran las pruebas unitarias sobre el paquete main.models.
- **androidTest:** en este paquete se encuentra las pruebas automatizadas de interfaz de usuario sobre la aplicación.

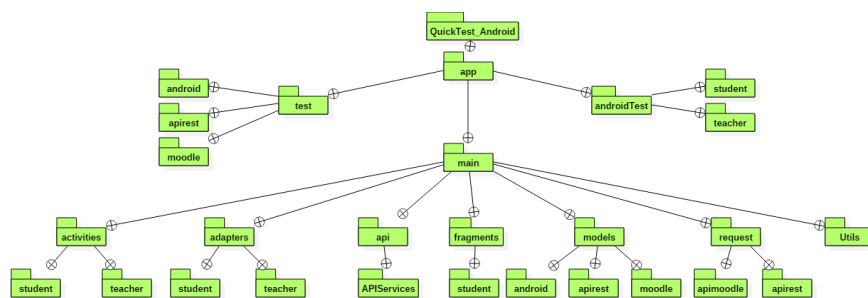


Figura C.7: Diagrama de paquetes de la aplicación Android

## C.4. Diseño procedimental

### Diagrama de secuencias

En esta sección se incluyen los diagramas de secuencias o de interacción, encargados de mostrar de una manera mas aproximada la interacción del sistema al realizar cada caso de uso.

#### Diagrama de secuencias que implementa CU1: Iniciar sesión

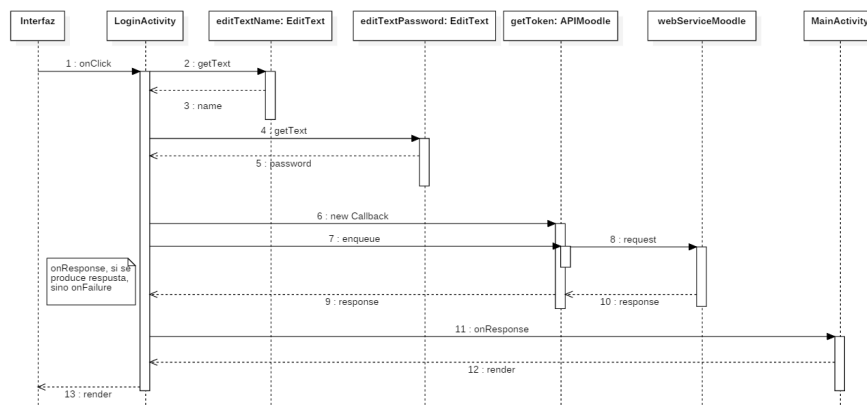


Figura C.8: Diagrama de secuencias caso de uso 1.

#### Diagrama de secuencias que implementa CU2: Olvidar campos

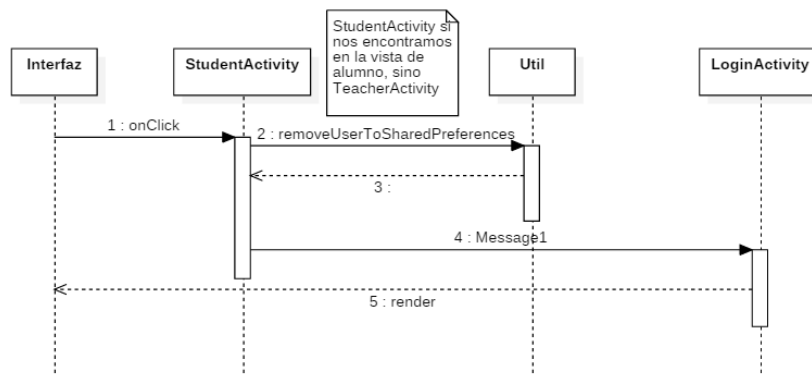


Figura C.9: Diagrama de secuencias caso de uso 2.

#### Diagrama de secuencias que implementa CU3: Recordar campos

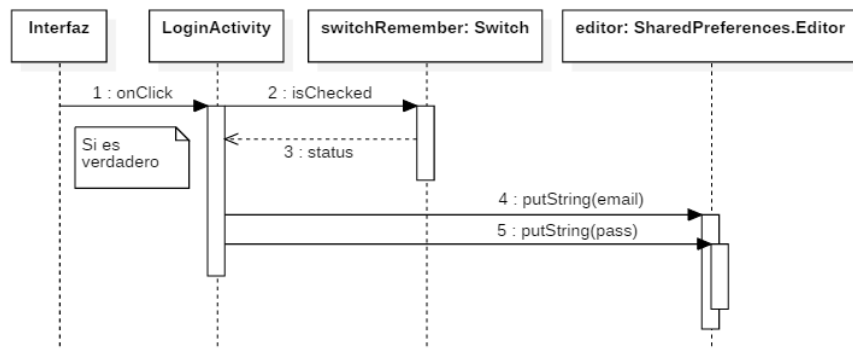


Figura C.10: Diagrama de secuencias caso de uso 3.

### Diagrama de secuencias que implementa CU4: Mostrar cursos

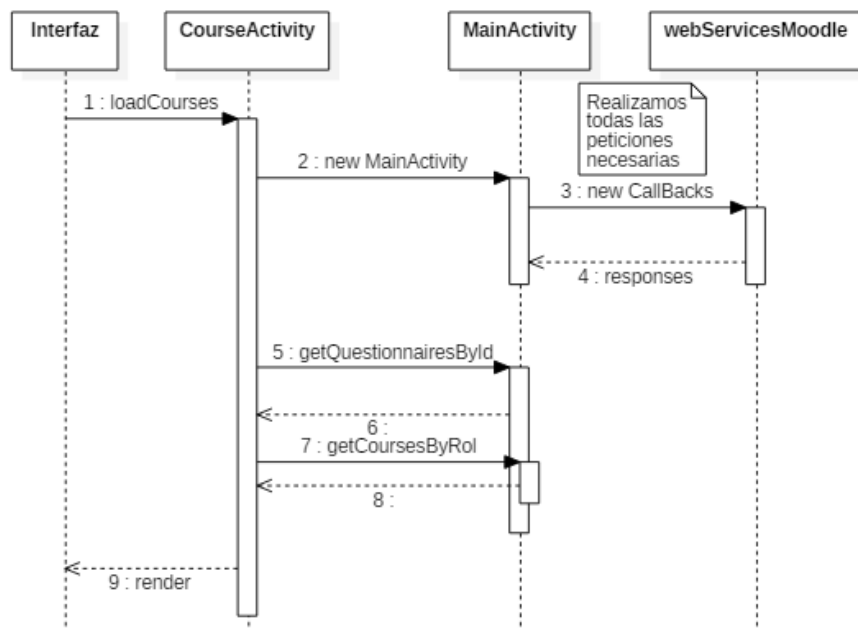


Figura C.11: Diagrama de secuencias caso de uso 4.

### Diagrama de secuencias que implementa CU5: Mostrar cuestionarios

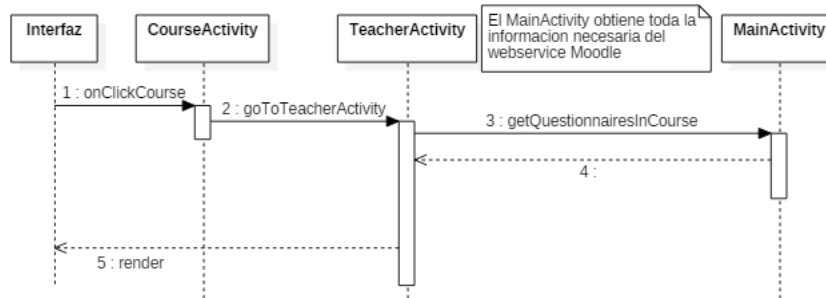


Figura C.12: Diagrama de secuencias caso de uso 5 desde el rol de profesor.

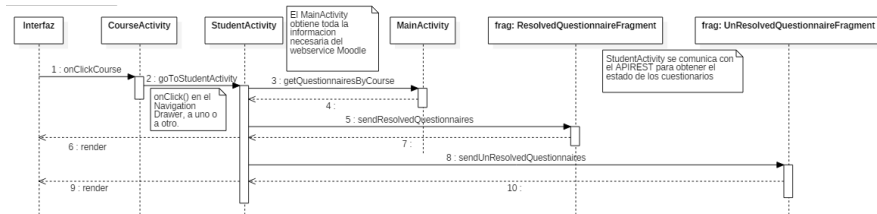


Figura C.13: Diagrama de secuencias caso de uso 5 desde el rol de alumno.

### Diagrama de secuencias que implementa CU6: Resolver cuestionario

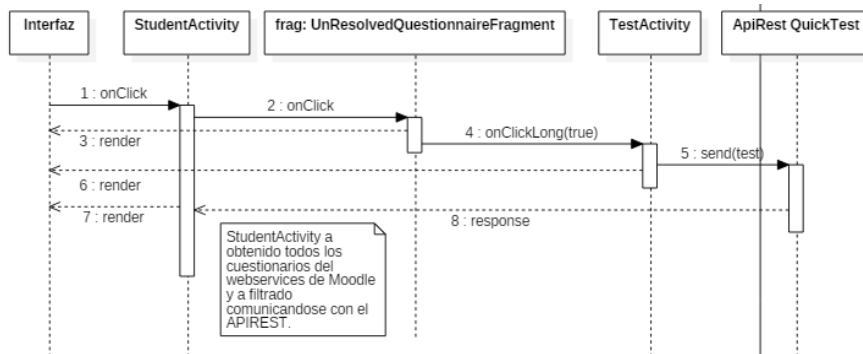


Figura C.14: Diagrama de secuencias caso de uso 6.

### Diagrama de secuencias que implementa CU7: Mostrar cuestionarios resueltos

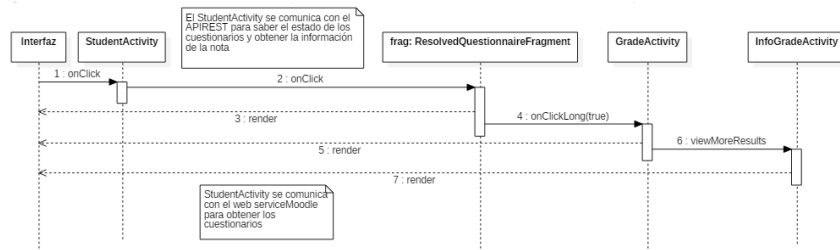


Figura C.15: Diagrama de secuencias caso de uso 7.

### Diagrama de secuencias que implementa CU8: Mostrar calificación de los alumnos

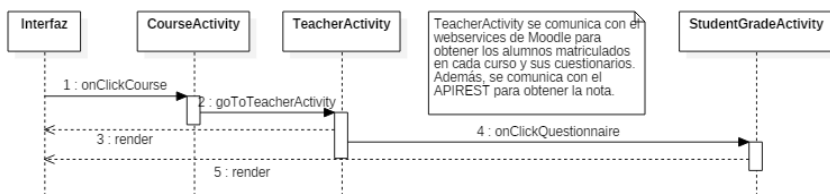


Figura C.16: Diagrama de secuencias caso de uso 8.

### Diagrama de secuencias que implementa CU9: Ver cuestionarios

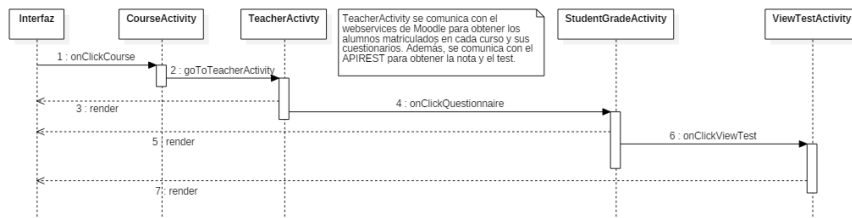


Figura C.17: Diagrama de secuencias caso de uso 9.

## C.5. Diseño arquitectónico

A continuación, se muestra el diagrama de despliegue de todo el proyecto, incluyendo la funcionalidad del proyecto de partida. Para ello se utiliza el mismo criterio de color anteriormente utilizado.

### Diagrama de despliegue

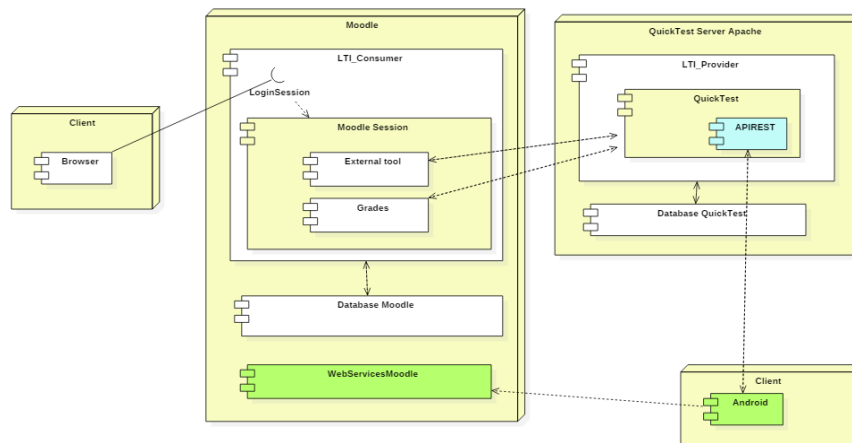


Figura C.18: Diagrama de despliegue del proyecto.

## Documentación técnica de programación

---

### D.1. Introducción

En esta sección se describen los requisitos mínimos para el funcionamiento de la aplicación, la configuración y las pruebas realizadas.

### D.2. Estructura de directorios

La estructura es la siguiente:

- `\`: contiene el proyecto completo incluyendo la aplicación Android, la documentación, el proyecto de partida y el archivo README.
- `\doc\latex\`: contiene la documentación del proyecto en formato  $\text{\LaTeX}$ .
- `\doc\docs\`: contiene la documentación del proyecto en formato PDF.
- `\src\_QuickTest\_TFG\`: contiene el proyecto de partida junto con el API-REST creado, las pruebas de dicho API y las pruebas correspondientes al proyecto inicial.
- `\src\_QuickTest\_TFG\app\apiREST\`: contiene toda la lógica del API-REST.
- `\src\_QuickTest\_TFG\app\apiREST\test_apiRest\`: contiene las pruebas unitarias para comprobar el correcto funcionamiento del BackEnd.
- `\src\InstalarBaseDeDatos\`: contiene la base de datos a importar.
- `\src\QuickTest_Android \`: contiene toda la lógica del FrontEnd.

- `\src\QuickTest_Android \doc\`: contiene el javadoc de la aplicación.
- `\src\QuickTest_Android \app\src\main\`: contiene todas las clases, layouts y imagenes que forman la aplicación.
- `\src\QuickTest_Android \app\src\test\`: contiene las pruebas unitarias sobre el FrontEnd.
- `\src\QuickTest_Android \app\src\androidTest\`: contiene las pruebas automatizadas de UI sobre el FrontEnd.
- `\tst\`: contiene los test del proyecto de partida.

## D.3. Manual del programador

### Entorno de desarrollo

Para el correcto funcionamiento del proyecto es necesario tener instalado las siguientes herramientas:

#### Java JDK

Java es un lenguaje de programación orientado a objetos y es el lenguaje mas popular para el desarrollo de aplicaciones Android. Para el desarrollo de este proyecto se ha utilizado la version 8. Esta versión se encuentra disponible en la web oficial de Java en el siguiente enlace: <http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html>

#### Android Studio

Android Studio es el entorno de desarrollo más utilizado para el desarrollo de aplicaciones Android. Además, es el IDE oficial recomendado por la plataforma. Cabe destacar que soporta la construcción basada en Gradle, incluye un propio emulador Android, refactorizaciones propias de Android, entre otras. Se encuentra disponible en el siguiente enlace: <https://developer.android.com/studio/index.html?hl=es-419>

#### XAMPP

XAMPP es un paquete de instalación de software libre utilizado como servidor de la aplicación. Este paquete contiene el servidor web Apache, base de datos MySQL, PHP y Perl. Se encuentra disponible en el siguiente enlace: <https://www.apachefriends.org/es/download.html>.



## Moodle

Moodle es un software orientado a la educación, cuyo principal objetivo es el de facilitar un entorno de calidad para proporcionar cursos. Esta herramienta se encuentra disponible en el siguiente enlace: <https://download.moodle.org/>

## Instalación de las herramientas

### Servidor

- Acceder a la web oficial de XAMPP en el siguiente enlace: <https://www.apachefriends.org/es/download.html>.
- Seleccionar la versión 5.6.30/PHP 5.6.30. Cabe destacar que es necesario instalar esta versión de XAMPP o inferiores, debido a que el proyecto de partida utiliza ciertas funcionalidades que se encuentran sin soporte en versiones superiores. Y comenzamos a descargar el instalador.

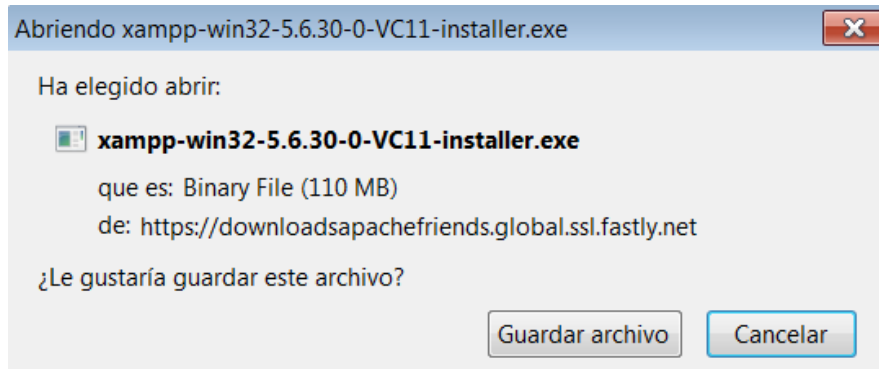


Figura D.1: Instalador de XAMMP

- Una vez descargado, procedemos a la instalación de esta herramienta. A continuación, procedemos a realizar la típica instalación de Siguiente – Siguiente y seleccionamos donde ubicar los archivos a instalar.
- Al finalizar la instalación se lanzará una alerta del firewall de Windows indicando que se necesita conceder acceso a Apache. Marcamos todas las opciones y finalizamos.

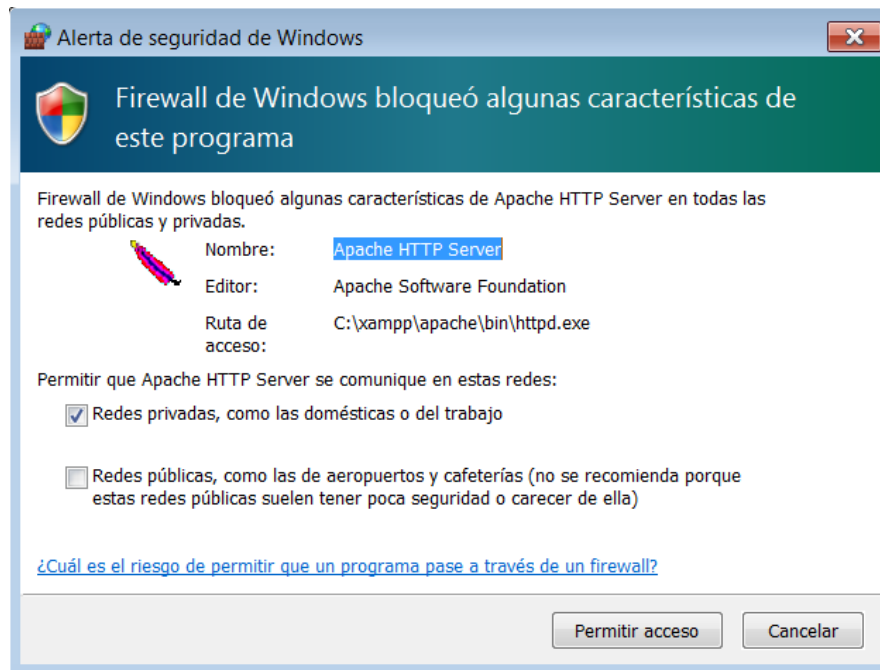


Figura D.2: Alerta del firewall de Apache

- Una vez instalado, ya tendremos instalado y disponible nuestro servidor XAMPP. Para el correcto funcionamiento del proyecto de partida y de QuickTest en Android es necesario arrancar Apache y MySQL al iniciar esta herramienta como podemos ver en la siguiente ilustración.

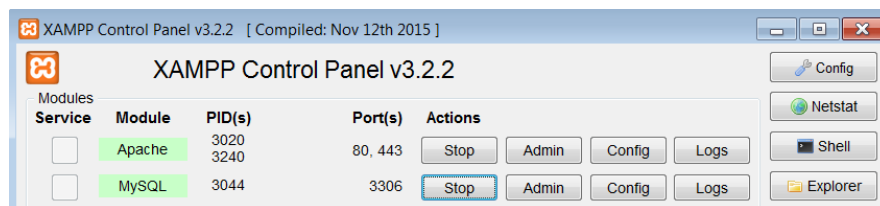


Figura D.3: Panel de XAMPP

- Además, al arrancar estos servicios nos volverá a saltar el firewall de Windows pidiendo conceder permisos a MySQL.

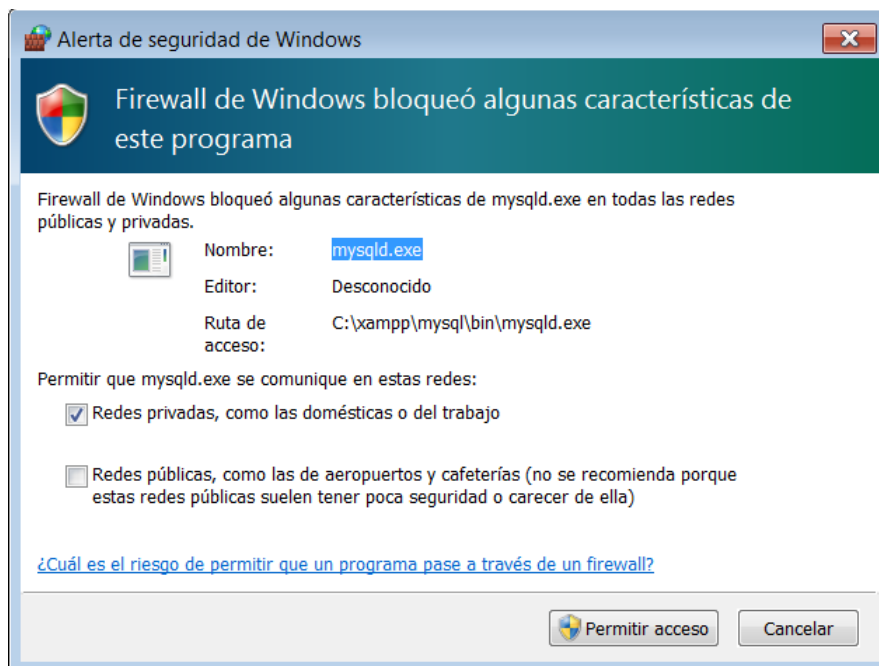


Figura D.4: Alerta del firewall de MySQL

- Concedemos los permisos y la instalación de nuestro servidor a finalizado.

## Moodle

Respecto al proyecto de partida se decidió utilizar como LMS, el mismo utilizado por la Universidad de Burgos, Moodle. Al tener que basarse en este proyecto no se ha realizado ningún cambio en la herramienta utilizada como LMS. No obstante, en esta ocasión la versión de Moodle empleada es la más actual hasta la fecha Moodle 3.2.2+. Este cambio en la versión utilizada es realizado al tener que utilizar las funciones que proporciona el propio web services de Moodle. De manera que cuanto más actual sea la versión, más funcionalidades tendremos a nuestra disposición.

- Descargamos el archivo comprimido de Moodle correspondiente a la versión 3.2.2+: <https://sourceforge.net/projects/moodle/files/Moodle/stable32/moodle-3.2.2.zip/download>
- Seleccionamos la versión comentada anteriormente y comenzamos la descarga del archivo comprimido.

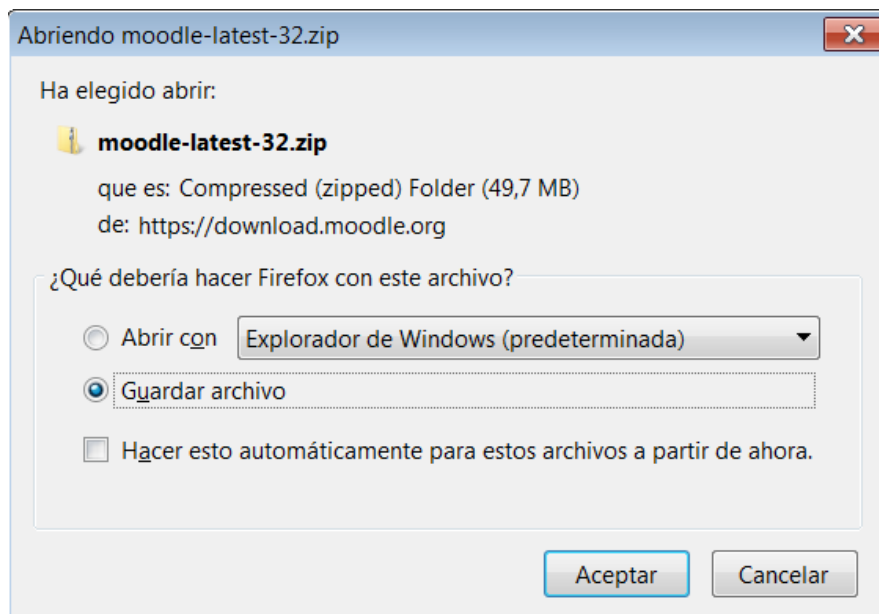


Figura D.5: Carpeta comprimida de Moodle

- Una vez finalizada la descarga, descomprimos el archivo en la carpeta htdocs. Esta carpeta se encuentra en el directorio raíz de XAMPP. Si la ruta de instalación de XAMPP se ha realizado con los campos por defecto, esta se encontrará en "C:\xampp\htdocs". Esta es la carpeta raíz del localhost, desde donde el servidor XAMPP comenzará a buscar cuando le mandemos una petición.
- Una vez descomprimido la carpeta abrimos un navegador y realizamos una petición al servidor indicando que queremos acceder a la URL de Moodle.

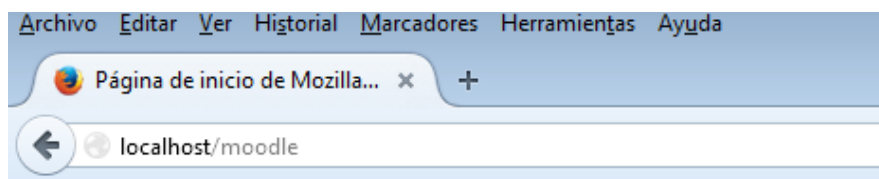
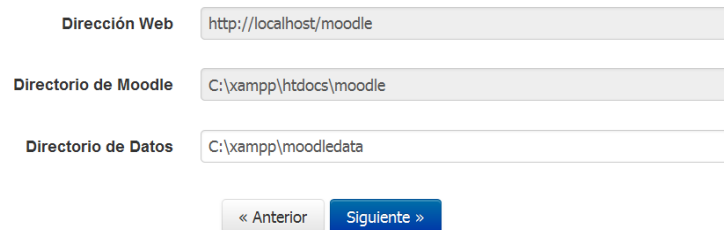


Figura D.6: url de Moodle

- Al no tener instalada ni configurada la herramienta, nos pedirá seleccionar el idioma deseado. Una vez seleccionado el idioma, tendremos que seleccionar la ruta de instalación de los nuevos componentes. En

este caso dejamos la configuración por defecto como podemos ver en la siguiente ilustración.



The screenshot shows the 'Dirección Web' (Web Address) field set to 'http://localhost/moodle'. The 'Directorio de Moodle' (Moodle Directory) field is set to 'C:\xampp\htdocs\moodle'. The 'Directorio de Datos' (Data Directory) field is set to 'C:\xampp\moodldata'. At the bottom, there are two buttons: '« Anterior' (Previous) and 'Siguiente »' (Next).

Figura D.7: Ubicación de donde instalar los directorios

- Posteriormente, tendremos que seleccionar el tipo de base de datos a utilizar. En este caso la opción elegida es MariaDB para evitar problemas de compatibilidad.



The screenshot shows the 'Seleccione el controlador de la base de datos' (Select the database controller) screen. A message box states: 'Moodle soporta varios tipos de servidores de base de datos. Por favor, póngase en contacto con el administrador del servidor si no sabe qué tipo usar.' Below this, the 'Tipo' (Type) dropdown menu is set to 'MariaDB (nativo/mariadb)'. At the bottom, there are two buttons: '« Anterior' (Previous) and 'Siguiente »' (Next).

Figura D.8: Selección de la base de datos

- Al finalizar la selección de la base de datos la instalación nos permitirá modificar la configuración por defecto de la misma. En este caso solamente añadiremos root en la sección Usuario de la base de datos como podemos ver en la siguiente ilustración y aceptamos la licencia.

Servidor de la base de datos	<input type="text" value="localhost"/>
Nombre de la base de datos	<input type="text" value="moodle"/>
Usuario de la base de datos	<input type="text" value="root"/>
Contraseña de la base de datos	<input type="password"/>
Prefijo de tablas	<input type="text" value="mdl_"/>
Puerto de la base de datos	<input type="text"/>
<div>« Anterior    Siguiente »</div>	

Figura D.9: Configuración de Moodle

- Antes de pulsar en el botón de siguiente tendremos que abrir una nueva pestaña en nuestro navegador e introducir el siguiente enlace localhost/phpmyadmin.

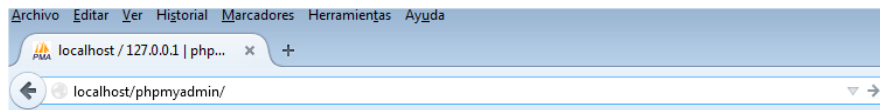


Figura D.10: url de MySql

- Una vez nos encontremos en la página, tendremos que cambiar el cotejamiento de la conexión al servidor como podemos ver en la siguiente ilustración.

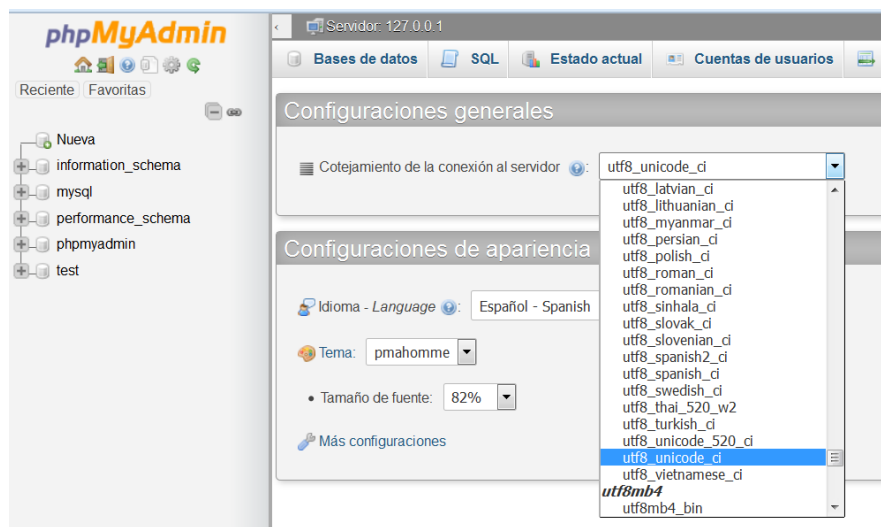


Figura D.11: Cambiamos el formato de la base de datos

- En esa misma página seleccionamos en Base de datos. Dentro de esta sección en el apartado Crear base de datos introducimos la siguiente información y la creamos.

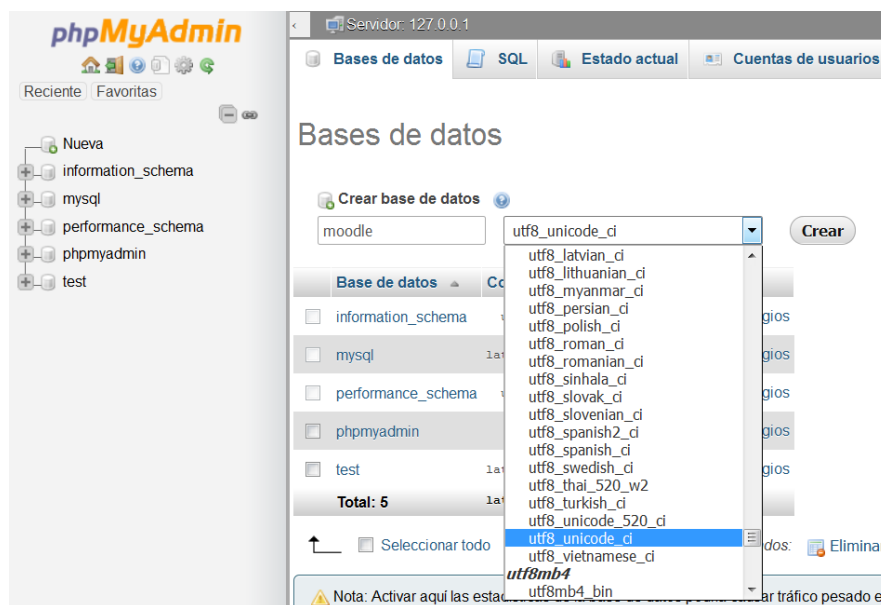


Figura D.12: Creación de la tabla Moodle con el formato deseado

- Una vez realizada la creación de la base de datos de manera manual, podemos volver a la pestaña de instalación de Moodle y proseguir con

la instalación.

- Si hemos realizado los anteriores pasos de forma correcta el sistema nos comunicará que el entorno cumple todos los requerimientos mínimos y continuamos con la instalación. Este proceso puede durar varios minutos dependiendo de la conexión a Internet.
- Una vez realizada la instalación de Moodle, es necesario la configuración de la plataforma. Para ello es necesario establecer un administrador de la misma. En nuestro caso la configuración es la que se puede ver en la siguiente ilustración.

The screenshot shows the 'General' configuration page of Moodle. It includes the following fields and values:

- Nombre de usuario:** admin
- Escoger un método de identificación:** Cuentas manuales
- Nueva contraseña:** Asdf1234! (with a strength indicator and a 'Forzar cambio de contraseña' checkbox)
- Nombre:** Admin
- Apellido(s):** Usuario
- Dirección de correo:** amg0098@alu.ubu.es

Figura D.13: Configuramos los datos del administrador de Moodle

- A continuación, tendremos que configurar el sitio. Para ello introducimos la siguiente información.

## Instalación

### Nuevos ajustes - Ajustes de la página principal

The screenshot shows the 'Nuevos ajustes - Ajustes de la página principal' form. It includes the following fields and values:

- Nombre completo del sitio (fullname):** UBUVirtual TFG
- Nombre corto para el sitio (una palabra) (shortname):** UBUVirtual

Figura D.14: Configuramos el sitio

- Una vez realizamos de forma correcta todos los anteriores pasos, la instalación y configuración de Moodle ha finalizado



## Configuración del Webservice Moodle

Antes de comenzar a la explicación de los distintos pasos a realizar, tendremos que encontrarnos logeado con los credenciales del administrador del sitio.

- Una vez logeado nos dirigimos a Administración del sitio. Allí seleccionamos la categoría Extensiones y dentro de esta categoría en Servicios Web.
- Si hemos realizado el punto anterior de manera correcta, nos encontramos en panel central de la configuración del Web Service. Este entorno de configuración se puede observar en la siguiente ilustración.

Pasos	Estado	Descripción
1. <a href="#">Habilitar Servicios Web</a>	No	Los servicios Web deben estar habilitados en las características avanzadas.
2. <a href="#">Habilitar los protocolos</a>	Ninguno	Al menos un protocolo debe estar habilitado. Por razones de seguridad, los protocolos que se van a utilizar deben estar habilitados.
3. <a href="#">Crear un usuario específico</a>		Se necesita un usuario de servicios web para representar el sistema que controla Moodle.
4. <a href="#">Comprobar privilegios del usuario</a>		El usuario debe tener las capacidades (privilegios) apropiadas de acuerdo al protocolo empleado, por ejemplo webservice/rest:use, webservice/soap:use. Para lograr esto, cree un rol de servicios web con las capacidades del protocolo habilitadas y asígnelo a usuario de servicios web como un rol del sistema.
5. <a href="#">Seleccione un servicio</a>		Un servicio es un conjunto de funciones de 'servicios web'. Permitirá acceder al usuario a un nuevo servicio. En la página "Añadir servicio" marque las opciones 'Habilitado' y 'Usuario autorizado'. Seleccione 'No se requiere permiso'.
6. <a href="#">Agregar funciones</a>		Seleccione las funciones requeridas para el servicio de nueva creación.
7. <a href="#">Seleccione un usuario específico</a>		Agregue el usuario de servicios web como un usuario autorizado.
8. <a href="#">Crear ficha para un usuario</a>		Crear ficha para usuario de servicios web.
9. <a href="#">Habilitar documentación del desarrollador</a>	No	Está disponible para los protocolos habilitados una documentación detallada sobre los servicios de la web.
10. <a href="#">Comprobar el servicio</a>		Simula el acceso externo al servicio usando el cliente de prueba de servicio web. Use un protocolo permitido con autenticación por ficha (token). <b>ADVERTENCIA: ¡Las funciones que usted pruebe SERÁN EJECUTADAS para este usuario, por lo que debe ser cauteloso con lo que decida probar!</b>

Figura D.15: Panel de configuración del Web Service de Moodle

- Para llevar a cabo la correcta configuración vamos a realizar los pasos de manera ordenada de menor a mayor como se puede ver en la anterior ilustración.
  - **Habilitar los Servicios Web:** para ello entramos en el paso 1, y activamos el checkbox y guardamos los cambios.
  - **Administrar protocolos:** para poder interactuar con el Web Service tendremos que habilitar al menos uno de ellos. En nuestro caso

utilizaremos REST al ser el mas común utilizado en la comunidad de los Web Service de Moodle. Además, permite mostrar el contenido de retorno en formato JSON, el cual es muy ventajoso para el el tratamiento de los datos en la aplicación Android. Por último, es aconsejable habilitar la documentación al ser un gran material de apoyo y guardamos los cambios.

- Como se ha comentado anteriormente, vamos a realizar la configuración de manera ordenada. No obstante, en el este ejemplo vamos a ignorar los pasos 3, 4, 5 y 6. En estos pasos, se crea un usuario específico para interactuar con el Web Service, este usuario debe de tener ciertos privilegios para poder desempeñar esta funcionalidad, se crea un servicio y se agregan las funcionalidades de interés a dicho servicio. Estos pasos son ignorados, ya que se va a utilizar el propio usuario administrador del sitio, al tener todos los privilegios necesarios y de esta manera no limitaremos las funcionalidades del Web Service al incluir todas ellas. Destacar, que no es recomendable utilizar el mismo usuario administrador de Moodle por motivos de seguridad, para interactuar con el Web Service. Al ser un trabajo meramente didáctico, utilizaremos este mismo usuario para evitar complicaciones. Por último, estos pasos omitidos se encuentran explicados en el siguiente enlace: <http://www.vitalinnova.com/servicios-web-en-moodle-2-0/>
- **Seleccione un servicio:** para poder habilitar el servicio externo tendremos que seleccionar el paso 5. Una vez allí, click sobre editar en servicio *Moodle mobile web service* y activamos el checkbox.

La configuración del Web Service ha finalizado. Para poder comprobar el correcto funcionamiento del mismo, vamos a pedir a este, el token del administrador. Para ello abrimos un nuevo navegador e introducimos la siguiente url: [http://localhost/moodle/login/token.php?username=admin&password=Asdf1234!&service=moodle\\_mobile\\_app](http://localhost/moodle/login/token.php?username=admin&password=Asdf1234!&service=moodle_mobile_app). Si todo ha ido correctamente la respuesta será la siguiente:

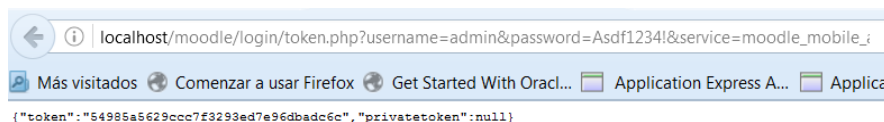


Figura D.16: Obtención del token del administrador

## Instalar base de datos

A continuación, vamos a importar la base de datos que utiliza QuickTest. Esta base de datos es la misma proporcionada por el proyecto de partida,

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN 52

pero se ha incluido una nueva tabla para el almacenamiento de las notas de los cuestionarios resueltos desde la aplicación móvil. Para llevar a cabo esta tarea tendremos que seguir los siguientes pasos:

- Dentro del proyecto nos situamos en la ruta `src/InstalarBaseDeDatos`.
- En esta carpeta tendremos un único archivo llamado: `instalar_BaseDatosQuickTest.sql`
- Abrimos una nueva pestaña en nuestro navegador e introducimos el siguiente enlace.

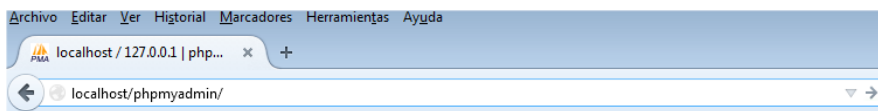


Figura D.17: Url de Mysql

- Una vez nos encontremos en el panel de administración de la base de datos tendremos que importar el archivo del punto 2. Para ello seguimos los siguientes pasos.

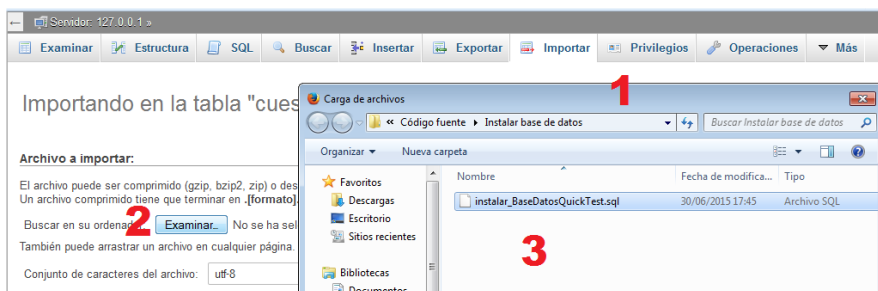


Figura D.18: Añadimos el fichero de la base de datos

- Una vez importado el archivo seleccionamos continuar y el proceso de importación de la base de datos de QuickTest ha finalizado.

### Instalar/Configurar QuickTest

Para instalar QuickTest tendremos que copiar la carpeta `_QuickTest.TFG` en la carpeta raíz del localhost de XAMPP para que el servidor pueda buscar el directorio pedido al mandarle una petición. Para ello tendremos que:

- Copiar la carpeta `_QuickTest.TFG` en la carpeta `htdocs` de XAMPP.

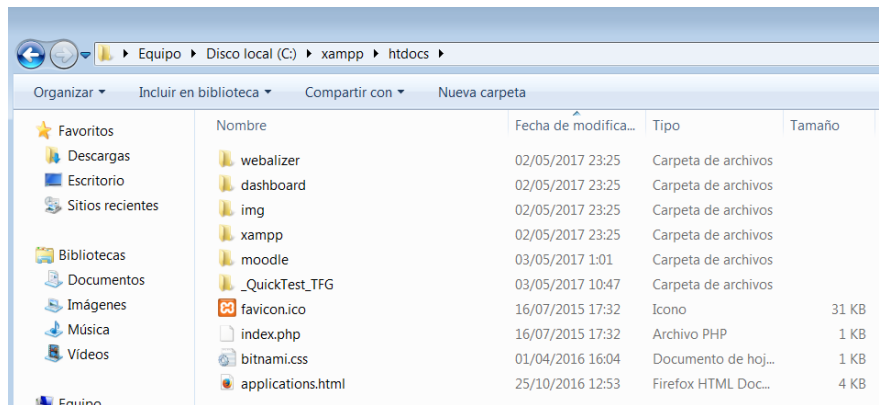


Figura D.19: Añadimos la carpeta al htdocs

## Registrar un usuario en QuickTest

Para poder utilizar QuickTest desde un LMS, en este caso Moodle, tendremos que estar previamente registrado en QuickTest. Para ello tendremos que:

- Abrir una nueva pestaña en el navegador e introducir el siguiente enlace:  
[http://localhost/\\_QuickTest\\_TFG/app/views/managementView/startQuickTest\\_View.php](http://localhost/_QuickTest_TFG/app/views/managementView/startQuickTest_View.php)

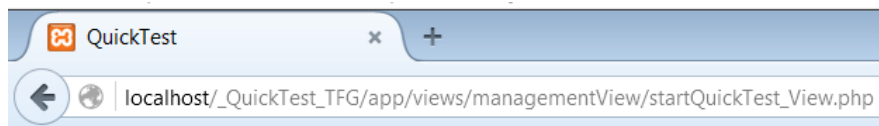


Figura D.20: Abrimos la página de inicio de QuickTest

- Seleccionamos Acceder/Registrarse. Una vez seleccionado este botón aparecerá una ventana emergente que permitirá Entrar o Registrarse.
- Para llevar a cabo el registro de un nuevo usuario en QuickTest tendremos que seguir dos criterios importantes.
  - En el campo Email tendremos que usar el mismo que en Moodle.
  - En el campo Contraseña podremos utilizar cualquiera siempre y cuando la contraseña este formada entre 5 y 10 caracteres.

En la siguiente ilustración podemos observar un inicio de sesión. Este usuario se encuentra por defecto en la base de datos inicial por lo que si

se desea podemos saltarnos el paso de registrar un usuario. La contraseña es 12345.



The image shows a login interface. At the top, there are three links: 'Entrar' (highlighted with a light blue background), 'Registrarse', and '¿Por qué?'. Below these links, there is a label 'Email:' followed by a text input field containing the email address 'amg0098@alu.ubu.es'. Underneath the email field is a label 'Contraseña:' followed by a password input field with five dots representing the password. Below the password field is a blue button labeled 'Entrar'. At the bottom of the form, there is a grey button labeled 'Cerrar'.

Figura D.21: Entramos en QuickTest

- Una vez realizado el registro y su posterior inicio de sesión el sistema informará con una ventana de bienvenida.

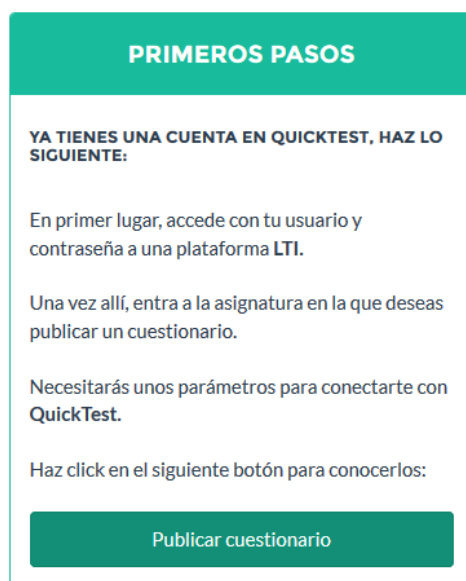


Figura D.22: Panel de primeros pasos

- Al pinchar sobre Publicar cuestionario se abrirá una ventana emergente. Es, en esta ventana, donde se encuentran los tres campos necesarios para poder publicar y enlazar nuestros cuestionarios con Moodle. La información necesaria es la siguiente.

**PARA PUBLICAR, SIGUE ESTOS PASOS:** ×

---

Crea una actividad desde tu plataforma LTI de tipo: **ACTIVIDAD EXTERNA**

Y copia estos datos en las casillas correspondientes

URL de inicio	Clave de cliente	Shared secret
localhost/_QuickTest_TFG/index.php	drQ\$1mF	Tu contraseña de QuickTest

**IMPORTANTE:**

En la pestaña 'privacidad', no olvides marcar:

- ☒ Aceptar calificaciones desde la herramienta
- ☒ Compartir el e-mail del usuario con la herramienta
- ☒ Aceptar calificaciones desde la herramienta

**Close**

Figura D.23: Ventana emergente de publicar un cuestionario

## Configurar/Publicar un cuestionario en Moodle

En esta sección vamos a configurar Moodle para poder utilizar QuickTest. Para llevar a cabo esta tarea tendremos que seguir los siguientes pasos.

### Configuración

- Abrimos una nueva pestaña del navegador y accedemos a Moodle.
- Una vez allí tendremos que iniciar sesión con el nombre de usuario y contraseña propios del administrador del sistema, en este caso son las credenciales utilizadas en la sección D.3.



Figura D.24: Inicio de sesión de Moodle

- Una vez iniciada la sesión tendremos que crear un curso donde añadir nuestro cuestionario, para ello tendremos que:
  - Entramos en Administración del sitio.
  - Una vez allí entramos en la categoría de Cursos y dentro de esa sección en Administrar cursos y categorías.

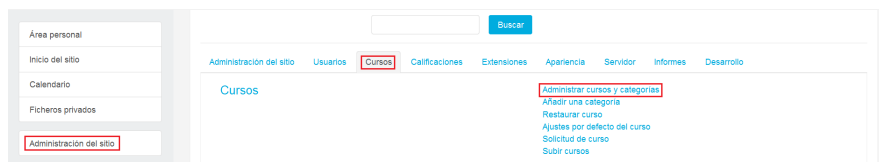


Figura D.25: Creamos un curso

- Una vez en esa sección seleccionamos Crear un nuevo curso y rellenamos los campos con la siguiente información. Al finalizar guardamos la información.

## Crear un nuevo curso

▼ General

Nombre completo del curso \* ?

Nombre corto del curso \* ?

Figura D.26: Rellenamos los campos del nuevo curso

- Al crear el curso tendremos que matricular a los usuarios. Para ello vamos a crear dos usuarios.
  - Tendremos que entrar en Administración del sitio, una vez allí tendremos que ir a la categoría de Usuarios y dentro de allí a la sección Crear un nuevo usuario.

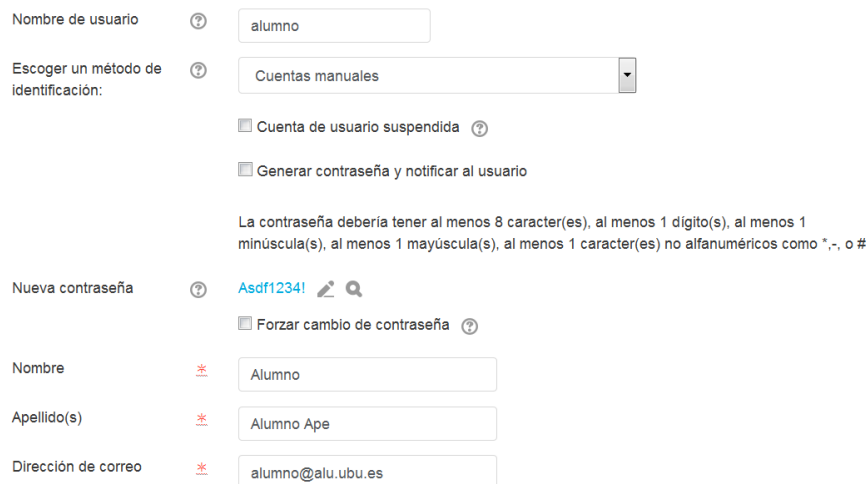


Figura D.27: Creamos los nuevos usuarios

- A continuación, tendremos que crear dos usuarios como podemos ver en las siguientes ilustraciones.



## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN 58



Nombre de usuario ⓘ alumno

Escoger un método de identificación: ⓘ Cuentas manuales

☐ Cuenta de usuario suspendida ⓘ

☐ Generar contraseña y notificar al usuario

La contraseña debería tener al menos 8 caracter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 caracter(es) no alfanuméricos como \*, -, o #

Nueva contraseña ⓘ Asdf1234! ⓘ

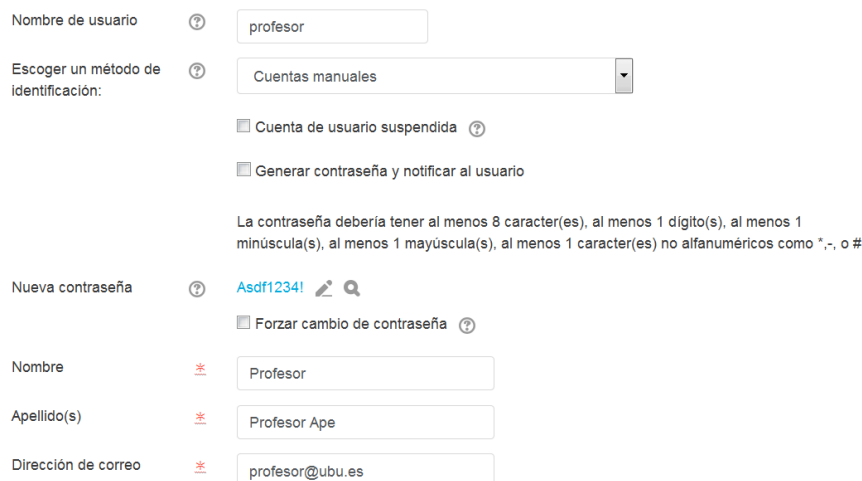
☐ Forzar cambio de contraseña ⓘ

Nombre \* Alumno

Apellido(s) \* Alumno Ape

Dirección de correo \* alumno@alu.ubu.es

Figura D.28: Rellenamos los campos del primer usuario



Nombre de usuario ⓘ profesor

Escoger un método de identificación: ⓘ Cuentas manuales

☐ Cuenta de usuario suspendida ⓘ

☐ Generar contraseña y notificar al usuario

La contraseña debería tener al menos 8 caracter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 caracter(es) no alfanuméricos como \*, -, o #

Nueva contraseña ⓘ Asdf1234! ⓘ

☐ Forzar cambio de contraseña ⓘ

Nombre \* Profesor

Apellido(s) \* Profesor Ape

Dirección de correo \* profesor@ubu.es

Figura D.29: Rellenamos los campos del segundo usuario

- Una vez creados los usuarios, tendremos que matricular dichos usuarios en el curso creado para ello tendremos que entrar en Administración del sitio, una vez allí tendremos que ir a la categoría de Cursos y dentro de allí a la sección Administrar cursos y categorías.
- Allí seleccionamos el nuevo curso creado y pinchamos en Usuarios matriculados.
- A continuación, matriculamos ambos usuarios creados anteriormente, el alumno con rol de Estudiante y el profesor con el rol de Profesor.

## Publicar

Para llevar a cabo la publicación de un cuestionario tendremos que:

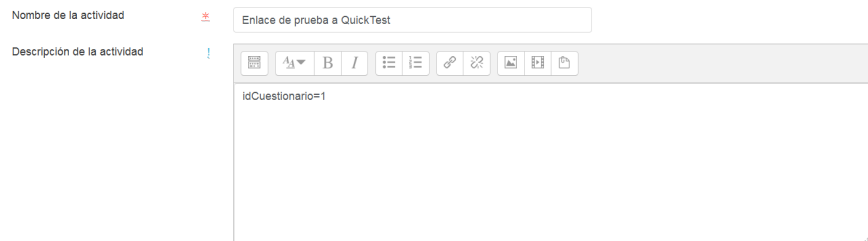
- Tendremos que iniciar sesión con algún usuario con rol de profesor.
- Una vez allí, seleccionamos un curso, activamos la edición y añadimos una herramienta externa.
  - Para ello, tendremos que seleccionar Agregue una actividad o recurso.



Figura D.30: Agregamos una nueva herramienta externa

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN60

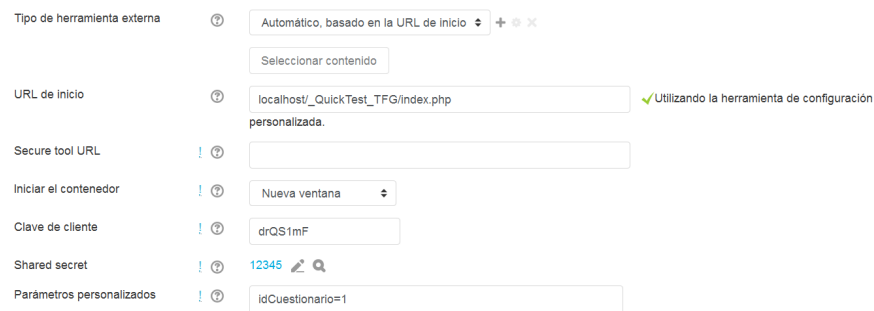
- En este instante comenzamos a configurar la herramienta externa para poder utilizar QuickTest. Tendremos que rellenar los campos con la siguiente información.



Nombre de la actividad: Enlace de prueba a QuickTest

Descripción de la actividad: idCuestionario=1

Figura D.31: Configuramos la nueva herramienta externa 1



Tipo de herramienta externa: Automático, basado en la URL de inicio

URL de inicio: localhost/\_QuickTest\_TFG/index.php

Secure tool URL:

Iniciar el contenedor: Nueva ventana

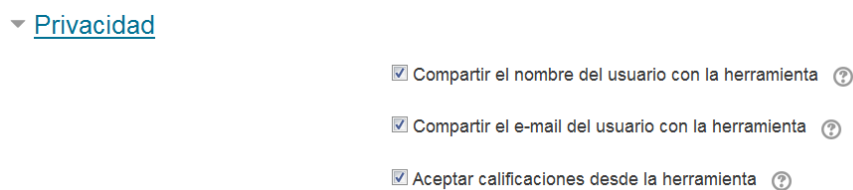
Clave de cliente: drQS1mF

Shared secret: 12345

Parámetros personalizados: idCuestionario=1

Figura D.32: Configuramos la nueva herramienta externa 2

- Además, revisar que en el apartado Privacidad se encuentren activas las siguientes opciones.



▼ Privacidad

☒ Compartir el nombre del usuario con la herramienta

☒ Compartir el e-mail del usuario con la herramienta

☒ Aceptar calificaciones desde la herramienta

Figura D.33: Campos de la privacidad

- En este instante la publicación ha finalizado.

## D.4. Pruebas del sistema

Para comprobar el correcto funcionamiento se han desarrollado un conjunto de pruebas.

### APIREST

En esta sección se recogen el conjunto de pruebas utilizadas para comprobar el correcto funcionamiento del BackEnd de la aplicación. Destacar que solamente se han realizado pruebas unitarias sobre el APIREST, ya que el único cometido de este es el de obtener y añadir información.

#### Pruebas unitarias

Las pruebas unitarias comprueban el funcionamiento de una única función, independientemente del resto. Para la construcción de este tipo de pruebas se ha utilizado PHPUnit. Este framework se instala al instalar XAMPP.

#### Ejecución de las pruebas unitarias

Las pruebas unitarias se encuentran dentro del propio paquete que contiene al APIREST añadido al proyecto de partida dentro de la carpeta `apiREST.test_apiRest`.

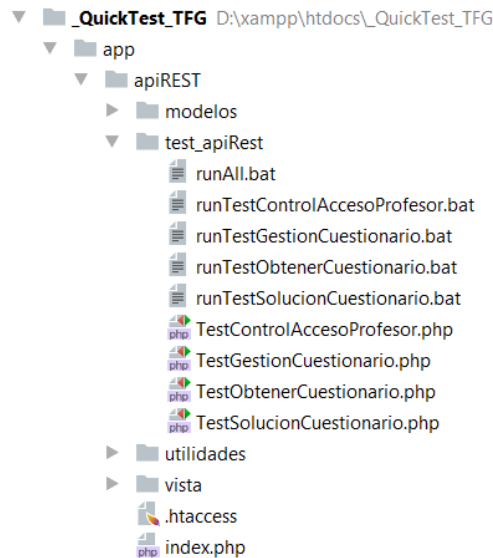


Figura D.34: Contenido de la carpeta `test_apiRest`

Dentro de este se encuentran las cuatro pruebas creadas para comprobar el correcto funcionamiento de cada una de las clases. Para llevar cabo la ejecu-

ción de dichas pruebas, podemos ejecutarlas de manera individual utilizando el respectivo archivo .bat de la clase o ejecutar todos mediante el archivo runAll.bat

## Android

En esta sección se recogen el conjunto de pruebas utilizadas para comprobar el correcto funcionamiento del FrontEnd de la aplicación.

### Pruebas unitarias

Como ya hemos comentado anteriormente, las pruebas unitarias comprueban el correcto funcionamiento de un método o funcionalidad independientemente del resto. Es por esto, que se ha decidido realizar pruebas sobre todas aquellas estructuras encargadas de contener la información proporcionada por el web services de Moodle, el propio APIREST o las propias estructuras de la aplicación.

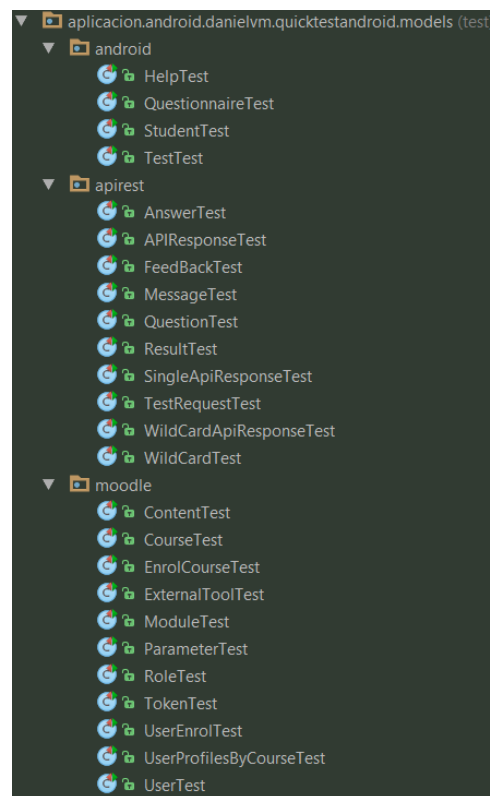


Figura D.35: Conjunto de pruebas unitarias del FrontEnd.

Estas pruebas podrán ejecutarse de dos maneras distintas:

- *Desde la propia consola de Windows:*
  - Situarnos en la raíz del proyecto QuickTest\_Android.
  - Ejecutar: `gradlew.bat testReleaseUnitTest`
  - Para visualizar los resultados abrir el archivo `index.html` en: `QuickTest_Android.app.build.reports.tests.testReleaseUnitTest`
- *Desde Android Studio:* para ello tendremos que utilizar Android Studio, situarnos en la carpeta en la se encuentra dichas pruebas, botón derecho y Run Test.

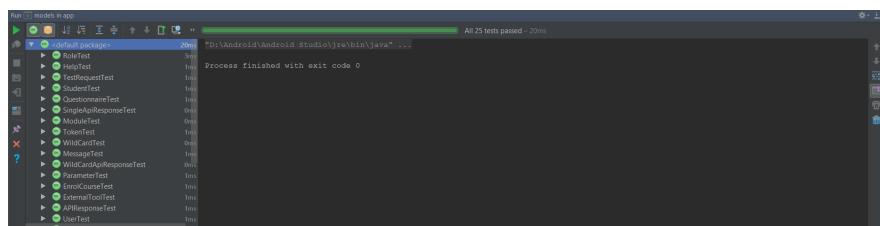


Figura D.36: Ejecución y resultado de las pruebas unitarias sobre el FrontEnd desde Android Studio.

## Pruebas automatizadas de UI

Este tipo de pruebas consiste en crear una batería de pruebas automatizada, que comprueben el correcto funcionamiento de la interfaz de usuario y valide los controles de la misma.

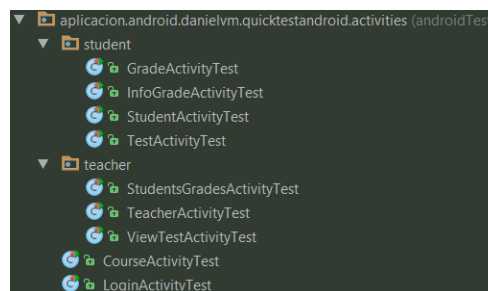


Figura D.37: Conjunto de interfaz de usuario del FrontEnd.

Como podemos observar se ha decidido crear un caso de prueba por caso de uso. Al necesitar un emulador o un dispositivo conectado para poder ejecutar estas pruebas, tendremos que utilizar Android Studio. Para ello, nos situamos en la carpeta en la que se encuentran dichas pruebas, botón derecho sobre la prueba a ejecutar y Run Test.

### Pruebas de estrés

Las pruebas de estrés se utilizan para comprobar el funcionamiento de la aplicación bajo grandes cargas de trabajo. Es por esto, que se ha decidido realizar estas pruebas en la aplicación para localizar posibles pérdidas de información.

Para ello se ha utilizado la herramienta Monkey de Android. Esta herramienta lanza en un emulador o dispositivo, un conjunto de flujos o eventos pseudo-aleatorias sobre la propia interfaz de la aplicación.

Para llevar a cabo la ejecución de esta prueba tendremos que:

1. Tener iniciado el emulador de Android Studio.
2. Ejecutar el archivo testMonkey.bat, situado en la raíz del proyecto Android.

### Pruebas de cobertura

Las pruebas de cobertura son las encargadas de medir la cantidad de código de un programa que nuestras pruebas cubren. En nuestro caso al realizar únicamente pruebas sobre las estructuras de almacenamiento de datos el nivel de código cubierto será pequeño. Para realizar este tipo de pruebas se ha utilizado el propio entorno de Android Studio.

1. Añadir al build.gradle de la aplicación.

```
buildTypes {  
    debug {  
        testCoverageEnabled = true  
    }  
}
```

Figura D.38: Habilitamos las pruebas de cobertura.

De esta manera permitimos la ejecución de este tipo de pruebas desde el propio IDE.

2. Nos situamos en la carpeta que contiene los fuentes, botón derecho, Run Test Coverage.

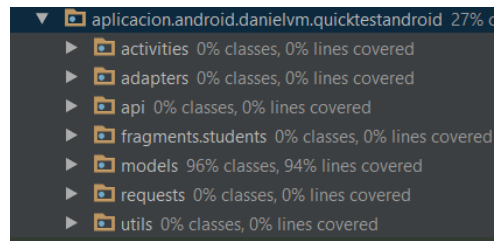


Figura D.39: Ejecución de las pruebas de cobertura sobre el paquete models.

### Pruebas de integración continua

Las pruebas de integración continua permite durante el desarrollo de una aplicación detectar posibles fallos cuantos antes. Para ello se ha decidido utilizar la herramienta sonarQube en su distribución online. Para ello es necesario incluir en el proyecto:

1. Seguir las indicaciones del vídeo proporcionado por el tribunal para la creación de una cuenta, generar el token, etc.
2. Dentro del proyecto tendremos que añadir:
  - a) Añadir al build.gradle del proyecto.

```
plugins {
    id "org.sonarqube" version "2.5"
}
```

Figura D.40: Añadimos la dependencia de sonarQube.

- b) Añadir al gradle.properties los parametros necesarios para su funcionamiento.

```
systemProp.sonar.host.url=https://sonarqube.com
systemProp.sonar.login=00f9883f372438436d54a845d3748fb8188a018c
systemProp.sonar.organization=danielpuente-dpg-github
```

Figura D.41: Configuración de SonarQube.

3. Para ejecutar esta prueba dentro de Android Studio bastará con introducir el siguiente comando.



## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN66

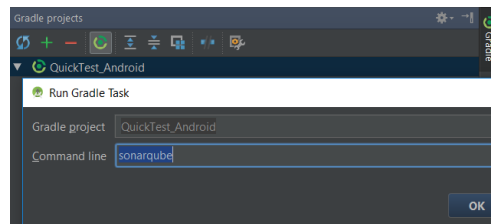


Figura D.42: Ejecución de SonarQube

4. El resultado de la prueba podrá visualizarse dentro de la distribución online de la herramienta.

---

## Documentación de usuario

---

### E.1. Introducción

Este manual se presentan y explican los requisitos necesarios para utilizar la aplicación, como instalar en el un dispositivo físico y otras pautas de configuración para su correcto funcionamiento.

### E.2. Requisitos de usuarios

Para poder utilizar la aplicación es necesario cumplir una serie de requisitos mínimos:

- Tener instalado las herramientas explicadas en [D.3](#).
- Configurar las herramientas como se encuentran explicadas en [D.3](#).

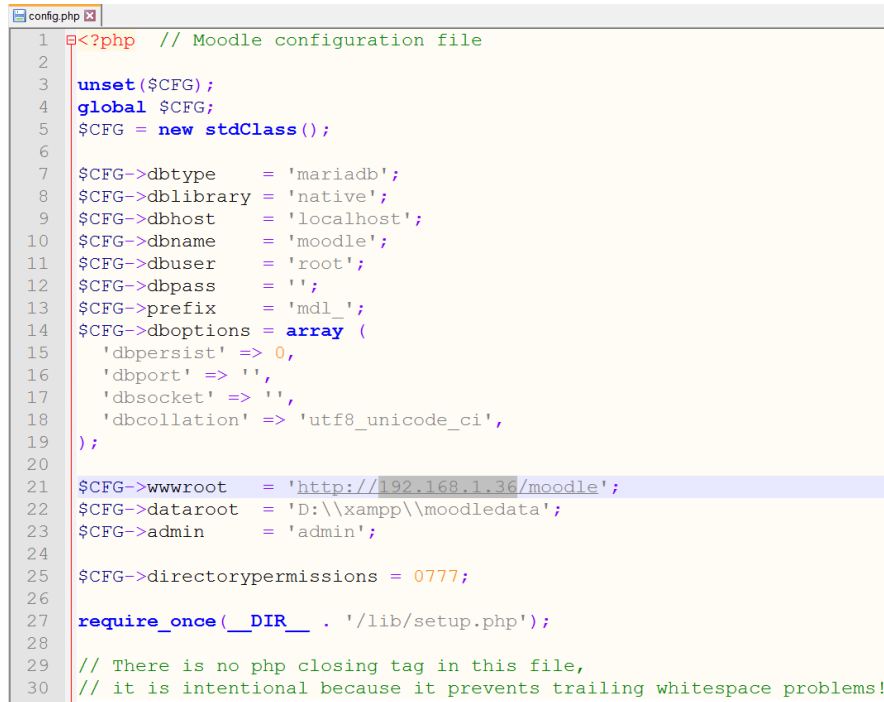
### E.3. Instalación

#### Configuración

La aplicación ha sido desarrollada para dispositivos Android 4.1 (Jelly Bean) o superiores. Antes de iniciar la aplicación es necesario realizar unos pequeños cambios en la configuración.

1. Modificar la dirección ip predeterminada del archivo de configuración de Moodle. Para ello tendremos que:
  - a) Situarnos dentro de la propia carpeta de la herramienta:  
D:\xampp\htdocs\moodle

- b) Una vez situados en dicha ruta tendremos que modificar la dirección del host: *localhost* por la dirección ip de la propia máquina del archivo config.php como podemos ver en la siguiente ilustración.



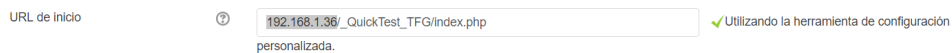
```

1  <?php // Moodle configuration file
2
3  unset($CFG);
4  global $CFG;
5  $CFG = new stdClass();
6
7  $CFG->dbtype      = 'mysqli';
8  $CFG->dblibrary    = 'native';
9  $CFG->dbhost      = 'localhost';
10 $CFG->dbname       = 'moodle';
11 $CFG->dbuser       = 'root';
12 $CFG->dbpass       = '';
13 $CFG->prefix       = 'mdl_';
14 $CFG->dboptions     = array (
15     'dbpersist' => 0,
16     'dbport' => '',
17     'dbsocket' => '',
18     'dbcollation' => 'utf8_unicode_ci',
19 );
20
21 $CFG->wwwroot      = 'http://192.168.1.36/moodle';
22 $CFG->dataroot      = 'D:\\xampp\\moodledata';
23 $CFG->admin         = 'admin';
24
25 $CFG->directorypermissions = 0777;
26
27 require_once(__DIR__ . '/lib/setup.php');
28
29 // There is no php closing tag in this file,
30 // it is intentional because it prevents trailing whitespace problems!

```

Figura E.1: Archivo config.php de Moodle.

2. Al haber modificado el host de Moodle tendremos que modificar la dirección de inicio de todas nuestras herramientas externas QuickTest por la dirección ip de la máquina. Esto es porque tenemos QuickTest y Moodle en la misma máquina, sino no sería necesario.



URL de Inicio  Utilizando la herramienta de configuración personalizada.

Figura E.2: Modificando la URL de inicio de las herramientas externas.

3. A continuación, tendremos que abrir nuestro proyecto en Android Studio y modificar los siguientes archivos dentro del paquete api.

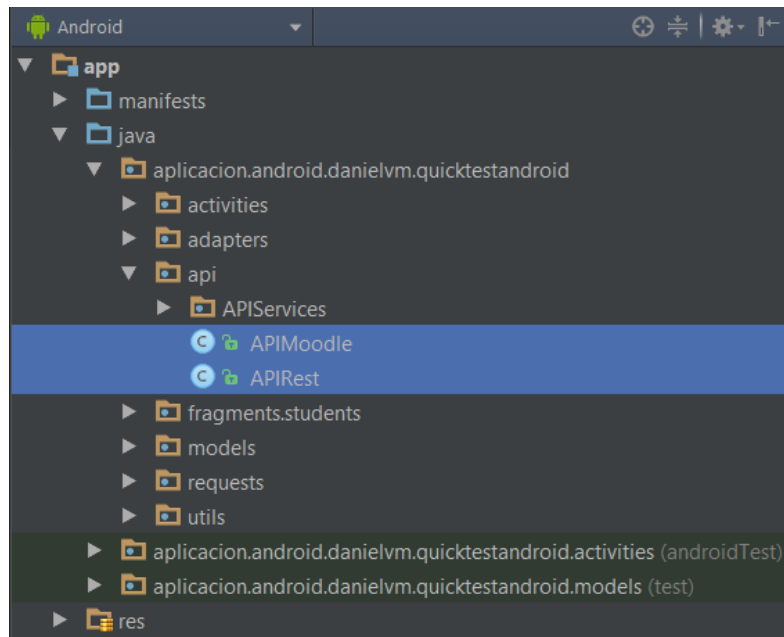


Figura E.3: Archivos a modificar dentro del paquete api.

- a) *APIMoodle*: tendremos que modificar la variable *BASE* por la dirección de la máquina y las variables *USERNAME*, *PASSWORD* por el nombre y contraseña del usuario encargado de manejar el web services de Moodle, ya que el es el único con privilegios para realizar las peticiones de información.

```
public class APIMoodle {

    public static final String BASE = "http://192.168.1.36/";
    public static final String BASE_URL = BASE + "moodle/";
    public static final String APP = "moodle_mobile_app";

    // Funciones del web service
    public static final String GET_EXTERNAL_TOOL = "mod_lti_get_tool_launch_data";
    public static final String FORMAT_JSON = "json";
    public static final String GET_COURSES = "core_course_get_courses";
    public static final String GET_CONTENT_COURSE = "core_course_get_contents";
    public static final String GET_USER_BY_FIELD = "core_user_get_users_by_field";
    public static final String GET_COURSES_BY_USER_ID = "core_enrol_get_users_courses";
    public static final String GET_PROFILES_USER_BY_USER = "core_user_get_course_user_profiles";
    public static final String GET_USERS_ENROLLED_IN_COURSE = "core_enrol_get_enrolled_users";

    // Campos del usuario que maneja el web servie
    public static final String USERNAME = "admin";
    public static final String PASSWORD = "Aadfl234!";
}
```

Figura E.4: Modificamos las variables de la clase APIMoodle

- b) *APIRest*: tendremos que modificar la variable *BASE* por la dirección de la máquina.

```
public class APIRest {  
  
    public static final String BASE = "http://192.168.1.36/";  
    public static final String BASE_URL = BASE + "_QuickTest_TFG/app/apiRest/";  
}
```

Figura E.5: Modificamos la variable de la clase APIRest.

Una vez realizada esta breve modificación en la configuración, la aplicación Android se encuentra completamente operativa para su ejecución desde el propio emulador de Android Studio o desde nuestro propio dispositivo Android.

### Instalación en un dispositivo Android físico

Para llevar cabo la instalación de la aplicación en nuestro propio dispositivo Android es necesario:

- Haber realizado la breve configuración previa del punto anterior. [E.3](#).
- Conectar nuestro dispositivo al ordenador.
- Nuestro dispositivo pertenezca a la misma red que nuestro equipo.

Una vez realizado estos pasos tendremos que:

1. Arrancar el proyecto en Android Studio.
2. Ejecutar la aplicación y seleccionar como dispositivo de destino, nuestro móvil.

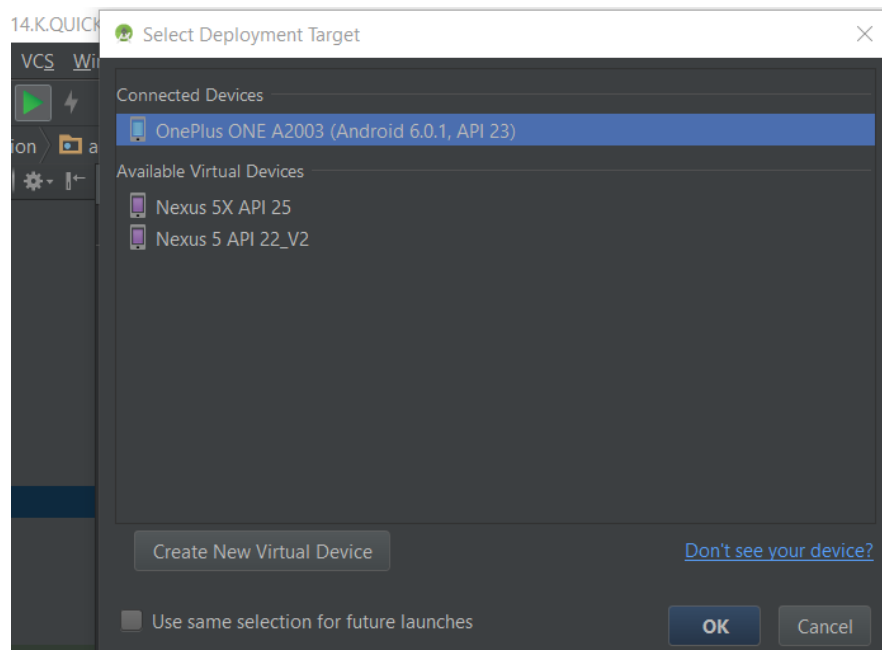


Figura E.6: Instalación de la aplicación en un dispositivo físico.

Una vez realizados estos pasos podremos utilizar nuestra aplicación desde nuestro propio dispositivo Android.

## E.4. Manual del usuario

En esta sección se describen el funcionamiento de las funcionalidades desarrolladas de la aplicación.

### Iniciar Sesión

Al iniciar la aplicación por primera vez nos aparecerá la ventana de *inicio de sesión*. Una vez allí tendremos que:

1. Introducir nuestro nombre de usuario y contraseña de Moodle.
2. Si se quiere recordar estas credenciales para que en futuros acceso esta ventana no se active, bastará con activar la opción *Recordar campos*.
3. Click en *Iniciar Sesión*.

Una vez iniciada sesión la aplicación nos mostrará los cursos en los que el usuario se encuentra matriculado independientemente de su rol en los mismos.

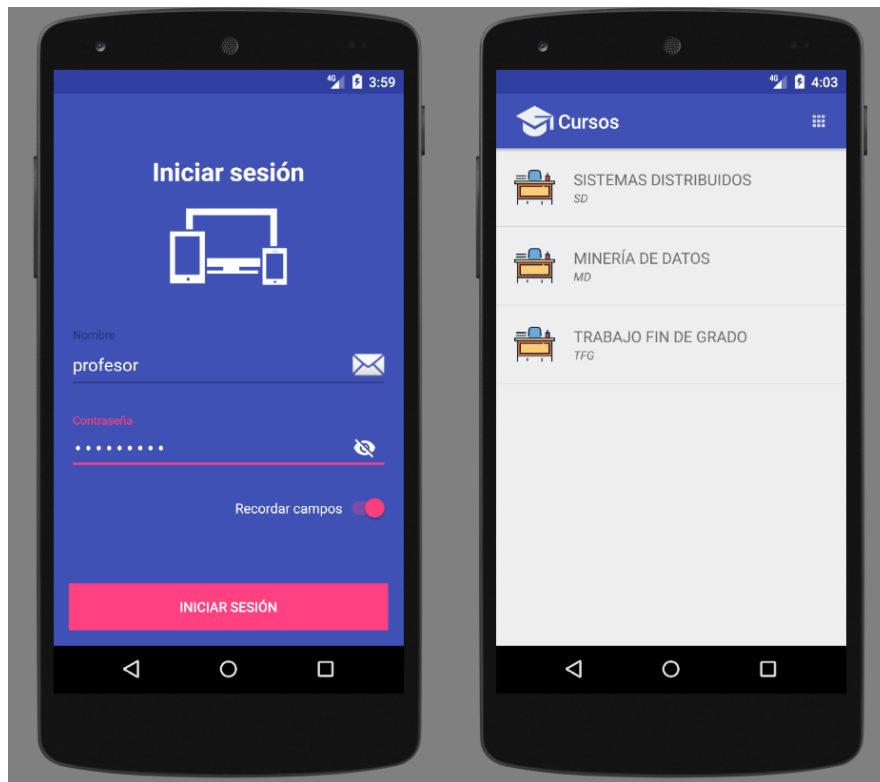


Figura E.7: Iniciar sesión en la aplicación.

### Obtener la calificación de los alumnos

Para poder comprobar la calificación de los alumnos de un profesor simplemente tendremos que:

1. Entrar en el curso.
2. Seleccionar el cuestionario a comprobar.
3. Comprobar la calificación de cada alumno.

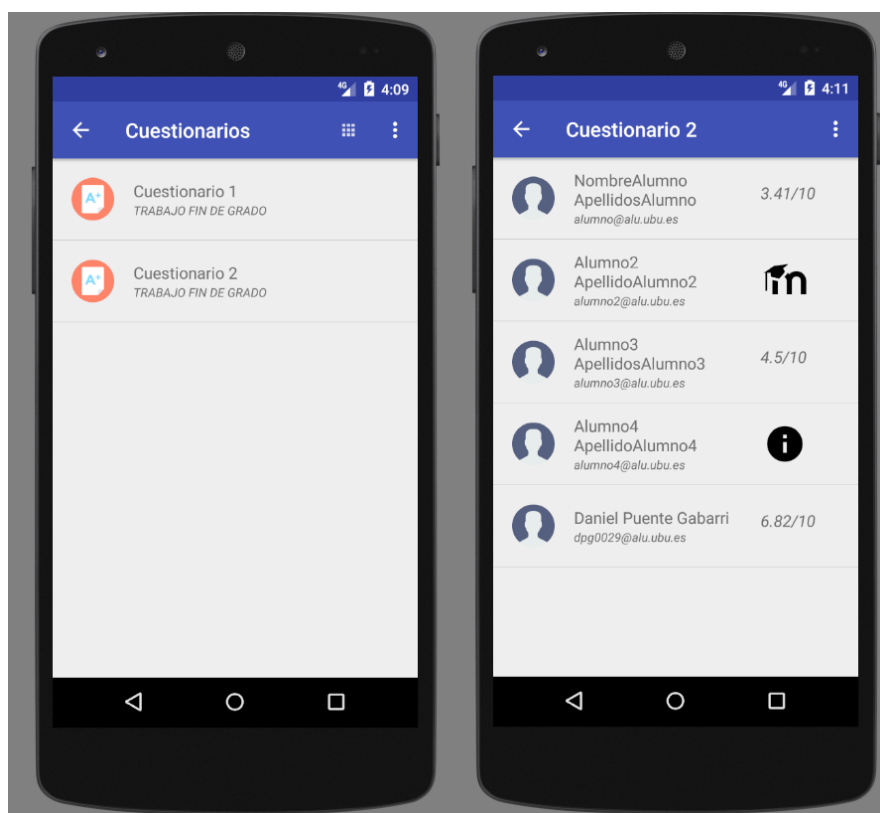


Figura E.8: Obtener la calificación de los alumnos matriculados en un curso.

Como podemos observar en la anterior ilustración, se utiliza el siguiente criterio para determinar que cuestionarios han sido resueltos de la aplicación web, desde la aplicación Android o aun no han sido resueltos.

- *Icono de Moodle*: cuestionario resuelto desde la aplicación web.
- *Icono de información*: cuestionario sin resolver.
- *Calificación*: cuestionario resuelto desde la aplicación Android.

### Mostrar cuestionario

Para obtener el cuestionario ya completado por los alumnos tendremos que:

1. Entrar en el curso.
2. Seleccionar el cuestionario a comprobar.



3. Seleccionar, *Ver cuestionario*.

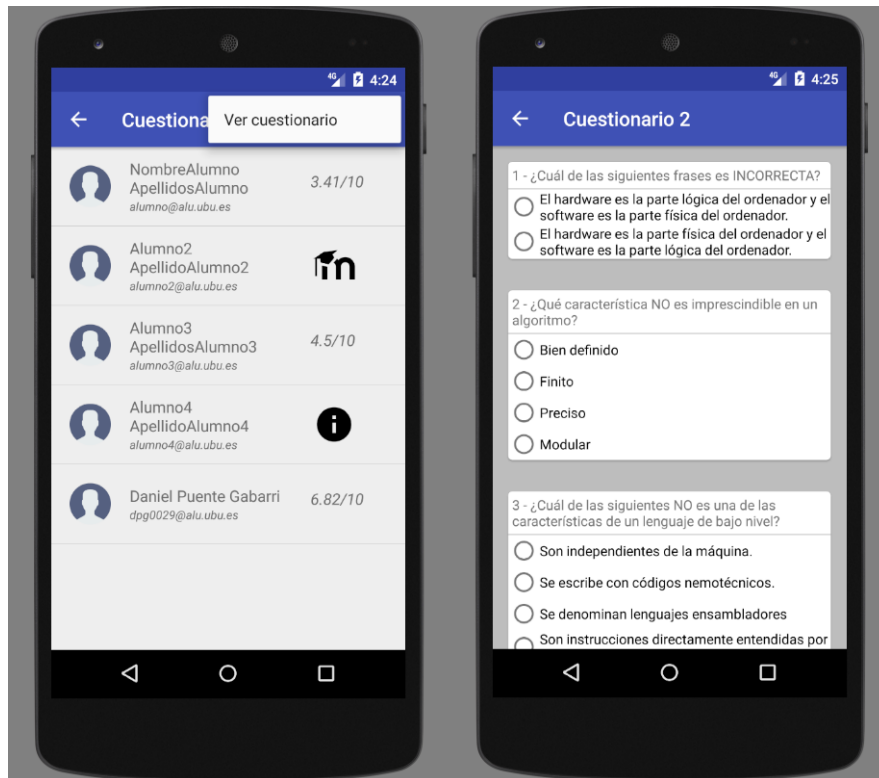


Figura E.9: Obtenemos el cuestionario al que se han enfrentado los alumnos.

## Resolver un cuestionario

Para resolver un cuestionario tendremos que:

1. Entrar en el curso.
2. Posicionarnos en *Mis cuestionarios sin resolver* en el *Burger Menu*.
3. Click largo sobre el cuestionario a resolver y seleccionamos *Resolver*.
4. Una vez resueltas las preguntas, click en *Enviar y finalizar cuestionario* y en la ventana emergente seleccionar *Enviar*.

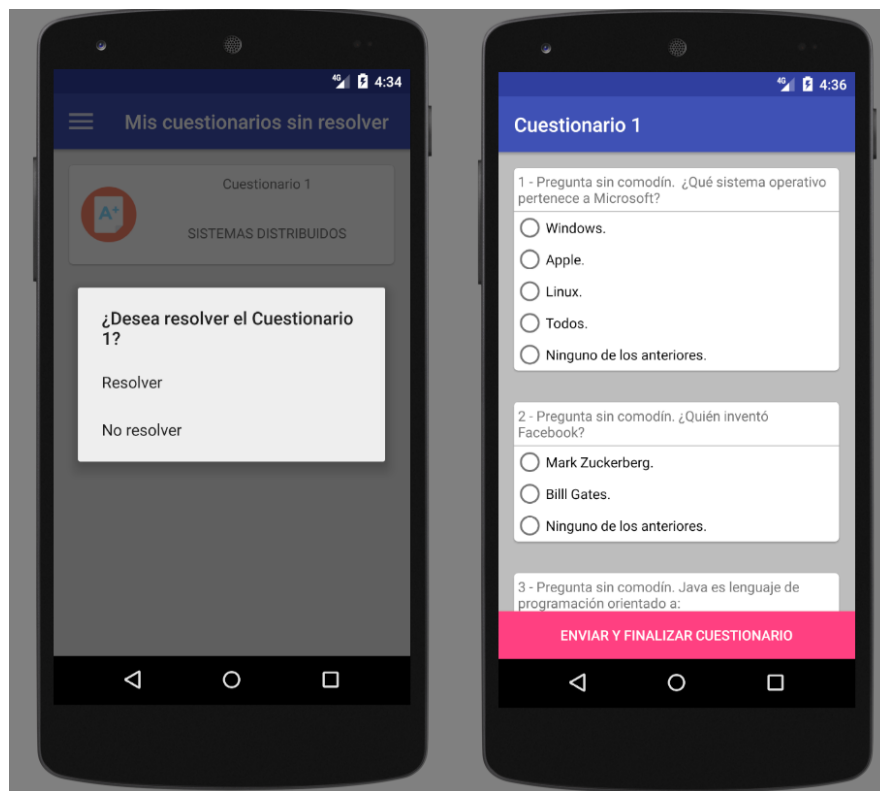


Figura E.10: Resolvemos un cuestionario.

### Revisar un cuestionario

Para revisar un cuestionario resuelto tendremos que:

1. Entrar en el curso.
2. Posicionarnos en *Mis cuestionarios resueltos* en el *Burger Menu*.
3. Click largo sobre el cuestionario a revisar y seleccionamos *Revisar*.
4. Si deseamos obtener información adicional sobre como ha sido resuelto el cuestionario, hacer click en *Ver resultados*

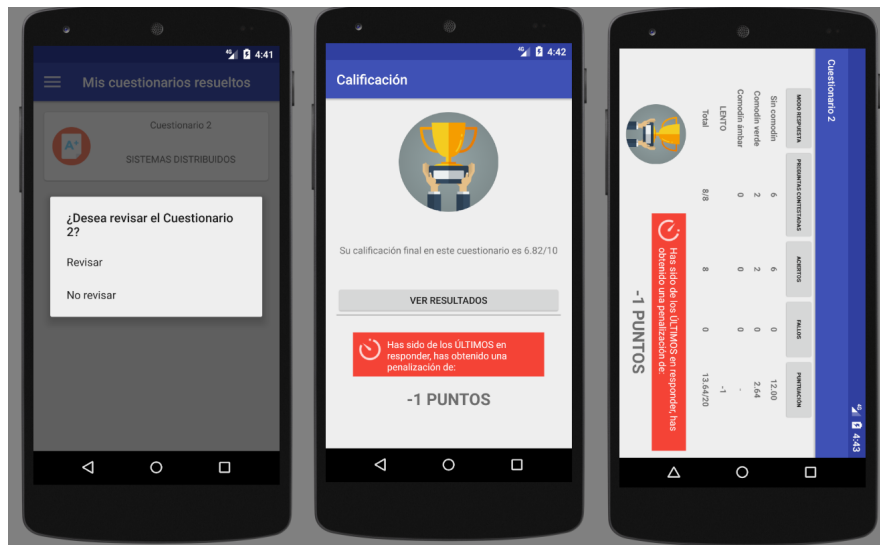


Figura E.11: Revisar un cuestionario.

## Cerrar sesión

Para cerrar sesión u olvidar y cerrar sesión tendremos que:

- *Rol alumno:*

1. Entrar en el curso.
2. Abrir en *Burger Menu* y seleccionar *Cerrar sesión u Olvidar y cerrar sesión*.

- *Rol Profesor:*

1. Entrar en el curso.
2. Abrir el botón de ajustes y seleccionar *Cerrar sesión u Olvidar y cerrar sesión*.

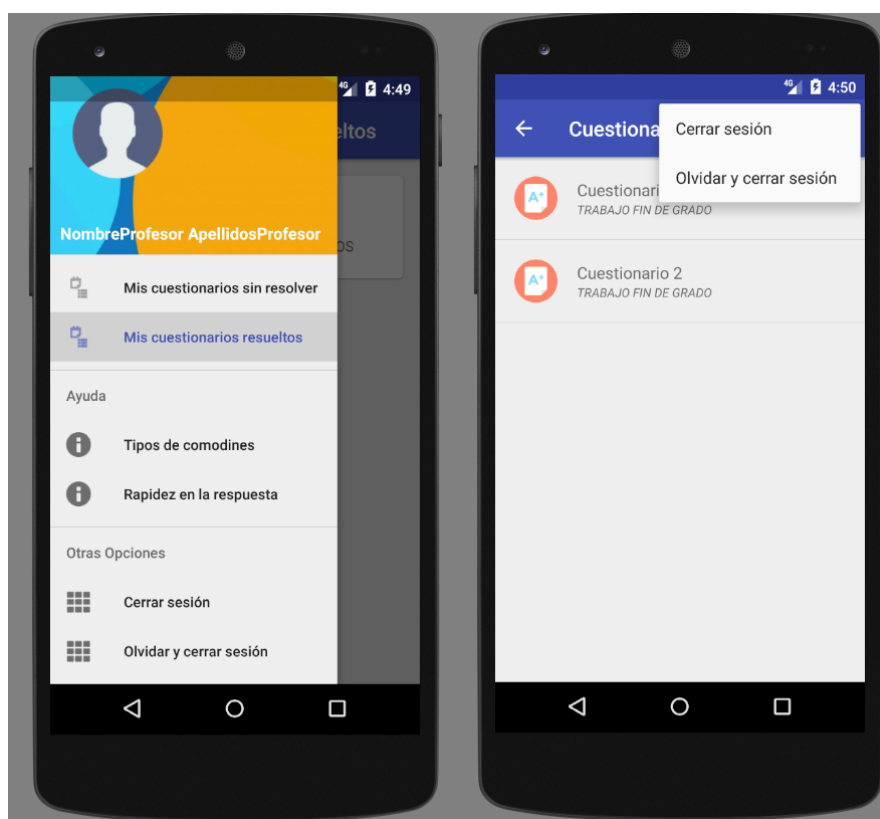


Figura E.12: Cerrar sesión desde ambos roles.

---

## Bibliografía

---

- [1] PHP. Php — referencia del lenguaje. URL <http://uk1.php.net/manual/es/langref.php>. [Internet; descargado 29-marzo-2015].
- [2] CakePhp. Cakephp — cakephp cookbook. URL <https://cakephp.org/>. [Internet; descargado 29-marzo-2015].
- [3] Laravel. Laravel — love beautiful code? we do too. URL <https://laravel.com/>. [Internet; descargado 29-marzo-2015].
- [4] Symphony. Symphony — the platform that powers work. URL <https://symphony.com/>. [Internet; descargado 29-marzo-2015].
- [5] Slim. Slim — slim is a php micro framework. URL <https://www.slimframework.com/>. [Internet; descargado 29-marzo-2015].
- [6] Wikipedia. Android studio — wikipedia, la enciclopedia libre, 2017. URL [https://es.wikipedia.org/w/index.php?title=Android\\_Studio&oldid=99905889](https://es.wikipedia.org/w/index.php?title=Android_Studio&oldid=99905889). [Internet; descargado 21-junio-2017 ].
- [7] Wikipedia. Xampp — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=XAMPP&oldid=99919164>. [Internet; descargado 21-junio-2017 ].
- [8] TortoiseGit. Tortoisegit — the power of git – in a windows shell, 2017. URL <https://tortoisegit.org/>. [Online; accessed 27-June-2017 ].
- [9] Wikipedia. Moodle — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=Moodle&oldid=99855535>. [Internet; descargado 21-junio-2017 ].
- [10] Google. Android — android. guía de desarrollo de aplicaciones para smartphones y tabletas - 2ª edición, . URL <https://www.amazon>.

- [es/Android-Desarrollo-Aplicaciones-Smartphones-Tabletas/dp/2746092298](#). [Internet; descargado 2-marzo-2017].
- [11] Lavanguardia. Moodle — el australiano que cambió la forma de aprender en todo el mundo. URL <http://www.lavanguardia.com/vida/20160401/40813172673/martin-dougiamas-moodle-universitat-vict-doctor-honoris-causa-agora-barcelona-t.html>. [Internet; descargado 2-marzo-2017].
- [12] Moodle. Moodle — moodle statistics. URL <https://moodle.net/stats/>. [Internet; descargado 2-marzo-2017].
- [13] Wikijuegos. Licencia comercial — definición. URL [http://es.videojuegos.wikia.com/wiki/Licencia\\_comercial](http://es.videojuegos.wikia.com/wiki/Licencia_comercial). [Internet; descargado 2-marzo-2017].
- [14] Google. Licencia comercial — app licensing, . URL <https://developer.android.com/google/play/licensing/index.html>. [Internet; descargado 2-marzo-2017].