



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Aplicación para el Acceso a
Plataformas de ELearning desde
Dispositivos Móviles**



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



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



D^a. María Belén Vaquerizo García y D. Bruno Baruque Zanón, profesores del departamento de Ingeniería Civil, área de Lenguajes y Sistemas informáticos.

Expone:

Que el alumno D. Daniel Puente Gabarri, con DNI 71347273-P, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Aplicación para el Acceso a Plataformas de ELearning desde Dispositivos Móviles.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 28 de junio de 2017

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

D. Bruno Baruque Zanón

D^a. María Belén Vaquerizo García

Resumen

No es ninguna locura afirmar que en la actualidad somos dependientes de las TIC, de nuestros dispositivos móviles, de Internet. Como tampoco que los humanos somos competitivos por naturaleza o que vivimos continuamente aprendiendo. Entonces, ¿Por que no agrupar estas necesidades?.

QuickTest, permite que los usuarios puedan enfrentarse a un mismo cuestionario utilizando su propio dispositivo móvil u ordenador en el aula al finalizar una clase de manera competitiva, gracias a la utilización de comodines o premios en función del orden de respuesta del alumno.

Descriptores

PHP, Moodle, Web service, APIREST, aplicación Android, XAMPP, Gamificación, Material Design.

Abstract

It is no foolish to say that we are currently dependent on ICT, our mobile devices, the Internet. Neither is it that humans are competitive by nature or that we live continuously learning. So, why not group these needs?

QuickTest allows users to face the same questionnaire using their own mobile device or computer in the classroom at the end of a class in a competitive way, thanks to the use of wildcards or prizes depending on the order of response of the student.

Keywords

PHP, Moodle, Web service, APIREST, Android application, XAMPP, Gamification, Material Design.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Contenido del proyecto	1
Objetivos del proyecto	3
2.1. Objetivos funcionales	3
2.2. Objetivos de carácter técnico	4
2.3. Objetivos personales	5
Conceptos teóricos	6
3.1. Web API	6
3.2. LTI	7
3.3. REST	8
Técnicas y herramientas	10
4.1. Lenguajes	10
4.2. Herramientas	11
4.3. Técnicas	14
Aspectos relevantes del desarrollo del proyecto	16
5.1. Inicio del proyecto	16
5.2. Metodología	16
5.3. Formación	17
5.4. Desarrollo del API Rest	18
5.5. Comunicación con el Web Service de Moodle	18

5.6. Desarrollo de la app	19
5.7. Pruebas software	19
Trabajos relacionados	20
6.1. Proyecto de partida	20
6.2. Cuestionarios de Moodle	20
6.3. Big Web Quiz	21
Conclusiones y Líneas de trabajo futuras	24
7.1. Conclusiones	24
7.2. Lineas de trabajo futuras	25
Bibliografía	26

Índice de figuras

6.1. Big Web Quiz en el Play Store.	22
---------------------------------------------	----

Índice de tablas

Introducción

Este trabajo de final de grado nace de la base de un proyecto anterior, debido a la necesidad de ampliar la funcionalidad de dicho proyecto para poder ser empleado también desde una aplicación móvil desarrollada en Android, que permita llevar a cabo las mismas funcionalidades y acceder a los mismos contenidos a disposición del alumno por medio del sistema E-Learning de la Universidad de Burgos.

1.1. Contenido del proyecto

Memoria

- **Introducción:** descripción breve sobre el proyecto a desarrollar.
- **Objetivos del proyecto:** principales objetivos del proyecto que posteriormente derivarán en las historias de usuario.
- **Conceptos teóricos:** principales conceptos teóricos necesarios para la comprensión del proyecto y del producto creado.
- **Aspectos relevantes del desarrollo del proyecto:** principales aspectos del proyecto.
- **Trabajos relacionados:** principales proyectos relacionados con producto creado.
- **Conclusiones y Líneas de trabajo futuras:** resumen sobre los conocimientos adquiridos y aspectos de mejora posibles en un futuro.

Anexos

- **Plan de Proyecto Software:** explicación sobre la planificación seguida durante el desarrollo del proyecto, junto con un estudio de viabilidad legal.
- **Especificación de Requisitos:** explicación de las historias de usuario, casos de uso, etc.
- **Especificación de diseño:** explicación y descripción del diseño de la aplicación.
- **Documentación técnica de programación:** explicación de los requisitos mínimos y configuración para el funcionamiento del proyecto.
- **Documentación de usuario:** explicación sobre el funcionamiento de la aplicación.

Proyecto

El proyecto se encuentra disponible en el siguiente enlace:

<https://github.com/danielpuente-dpg/GII14.K.QUICKTEST>

Objetivos del proyecto

A continuación, se va a llevar a cabo una citación y argumentación de los distintos objetivos a realizar en este proyecto con el objetivo de transmitir la finalidad del mismo.

2.1. Objetivos funcionales

Antes de comenzar con la explicación de los objetivos del proyecto tendremos que mencionar la principales funcionalidades del proyecto de partida, QuickTest.

- Generar un aplicación web que permita a los docentes: *crear, editar, eliminar, duplicar o publicar un cuestionario Moodle*.
- Generar una aplicación web que permita a los alumnos: *resolver un cuestionario publicado en Moodle, responder a las preguntas del cuestionario utilizando el sistema de comodines y conocer la calificación obtenida*.
- La aplicación web sea integrable en Moodle.

Partiendo del proyecto de partida identificamos los siguientes nuevos objetivos.

Docentes

- Estos cuestionarios permitirán a los docentes poder llevar a cabo una evaluación de los conocimientos adquiridos por su alumnado.
- Los profesores podrán ver la calificación de sus alumnos desde su dispositivo.

- Los docentes podrán ver el cuestionario al que se enfrentan sus alumnos.
- Todas estas acciones podrán realizarse desde el propio dispositivo del profesor.

Alumnos

- Podrán resolver los distintos cuestionarios a los que tenga que enfrentarse desde su dispositivo.
- Podrán ver la calificación obtenida y revisar como han resuelto el cuestionario.
- Las calificaciones de los alumnos variarán en función de las distintas recompensas:
 - Los alumnos serán evaluados en función de su orden de respuesta al cuestionario.
 - Los alumnos podrán utilizar comodines si lo consideran necesarios y su calificación se vera afectada en caso de uso.

2.2. Objetivos de carácter técnico

- Desarrollar una aplicación Android para la resolución de estos cuestionarios que sea compatible con el mayor numero de dispositivos.
- Desarrollar un API que se capaz de proporcionar una interfaz común para la comunicación del proyecto de partida, es decir, *"la aplicación web"* con la aplicación a implementar.
- Utilizar un sistema de control de versiones como *Git* que permita registrar los cambios realizados durante el desarrollo del proyecto y la utilización de una plataforma que permita alojar este tipo de proyectos como *GitHub*.
- Desarrollar una aplicación Android que cumpla el diseño Material Design.
- Utilizar el sistema de compilación Gradle en Android para: *automatizar, administrar el proceso de compilación del proyecto e incluir en el mismo, ciertas configuraciones de compilación personales.*
- Realizar el proyecto mediante la utilización de la metodología ágil: *Scrum*.
- Utilización de los servicios externos de Moodle para obtener la información necesaria por la aplicación.

2.3. Objetivos personales

- Enfrentarme a nuevos retos que pongan a prueba todos mis conocimientos adquiridos a lo largo del grado.
- Enfrentarme a un posible proyecto o reto que se asemejen a los del mundo laboral.
- Aprender a desarrollar aplicaciones Android.
- Construir y comprender un API.
- Enfrentarme a un proyecto utilizando una metodología ágil.
- Mejorar el proyecto de partida, proporcionando una nueva forma de enfrentarse a los cuestionarios.

Conceptos teóricos

En esta sección se va proceder a la explicación de ciertos conceptos teóricos necesarios para la correcta comprensión de este trabajo. Además, dichos conceptos han sido necesarios para llevar a cabo la toma de decisiones sobre cómo resolver el trabajo junto con la realización del mismo.

3.1. Web API

Antes de hablar de que es un Web API tendremos que explicar que es un API, ya que la funcionalidad de un Web API es similar a la de un API, pero orientada a la Web.

API

Una posible definición podría ser la siguiente: *Un API (siglas de Application Programming Interface) es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.* [1].

Es decir, permite la comunicación entre distintos componentes software. La mayor ventaja es que permite reutilizar métodos escritos en un determinado lenguaje o software, de esta manera evitamos la existencia de duplicidad de una misma funcionalidad en los diferentes componentes. Además, estas funcionalidades se encuentran testeadas y funcionan de forma adecuada en un determinado componente software.

Web API

En este caso y relacionado con lo anteriormente explicado, la lógica de un Web API es la misma que un API salvo que en este caso esta comunicación, es decir, el intercambio de información se realiza entre un servicio web y una aplicación mediante una URL. Para llevar a cabo esta comunicación, se utilizan

peticiones HTTP o HTTPS y toda esta información se encuentra encapsulada generalmente en XML o JSON. Existen principalmente cuatro tipos de Web API.

SOAP

Es un protocolo estándar de intercambio de información y datos en XML entre dos objetos, cuyas siglas son: *Simple Object Access Protocol*. [2] [3]

XML-RPC

Es un protocolo que llama a un procedimiento remoto, que utiliza XML para encapsular los datos y llamadas HTTP para llevar a cabo la comunicación.

JSON-RPC

Es un protocolo cuya lógica es igual que el protocolo explicado anteriormente, salvo que en este caso utiliza el formato JSON para encapsular los datos.

REST

Es una arquitectura software para sistemas hipermedia en la World Wide Web. Además, esta arquitectura utiliza el protocolo HTTP para llevar a cabo la comunicación. No obstante, en esta misma sección se dedicará un apartado donde se profundizará más en este protocolo [4]. 3.3

3.2. LTI

Learning Tools Interoperability es un estándar desarrollado por *IMS Global Learning Consortium* que permite utilizar un framework o marco para la integración de aplicaciones educativas con un entorno de aprendizaje. [5] Un ejemplo podría ser el siguiente:

- **Entorno de aprendizaje** Moodle.
- **Marco:** LTI.
- **Aplicación educativa:** Proyecto de partida.

Términos

A continuación, se enumeran y describen los términos involucrados en el funcionamiento.

- **Consumer:** este será el consumidor de aplicaciones y enviará la información necesaria al proveedor para realizar la comunicación entre ambas. Un ejemplo de consumer sería: *Moodle o UbuVirtual*.
- **Provider:** este será el encargado de proporcionar la aplicación al consumidor. Para ello, será necesario establecer un canal de comunicación para que el consumidor le proporcione cierta información vital para su correcto funcionamiento. Un ejemplo de provider sería: el proyecto de partida, QuickTest.
- **Learning Management System:** es un software alojado en el servidor utilizado para realizar toda las tareas de gestión de las actividades de formación no presencial. Un ejemplo sería: *Moodle o UbuVirtual*. [6].
- **Consumer Key:** clave generada por el proveedor, la cual permite al consumidor identificar al proveedor autorizado cuando este realice peticiones al consumidor.
- **Shared secret:** clave o contraseña común al proveedor y consumidor.
- **Context:** conjunto de información proporcionada por el consumidor, al realizar una petición al proveedor. Un ejemplo sería: *Click en una herramienta externa en Moodle*.
- **Resource link:** enlace asociado a un determinado recurso en el proveedor, al que el sistema nos redirecciona al seleccionar una herramienta externa en el consumidor.

3.3. REST

Como ya hemos explicado anteriormente, *es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web* [7], cuyas siglas son las siguientes **REpresentational State Transfer** o en castellano, Transferencia de Estado Representacional.

En la actualidad, el termino REST se utiliza para aquellas interfaces que utilicen HTTP para el intercambio de información entre sistemas, pudiendo utilizar cualquier formato para encapsular los datos, aunque generalmente los más utilizados son XML y JSON. Además, actualmente, los sistemas que siguen las pautas o principios REST también se les suelen denominar RESTful. Para que un sistema se considere RESTful debe de cumplir con las siguientes pautas o principios. [4]

- **Protocolo cliente/servidor sin estado:** de manera que cada petición HTTP debe contener toda aquella información necesaria para poder ejecutarse. De esta forma, evitamos que tanto el cliente como el servidor

tengan que almacenar información sobre el estado previo de la misma para poder llevarla a cabo.

- **Operaciones:** un sistema REST debe poder llevar cabo las siguientes operaciones: *POST* (crear), *GET* (leer o consultar), *PUT* (editar) y *DELETE* (borrar). Estas operaciones, como podemos observar, se asemejan en gran parte a las operaciones CRUD en bases de datos.
- **Sistema de capas:** el sistema deberá de utilizar una arquitectura jerárquica entre los distintos componentes que la formen, de esta manera, garantizamos que cada una de estas capas se encargue de llevar a cabo una única funcionalidad.
- **Manipulación de recursos:** para realizar la manipulación de los objetos se lleva a cabo mediante la URI. Dicha URI se utiliza como el identificador único para cada recurso, un recurso es un elemento de información. De esta manera se simplifica el acceso a la información para su posterior manipulación.
- **Uso de hipermedios:** para las transiciones entre los distintos estados de la aplicación y para la información de la misma.

Técnicas y herramientas

En esta sección se llevará a cabo una mención y breve explicación sobre el conjunto técnicas y herramientas utilizadas durante el desarrollo del proyecto.

4.1. Lenguajes

PHP

PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML [8] cuyas siglas es un acrónimo recursivo de PHP Hypertext Preprocessor. Este lenguaje se utiliza en el lado del servidor y está enfocado en el desarrollo web para el contenido dinámico [9].

Este lenguaje se ha decidido utilizar frente a otras alternativas en el lado del servidor, debido a que, el proyecto de partida se basa en este lenguaje en el lado del servidor. Además, cabe destacar que, desde mi punto de vista me parece una gran decisión al ser uno de los lenguajes más utilizados en el lado del servidor, existe una gran documentación al respecto y la curva de aprendizaje es menos costosa al asemejarse a otros lenguajes orientados a objetos.

JAVA

Java es un lenguaje de programación orientado a objetos, diseñado para disminuir el número de dependencias [10].

Además, es el lenguaje más utilizado para el desarrollo de aplicaciones Android y el lenguaje oficial recomendado. Por todo esto, se ha decidido utilizar este lenguaje para la creación de la aplicación para dispositivos Android.

JSON

JSON es un formato ligero de intercambio de datos cuyas siglas es un acrónimo de JavaScript Object Notation, es decir, Notación de Objetos de JavaScript [11].

JSON es un formato de texto independientemente del lenguaje y se le considera una gran alternativa frente al formato de intercambio de datos XML, al ser mucho más sencillo de parsear por un analizador sintáctico [12].

Se ha decidido utilizar para la captación y devolución de los diferentes datos al proyecto de partida y a la aplicación Android respectivamente. De esta forma, la comunicación de la aplicación con el proyecto se simplifica, gracias a la utilización de ciertas librerías que facilitan el tratamiento de los datos en Android, como es *Gson*.

4.2. Herramientas

XAMPP

XAMPP es un paquete de instalación independiente de la plataforma y de software libre. Dicho paquete contiene el sistema de gestión de bases de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script: PHP y Perl [13]. Su nombre proviene del acrónimo X, indicando que es **compatible para cualquier sistema operativo** y el resto de siglas AMPP, hace referencia a cada uno de los elementos que contiene el paquete, **Apache, MySQL, PHP, Perl** respectivamente.

Se ha decidido utilizar esta herramienta como servidor, al ser utilizada en el proyecto de partida. Además, cabe destacar, que me parece una gran decisión ya que el propio paquete contiene todo lo necesario para poder crear un servidor web.

GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando un sistema de control de versiones Git [14].

Se ha decidido optar por esta herramienta, al haber sido anteriormente utilizada, ser una de las más utilizadas para el desarrollo colaborativo y al existir el plugin ZenHub para la gestión desde el propio repositorio de las distintas tareas a realizar en un panel de trabajo.

Además, cabe destacar que el repositorio permite dos tipos de formato para el repositorio: público y privado. En nuestro caso se encuentra público. <https://github.com/danielpuente-dpg/GII14.K.QUICKTEST>

ZenHub

ZenHub es la herramienta utilizada para la gestión del proyecto. Esta herramienta, permite su integración en la gestión de proyectos en GitHub, proporciona un tablero kanban para gestión de las tareas y automatiza la creación de los gráficos burndown para cada iteración. Gracias a este plugin gratuito, se facilita la gestión del proyecto de manera ágil. [15]

TortoiseGit

TortoiseGit es un software libre utilizado para llevar a cabo el control de versiones del proyecto sobre GitHub [16]. Implementa una extensión del shell de Windows, por lo que el control de versiones no tiene por qué realizarse mediante línea de comandos, sino que se puede realizar mediante la interfaz gráfica que proporciona esta herramienta, la cual es muy fácil de utilizar.

Android

Para llevar a cabo el desarrollo se decidió utilizar el propio entorno de desarrollo oficial y recomendado por Google: *Android Studio*.

Android Studio, es un IDE que soporta la construcción basada en Gradle, incluye su propio emulador Android y refactorizaciones propias, entre otras características.

Moodle

Moodle es un software orientado a la educación, cuyo principal objetivo es el de facilitar un entorno de calidad para proporcionar cursos. [17]

Se ha decidido utilizar este software al ser utilizado por el proyecto de partida como consumidor del proyecto.

PhpStorm

PhpStorm es el entorno de desarrollo comercial para PHP, HTML y JavaScript de JetBrains IntelliJ IDEA utilizado para el desarrollo del API. [18]

Se ha decidido utilizar este IDE frente a otras alternativas como **Notepad++** [19] al ser un entorno propio para el desarrollo en PHP, poseer refactorizaciones más potentes, atajos de teclado propios de un entorno de desarrollo, entre otras.

PHPUnit

PHPUnit es un framework para el desarrollo de pruebas unitarias en PHP. [20]. Permite construir y ejecutar pruebas unitarias o de cobertura código PHP.

Se ha decidido utilizar este framework de pruebas al venir integrado en la propia distribución de XAMPP.

SonarQube

SonarQube es una plataforma libre que permite realizar una integración continua de nuestro desarrollo software. Informa sobre la duplicidad de código, errores en el no cumplimiento de estándares de configuración, errores en el diseño del software, pruebas unitarias, etc. [21]

Se ha decidido utilizar esta herramienta al ser propuesta por los tutores debido a la gran funcionalidad que aporta.

Advanced REST Client

Advanced REST Client software que permite realizar peticiones REST sobre nuestro propio servidor. [22]

Se ha decidido utilizar esta herramienta frente a otras alternativas como *Postman* [23], al ser enfocada únicamente para un servidor REST. Esta herramienta ha sido utilizada para depurar de manera manual cada una de las peticiones realizadas al APIREST.

StarUML

StarUML es un software de diseño UML que permite la construcción de diagramas de clases, secuencias, paquetes, entre otras. [24]

Se ha decidido utilizar esta herramienta frente a otras como *Astah** [25] al ser utilizado anteriormente.

Librerías

A continuación, se incluye una pequeña explicación sobre las librerías utilizadas para el desarrollo del proyecto.

Consumir peticiones

Durante el desarrollo de la aplicación se han barajado ambas librerías para el consumo de las peticiones al propio API y al webservice de Moodle. Finalmente, se ha decido optar por la librería Retrofit al ser más fácil de utilizar, más intuitiva a la hora de crear y configurar las distintas peticiones, más flexible y porque suele aportar mejores resultados que Volley.

- **Retrofit:** Retrofit proporciona un framework para poder interactuar con APIs y enviar peticiones HTTP de forma fiable desde aplicaciones

Android. Para ello proporciona un cliente HTTP encargado de interactuar con APIs y de gestionar las peticiones de manera transparente. Esta librería es utilizada para el tratamiento de las peticiones desde la aplicación Android. Además, permite consumir peticiones de manera síncrona y asíncrona. [26]

- **Volley:** Al igual que la librería anterior, Volley es una librería desarrollada por Google para enviar peticiones HTTP desde aplicaciones Android. [27] [28]
- **Gson:** Esta librería proporcionada por Google, permite trabajar con JSON a la hora de serializar y deserializar los objetos. Es utilizada para convertir las respuestas JSON de las APIs en objetos Java de manera muy sencilla, eliminando toda esta carga de conversión al programador. Además, también permite el paso contrario a la hora de enviar objetos a las APIs, al encargarse de transformar estos objetos en respuestas en formato JSON. [29]

JUnit

JUnit es un framework para realizar pruebas unitarias sobre aplicaciones Java. [30]

Se ha utilizado esta herramienta para realizar las pruebas unitarias sobre la aplicación Android.

Espresso

Espresso es el propio framework de Android para la realización de pruebas de interfaz de usuario de manera automatizada. [31]

Se ha decidido utilizar esta herramienta al ser la recomendada para realizar la automatización de pruebas de UI.

4.3. Técnicas

SCRUM

SCRUM es un modelo de referencia que define un conjunto de práctica y roles, que pueden utilizarse como punto de partida para llevar a cabo la definición de como se elaborará el proceso de desarrollo del proyecto [32]. Cabe destacar que este modelo se encuentra dentro de los marcos de las metodologías ágiles y que es de los más utilizados actualmente. Los roles principales son:

- **Scrum Master:** se encarga de gestionar los cambios y procura facilitar la aplicación de esta metodología. En este proyecto este rol lo han desempeñado los tutores.

- **Product Owner:** en este grupo se encontrarán el personal interno o externo que representa al cliente. Este grupo se encargará de que el trabajo se realice de forma acorde con las necesidades y peticiones del cliente. Este rol también ha sido desempeñado por los tutores.
- **Team:** representa al equipo de desarrollo encargado de ejecutar el desarrollo y de entregar el producto deseado. Este rol ha sido desempeñado por el propio alumno al ser el único integrante del equipo.
- **Team Lider:** es el responsable del funcionamiento de Scrum en el proyecto. Este rol también es desempeñado por el alumno.

Para realizar el desarrollo del producto se deben de definir un Product Backlog, el cual contendrá todas las historias de usuario a realizar. Dichas historias de usuario se asignarán a los distintos Sprints o iteraciones que se realicen a lo largo del desarrollo del producto hasta finalizar con todas las historias de usuario.

Aspectos relevantes del desarrollo del proyecto

En este apartado se recogen los aspectos mas relevantes que han surgido a lo largo del desarrollo del proyecto, es decir todos aquellos puntos de inflexión que han marcado los pasos a seguir o la toma de decisiones frente a diferentes problemáticas.

5.1. Inicio del proyecto

La selección de este trabajo final de grado, nace de la necesidad propia de la construcción de una aplicación en Android. Una vez publicados las distintas propuestas, busque aquellas que mas se acercarán a mis necesidades. Aquí, es donde este proyecto destacó sobre las demás.

Una vez realizada mi elección de proyecto, comunique a mis tutores mi interés en el mismo y que me explicarán en mayor profundidad las funcionalidades a desarrollar. Tras las explicaciones y aclaraciones pertinentes nos pusimos manos a la obra.

5.2. Metodología

Para la correcta gestión del proyecto se decidió utilizar una metodología ágil, en este caso Scrum [4.3](#). Destacar que no se siguió al pie de la letra, ya que el equipo de proyecto solamente estaba formado por 3 personas. No obstante, se siguieron las siguientes pautas:

- Desarrollo incremental del proyecto basado en iteraciones o sprints.
- La duración ideal de las distintas iteraciones fue de 15 días.

- Al finalizar cada iteración, se realiza una revisión del sprint anterior y al finalizar esta, se realiza la planificación de las nuevas tareas a entregar en la siguiente iteración.
- Para manipular estas tareas se utiliza un tablero Kanban. Este tablero se incluye en GitHub gracias a la utilización de ZenHub [4.2](#).
- Utilización de gráficos Burndown para monitorizar el progreso de proyecto.

5.3. Formación

La realización de este proyecto requería una serie de conocimientos que al comienzo del mismo no disponía. Para ello en las primeras iteraciones se realizaron estudios sobre estos conocimientos técnicos. Para ello se realizó:

- Para la formación en PHP:
 - Lectura de la documentación oficial que proporciona PHP. [\[33\]](#)
 - Curso de aprendizaje de PHP. [\[34\]](#)
 - Aprendizaje sobre el funcionamiento de PHPUnit. [\[35\]](#)
- Para la formación en la plataforma Moodle:
 - Lecturas varias, del funcionamiento de la misma.
 - Documentación del proyecto de partida.
 - Activación y uso del web service de Moodle.
- Para la formación en Android:
 - Programación de Android desde Cero + 35 horas (Udemy).
 - Android. Guía De Desarrollo De Aplicaciones Para Smartphones Y Tablet as [\[27\]](#) Libro proporcionado por mis tutores.
 - Pruebas unitarias en Android mediante JUnit.
 - Pruebas automatizadas de interfaz de usuario. [\[36\]](#)
 - Pruebas de estrés. [\[37\]](#)

5.4. Desarrollo del API Rest

Para poder comunicar la aplicación Android con las propias funcionalidades del proyecto de partida, se utiliza un API. Las principales funcionalidades de este API, han sido desarrolladas durante las primeras iteraciones del proyecto, no obstante otras han sido desarrolladas fruto de la necesidad de las mismas.

Se ha decidido utilizar una arquitectura Rest para este API, ya que:

- **Separación cliente/servidor:** de esta manera al ser independiente, la comunicación entre cliente y servidor se realiza mediante un lenguaje común de intercambio, en este caso JSON [4.1](#). Gracias a esta idea, podemos realizar la construcción de un API que funcione de adaptador entre las funcionalidades del proyecto de partida y la información que la propia aplicación Android necesita. Además, de esta manera no es necesario modificar ninguna funcionalidad del proyecto de partida.
- **Lenguajes independientes:** permite la comunicación de información entre distintos lenguajes. En nuestro caso, este API está desarrollado en PHP y el cliente que lo consume en Java.
- **Flexibilidad:** el cambio de alguno de estos nexos no implica la transformación de los mismos, siempre y cuando la devolución de los datos se realice de forma correcta.
- **Sin estado:** al no tener estado, no requiere un almacenamiento de este y por lo tanto permite un mayor número de peticiones.

Es por esto que la visión que podemos tener de este, es la del BackEnd de la app, es decir es el proveedor de los datos a mostrar en la aplicación.

5.5. Comunicación con el Web Service de Moodle

La aplicación Android nos permitirá resolver los mismos cuestionarios del proyecto de partida. Para ello, es necesario conocer cierta información que el propio API desarrollado no tiene. Es por esto, que la aplicación tendrá que comunicarse de alguna con Moodle para conocer estos datos, es aquí donde entra en juego los Web Service de Moodle.

Para llevar a cabo esta comunicación se han tenido que realizar previamente la instalación y configuración de estos. Una vez realizado esto, la aplicación Android podrá comunicarse con Moodle.

5.6. Desarrollo de la app

Una vez desarrolladas las principales funcionalidades del API en las primeras iteraciones, se comenzó con el desarrollo de la aplicación Android. La versión utilizada en el proyecto es Android 4.1 (Jelly Bean) al ser actualmente la que mayor soporte cubre, con un 95,2 % al inicio del desarrollo de la app.

La primera problemática que surgió fue la del tratamiento de las peticiones desde la app. Para ello y después de mucha investigación y pruebas se decidió utilizar la librería Retrofit de Square [26] entre otras alternativas como Volley [28] de Google, al permitir esta la utilización de peticiones síncronas y la librería Gson de Google [29] para facilitar la transformación de los datos en formato JSON proporcionados por el propio API o el Web Service de Moodle, a objetos en Java.

Durante las últimas fases de construcción de la aplicación, el proyecto se encontró frente a una gran problemática que no se ha podido solucionar: no se ha encontrado ninguna forma de realizar el almacenamiento de la calificación del alumno que esta resolviendo un cuestionario desde la aplicación Android en el apartado de calificaciones de Moodle, pero si en la propia base de datos de QuickTest.

Esta problemática esta causada al no existir bibliotecas en Android que permitan encapsular el envío de los datos a Moodle cumpliendo el estándar LTI.

Debido a esto y como solución intermedia, la calificación solamente se puede conocer desde la propia aplicación Android. De esta manera, los docentes podrán conocer la calificación de aquellos alumnos que hayan resuelto el cuestionario desde la aplicación Android e incluirla de manera manual en el apartado de calificaciones del propio alumno.

5.7. Pruebas software

A lo largo del desarrollo del proyecto se realizaron diferentes tipos de pruebas para comprobar el correcto funcionamiento de la aplicación.

- *API Rest*: una vez desarrolladas las funcionalidades necesarias se realizaron pruebas unitarias sobre las mismas.
- *Aplicación Android*: al finalizar el desarrollo de la misma se realizaron: pruebas unitarias sobre el modelo, pruebas automatizadas sobre la interfaz de usuario, pruebas de cobertura y pruebas de estrés.

Trabajos relacionados

Durante el desarrollo del proyecto se analizó el comportamiento y diseño de ciertas aplicaciones. No obstante, cabe destacar las siguientes.

6.1. Proyecto de partida

Como se ha comentado anteriormente, este proyecto nace por la necesidad de ampliar la funcionalidad de la resolución de este tipo de cuestionarios desde nuestro propio dispositivo Android, ya que presenta cierta problemática en el renderizado de la aplicación en un dispositivo móvil y al utilizar una aplicación nativa su funcionamiento sera más natural y fluido.

Al tratarse de una ampliación de este proyecto tanto las ventajas como inconvenientes que presenta el proyecto de partida serán los mismo que la aplicación desarrollada.

6.2. Cuestionarios de Moodle

El propio entorno educativo de Moodle incorpora sus propios cuestionarios.

Ventajas

A diferencia de los propios cuestionarios de Moodle la aplicación permite:

- **Dinamismo:** la herramienta construida permite que los alumnos se enfrenten a los cuestionarios de una manera más divertida, debido al sistema de recompensa empleado: *comodines y orden*.
- **Alumnos:** estos podrán autoevaluar su nivel de conocimiento de la asignatura.

- **Docentes:** estos podrán evaluar en tiempo real el conocimiento de la clase de lo que acaba de explicar. De esta manera, podrá aclarar ciertos conceptos lo antes posible.
- **Comodines y orden:** los alumnos se enfrentarán a estos cuestionarios no solamente con la motivación de superarlo, sino que, *si son los primeros obtendrán una recompensa, si son de los últimos en responder tendrán a su disposición un mayor número de comodines*, es decir *fomentará la competición entre los alumnos, ayudará a que los propios alumnos se relacionen fruto de la competición*, etc.
- **Móvil:** al permitir que los usuarios puedan resolver estos cuestionarios desde su dispositivo Android, estos cuestionarios podrán realizarse como una comprobación al finalizar cada clase de lo aprendido en la misma, al necesitar únicamente de sus teléfonos móviles.

Desventajas

A continuación, se citan algunas de las principales desventajas que presenta el producto creado frente a los cuestionarios nativos de Moodle.

- **Multirespuesta:** permite que en el número de respuestas de una pregunta sea mayor que uno.
- **Categorías:** permite el almacenamiento de las preguntas por categorías.
- **Reintentos:** el profesor puede configurar el número de intentos de un mismo cuestionario.
- **Tiempo:** permite incluir un tiempo en el que el cuestionario tiene que resolverse.
- **Imágenes:** permite incluir en la propia pregunta una imagen.
- **Aleatoriedad:** permite que en un mismo cuestionario el número de preguntas varíen entre alumnos o intentos.

6.3. Big Web Quiz

[38]

Durante el desarrollo del proyecto se realizó una búsqueda de alguna herramienta de resolución de cuestionarios en dispositivos Android como complemento de Moodle pero, no se encontró ninguna al respecto. No obstante, se ha elegido esta herramienta: *Big Web Quiz* al ser desarrollada por Google y al fomentar, al igual que la aplicación construida: *la competición entre los participantes*.

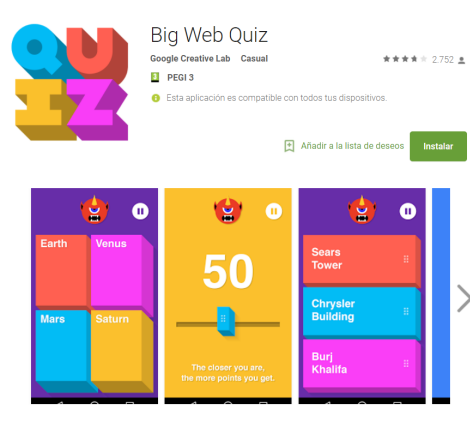


Figura 6.1: Big Web Quiz en el Play Store.

Ambas aplicaciones destacan por:

- Fomentar la competición.
- Proporcionan una forma más dinámica de enfrentarse a un cuestionario.
- Permiten resolver los cuestionarios desde un dispositivo móvil Android.
- Fomentar la gamificación durante la resolución de los cuestionarios, es decir aprender divirtiéndose.

Ventajas

- **Moodle:** la aplicación desarrollada es compatible con LTI y por tanto con Moodle. Sin embargo, Big Web Quiz no. [3.2](#).
- **Docentes:** la aplicación desarrollada permite a los docentes obtener la calificación de sus alumnos.
- Para su funcionamiento no es necesario un Chromecast.
- Big Web Quiz solamente permite enfrentarse a un cuestionario desde un dispositivo móvil.
- La aplicación desarrollada permite que los alumnos puedan obtener información sobre como han resuelto un cuestionario y por medio de los comodines, relacionar su conocimiento con el del resto de la clase.
- Permite a los docentes crear sus propios cuestionarios.

Desventajas

A diferencia de la aplicación desarrollada, Big Web Quiz proporciona:

- Permite varias formas de respuesta: multirespuesta, preguntas con imágenes, proporciona gráficos, entre otras.
- Permite la selección de avatares al iniciar un cuestionario.
- Fomenta la competición de manera agrupada en varios cuestionarios.

Conclusiones y Líneas de trabajo futuras

En esta sección se incluyen todas las conclusiones derivadas del desarrollo del proyecto y las posibles líneas de trabajo futuras.

7.1. Conclusiones

- Desde mi punto de vista, considero que los objetivos principales del proyecto han sido superados durante el desarrollo del mismo.
- Al haber tenido que basarse en un proyecto ya desarrollado, he tenido muchas ventajas al tener muchas funcionalidades necesarias ya implementadas, pero también he tenido el inconveniente de que al crear nuevas funcionalidades estas no alteren el funcionamiento inicial.
- La creación de un API sobre un proyecto web ha aportado no solamente nuevos conocimientos sobre: *PHP, lado del servidor, JavaScript, etc.* Sino, que me ha permitido comprender que prácticamente todas las aplicaciones Android de productos que anteriormente habían sido desarrollados para la web, han tenido que crear un API para manejar esta lógica.
- Se ha obtenido un nuevo conocimiento sobre la programación en Android, PHP o los web services que hasta la realización de este se desconocían.
- Se ha comprendido el verdadero potencial de la utilización de herramientas para: *el control de versiones, la utilización de una metodología ágil, utilización de sistemas de construcción basado en dependencias como Gradle, herramienta de integración continua como SonarQube, etc.*

- Se ha comprendido como puede ser la dinámica en el mundo laboral en un futuro, es decir la asignación de un proyecto, su gestión, nuevos conocimientos a adquirir, metodología de trabajo, etc.

7.2. Líneas de trabajo futuras

■ Proyecto de partida:

- Modificar la clase encargada de manejar la conexión a la base de datos implementando un patrón Singleton. De esta forma, limitaremos a que exista un único hilo para acceder a la misma.
- Modificar todas aquellas consultas a la base de datos que no se realicen mediante sentencias preparadas. De esta manera, evitaremos posibles problemas de seguridad y mejoraremos el mantenimiento.
- Mejorar la forma de crear un cuestionario, ya que el método encargado de esta lógica se ocupa de tareas: *crear y editar un cuestionario*.
- Incorporar un sistema que permita implementar la concurrencia.

■ Aplicación Android:

- Permitir que la calificación de los alumnos pueda ser almacenada en el apartado de calificaciones del curso, mediante alguna biblioteca en Android que permita el envío de datos a Moodle cumpliendo el estándar LTI.
- Posibilite al usuario una retroalimentación sobre las respuestas realizadas y la respuesta correcta.
- Permita la creación de cuestionarios multirespuesta o que las preguntas puedan adjuntar una imagen.
- Nuevas formas de enfrentarse a un cuestionario.
- Los docentes puedan obtener ciertos gráficos sobre: *las respuestas más contestadas, aquellas preguntas con derecho a comodín, entre otras*.
- Permita que los docentes puedan crear, modificar, duplicar o eliminar sus cuestionarios. Destacar que los métodos principales para desarrollar esta lógica, se encuentran implementados en el propio API.
- Realizar una internacionalización de la aplicación.
- La aplicación incluya soporte para personas con dificultades como: *problemas de vista o auditivas*. Algunos ejemplos podrían ser: *asistente de lectura de voz o paleta de colores propia para personas con problemas visuales*.

Bibliografía

- [1] Wikipedia. Interfaz de programación de aplicaciones — wikipedia, la enciclopedia libre, 2017. URL https://es.wikipedia.org/w/index.php?title=Interfaz_de_programaci%C3%B3n_de_aplicaciones&oldid=97439557. [Internet; descargado 27-junio-2017].
- [2] Wikipedia. Simple object access protocol — wikipedia, la enciclopedia libre, 2017. URL https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&oldid=99018786. [Internet; descargado 22-junio-2017].
- [3] DesarrolloWeb. Soap – soap. simple object access protocol. URL <https://desarrolloweb.com/articulos/1853.php>. [Internet; descargado 23-febrero-2017].
- [4] BBVAopen4u. Api rest – qué es y cuáles son sus ventajas en el desarrollo de proyectos. URL <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>. [Internet; descargado 23-febrero-2017].
- [5] IMS GLOBAL. Ims global — learning tools interoperability® background, 2016. URL <https://www.msglobal.org/activity/learning-tools-interoperability>. [Internet; descargado 27-junio-2017].
- [6] Wikipedia. Sistema de gestión de aprendizaje — wikipedia, la enciclopedia libre, 2017. URL https://es.wikipedia.org/w/index.php?title=Sistema_de_gesti%C3%B3n_de_aprendizaje&oldid=99344842. [Internet; descargado 22-junio-2017].
- [7] Wikipedia. Transferencia de estado representacional — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index>.

- [php?title=Transferencia_de_Estado_Representacional&oldid=99334337](#). [Internet; descargado 22-junio-2017].
- [8] PHP. Php – ¿qué es php?, . URL <http://php.net/manual/es/intro-what-is.php>. [Internet; descargado 2-marzo-2017].
- [9] Wikipedia. Php — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=PHP&oldid=99804274>. [Internet; descargado 22-junio-2017].
- [10] Wikipedia. Java (lenguaje de programación) — wikipedia, la enciclopedia libre, 2017. URL [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=99778339](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=99778339). [Internet; descargado 22-junio-2017].
- [11] JSON. Json – introducción a json. URL <http://www.json.org/json-es.html>. [Internet; descargado 2-marzo-2017].
- [12] Wikipedia. Json — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=JSON&oldid=99853092>. [Internet; descargado 22-junio-2017].
- [13] Wikipedia. Xampp — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=XAMPP&oldid=99919164>. [Internet; descargado 22-junio-2017].
- [14] Wikipedia. Github — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=GitHub&oldid=98817555>. [Internet; descargado 22-junio-2017].
- [15] ZenHub. Zenhub. URL <https://www.zenhub.com/product>. [Internet; descargado 21-junio-2017].
- [16] TortoiseGit. Tortoisegit — the power of git – in a windows shell, 2017. URL <https://tortoisegit.org/>. [Online; accessed 27-June-2017].
- [17] Wikipedia. Moodle — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=Moodle&oldid=99855535>. [Internet; descargado 22-junio-2017].
- [18] Wikipedia. Phpstorm — wikipedia, the free encyclopedia, 2017. URL <https://en.wikipedia.org/w/index.php?title=PhpStorm&oldid=784359638>. [Online; accessed 22-June-2017].
- [19] Wikipedia. Notepad++ — wikipedia, the free encyclopedia, 2017. URL <https://en.wikipedia.org/w/index.php?title=Notepad%2B%2B&oldid=786845183>. [Online; accessed 22-June-2017].

- [20] Wikipedia. Phpunit — wikipedia, la enciclopedia libre, 2016. URL <https://es.wikipedia.org/w/index.php?title=PHPUnit&oldid=89669740>. [Internet; descargado 22-junio-2017].
- [21] Wikipedia. Sonarqube — wikipedia, la enciclopedia libre, 2017. URL <https://es.wikipedia.org/w/index.php?title=SonarQube&oldid=98980020>. [Internet; descargado 22-junio-2017].
- [22] Google. Advanced rest client — chromerestclient, . URL <https://github.com/jarrodek/ChromeRestClient>. [Internet; descargado 21-junio-2017].
- [23] PostMan. Postman. URL <https://github.com/postmanlabs/postman-app-support>. [Internet; descargado 21-junio-2017].
- [24] Wikipedia. Staruml — wikipedia, the free encyclopedia, 2017. URL <https://en.wikipedia.org/w/index.php?title=StarUML&oldid=774075865>. [Online; accessed 22-June-2017].
- [25] Wikipedia. Astah* — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Astah*&oldid=749397298. [Online; accessed 22-June-2017].
- [26] SQUARE. Retrofit — a type-safe http client for android and java. URL <http://square.github.io/retrofit/>. [Internet; descargado 4-mayo-2017].
- [27] 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].
- [28] Google. Volley — volley is an http library, . URL <https://github.com/google/volley>. [Internet; descargado 4-mayo-2017].
- [29] Google. Gson — google-gson, . URL <https://github.com/google/gson>. [Internet; descargado 4-mayo-2017].
- [30] Wikipedia. Junit — wikipedia, la enciclopedia libre, 2015. URL <https://es.wikipedia.org/w/index.php?title=JUnit&oldid=79868227>. [Internet; descargado 22-junio-2017].
- [31] Google. Espresso — testing ui for a single app, . URL <https://developer.android.com/training/testing/ui-testing/espresso-testing.html>. [Internet; descargado 21-junio-2017].

- [32] DocPlayer. Docplayer — scrum manager. gestión de proyectos rev. 1.4, 2017. URL <http://docplayer.es/1449776-Scrum-manager-gestion-de-proyectos-rev-1-4.html>. [Internet; descargado 22-junio-2017].
- [33] PHP. Php — referencia del lenguaje, . URL <http://uk1.php.net/manual/es/langref.php>. [Internet; descargado 16-mayo-2017].
- [34] Codejobs. Php — curso: Aprende php desde cero hd. URL <https://www.youtube.com/watch?v=89f9vC4hZh8&t=787s>. [Internet; descargado 16-mayo-2017].
- [35] Symfony Madrid. Symfony madrid — breve introducción a phpunit, 2017. URL <https://www.youtube.com/watch?v=CZ15irnqpxE>. [Internet; descargado 22-junio-2017].
- [36] Android Developers. Android developers — android testing support - android testing patterns. URL https://www.youtube.com/watch?v=W8LJjfkTKik&list=PLWz5rJ2EKKc-6HWg_jyPOU1zrVLHn65b2. [Internet; descargado 22-junio-2017].
- [37] Android Studio. Android studio — ui/application exerciser monkeys. URL <https://developer.android.com/studio/test/monkey.html>. [Internet; descargado 22-junio-2017].
- [38] Google Creative Lab. Big web quiz — big web quiz brings trivia night to your tv! URL <https://play.google.com/store/apps/details?id=com.google.android.apps.bigwebquiz.app.prod&hl=es>. [Internet; descargado 22-junio-2017].