|  |  |  |
| --- | --- | --- |
|  | UNIVERSIDAD DE BURGOS  ESCUELA POLITÉCNICA SUPERIOR  Grado en Ingeniería en Informática |  |

MEMORIA

|  |  |
| --- | --- |
| **TFG del Grado en Ingeniería en Informática**  **Aplicación para el Acceso a Plataformas de ELearning desde Dispositivos Móviles** |  |

Presentado por Daniel Puente Gabarri en Universidad de Burgos — dd de mm de YYYY Tutor: María Belén Vaquerizo García y Bruno Baruque Zanón

|  |  |  |
| --- | --- | --- |
|  | UNIVERSIDAD DE BURGOS  ESCUELA POLITÉCNICA SUPERIOR  Grado en Ingeniería en Informática |  |

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone: Que el alumno D. Daniel Puente Gabarri, con DNI 71347273-P, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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, 10 de febrero de 2017

|  |  |
| --- | --- |
| Vº. Bº. del Tutor | Vº. Bº. del co-tutor |
| D. nombre tutor | D. nombre co-tutor |

**Agradecimientos**

**Resumen**

**Descriptores**

**Abstract**

**Keywords**

# Índice general

[MEMORIA 1](#_Toc475443087)

[Índice general 7](#_Toc475443088)

[Índice de ilustraciones 10](#_Toc475443089)

[Índice de tablas 11](#_Toc475443090)

[1. Introducción 12](#_Toc475443091)

[2. Objetivos del proyecto 12](#_Toc475443092)

[2.1. Objetivos funcionales 12](#_Toc475443093)

[2.2. Objetivos de carácter técnico 13](#_Toc475443094)

[2.3. Objetivos personales 13](#_Toc475443095)

[3. Conceptos teóricos 14](#_Toc475443096)

[3.1. Web API 14](#_Toc475443097)

[3.1.1. API 14](#_Toc475443098)

[3.1.2. Web API 14](#_Toc475443099)

[3.1.2.1. SOAP 14](#_Toc475443100)

[3.1.2.2. XML-RPC 14](#_Toc475443101)

[3.1.2.3. JSON-RPC 15](#_Toc475443102)

[3.1.2.4. REST 15](#_Toc475443103)

[3.2. Soap 15](#_Toc475443104)

[3.3. Rest 16](#_Toc475443105)

[4. Técnicas y herramientas 18](#_Toc475443106)

[4.1. Lenguajes 18](#_Toc475443107)

[4.1.1. PHP 18](#_Toc475443108)

[4.1.2. JSON 18](#_Toc475443109)

[4.1.3. Android 18](#_Toc475443110)

[4.2. Herramientas 18](#_Toc475443111)

[4.2.1. XAMPP 18](#_Toc475443112)

[4.2.2. GitHub 18](#_Toc475443113)

[4.2.3. TortoiseSVN 18](#_Toc475443114)

[4.2.4. Moodle 18](#_Toc475443115)

[4.2.5. PhpStorm 18](#_Toc475443116)

[4.2.6. phpMyAdmin 18](#_Toc475443117)

[4.2.7. PHPUnit 18](#_Toc475443118)

[4.2.8. SonarQube 18](#_Toc475443119)

[4.2.9. Modelio 18](#_Toc475443120)

[4.3. Técnicas 18](#_Toc475443121)

[4.3.1. LTI 18](#_Toc475443122)

[4.3.2. MVC 18](#_Toc475443123)

[4.3.3. SCRUM 18](#_Toc475443124)

[5. Trabajos relacionados 19](#_Toc475443125)

[6. Aspectos relevantes del desarrollo del proyecto 19](#_Toc475443126)

[7. Conclusiones y líneas de trabajo futuras 19](#_Toc475443127)

[ANEXOS 20](#_Toc475443128)

[I. Plan de proyecto software 21](#_Toc475443129)

[I.1. Introducción 21](#_Toc475443130)

[I.2. Planificación temporal del proyecto 21](#_Toc475443131)

[I.3. Iteración 1 (13 Nov 2014 a 27 Nov 2014) 21](#_Toc475443132)

[I.4. Estudio de viabilidad 21](#_Toc475443133)

[I.5. Estudio de viabilidad económica. 21](#_Toc475443134)

[I.6. Estudio de viabilidad legal. 21](#_Toc475443135)

[II. Especificación de requisitios 22](#_Toc475443136)

[II.1. Introducción 22](#_Toc475443137)

[II.2. Objetivos generales 22](#_Toc475443138)

[II.3. Catálogo de requisitos 22](#_Toc475443139)

[II.3.1. Requisitos funcionales 22](#_Toc475443140)

[I.1.1. Requisitos no funcionales 23](#_Toc475443141)

[I.1.2. Restricción de requisitos 23](#_Toc475443142)

[II.4. Especificación de requisitos 23](#_Toc475443143)

[I. Especificación de diseño 24](#_Toc475443144)

[I.1. Introducción 24](#_Toc475443145)

[I.2. Diseño de datos 24](#_Toc475443146)

[I.3. Diseño procedimental 24](#_Toc475443147)

[I.4. Diseño arquitectónico 24](#_Toc475443148)

[II. Manual del programador 24](#_Toc475443149)

[II.1. Introducción 24](#_Toc475443150)

[II.2. Estructura de directorios 24](#_Toc475443151)

[II.3. Manual del programador 24](#_Toc475443152)

[II.4. Compilación, instalación y ejecución del proyecto 24](#_Toc475443153)

[II.5. Pruebas del sistema 24](#_Toc475443154)

[III. Manual del usuario 24](#_Toc475443155)

[III.1. Introducción 24](#_Toc475443156)

[III.2. Requisitos de usuarios 24](#_Toc475443157)

[III.3. Instalación 24](#_Toc475443158)

[III.4. Manual del usuario 24](#_Toc475443159)

[Bibliografía 25](#_Toc475443160)

# Índice de ilustraciones

**No se encuentran elementos de tabla de ilustraciones.**

# Índice de tablas

No se encuentran elementos de tabla de ilustraciones.

# Introducción

Este trabajo de final de grado nace de la base de un proyecto anterior <nombre\_del proyecto>, debido a la necesidad de ampliar la funcionalidad de dicho proyecto a una aplicación móvil 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.

# 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.

## Objetivos funcionales

Partiendo del proyecto de partida identificamos los siguientes nuevos objetivos:

* + 1. Creación de una aplicación móvil que permita acceder y realizar las mismas actividades que permite la aplicación Web. Estas actividades varían en función del tipo de usuario que las desempeñe.
       1. Docentes:
          1. Llevarán a cabo las tareas de crear, editar, publicar o duplicar un cuestionario en Moodle.
          2. Estos cuestionarios permitirán a los docentes poder llevar a cabo una evaluación de los conocimientos adquiridos por su alumnado.
          3. Al duplicar los cuestionarios estos podrán reutilizarse por los docentes para los distintos grupos o asignaturas que dicho docente imparta.
          4. Además, gracias a esta aplicación el docente podrá llevar a cabo todas estas funcionalidades con total comodidad desde su Smartphone.
       2. Alumnos:
          1. Ofrecer la posibilidad de resolver los distintos cuestionarios a los que tenga que enfrentarse desde su Smartphone.
          2. Al finalizar el mismo, el sistema notificará al alumno de la calificación obtenido junto con una retroalimentación de las diferentes preguntas.
          3. Estas calificaciones podrán variar en función de distintas recompensas a la hora de enfrentarse al cuestionario.
          4. Estas recompensas o *comodines* a partir de ahora, permitirán al alumnado enfrentarse a la prueba de una manera más amigable al enmascarar la verdadera finalidad del cuestionario.

## Objetivos de carácter técnico

* + 1. La aplicación móvil deberá de poder ser lo suficientemente amigable para los distintos usuarios para facilitar su correcta utilización y finalidad.
    2. La aplicación móvil a desarrollar se llevará a cabo para Android.
    3. Para llevar a cabo la creación de la aplicación se utilizará el entorno de desarrollo Android Studio.
    4. <Versión: de la aplicación>
    5. Como conectaremos Android con la Web
    6. aPI APLICACIÓN, FUNCIONE EN UN DISPOSITIVOS. VERSION

## Objetivos personales

* + 1. Destacar que uno de los objetivos principales de este proyecto es adquirir nuevos conocimientos dentro del desarrollo *Android*, junto con otros conocimientos necesarios para poder llevar a cabo la correcta integración del proyecto de partida a el proyecto a desarrollar.
    2. Afrontarme a nuevos retos que pongan a prueba todos mis conocimientos adquiridos a lo largo del grado.
    3. Enfrentarme a un posible trabajo o proyecto que se asemeje a mi vida profesional.

# Conceptos teóricos

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

## 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 reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software”.* [1]

Es decir, permite la comunicación entre distintos componentes software. La mayor ventaja es que permite reutilizar métodos escrito 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 las siguientes *Simple Object Access Protocol.* Posteriormente en esta misma sección se dedicará un apartado donde se explicará en mayor.

### 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

En este caso 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 para llevar a cabo una explicación en mayor detalle.

[2]

## LTI

## Soap

Anteriormente se ha llevado a cabo una pequeña explicación sobre este protocolo de comunicación. Relacionado con esto definíamos *SOAP* como un protocolo que posibilita el intercambio de información, es decir, la comunicación mediante internet entre aplicaciones. Para llevar a cabo este intercambio de datos se utiliza el formato *XML*, gracias a este el intercambio de datos se puede realizar independientemente de la plataforma o lenguaje utilizado.

Hoy en día, estos mensajes de tipo *SOAP* se envían generalmente mediante el protocolo *HTTP*, aunque también se pueden utilizar otros como *SMTP, TCP o JMS* entre otros. Otro aspecto importante es que es un pilar fundamental para los *Web Services*, ya que estos utilizan también el formato *XML* para describir los servicios. Además, permite el intercambio de datos o llamadas a procedimientos remotos mediante *RPC.*

Para poder llevar a cabo la correcta lectura de un mensaje *SOAP* el documento *XML* debe de tener una determinada estructura. Dicha estructura debe estar formada por las siguientes partes:

* **Envelope:** esta parte de la estructura del mensaje es obligatoria, ya que identifica el mensaje.
* **Header:** esta parte de la estructura permite enviar información adicional sobre cómo debe de procesarse el mensaje. Es por esto por lo que esta parte no es obligatoria, debido a que únicamente aporta cierta información relacionada con la lectura del mismo.
* **Body:** esta parte de la estructura es la encargada de almacenar toda la información del mensaje.
* **Fault:** esta parte de la estructura al igual que el *Header* es opcional, ya que aporte información relacionada con ciertos errores producidos durante el procesado del mensaje.

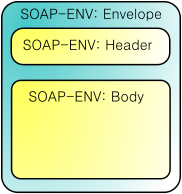


Ilustración 1 - Estructura de un mensaje soap [3]

[3] [4]

## Rest

Como ya hemos explicado anteriormente,es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.[5]cuyas siglas son las siguientes **RE**presentational **S**tate **T**ransfer 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.

* **Protocolo cliente/servidor sin estado:** de manera que cada petición *HTTP* debe contener toda aquella información necesaria para poder ejecutarse, es decir, 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.

[6]

# Técnicas y herramientas

## Lenguajes

## PHP

## JSON

## Android

## Herramientas

## XAMPP

## GitHub

## TortoiseSVN

## Moodle

## PhpStorm

## phpMyAdmin

## PHPUnit

## SonarQube

## Modelio

## Técnicas

## LTI

## MVC

## SCRUM

# Trabajos relacionados

# Aspectos relevantes del desarrollo del proyecto

# Conclusiones y líneas de trabajo futuras

ANEXOS

|  |  |  |
| --- | --- | --- |
|  | UNIVERSIDAD DE BURGOS  ESCUELA POLITÉCNICA SUPERIOR  Grado en Ingeniería en Informática |  |
|  |  |  |

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

Presentado por Daniel Puente Gabarri en Universidad de Burgos — dd de mm de YYYY Tutor: María Belén Vaquerizo García y Bruno Baruque Zanón

# Plan de proyecto software

## Introducción

## Planificación temporal del proyecto

## Iteración 1 (****13 Nov 2014**** a ****27 Nov 2014****)

## Estudio de viabilidad

## Estudio de viabilidad económica.

## Estudio de viabilidad legal.

# Especificación de requisitios

## Introducción

A continuación, en este apartado se va a llevar a cabo a la numeración y correspondiente explicación de los requisitos funcionales junto con los diagramas de casos de uso generados a partir de estos requisitos. También, se van a citar aquellos requisitos no funcionales que cumple la aplicación.

## Objetivos generales

## Catálogo de requisitos

### Requisitos funcionales

https://es.wikipedia.org/wiki/Requisito\_funcional

#### Rol profesor

|  |  |
| --- | --- |
| Nº RF | Descripción |
| RF-P1 | El profesor deberá registrarse en QuickTest utilizando como nombre de usuario, el mismo correo que en Moodle. |
| RF-P2 | Podrá iniciar sesión una vez logeado. |
| RF-P3 | Podrá crear un nuevo cuestionario. |
| RF-P4 | Podrá editar un cuestionario. |
| RF-P5 | Podrá eliminar un cuestionario. |
| RF-P6 | Podrá duplicar un cuestionario. |
| RF-P7 | Podrá configurar los comodines |
| RF-P8 | Podrá editar las preguntas de un cuestionario. |
| RF-P9 | Podrá configurar la puntuación de cada preguntar. |
| RF-P10 | Podrá elegir la respuesta correcta de cada pregunta. |
| RF-P11 | Podrá cerrar sesión. |
| RF-P12 | Podrá cerrar sesión y olvidar. |

#### Rol alumno

|  |  |
| --- | --- |
| Nº RF | Descripción |
| RF-A1 | El alumno deberá registrarse en QuickTest utilizando como nombre de usuario, el mismo correo que en Moodle. |
| RF-A2 | El alumno podrá iniciar sesión una vez logeado. |
| RF-A3 | Podrá responder al cuestionario. |
| RF-A4 | Podrá decidir si usar los comodines. |
| RF-A5 | Por cada pregunta del cuestionario podrá guardar su respuesta. |
| RF-A6 | Por cada pregunta del cuestionario podrá anular la respuesta guardada. |
| RF-A7 | Podrá finalizar el cuestionario y acabar. |
| RF-A8 | Podrá guardar el estado de todas las preguntas y finalizar y acabar. |
| RF-A9 | Podrá cerrar sesión. |
| RF-A10 | Podrá cerrar sesión y olvidar. |
| RF-A11 | Podrá revisar sus respuestas. |
| RF-A12 | Podrá ver su calificación. |
| RF-A13 | Si el cuestionario está cerrado podrá revisarlo y ver su calificación. |

### Requisitos no funcionales

https://es.wikipedia.org/wiki/Requisito\_no\_funcional

### Restricción de requisitos

## Especificación de requisitos

Aquí D.Casos de uso para los RF

# Especificación de diseño

## Introducción

## Diseño de datos

## Diseño procedimental

## Diseño arquitectónico

# Manual del programador

## Introducción

## Estructura de directorios

## Manual del programador

## Compilación, instalación y ejecución del proyecto

## Pruebas del sistema

# Manual del usuario

## Introducción

## Requisitos de usuarios

## Instalación

## Manual del usuario

# Bibliografía

|  |  |
| --- | --- |
| [1] | M. Merino, «Ticbeat - API,» 19 2 2017. [En línea]. Available: http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/. |
| [2] | «bbvaopen4u - API,» [En línea]. Available: https://bbvaopen4u.com/es/actualidad/que-es-una-api-y-que-puede-hacer-por-mi-negocio. |
| [3] | «Wikipedia - SOAP,» [En línea]. Available: https://es.wikipedia.org/wiki/Simple\_Object\_Access\_Protocol. |
| [4] | O. F. Brea, «DesarroloWeb - SOAP,» [En línea]. Available: https://desarrolloweb.com/articulos/1853.php. |
| [5] | «Wikipedia - REST,» 19 2 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Transferencia\_de\_Estado\_Representacional. |
| [6] | «bbvaopen4u - REST,» [En línea]. Available: https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos. |