

**Universidad Carlos III de Madrid**  
Escuela Politécnica Superior



Grado en Ingeniería Informática  
Trabajo Fin de Grado

**Aplicación Android para  
interacción en educación  
presencial**

Autor: Alejandro Martín García

Tutor: Ángel García Olaya



# AGRADECIMIENTOS

Llegado este momento, uno observa los años anteriores y ve lo rápido que han pasado. Han sido cuatro años cargados de experiencias y anécdotas y en los que he hecho grandes amistades. Ahora, es justo nombrar a todas esas personas, tanto a las que me han acompañado desde muy pequeño como a las que he ido conociendo con el trascurso de los años, para agradecerles haber compartido tiempo y haber aprendido algo de cada una de ellas.

Particularmente, a las primeras personas que debo mencionar es a mis padres. Gracias a ellos puedo escribir estos agradecimientos. Desde pequeño me inculcaron una educación y unos valores de esfuerzo y pasión por el estudio que han conseguido que pueda llegar hasta aquí. Os doy las gracias. Por supuesto, también a mi hermana por preocuparse de mí. Gracias. Los siguientes a los que debo mencionar son mis abuelos. Gracias por todo lo que me habéis aportado.

En el terreno de la universidad, debo agradecer a Alfonso, Raúl y Roberto el haber sido más amigos que compañeros. A Alfonso por estar siempre disponible para echarme una mano y ser una gran persona. A Raúl, con el que he compartido bastante tiempo de la carrera y por ser también un gran amigo. Y a Roberto, porque le conocí nada más entrar en la universidad y siempre ha estado ahí. Gracias a los tres. Y también al resto de compañeros: Andrea, Víctor, Álvaro, Georgi, Rubén, Carlos, Juanma, Sergio, Jorge y Ana.

Debo agradecer a mi tutor Ángel el tiempo prestado y su dedicación al trabajo. Muchas gracias. Tampoco deben faltar en estos agradecimientos Javier García ni Javier Sánchez por su colaboración y ayuda.

Finalmente, aunque no los mencione por su nombre en estas cortas líneas, deseo agradecer a todas aquellas personas que han aportado su granito de arena en todo este camino: a los compañeros y profesores del conservatorio, donde me he encontrado a gente genial, a los profesores del instituto que confiaron de mí y al resto de amigos.



# RESUMEN

El presente proyecto trata de facilitar la labor de la enseñanza con el uso de las tecnologías actuales. Hoy en día, la mayoría de los alumnos llevan siempre consigo un teléfono inteligente. El potencial de estos dispositivos es muy elevado, permiten realizar miles de tareas gracias a la ingente cantidad de aplicaciones que existe para ellos. Dentro del aula, pueden permitir realizar múltiples actividades, facilitando la comunicación con el profesor y que este tenga constancia del desarrollo de la sesión.

Una de las actividades más comunes llevadas a cabo en las clases, que permite al profesor recibir información sobre el seguimiento, es la realización de cuestionarios. Con ellos, el profesor podrá analizar si los alumnos han aprendido correctamente los contenidos explicados. Otro uso de estos cuestionarios es realizar evaluaciones de los alumnos. Tanto en un caso como en otro, resulta incuestionable la practicidad de esta herramienta.

Actualmente, la Universidad Carlos III de Madrid cuenta con un sistema de gestión de aprendizaje basado en Moodle, llamado Aula Global, que permite crear cuestionarios por los profesores para que sean respondidos por los alumnos. Sin embargo, la utilización dentro del aula de las funciones de este sistema se reduce a acceder al mismo únicamente desde un ordenador. Existen múltiples aplicaciones para dispositivos móviles que permiten realizar tareas de esta plataforma, pero ninguna responder a los cuestionarios. En esto reside el objetivo principal de este proyecto: la implementación de una aplicación, para uno de los sistemas operativos de dispositivos móviles más extendido, que permita la realización de cuestionarios de Moodle. Gracias a esta aplicación, los alumnos podrán contestar desde sus dispositivos a los cuestionarios creados por los profesores en la plataforma.

Debido a que la universidad ya cuenta con un sistema que incluye aplicaciones para dispositivos móviles que permiten realizar preguntas de tipo test, se determinó efectuar una ampliación de este sistema para que recuperase los cuestionarios de Moodle y que los alumnos pudieran contestar las preguntas que contenían desde sus dispositivos. Este sistema se denomina mSuite y ha sido desarrollado por el grupo de investigación miSEL, que pertenece al Software Engineering Lab. Está formado por un conjunto de aplicaciones para dispositivos con sistemas operativos iOS y Android que ayudan a los profesores en la preparación e impartición de clases, la realización de preguntas de tipo test a los alumnos y la integración con redes sociales para hacerles llegar información durante la clase.

En esta ampliación, se ha implementado una aplicación para el alumno para dispositivos Android, se ha desarrollado un complemento para Moodle que permita obtener los cuestionarios creados por los profesores desde otros sistemas y se han realizado una serie de modificaciones en la estructura de este sistema.

**Palabras clave:** Android, LMS, mSuite, Moodle, dispositivos móviles, cuestionarios.



# ABSTRACT

The present work tries to make easier the educational work using the current technologies. Nowadays, most students carry a smartphone. These devices have a huge potential, allowing to perform thousands of tasks through the large amount of applications that exist for them. Within the classroom, they make multiples activities possible, improving communication with the professor so that he/she can receive feedback.

One of the most common activities carried out in the classroom, which allows the professor receiving information about how the students are following the class, is to answer questionnaires. With them, the professor can analyze if the students have correctly acquired the knowledge explained. Another use of this tool is to assess students. In both cases, the usefulness of the questionnaires is beyond question.

The Universidad Carlos III de Madrid currently has a Learning Management System based on Moodle which is called Aula Global. This system allows the professors to create questionnaires to be answered by the students. However, the use inside the classroom of the LMS is limited to accessing it from a computer. There are numerous applications for mobile devices that permit to perform diverse tasks of this system, but none allows answering questionnaires. Here lies the main objective of this project: the implementation of an application to one of the most extended operative systems for mobile devices to answer Moodle questionnaires. Thanks to this application, the students may answer the questions, created in Moodle by the professors, from their devices.

Because the university has already a system with applications to answer multiple-choice questions, it was decided to expand this system to get the questionnaires from Moodle, so the students can answer the questions contained in them from student's applications of this system. It is named mSuite and has been developed by the research group miSEL which belongs to the Software Engineering Lab. It is composed of applications set for iOS and Android operating systems that help the professors planning and delivering their classes, asking questions to students and integrating with social networks to send information during the class.

In this extension, an Android application for the students has been developed, a plugin has been created to recover questionnaires created by the professors in Moodle and some changes on the architecture have been performed.

**Keywords:** Android, LMS, mSuite, Moodle, mobile devices, questionnaires.





# ÍNDICE GENERAL

1. Introducción .....	15
1.1 Descripción del problema .....	15
1.2 Motivación.....	16
1.3 Objetivos del trabajo.....	17
1.4 Estructura del documento.....	18
2. Estado del arte .....	19
2.1 Arquitectura Orientada a Servicios .....	19
2.1.1 Servicios web.....	20
2.2 Tecnologías de la Información y la Comunicación en el aprendizaje.....	21
2.3 Sistemas de Gestión de Aprendizaje .....	22
2.3.1 Blackboard .....	22
2.3.2 ConnectEDU .....	23
2.3.3 Edmodo.....	23
2.4 Moodle .....	23
2.4.1 Clientes de Moodle .....	25
2.4.1.4 MY MOODLE .....	27
2.4.2 Comparación de los clientes de Moodle.....	28
2.4.3 Servicios web de Moodle .....	29
2.5 mSuite .....	29
2.5.1 mTeachPlan.....	30
2.5.2 mTeachClass.....	31
2.5.3 mClassTest .....	31
2.6 Plataformas de desarrollo para dispositivo móvil.....	33
2.6.1 Windows Phone .....	34
2.6.2 iOS.....	34
2.6.3 Android.....	35
2.6.4 Comparación de las plataformas de desarrollo .....	37
2.6.5 Aplicaciones nativas, híbridas y web.....	37
2.6.6 Entorno de desarrollo Android SDK .....	39
2.7 Play Framework.....	40

3.	Análisis de la solución.....	41
3.1	Descripción de la solución.....	41
3.1.1	Propuesta inicial.....	41
3.1.2	Problemáticas y cambios en la propuesta inicial .....	41
3.1.3	Descripción de la solución final.....	43
3.2	Restricciones del sistema .....	45
3.3	Entorno operacional.....	46
3.4	Especificación de casos de uso.....	47
3.4.1	Diagramas de casos de uso .....	47
3.4.2	Descripción de los atributos de los casos de uso.....	49
3.4.3	Descripción textual de los casos de uso.....	50
3.5	Especificación de requisitos .....	56
3.5.1	Descripción de los atributos de los requisitos .....	56
3.5.2	Especificación de requisitos.....	58
4.	Diseño.....	68
4.1	Descripción general de la arquitectura .....	68
4.2	Alternativas de diseño.....	69
4.3	Arquitectura detallada del sistema .....	70
4.3.1	SELQuizServer .....	70
4.3.2	Aplicación Android .....	73
4.3.3	Extensión para Moodle .....	78
4.4	Funcionamiento del sistema .....	81
4.4.1	Identificación con Twitter .....	81
4.4.2	Listar sesiones .....	82
4.4.3	Listar preguntas.....	84
4.4.4	Recibir notificación.....	85
4.5	Diseño de la interfaz gráfica de la aplicación Android .....	85
4.5.1	Prototipo de la aplicación Android .....	87
5.	Implementación y pruebas del sistema .....	92
5.1	Implementación .....	92
5.1.1	Problemas encontrados en la implementación .....	92
5.2	Pruebas del sistema .....	93

5.2.1	Entorno de pruebas del sistema .....	93
5.2.2	Descripción de los atributos de las pruebas del sistema .....	94
5.2.3	Especificación de las pruebas del sistema .....	95
5.2.4	Trazabilidad entre pruebas del sistema y requisitos .....	102
6.	Gestión del proyecto .....	103
6.1	Metodología de desarrollo .....	103
6.2	Planificación del proyecto .....	105
6.3	Presupuesto del proyecto .....	107
6.3.1	Costes de recursos humanos .....	107
6.3.2	Costes de material .....	107
6.3.3	Presupuesto total del proyecto.....	108
7.	Conclusiones y trabajos futuros .....	109
7.1	Conclusiones.....	109
7.2	Trabajos futuros .....	110
8.	Bibliografía .....	112
	Anexos.....	115
	Anexo A: Manual de usuario de la aplicación mClassTest para Android .....	116
	Anexo B: Manual de instalación y uso de la extensión para Moodle .....	122
	Anexo C: Instalación del servidor SELQuizServer .....	127
	Anexo D: Project abstract .....	130
	Glosario de términos .....	140

# ÍNDICE DE FIGURAS

<i>Figura 1 - Diagrama del funcionamiento de la Arquitectura Orientada a Servicios .....</i>	<i>21</i>
<i>Figura 2 - Tipos de preguntas admitidas en cuestionarios de Moodle.....</i>	<i>24</i>
<i>Figura 3 - Realización de test con Moodle Mobile App.....</i>	<i>25</i>
<i>Figura 4 - Realización de cuestionarios en mTouch .....</i>	<i>27</i>
<i>Figura 5 - Pantalla principal de mTeachPlan.....</i>	<i>30</i>
<i>Figura 6 - Pantalla principal de la aplicación mTeachClass.....</i>	<i>31</i>
<i>Figura 7 - mClassTest para Android en HTML5.....</i>	<i>32</i>
<i>Figura 8 - mClassTest para iOS.....</i>	<i>33</i>
<i>Figura 9 - Gráfico de cuotas de mercado de las plataformas móviles en España.....</i>	<i>33</i>
<i>Figura 10 - Capas en la arquitectura de Android .....</i>	<i>36</i>
<i>Figura 11 - Entorno de desarrollo Android SDK.....</i>	<i>40</i>
<i>Figura 12 - Diagrama general de la solución final .....</i>	<i>45</i>
<i>Figura 13 - Diagrama de casos de uso del alumno .....</i>	<i>48</i>
<i>Figura 14 - Diagrama de casos de uso del usuario de la extensión de Moodle .....</i>	<i>48</i>
<i>Figura 15 - Diagrama general de la arquitectura del sistema .....</i>	<i>68</i>
<i>Figura 16 - Diagrama de componentes del servidor SELQuizServer.....</i>	<i>71</i>
<i>Figura 17 - Diagrama de la base de datos del SELQuizServer .....</i>	<i>72</i>
<i>Figura 18 - Diagrama de componentes de la aplicación Android .....</i>	<i>74</i>
<i>Figura 19 - Diagrama de clases 1 del componente mClassTest .....</i>	<i>74</i>
<i>Figura 20 - Diagrama de clases 2 del componente mClassTest .....</i>	<i>75</i>
<i>Figura 21 - Diagrama de clases del componente Gestor de sesiones .....</i>	<i>76</i>
<i>Figura 22 - Diagrama de clases del componente Gestor de preguntas y respuestas .....</i>	<i>77</i>
<i>Figura 23 - Diagrama de clases del componente Gestor de notificaciones .....</i>	<i>78</i>
<i>Figura 24 - Diagrama de clases de la extensión para Moodle .....</i>	<i>79</i>
<i>Figura 25 - Diagrama de secuencia 1 - Identificación con Twitter.....</i>	<i>82</i>
<i>Figura 26 - Diagrama de secuencia 2 - Listar sesiones.....</i>	<i>83</i>
<i>Figura 27 - Diagrama de secuencia 3 - Listar preguntas.....</i>	<i>84</i>
<i>Figura 28 - Diagrama de secuencia 4 - Recibir notificación .....</i>	<i>85</i>
<i>Figura 29 - Vista del listado de sesiones en la aplicación mClassTest para iOS .....</i>	<i>86</i>
<i>Figura 30 - Ejemplo de Drawer Layout en Android .....</i>	<i>87</i>
<i>Figura 31 - Prototipo de la aplicación Android - Vista de presentación.....</i>	<i>88</i>
<i>Figura 32 - Prototipo de la aplicación Android - Inicio de sesión con Twitter .....</i>	<i>88</i>
<i>Figura 33 - Prototipo de la aplicación Android - Vista de introducción del NIA .....</i>	<i>89</i>
<i>Figura 34 - Prototipo de la aplicación Android - Vista del listado de sesiones.....</i>	<i>89</i>
<i>Figura 35 - Prototipo de la aplicación Android - Vista del listado de preguntas.....</i>	<i>90</i>
<i>Figura 36 - Prototipo de la aplicación Android - Pregunta de respuesta múltiple .....</i>	<i>90</i>
<i>Figura 37 - Prototipo de la aplicación Android - Pregunta de respuesta simple.....</i>	<i>91</i>
<i>Figura 38 - Modelo en cascada realimentado.....</i>	<i>103</i>
<i>Figura 39 - Diagrama Gantt de la planificación del proyecto.....</i>	<i>106</i>

<i>Figura 40 - Vista de presentación de mClassTest para Android.....</i>	<i>117</i>
<i>Figura 41 - Vista de solicitud de inicio de sesión con Twitter de mClassTest para Android .....</i>	<i>117</i>
<i>Figura 42 - Vista inicio de sesión con Twitter en mClassTest para Android .....</i>	<i>118</i>
<i>Figura 43 - Vista de solicitud de NIA en mClassTest para Android.....</i>	<i>118</i>
<i>Figura 44 - Vista del listado de sesiones en mClassTest para Android.....</i>	<i>119</i>
<i>Figura 45 - Cuadro de diálogo en sesiones protegidas en mClassTest para Android.....</i>	<i>119</i>
<i>Figura 46 - Vista del menú en mClassTest para Android.....</i>	<i>120</i>
<i>Figura 47 - Vista del listado de preguntas en mClassTest para Android.....</i>	<i>120</i>
<i>Figura 48 - Vista de pregunta de respuesta simple en mClassTest para Android .....</i>	<i>121</i>
<i>Figura 49 - Vista de pregunta de respuesta múltiple en mClassTest para Android .....</i>	<i>121</i>
<i>Figura 50 - Reconocimiento de la nueva extensión para Moodle .....</i>	<i>123</i>
<i>Figura 51 - Activación de protocolo en Moodle .....</i>	<i>124</i>
<i>Figura 52 - Modificación del nombre corto de la extensión en la base de datos Moodle.....</i>	<i>125</i>
<i>Figura 53 - Orden de las llamadas hacia la extensión para Moodle .....</i>	<i>126</i>
<i>Figure 54 - General diagram of mSuite .....</i>	<i>135</i>

# ÍNDICE DE TABLAS

<i>Tabla 1 - Comparación de los clientes de Moodle.....</i>	<i>28</i>
<i>Tabla 2 - Distribución de versiones de Android [28].....</i>	<i>36</i>
<i>Tabla 3 - Comparación de plataformas móviles.....</i>	<i>37</i>
<i>Tabla 4 - Comparativa de aplicaciones nativas, híbridas y web.....</i>	<i>38</i>
<i>Tabla 5 - Tabla de ejemplo de especificación de caso de uso .....</i>	<i>49</i>
<i>Tabla 6 - Caso de uso CU-001 - Identificarse en aplicación.....</i>	<i>50</i>
<i>Tabla 7 - Caso de uso CU-002 - Ver sesiones.....</i>	<i>50</i>
<i>Tabla 8 - Caso de uso CU-003 - Ver preguntas.....</i>	<i>51</i>
<i>Tabla 9 - Caso de uso CU-004 - Suscribirse a una sesión.....</i>	<i>51</i>
<i>Tabla 10 - Caso de uso CU-005 - Cancelar suscripción a una sesión.....</i>	<i>52</i>
<i>Tabla 11 - Caso de uso CU-006 - Suscribirse a una sesión con contraseña.....</i>	<i>52</i>
<i>Tabla 12 - Caso de uso CU-007 - Responder pregunta simple.....</i>	<i>53</i>
<i>Tabla 13 - Caso de uso CU-008 - Responder pregunta múltiple.....</i>	<i>54</i>
<i>Tabla 14 - Caso de uso CU-009 - Cerrar sesión en la aplicación.....</i>	<i>54</i>
<i>Tabla 15 - Caso de uso CU-010 - Recibir notificación.....</i>	<i>55</i>
<i>Tabla 16 - Caso de uso CU-011 - Obtener listado de alumnos de un curso.....</i>	<i>55</i>
<i>Tabla 17 - Caso de uso CU-012 - Obtener listado de cursos.....</i>	<i>55</i>
<i>Tabla 18 - Caso de uso CU-013 - Obtener cuestionario.....</i>	<i>56</i>
<i>Tabla 19 - Tabla de ejemplo de especificación de requisito.....</i>	<i>57</i>
<i>Tabla 20 - Requisito RF-001 - Identificación mediante Twitter.....</i>	<i>58</i>
<i>Tabla 21 - Requisito RF-002 - Solicitud de NIA.....</i>	<i>58</i>
<i>Tabla 22 - Requisito RF-003 - Lista sesiones.....</i>	<i>58</i>
<i>Tabla 23 - Requisito RF-004 - Lista preguntas.....</i>	<i>59</i>
<i>Tabla 24 - Requisito RF-005 - Sincronización de la aplicación Android con servidor.....</i>	<i>59</i>
<i>Tabla 25 - Requisito RF-006 - Suscripción de sesiones.....</i>	<i>59</i>
<i>Tabla 26 - Requisito RF-007 - Sesiones protegidas con contraseña.....</i>	<i>60</i>
<i>Tabla 27 - Requisito RF-008 - Responder pregunta.....</i>	<i>60</i>
<i>Tabla 28 - Requisito RF-009 - Cerrar sesión en la aplicación.....</i>	<i>60</i>
<i>Tabla 29 - Requisito RF-010 - Servicio de notificaciones.....</i>	<i>60</i>
<i>Tabla 30 - Requisito RF-011 - Menú en la aplicación.....</i>	<i>61</i>
<i>Tabla 31 - Requisito RF-012 - Vista predeterminada.....</i>	<i>61</i>
<i>Tabla 32 - Requisito RF-013 - Intento de envío de preguntas sin proporcionar respuesta.....</i>	<i>61</i>
<i>Tabla 33 - Requisito RF-014 - Una única sesión suscrita simultáneamente.....</i>	<i>61</i>
<i>Tabla 34 - Requisito RF-015 - Obtención de un listado de estudiantes asociados a un curso de Moodle.....</i>	<i>62</i>
<i>Tabla 35 - Requisito RF-016 - Obtención de un listado de cursos de Moodle.....</i>	<i>62</i>
<i>Tabla 36 - Requisito RF-017 - Obtención de cuestionarios de Moodle.....</i>	<i>62</i>
<i>Tabla 37 - Requisito RF-018 - Cambios en SELQuizServer.....</i>	<i>63</i>
<i>Tabla 38 - Requisito RNF-001 - Servicio de notificaciones con GCM.....</i>	<i>63</i>

<i>Tabla 39 - Requisito RNF-002 - Diseño menú en aplicación Android .....</i>	<i>63</i>
<i>Tabla 40 - Requisito RNF-003 - Versión mínima de Android .....</i>	<i>63</i>
<i>Tabla 41 - Requisito RNF-004 - Normas de estilo, diseño y programación .....</i>	<i>64</i>
<i>Tabla 42 - Requisito RNF-005 - Uso de Switch en las sesiones.....</i>	<i>64</i>
<i>Tabla 43 - Requisito RNF-006 - Un único intento de respuesta.....</i>	<i>64</i>
<i>Tabla 44 - Requisito RNF-007 - Único Switch activo.....</i>	<i>64</i>
<i>Tabla 45 - Requisito RNF-008 - Uso de CheckBox en el listado de preguntas .....</i>	<i>65</i>
<i>Tabla 46 - Requisito RNF-009 - Distintas resoluciones.....</i>	<i>65</i>
<i>Tabla 47 - Requisito RNF-010 - Vista vertical.....</i>	<i>65</i>
<i>Tabla 48 - Requisito RNF-011 - Actualización de contenido en la aplicación.....</i>	<i>65</i>
<i>Tabla 49 - Requisito RNF-012 - Protocolo de llamadas a la extensión para Moodle .....</i>	<i>66</i>
<i>Tabla 50 - Requisito RNF-013 - Lenguaje de implementación de la extensión para Moodle.....</i>	<i>66</i>
<i>Tabla 51 - Requisito RNF-014 - Implementación de la extensión para Moodle en distintas llamadas.....</i>	<i>66</i>
<i>Tabla 52 - Requisito RNF-015 - Control de roles.....</i>	<i>66</i>
<i>Tabla 53 - Requisito RNF-016 - Token para la extensión para Moodle.....</i>	<i>67</i>
<i>Tabla 54 - Matriz de trazabilidad entre requisitos y casos de uso .....</i>	<i>67</i>
<i>Tabla 55 - Tabla de ejemplo de especificación de prueba del sistema .....</i>	<i>95</i>
<i>Tabla 56 - Prueba del sistema PS-001 - Identificación en Twitter y registro en el SELQuizServer .....</i>	<i>95</i>
<i>Tabla 57 - Prueba del sistema PS-002 - Recuperación y visualización de sesiones.....</i>	<i>96</i>
<i>Tabla 58 - Prueba del sistema PS-003 - Suscripción de sesi .....</i>	<i>96</i>
<i>Tabla 59 - Prueba del sistema PS-004 - Solicitud de contraseña para sesiones protegidas.....</i>	<i>97</i>
<i>Tabla 60 - Prueba del sistema PS-005 - Respuesta a preguntas .....</i>	<i>97</i>
<i>Tabla 61 - Prueba del sistema PS-006 - Cierre de sesión.....</i>	<i>98</i>
<i>Tabla 62 - Prueba del sistema PS-007 - Recepción de notificaciones.....</i>	<i>98</i>
<i>Tabla 63 - Prueba del sistema PS-008 - Intercambio de vistas de preguntas y sesiones .....</i>	<i>98</i>
<i>Tabla 64 - Prueba del sistema PS-009 - Vista predeterminada.....</i>	<i>98</i>
<i>Tabla 65 - Prueba del sistema PS-010 - Imposibilidad de enviar respuesta vacía.....</i>	<i>99</i>
<i>Tabla 66 - Prueba del sistema PS-011 - Suscripción única .....</i>	<i>99</i>
<i>Tabla 67 - Prueba del sistema PS-012 - Obtención de listado de estudiantes de Moodle .....</i>	<i>100</i>
<i>Tabla 68 - Prueba del sistema PS-013 - Obtención de listado de cursos de Moodle.....</i>	<i>100</i>
<i>Tabla 69 - Prueba del sistema PS-014 - Obtención de listado de cuestionarios de Moodle.....</i>	<i>101</i>
<i>Tabla 70 - Prueba del sistema PS-015 - Obtención de listado de preguntas y respuestas de Moodle .....</i>	<i>101</i>
<i>Tabla 71 - Matriz de trazabilidad entre pruebas del sistema y requisitos .....</i>	<i>102</i>
<i>Tabla 72 - Costes de los recursos humanos del proyecto.....</i>	<i>107</i>
<i>Tabla 73 - Costes de los recursos materiales del proyecto.....</i>	<i>108</i>

# 1. INTRODUCCIÓN

En este primer capítulo se realiza una presentación del trabajo, indicando el problema al que va a dar solución, se plantean una serie de objetivos y se describe la estructura del documento.

## 1.1 DESCRIPCIÓN DEL PROBLEMA

Analizando las últimas décadas en la enseñanza, se puede ver como se ha ido introduciendo un elemento externo, cada vez con más fuerza: la tecnología. Los beneficios de este elemento son muy numerosos [1]. El uso de la tecnología proporciona un gran espectro de posibilidades para llevar a cabo la labor de la enseñanza. Por ejemplo, el uso de un proyector en el aula se encuentra muy extendido, particularmente en el entorno universitario, siendo una herramienta en la que muchos profesores se apoyan para impartir sus clases. Siempre que se haga un buen uso del mismo, evitando distracciones, facilita una herramienta con la que mostrar nuevos contenidos a los alumnos y de variado formato.

Otro ejemplo de irrupción de la tecnología en el entorno educativo, son las plataformas online de apoyo a la docencia utilizadas en las universidades, como es el caso de Aula Global en la Universidad Carlos III de Madrid. De ella se sirven los profesores para hacer llegar los distintos recursos utilizados en una asignatura a los alumnos, así como para comunicarse o incluso para hacer exámenes. Hoy en día, la mayoría de centros universitarios cuentan con sistemas similares como apoyo a la enseñanza.

Durante el transcurso de una clase, un profesor puede necesitar recoger cierta información de los alumnos, ya sea para evaluarlos, para tomar información del seguimiento de la sesión o para realizar una encuesta entre otras muchas opciones. Para esto, la opción frecuentemente escogida es utilizar un papel donde el alumno escribe la información requerida y que posteriormente se entrega al profesor. Obviamente, este proceso requiere de una pérdida de tiempo notable, además de no poder analizar la información recogida instantáneamente.

Una posibilidad para atajar este problema, podría pasar por utilizar la plataforma educativa online de la universidad. Haciendo uso de ella, se podría recoger información de los alumnos mediante cuestionarios o encuestas, pudiendo recibir la información en un tiempo muy reducido. Además, se añade la posibilidad de visualizar dicha información ya analizada informáticamente. A modo de ejemplo, un profesor podría ver la tasa de acierto global de la clase a una pregunta realizada. Incluso se puede realizar la corrección automática de estas pruebas y la integración, también automática, de los resultados con el resto de notas del alumno.



Sin embargo, esta solución genera otros problemas. No todos los alumnos disponen de un ordenador portátil en el aula con el que acceder a la web de la plataforma educativa. Incluso cuando una clase se imparte en un aula con equipos informáticos, no siempre hay ordenadores suficientes para ser utilizados individualmente. Estos motivos hacen que sea muy dificultoso el uso de ordenadores como vía para la comunicación en clase.

Paralelamente a la entrada de la tecnología en el aula, y por parte de los alumnos, existen unos dispositivos que se han impuesto decisivamente en los últimos tiempos: las tabletas o teléfonos móviles. Es este último elemento el que más extendido se encuentra, llevando la práctica totalidad de los estudiantes uno en su bolsillo. Los teléfonos móviles, en los últimos siete años, han sufrido una revolución vertiginosa. Dispositivos que permitían únicamente realizar llamadas y enviar mensajes se han convertido en pequeños ordenadores con los que realizar acciones de todo tipo gracias a las miles de aplicaciones disponibles. Son los llamados teléfonos inteligentes. A estos se suman las tabletas, que también crecen con fuerza.

El uso tan extendido de teléfonos inteligentes o tabletas entre los estudiantes ofrece un camino con numerosas posibilidades, gracias al potencial de los mismos. La conexión a Internet de estos dispositivos junto con la posibilidad de navegar por la web o utilizar diferentes aplicaciones, brinda la capacidad de interactuar telemáticamente con los alumnos. De esta forma, se puede dar respuesta al problema planteado del profesor que necesita recoger información durante una clase, evitando los inconvenientes comentados de los ordenadores portátiles. Se pueden desarrollar aplicaciones para estos dispositivos que permitan una recogida rápida y organizada de información.

### 1.2 MOTIVACIÓN

Este proyecto trata de acrecentar la unión de la tecnología con la enseñanza, con vistas a mejorar su desarrollo y su calidad. Desde el punto de vista del profesor, para tener herramientas que le permitan mejorar el proceso de la enseñanza y aumentar la atención de los alumnos. Desde el punto de vista del alumno, para mejorar el proceso de aprendizaje y asistir a clases participativas e interactivas.

Los teléfonos inteligentes, que poseen la mayoría de los alumnos, se pueden utilizar como eje central para crear nuevas herramientas y servicios. Gracias al potencial de estos dispositivos, se pueden desarrollar aplicaciones que permitan interactuar con los alumnos de una manera rápida y atractiva. Además, el uso de estos dispositivos posibilita el acceso a nuevos recursos, como son las redes sociales. Estas pueden ayudar al proceso de la enseñanza, ensanchando la comunicación entre las distintas personas involucradas.

En el apartado anterior se han mostrado algunas problemáticas que no cubre la tecnología comúnmente usada en la enseñanza, dejando lagunas susceptibles de mejora. Vistas las posibilidades de los teléfonos inteligentes, es lógico acercar ambos para lograr un perfeccionamiento en los métodos actuales. Entre estas problemáticas se encuentran los límites de la plataforma online utilizada. Hay ciertos recursos de la misma que únicamente pueden ser utilizados desde un navegador web en un ordenador. Con los teléfonos inteligentes se puede lograr extender la funcionalidad y accesibilidad de esta plataforma.

Desde el punto de vista de los alumnos, el uso de estos dispositivos comentados supone un acto muy cotidiano. Se contribuirá así a aumentar la atención de los alumnos en clase, lograr mejorar su participación y su asistencia. Estarán usando una tecnología que les resulta familiar y a la vez novedosa, debido a su poca utilización actual en el aula como una parte más del proceso educativo.

El interés en proyectos como este se acrecienta con el margen de posibilidades y mejoras que se puedan lograr en el futuro. Actualmente, es un terreno del que apenas se han aprovechado sus ventajas.

### 1.3 OBJETIVOS DEL TRABAJO

El objetivo central de este trabajo es permitir contestar a los cuestionarios creados por profesores en la plataforma educativa en línea de la Universidad (Aula Global, implementada sobre Moodle), desde una aplicación para dispositivos móviles. Los cuestionarios se podrán exportar desde Moodle para ser gestionados en un sistema externo (mSuite, ver sección 2.5). Desde este, se harán llegar las preguntas contenidas en los cuestionarios a los alumnos, que las recibirán en una aplicación desarrollada para su dispositivo móvil con sistema operativo Android. Desde la aplicación se podrá contestar a las preguntas recibidas y enviar posteriormente las respuestas.

A continuación se definen los principales hitos que componen el desarrollo de este proyecto:

- Creación de una aplicación para dispositivos Android: esta aplicación será utilizada por los alumnos para responder las preguntas definidas por los profesores en Aula Global. La aplicación será compatible tanto con preguntas de respuesta simple como de respuesta múltiple. Para conocer cuando hay nuevas preguntas disponibles, la aplicación implementará un servicio de recepción de notificaciones, para mostrar mensajes en el dispositivo del alumno.
- Creación de un módulo para la plataforma Moodle: se desarrollará un módulo que permita recuperar, desde otro sistema o aplicación, los cuestionarios creados por los profesores en Moodle y otros datos también necesarios, como el listado de cursos de un profesor o un listado de alumnos matriculados en un determinado curso.

- Establecer un control sencillo para vigilar la presencia del alumno en el aula correspondiente cuando conteste a un cuestionario, evitando que alumnos que no estén en el aula puedan responder a las preguntas planteadas en la clase.

### 1.4 ESTRUCTURA DEL DOCUMENTO

El documento se distribuye en siete apartados que detallan el desarrollo del proyecto más un octavo con la bibliografía utilizada. Además, se han añadido unos anexos finales con información sobre el uso e instalación de los diferentes elementos desarrollados en este proyecto y un resumen del mismo en inglés. Los siete capítulos mencionados son los siguientes:

- Capítulo uno, *Introducción*: este apartado contiene una presentación del trabajo, una descripción del problema a tratar y una explicación de los objetivos.
- Capítulo dos, *Estado del Arte*: se realiza una descripción de las tecnologías que afectan a este trabajo y un análisis de las aplicaciones ya existentes que deben ser tenidas en cuenta a la hora de elaborar una solución.
- Capítulo tres, *Análisis de la solución*: en este capítulo se plantean las posibles soluciones a adoptar, con los problemas encontrados en ellas, se describen las restricciones que tendrá el sistema y las características del entorno operacional y se realiza una definición de la solución finalmente escogida, formalizándola con una especificación de casos de uso y de requisitos.
- Capítulo cuatro, *Diseño*: en este apartado se diseña la solución planteada en el capítulo anterior, aportando diferentes diagramas que facilitarán la implementación. También se incluye un prototipo de la aplicación Android.
- Capítulo cinco, *Implementación y pruebas del sistema*: se realizan unas consideraciones sobre la implementación, indicando problemas encontrados en ella y se genera una serie de pruebas para evaluar el correcto funcionamiento del sistema.
- Capítulo seis, *Gestión del proyecto*: se describe el modelo de desarrollo de software seguido durante el proyecto, se aporta la planificación planteada y se formula un presupuesto del proyecto.
- Capítulo siete, *Conclusiones y trabajos futuros*: finalmente, se aportan unas conclusiones sobre el proyecto y se enumeran posibles modificaciones del sistema implementado en el futuro.

## 2. ESTADO DEL ARTE

El objetivo de este apartado es realizar una explicación de aquellas tecnologías y conceptos necesarios para entender el trabajo realizado. También se realizará un análisis comparativo de las diferentes herramientas o plataformas existentes que intervienen en alguna fase del proyecto.

En concreto, se hablará de las diferentes plataformas en línea de apoyo a la docencia más extendidas y en las que se incluye Moodle. También se comentarán los mecanismos utilizados por esta última plataforma para comunicarse con otros sistemas. A continuación, se explicará en qué consiste mSuite y para que se utilizan las aplicaciones que lo forman. En último lugar, se realizará un análisis de los sistemas operativos para dispositivos móviles más extendidos y de los modelos existentes de desarrollo de aplicaciones para estos.

### 2.1 ARQUITECTURA ORIENTADA A SERVICIOS

Es un concepto de definición compleja, debido a que varía en gran medida según la fuente escogida. Intentando dar una visión lo más objetiva posible, y desde el punto de vista del proyecto que detalla este documento, SOA, como se conoce por sus siglas en inglés, constituye un enfoque para el desarrollo de aplicaciones. En él, se establecen distintos servicios para ser accesibles desde una red [2], permitiendo así la comunicación y consiguiente envío de información entre diferentes sistemas.

Al no ser una tecnología concreta o un software, no existe una implementación específica. Su uso es cada vez mayor: cada vez más empresas lo utilizan y aprovechan sus ventajas. Para entender mejor en qué consiste esta arquitectura y por qué surgió, es necesario entender la evolución de los grandes sistemas informáticos. En sus primeras etapas, estos eran utilizados por las empresas para centralizar todo proceso, constatando un importante paso en la mejora de la productividad. Distintos procesos que dependían entre sí se podían ejecutar en un mismo sistema. Sin embargo, con el desarrollo de estos sistemas para manejar muchos otros subsistemas y un crecimiento en el volumen de datos manejado, su rendimiento se encontró con ciertos límites. Además, su infraestructura se vuelve muy complicada, dificultando cualquier modificación dentro de la misma.

Con la Arquitectura Orientada a Servicios se logra una separación entre los procesos de negocio de las funciones automatizadas, organizándolas en módulos individuales para facilitar su uso por toda la organización [3]. Se consigue aislar la implementación, facilitando el uso de estos servicios.

### 2.1.1 SERVICIOS WEB

Los servicios web constituyen una implementación de SOA. Se integran en aplicaciones para dotarlas de un mecanismo de comunicación con otras aplicaciones. Para su funcionamiento, definen una serie de funciones que serán accesibles desde la red mediante un protocolo de Internet, como HTTP. El objetivo de estas funciones es establecer un intercambio de información, no disponiendo de interfaz gráfica.

Cuando una aplicación realiza una llamada a una de las funciones definidas en unos servicios web, estos, en función de la información recibida de la aplicación que ha iniciado la comunicación, devolverán una respuesta. Un ejemplo de implementación de servicios web es un servidor de correo al que se conectan diversas aplicaciones cliente. Mediante llamadas a sus servicios web, las aplicaciones cliente podrán obtener los correos de cada usuario.

Los servicios web nacieron de la necesidad de comunicar distintas aplicaciones en la red, independientemente de la plataforma de cada una, siendo esto una de sus grandes ventajas. Además, no es necesario conocer los detalles de su implementación para realizar una llamada a los mismos.

Los servicios web tienen asociados cuatro conceptos fundamentales [4]:

- XML: es el lenguaje de marcas utilizado para codificar los datos que se intercambian entre cliente y servidor. También se utiliza para describir las funciones de los servicios web. Es un estándar que permite intercambiar datos entre distintas plataformas, aunque operen con distintos lenguajes.
- SOAP: es el protocolo utilizado para el envío de los datos entre los extremos y es independiente la plataforma. Este protocolo define cómo debe ser enviado y cómo debe estar construido un fichero XML. Los mensajes intercambiados estarán compuestos por un encabezado (contiene información para el enrutamiento del mensaje), el cuerpo (contiene los datos) y el sobre (define el contenido del mensaje).
- WSDL: siglas de *Web Services Description Language*. Es una especificación de XML para definir unos servicios web. Sirve para indicar los servicios que se proveen y desde qué dirección y otros detalles sobre su utilización [5]. La principal utilidad del uso de WSDL es que las aplicaciones que quieran acceder a los servicios web podrán obtener una descripción de los servicios y de cómo deben ser utilizados.
- UDDI: es un directorio distribuido para publicar los servicios web de una aplicación. Permite almacenar de forma centralizada un registro de sistemas y los servicios que ofrecen. Almacena información técnica y del proveedor de los servicios.

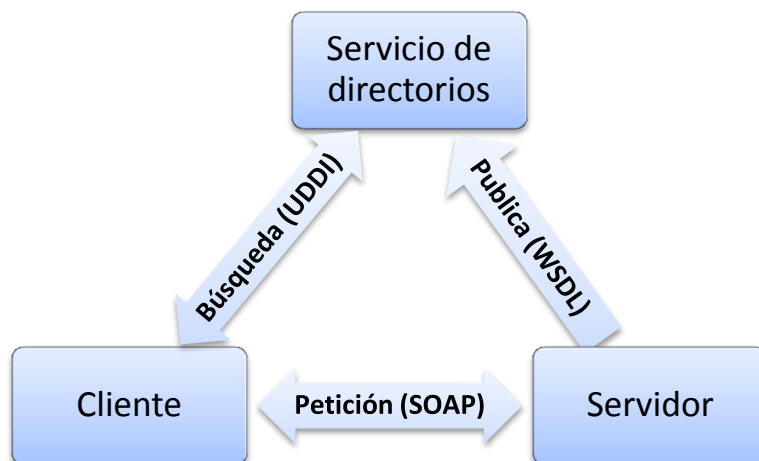


Figura 1 - Diagrama del funcionamiento de la Arquitectura Orientada a Servicios

En la Figura 1 se muestra un diagrama donde se representan los conceptos explicados y las relaciones que se establecen entre ellos. Una aplicación cliente puede acceder a UDDI para realizar una búsqueda de unos servicios web, con el objetivo de conocer cómo utilizarlos. El intercambio de datos entre el servidor y la aplicación cliente se realiza mediante el protocolo SOAP. Por último, el servidor publica en el servicio de directorios (UDDI) la especificación de sus servicios mediante WSDL.

## 2.2 TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN EN EL APRENDIZAJE

El avance de la Tecnologías de la Información y la Comunicación (TIC) dentro de los entornos educativos es innegable. Cada año crece su alcance, sus capacidades y su aceptación por parte de los alumnos. Los beneficios que aportan a la enseñanza pasan por un mayor número de recursos disponibles para los alumnos, mayores facilidades para acceder a los mismos desde cualquier lugar o mejoras en la comunicación entre estudiantes y profesores, entre muchas otras ventajas. Así mismo, el uso de las TIC ofrece gran cantidad de nuevas opciones en la enseñanza, siendo una de las principales la posibilidad de impartir clases no presenciales [6].

Dentro de las TIC, se establecen los Entornos Personales de Aprendizaje (PLE, del inglés *Personal Learning Environment*). Se trata de sistemas, que apoyándose en las nuevas tecnologías, facilitan a un alumno el aprendizaje con cierta autonomía, de modo que pueda gestionarlo según sus intereses. Estos sistemas no están ligados obligatoriamente al uso de la web y pueden ser una aplicación de escritorio [7].

## 2.3 SISTEMAS DE GESTIÓN DE APRENDIZAJE

Como evolución de los PLE, se encuentran los Sistemas de Gestión de Aprendizaje (LMS, del inglés *Learning Management System*). Estos se basan fundamentalmente en un uso mediante la web, utilizando todo el potencial de esta última para aprovechar las ventajas de las TIC anteriormente comentadas. Los LMS se basan en cuatro funcionalidades [8]:

- Gestión de profesores y alumnos: permiten organizar y manejar toda la información relativa a las distintas personas involucradas.
- Gestión de cursos y grupos: permiten establecer asignaturas, grupos de alumnos y cursos entre otras opciones para suministrar de la forma más organizada posible multitud de recursos electrónicos de distintos tipos.
- Comunicación: permiten la comunicación entre los distintos miembros de la comunidad educativa, la creación de foros de debate y encuestas o el envío de distintos tipos de información.
- Herramientas de evaluación: ofrecen mecanismos para la gestión y entrega de trabajos por parte de los alumnos y la posibilidad de evaluarlos de forma presencial o no presencial (funcionalidad en la que se enmarca el presente proyecto).

Estos sistemas se instalan en un servidor al que accederán los distintos usuarios mediante una web de acceso o mediante clientes para distintos tipos de dispositivos, como teléfonos inteligentes o tabletas.

En los siguientes puntos se describen los LMS más utilizados mundialmente [9].

### 2.3.1 BLACKBOARD<sup>1</sup>

Se trata de un LMS comercial muy extendido, numerosas universidades de todo el mundo cuentan con este sistema. En España, la Universidad de Alcalá o la Universidad de Zaragoza lo tienen implantado. Ofrece múltiples funcionalidades para la realización de cualquier tarea relacionada con el ámbito de la enseñanza. Es un sistema robusto, con una arquitectura que ofrece gran calidad y rendimiento. Como gran ventaja, su arquitectura es abierta, de forma que los usuarios pueden implementar nuevas extensiones para el entorno, opción conocida por *Extend Blackboardlearn* [10].

Sus funcionalidades incluyen las propias de cualquier LMS: gestión de usuarios con un control de roles, crear cursos y asociar alumnos a los mismos, compartir recursos electrónicos o establecer una vía de comunicación entre los distintos usuarios. Es el cuarto LMS más popular a nivel mundial [9].

---

<sup>1</sup> <http://www.blackboard.com/>

### 2.3.2 CONNECTEDU<sup>2</sup>

Está compuesto por un conjunto de herramientas centradas en la enseñanza. Una de estas se denomina Connect! y tiene como principal objetivo realizar un seguimiento de los alumnos durante un largo período de su formación. Esto permite generar un plan de futuro laboral y académico adaptado a cada alumno teniendo en cuenta sus aspiraciones.

Otra de las muchas herramientas que incluye es Epsilen Learning Management System, el Sistema de Gestión de Aprendizaje ofrecido por esta plataforma. Posee las características propias de cualquier otro LMS, entre las que se incluye la gestión de cursos o las comunicación entre estudiantes y profesores. Se encuentra en la tercera posición, según el grado de popularidad, a escala mundial [9].

### 2.3.3 EDMODO<sup>3</sup>

La principal característica de este LMS gratuito es su similitud a una red social, por ejemplo Facebook. Consiste en una única web accesible por cualquier persona. No necesita de una instalación en un servidor por parte del centro educativo que desee utilizarlo, así, su instalación es más sencilla comparada con otros LMS. Requiere de un registro por parte de los profesores, que pueden crear grupos de alumnos en la plataforma, obteniendo un código único que distribuirán a sus alumnos para que introduzcan en el sistema. Una vez establecido el grupo, este será privado, no siendo posible el acceso a su información por terceras personas [11].

En cuanto a sus funcionalidades, ofrece las comunes de cualquier LMS, sumando algunas de gran utilidad, como el acceso desde dispositivos móviles de la plataforma Android o iOS. También admite utilizar la herramienta WallWisher, que permite la realización de una lluvia de ideas, de forma online, por un grupo de personas [12]. Ha establecido la segunda posición como Sistema de Gestión de Aprendizaje más popular a nivel mundial [9].

## 2.4 MOODLE<sup>4</sup>

Moodle, término que proviene de *Modular Object Oriented Dynamic Learning Environment*, es el LMS más popular en todo el mundo [9]. Se basa en software libre bajo Licencia Pública GNU. Es un sistema que ofrece gran escalabilidad, estabilidad, capacidad de personalización, seguridad, con multitud de opciones de configuración y gran cantidad de funcionalidades. Sus inicios se remontan a 1999, de la mano de Martin Dougiamas.

---

<sup>2</sup> <http://www.connectedu.com/>

<sup>3</sup> <http://www.edmodo.com>

<sup>4</sup> <https://moodle.org/>



Para establecer Moodle en un centro educativo, es necesaria su instalación en un servidor. Este proceso es sencillo y no hacen falta grandes conocimientos informáticos para ello. Una vez establecido, ofrece una configuración paso a paso que permite lograr un sistema adecuado al entorno concreto. Para acceder al mismo, se necesita de un navegador web. En las últimas versiones existen aplicaciones para clientes de distintas plataformas, no obstante, los contenidos a los que pueden acceder están limitados.

La Universidad Carlos III de Madrid utiliza este LMS, con el nombre de Aula Global, como sistema para gestionar toda la comunicación entre profesores y alumnos, distribuir los contenidos de las asignaturas o realizar evaluaciones de los alumnos. La versión instalada en la actualidad es la 2.4.

Dentro de la plataforma, el contenido se organiza en cursos o asignaturas, donde se matricularán los alumnos. Dentro de estos cursos, Moodle permite insertar distintos tipos de actividades o recursos, como cuestionarios, foros, glosarios, tareas (para enviar trabajos los alumnos), referencias bibliográficas o documentos de diversos formatos.

Los cuestionarios de Moodle forman un recurso con múltiples posibilidades. Permiten realizar una evaluación de los alumnos, un control *in-situ* del seguimiento de los alumnos en una clase o realizar una encuesta personalizada. Además, permiten establecer unos criterios de calificación por parte del profesor, de forma que los cuestionarios se autocorregirán y se guardarán automáticamente las calificaciones de cada alumno.

Cuando se crea un cuestionario, Moodle ofrece una ventana con múltiples opciones de configuración. Por ejemplo, permite crear textos de retroalimentación, establecer las fechas de apertura y fin del cuestionario o fijar un límite de tiempo para resolverlo. También ofrece opciones de control de la presencia del alumno en clase, como limitar el rango de direcciones IP desde el que puede realizarse el cuestionario (útil para obligar a realizarlo únicamente desde una dirección accesible desde el centro educativo) o fijar una contraseña para poder acceder al cuestionario (la contraseña se proporciona dentro de la clases). Después, se podrán agregar distintos tipos de preguntas, como se puede apreciar en la Figura 2.

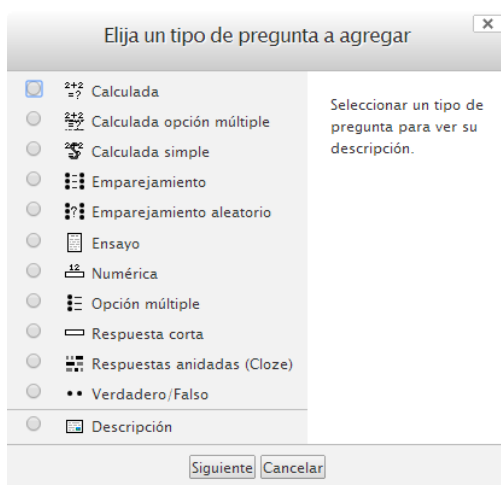


Figura 2 - Tipos de preguntas admitidas en cuestionarios de Moodle

## 2.4.1 CLIENTES DE MOODLE

Con el fin de conocer si los propósitos de este proyecto ya están cubiertos, total o parcialmente, se realiza un estudio de todas aquellas aplicaciones que guardan cierta similitud, las cuales deben ser consideradas antes de definir una solución. Existen muchos clientes para sistemas operativos como Android o iOS [13], pero como se irá comentando en cada una de ellas, ninguna se ajusta exactamente a los requerimientos.

### 2.4.1.1 MOODLE MOBILE

Esta aplicación es la ofrecida oficialmente por Moodle para el uso en dispositivos móviles. Ha sido desarrollada por Juan Leyva con el soporte de Jerome Mouneyrac y Moodle entre otros. Existen versiones tanto para Android como para iOS, funcionales con teléfonos inteligentes y con tabletas. También existen versiones para Symbian, webOS y Windows Phone, pero sin soporte oficial. Su código fuente es de libre acceso, se basa en HTML5 y requiere de la activación de algunos servicios web en el lado servidor.

Algunas de las funciones más destacadas de esta aplicación son [14]:

- Subir una imagen
- Grabar un archivo de audio y subirlo a la plataforma
- Enviar mensajes privados a otros compañeros
- Localizar a un compañero en Google Maps
- Descargar y ver archivos
- Cambiar el diseño de la aplicación de forma remota, a través de hojas de estilo CSS

La funcionalidad principal que atañe a este trabajo, la realización de cuestionarios, no está incluida en la aplicación oficial de Moodle, tal como se ha comprobado. Al intentar acceder a un test disponible en la plataforma Moodle, se abre automáticamente una ventana (Figura 3) de navegador que permite visualizar la página web de acceso, tal y como se muestra en cualquier navegador web, perdiendo las ventajas del uso de una aplicación dedicada.

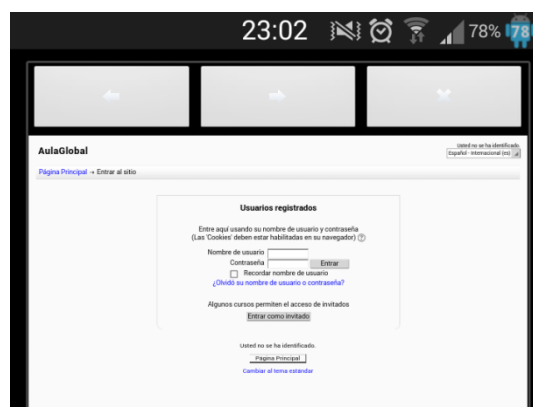


Figura 3 - Realización de test con Moodle Mobile App

Intentando acceder a Aula Global con Moodle Mobile, esta aplicación informa de que los servicios web no están habilitados, por lo que la aplicación no podrá conectarse a Moodle y recibir información.

### 2.4.1.2 mTOUCH, mTOUCH+, MOODLEZ Y mTOUCH-U

Son aplicaciones para teléfonos inteligentes y tabletas destinadas al manejo de parte de las funcionalidades de Moodle. Han sido desarrolladas por el equipo PragmaTouch, que está unido a la Universidad Bilgi de Estambul. Tanto mTouch, mTouch+ como moodIEZ, son aplicaciones de pago disponibles únicamente para el sistema operativo iOS. mTouch-U es una aplicación web gratuita [15].

mTouch es la aplicación destinada a iPhone y a iPod Touch, mientras que mTouch+ es la correspondiente para iPad. Ambas proporcionan únicamente servicios para estudiantes, no pudiendo realizarse tareas propias de los profesores y administradores, que quedan reservadas para moodIEZ. Esta aplicación existe únicamente en versión para iPad e incluye también funciones para estudiantes.

Al igual que otras aplicaciones, dispone de un campo en la pantalla de inicio para introducir la dirección web donde se encuentra instalada la plataforma Moodle a la que se quiere realizar la conexión. Al iniciar sesión, se muestra una pantalla con las asignaturas. De aquí surgen dos problemáticas de esta aplicación. Por un lado, los nombres de los cursos aparecen en lengua inglesa, no siendo posible el cambio de idioma. Por otro lado, los cursos que se muestran no son únicamente los actuales, sino que también aparecen asignaturas ya superadas, ofreciendo una discordancia entre el contenido de Aula Global y el de la aplicación.

Respecto a la funcionalidad de realizar cuestionarios, mTouch-U no está preparada para ello, mostrando un mensaje de error (Figura 4) en caso de que se intente, que a la vez proporciona un enlace a la página concreta Aula Global en la que se posiciona el test, para poder acceder con un navegador. En el caso de mTouch sucede lo mismo, no permite responderlos desde la propia aplicación. Además, esta última aplicación es de pago.

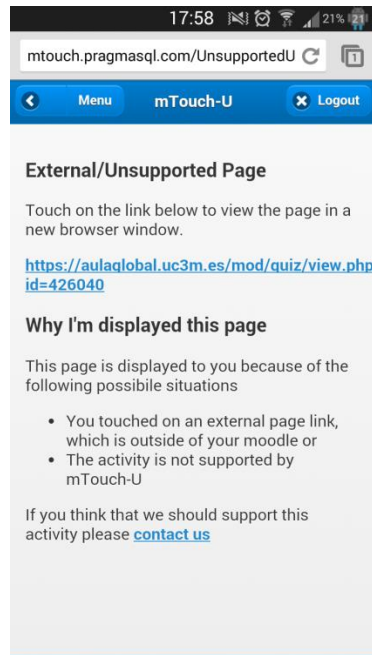


Figura 4 - Realización de cuestionarios en mTouch

Esta aplicación contiene diversas funcionalidades, pero no cubre la que se trata en este trabajo. Además, el código fuente de esta aplicación no es abierto, impidiendo la adaptación de la misma para lograr los objetivos planteados

### 2.4.1.3 MPAGE

Creada por Mass Media HK, es una de las antecesoras de Moodle Mobile y actualmente está desfasada. Inicialmente creada para iOS, más tarde salió la versión para Android de tipo híbrida (combinación de HTML5 y aplicación nativa). Para poder ser utilizada, debía instalarse una extensión junto con Moodle en el lado servidor. El uso de esta extensión permitía reducir la información a enviar entre el dispositivo y el servidor, acelerando este proceso [16].

### 2.4.1.4 MY MOODLE

Esta aplicación fue la anterior a Moodle Mobile, y ya está obsoleta. Estaba desarrollada como aplicación nativa para iOS e, igual que la actual, utilizaba también los servicios web como forma de intercambiar información con el servidor. Entre otras funcionalidades, permitía el intercambio de mensajes entre participantes de las asignaturas o dejar almacenado contenido en el dispositivo para verlo de forma *offline* [17].

### 2.4.1.5 MOODBILE

Es un proyecto de código abierto iniciado por SUSHITOS Research Group, perteneciente a la Universitat Politècnica de Catalunya. Requiere de la instalación de unos servicios web especiales en el servidor, para permitir la recuperación y el envío de datos desde el lado cliente. Para este, existen tres aplicaciones: un cliente HTML5, ejecutable desde un navegador web y un cliente nativo para Android. Sin embargo, desde el año 2012 no se han desarrollado nuevas versiones para esta aplicación [18]. Actualmente no se puede acceder a la web oficial [19] y, por tanto, a la descarga de la aplicación o del código fuente, imposibilitando su uso.

### 2.4.1.6 OTRAS APLICACIONES

Existen más aplicaciones destinadas al acceso, desde distintos dispositivos, a la plataforma Moodle. UC3M – Aula Global es la aplicación oficial que ofrece la universidad para el acceso a Aula Global a través de un dispositivo Android o iOS. Permite diversas acciones, como descargar documentos o acceso a enlaces, pero no permite la realización de cuestionarios.

Otros ejemplos son Mbot [20], que se encuentra desfasado, Moody - Moodle Web Client [21], que no contempla la realización de cuestionarios o iActive [22], sin versión disponible para la plataforma Android.

## 2.4.2 COMPARACIÓN DE LOS CLIENTES DE MOODLE

En la Tabla 1 se realiza una comparación entre los distintos clientes de Moodle, indicando si proporcionan las principales funcionalidades que se esperan de ellos y que afectan a este proyecto.

	Actualizada	Software libre	Gratuita	Versiones para distintos SO	Realización de cuestionarios directamente (sin navegador web)
Moodle Mobile	Si	Si	Si	Si	No
mTouch	Si	No	No	No	No
mPage	No	No	Si	Si	No
My Moodle	No	Si	Si	No	No
Moodbile	No	Si	Si	Si	No

Tabla 1 - Comparación de los clientes de Moodle

Como se puede apreciar en la tabla superior, ninguna de las aplicaciones analizadas se ajusta a los objetivos planteados en este proyecto. Además de no permitir responder cuestionarios directamente (no a través de un navegador que abra la página web de Moodle) tienen otros inconvenientes, como estar obsoletas o no ser software libre.

### 2.4.3 SERVICIOS WEB DE MOODLE

A partir de la versión 2.0, Moodle incorpora una nueva funcionalidad, los *Web Services*. Una de las principales ventajas de permitir la conexión a la plataforma desde otras vías, además de un navegador web, es poder crear aplicaciones para interactuar con la plataforma adaptadas a las peculiaridades de cada dispositivo. Así, podremos tener una aplicación para un teléfono inteligente con la que se pueda acceder a Moodle, pero adaptada a la forma de manejo de este tipo de dispositivo. Moodle incluye por defecto unos servicios web con una serie de funciones ya implementadas [23]. Entre todas ellas, no hay ninguna relacionada con la obtención de cuestionarios.

Moodle permite la instalación de *plug-ins* o extensiones, las cuales constituyen la definición e implementación de una serie de funciones, para ser utilizadas como servicios web [24]. Una vez incorporado al servidor donde resida la instalación de Moodle, este lo reconocerá automáticamente y realizará una instalación para hacerlo disponible desde otros servicios.

Para realizar una llamada a una de las funciones definidas en los servicios web de Moodle, ya sean los implementados por defecto o los añadidos mediante una extensión, es necesario suministrar un *token*. Estas claves se asocian a un usuario de Moodle para poder utilizar un servicio concreto. El administrador de la plataforma puede configurar qué usuarios pueden disponer de *token* y para qué servicio concreto. Los usuarios podrán ver en los ajustes de su perfil los *tokens* o claves de seguridad que puede usar.

## 2.5 mSUITE

mSuite está formado por un conjunto de aplicaciones desarrollado por el grupo de investigación miSEL, que pertenece al Software Engineering Lab de la Universidad Carlos III de Madrid. mSuite tiene como principal objetivo proporcionar un mecanismo integrado en todo lo relativo a la impartición de clases, permitiendo planificar las clases, impartirlas con la ayuda de redes sociales para el envío de información instantánea a los alumnos y la evaluación de estos últimos (mediante la realización de preguntas de respuesta simple). Estas aplicaciones se encuentran separadas de Aula Global, no existiendo un vínculo entre ambas. Se deben definir los dos roles que participan en este proceso:

- Profesor: planifica las clases subiendo material a la aplicación, como un PDF o un vídeo de Youtube, imparte posteriormente las clases y recupera información de los alumnos mediante preguntas.
- Alumno: asiste a la impartición de una clase, recibe información en tiempo real mediante redes sociales y contesta preguntas sobre el contenido de la clase.

Las aplicaciones que conforman mSuite están interconectadas mediante varios servidores, dos de ellos propios de mSuite y a través de los servicios almacenamiento de Dropbox, concretamente Dropbox Datastore y Dropbox Chooser. Los servidores propios son los denominados SelData y SELQuizServer.

Para comprender el funcionamiento de mSuite, se detallan las aplicaciones que la conforman, explicando más en detalle las acciones que ejecuta cada rol definido (profesor o alumno).

### 2.5.1 MTEACHPLAN

Aplicación únicamente utilizada por el profesor, con el objetivo de planificar las clases. Permite crear cursos (asignaturas) en los que introducir sesiones (que se corresponderán con las distintas clases impartidas a lo largo del curso). Dentro de estas sesiones se pueden crear preguntas para ser enviadas a los alumnos en el transcurso de la clase y que las contesten desde sus respectivos dispositivos con la aplicación mClassTest.

Estas preguntas se podrán agrupar también en conceptos, para una mejor organización de las mismas. Dentro de las sesiones también se pueden crear *tweets*, para tenerlos preparados y que puedan ser enviados en el transcurso de la clase.



Figura 5 - Pantalla principal de mTeachPlan

mTeachPlan también permite la visualización de resultados de las preguntas realizadas por los alumnos. Para que esta aplicación pueda operar conjuntamente con el resto de aplicaciones de mSuite, es necesaria su conexión con Dropbox y con el servidor SelData.

Por último a comentar sobre esta aplicación, existen versiones para Android y para iOS. Esta última fue desarrollada por Roberto Estrada Casarrubios en su Proyecto Fin de Carrera, presentado el 31 de Octubre de 2013.

## 2.5.2 mTEACHCLASS

Esta aplicación es la utilizada por el profesor para impartir las clases, estando pensada para que el dispositivo en el que se ejecute se conecte al proyector del aula y así mostrar el contenido (diapositivas o vídeos) directamente desde la aplicación. Permite la publicación de *tweets* en la cuenta asignada a la asignatura y también lanzar preguntas que serán respondidas por los alumnos. Tanto los *tweets* como las preguntas han sido previamente creados en la aplicación mTeachPlan. Esta aplicación también permite la visualización de resultados de las preguntas realizadas durante la propia clase.



Figura 6 - Pantalla principal de la aplicación mTeachClass

Al igual que la anterior aplicación, que permitía planificar las clases, mTeachClass también se encuentra disponible tanto para Android como para iOS. Esta aplicación también se conecta a los servicios de almacenamiento de Dropbox, al servidor SelData y también al SELQuizServer, que será el vínculo con las aplicaciones de los alumnos, mClassTest.

## 2.5.3 mCLASSTEST

Es la aplicación de mSuite utilizada por los alumnos para responder preguntas. Para utilizar la aplicación es necesario iniciar sesión con Twitter. El principal motivo del uso de esta red social es la apertura del conjunto de aplicaciones a ser utilizadas en los cursos en línea masivos y abiertos (MOOC), permitiendo un sistema de inicio no ligado estrictamente al centro que utilice mSuite.



Una vez identificado, la aplicación permite suscribirse a las sesiones que se encuentren disponibles y así contestar a las preguntas ya disponibles o que se vayan a lanzar durante una clase.

Esta aplicación se encuentra disponible para iOS en código nativo y en Android como aplicación híbrida en HTML5, por lo que la funcionalidad de esta última se encuentra muy limitada. En la versión de iOS, tiene la posibilidad de recibir notificaciones cuando hay nuevas preguntas disponibles.

La aplicación mClassTest, independientemente de si se trata de la versión para iOS o para Android, se conecta con el resto de elementos de mSuite mediante el servidor SELQuizServer. Es el punto de unión de estas aplicaciones con el resto de sistema, por lo que todo el envío y recepción de información de mClassTest se produce a través de este servidor.



Figura 7 - mClassTest para Android en HTML5

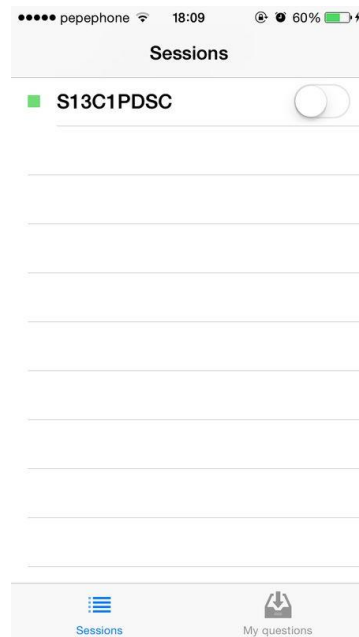


Figura 8 - mClassTest para iOS

## 2.6 PLATAFORMAS DE DESARROLLO PARA DISPOSITIVO MÓVIL

El uso de teléfonos inteligentes en la población sigue un camino en auge, cosechando año tras año cifras cada vez mayores. Por ejemplo, en el año 2013, ha sido el dispositivo más utilizado para acceder a Internet por los jóvenes entre 20 y 29 años, con una tasa del 96%, frente a un 88% de dispositivos fijos [25].

Dentro de estos dispositivos, existen diversos sistemas operativos. En el siguiente gráfico, se muestra la cuota de mercado de cada plataforma en España en el año 2013 [26]:

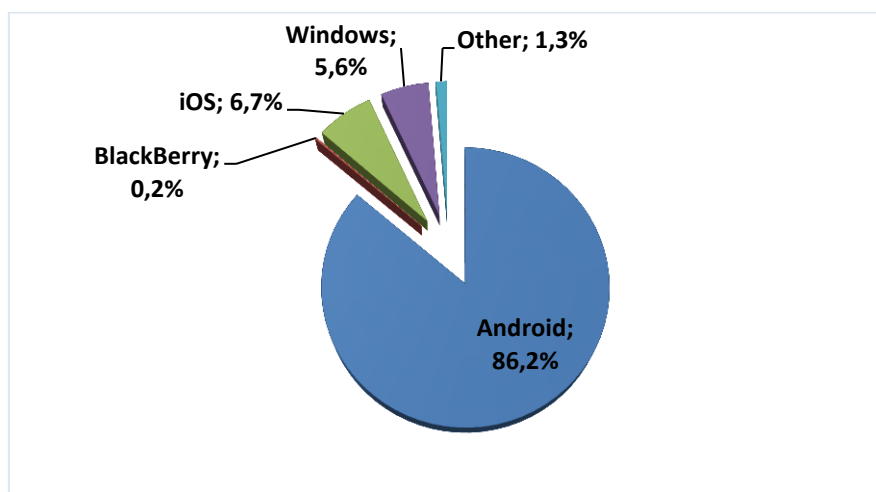


Figura 9 - Gráfico de cuotas de mercado de las plataformas móviles en España

Como se puede apreciar en la Figura 9, la cuota líder es para la plataforma Android, con un 86,2% de dispositivos vendidos. La ventaja respecto a sus competidores es muy elevada, no obstante, otras plataformas tienen unas tasas importantes.

A continuación se describen las tres plataformas más utilizadas: Windows Phone, iOS y Android.

### 2.6.1 WINDOWS PHONE

Fue presentado en Febrero de 2010 de la mano de Microsoft. Es software propietario. Su objetivo era ofrecer un sistema operativo para dispositivos móviles de diferentes fabricantes basado en el sistema operativo de la misma compañía para ordenadores personales Windows. Se encuentra en constante crecimiento en lo que a número de usuarios se refiere, alcanzando en 2013 una cuota de mercado del 5.6%.

Ofrece un entorno de desarrollo, SDK Visual Studio, que permite desarrollar aplicaciones tanto en Visual Basic como en C#. Este SDK incluye un emulador para probar la aplicación desarrollada. A partir de la versión Windows Phone 8, Microsoft ha definido como requisito tener un ordenador con Windows 8 para poder desarrollar, no siendo posible en Mac.

Entre las recientes novedades que incluye en su última versión Windows Phone 8.1, se encuentra la inclusión del asistente de voz Cortana o la mejora de la experiencia con la multitarea, reduciendo el tiempo de cambio entre aplicaciones.

### 2.6.2 IOS

Es el sistema operativo para dispositivos móviles propiedad de Apple. Es un sistema propietario que únicamente se encuentra en dispositivos de la propia marca, no siendo posible su instalación en dispositivos de terceros. Sus inicios se remontan al año 2007. En España, mantiene una cuota de mercado del 6,7%, cifra muy alejada del 49,7% que posee en Estados Unidos (en el año 2013). Este sistema es utilizado por diversos dispositivos por la compañía, como el iPhone o el iPad.

Para su desarrollo es necesario un Mac con el sistema operativo de la misma compañía Mac OS, estando imposibilitado el desarrollo en otros sistemas operativos. Apple proporciona un entorno de desarrollo gratuito, X-Code, que cuenta con emuladores de los diferentes dispositivos. Para probar la aplicación en un dispositivo físico o para publicar aplicaciones en su tienda de aplicaciones App Store, es necesario poseer una licencia de desarrollador de Apple, la cual es de pago. El desarrollo se lleva a cabo en el lenguaje de programación Objective C.

### 2.6.3 ANDROID

Es el sistema operativo más utilizado mundialmente. Originalmente desarrollado por Android Inc, años más tarde fue comprado por Google. Se trata de una plataforma totalmente abierta, de forma que puede ser modificada y adaptada. Puede ser utilizado por cualquier fabricante de dispositivos móviles, ya sean teléfonos inteligentes, tabletas o incluso cámaras de fotos o electrodomésticos. Multitud de dispositivos con diversas características y precios funcionan con Android. También permite portar las aplicaciones implementadas a muchas otras plataformas con gran facilidad, debido a que se desarrollan en Java.

Para el desarrollo de aplicaciones para esta plataforma, Google proporciona un SDK (Software Development Kit) que se basa en el entorno de desarrollo Eclipse, adaptado a las peculiaridades de Android. Ofrece, al igual que el resto de plataformas comentadas, un emulador para probar las aplicaciones. Las aplicaciones también se pueden desarrollar en C o en C++, con el Android NDK (Native Development Kit).

Para la mayoría de aplicaciones desarrolladas para Android, la herramienta más adecuada a utilizar es el Android SDK, puesto que ofrece una mayor abstracción y requiere de un nivel de conocimiento de programación menor. Por el contrario, Android NDK posibilita tener un mayor control sobre el hardware, desarrollando así aplicaciones más complejas.

Una de las características clave de Android es que las aplicaciones, programadas en Java, se ejecutan en una máquina virtual llamada Dalvik. Esto permite una adaptación de la aplicación al dispositivo para maximizar el rendimiento, lograr un buen funcionamiento y consumir la menor energía posible.

Su arquitectura está formada por cuatro capas, todas ellas de software libre [27]. En el nivel con mayor abstracción se encontraría la capa de aplicaciones, donde se encuentran las distintas aplicaciones que se ejecutan en el dispositivo. En el siguiente nivel se encuentra el entorno de aplicaciones, al que recurren las aplicaciones para acceder a las distintas funcionalidades del dispositivo. A continuación se encuentran las librerías nativas (por ejemplo, las dedicadas a los gráficos o a bases de datos) y el Runtime de Android (donde se ejecuta la mencionada máquina virtual Dalvik). En último lugar se encuentra el núcleo (Linux), que es el que accede directamente al hardware.

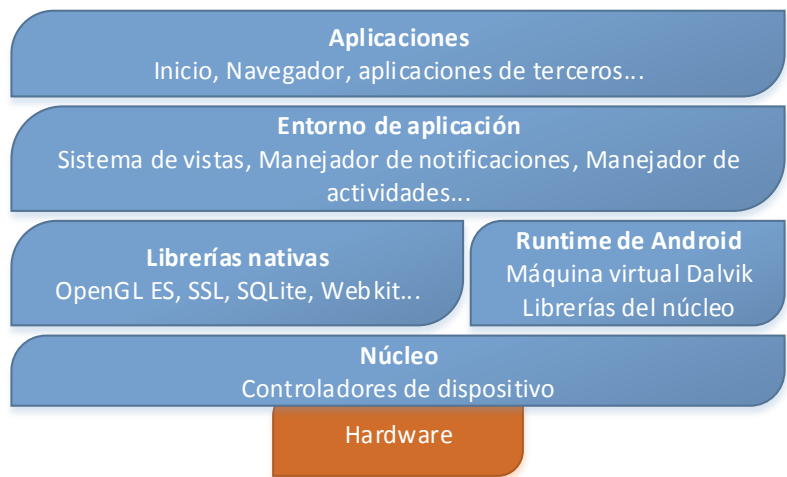


Figura 10 - Capas en la arquitectura de Android

Existen diferentes versiones de Android. Cuando se va a desarrollar una aplicación para este sistema, se debe tener en cuenta la versión mínima a partir de la cual será compatible. Esta decisión dependerá del número de dispositivos a los que se quiera llegar y de las nuevas funcionalidades que incluya cada versión.

Versión	Nombre	API	Distribución
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%

Tabla 2 - Distribución de versiones de Android [28]

La versión 2.3.3 es la primera, como se puede apreciar en la tabla, en sumar un importante porcentaje. Si se suman todas las versiones a partir de esa se obtiene un 99,2%, por lo que desarrollar para esta versión asegurará su funcionamiento en una amplia mayoría de dispositivos.

### 2.6.4 COMPARACIÓN DE LAS PLATAFORMAS DE DESARROLLO

A continuación se muestra una tabla con las principales diferencias entre las 3 plataformas enunciadas. Se utilizarán las variables que pueden ayudar a establecer cuál es la mejor plataforma para el desarrollo del presente proyecto.

	Windows Phone	iOS	Android
Dispositivos compatibles	De diversos fabricantes, pero cumpliendo ciertos requisitos mínimos	Únicamente en dispositivos compatibles de la marca Apple	Multitud de dispositivos de características muy variadas
Sistema operativo de desarrollo	Windows	Mac OS	Windows/Mac OS/Linux
Núcleo del sistema operativo	Windows NT	Mac OS	Linux
Documentación existente	Media	Muy elevada	Muy elevada
Seguridad	Muy elevada	Muy elevada	Media
Cuota de mercado (España)	5,6%	6,7%	86,2%
Lenguaje de programación	Visual Basic C#	Objective C	Java C
Licencia de Software	Propietaria	Propietaria	Libre

Tabla 3 - Comparación de plataformas móviles

Analizando la Tabla 3, Android es la plataforma que mejores datos aporta: la mayor cuota de mercado con gran distancia sobre las demás plataformas, se basa en software libre, posibilita el desarrollo en ordenadores con distintos sistemas operativos y existen multitud de dispositivos con este sistema instalado (lo que conlleva que existan terminales de precios muy variados). En contra, tiene una seguridad inferior, que se debe principalmente a un bajo control por parte de Google de las aplicaciones disponibles en su tienda (Google Play).

No solamente se ha escogido Android como plataforma de desarrollo por las cualidades comentadas en el primer párrafo, también se ha escogido por la ya existencia de un cliente para iOS. Se explicará este punto en la solución adoptada con mayor detalle.

### 2.6.5 APLICACIONES NATIVAS, HÍBRIDAS Y WEB

Existen tres tipos de aplicaciones que se pueden desarrollar para las distintas plataformas móviles.

Las aplicaciones web son aquellas desarrolladas en HTML5 y que se ejecutan a través de un navegador web. Para acceder a ellas se procede de la misma forma por la que accedemos a cualquier página web. Su ejecución se realiza en el servidor donde se alberga, dejando al terminal la visualización.

También pueden estar escritas conjuntamente en el lenguaje propio de un sistema operativo y en HTML5 (aplicaciones híbridas), y es el caso de Moodle Mobile. La parte esencial de la aplicación está desarrollada en HTML5, visualizándose del mismo modo que se realiza en las aplicaciones web, pero la ejecución en las híbridas recae sobre el propio dispositivo. En el código del sistema operativo se escribe únicamente el instalador y el inicio de la aplicación (que carga una ventana de un navegador web). Ayudándose del *framework Phonegap*, que se encuentra disponible para distintas plataformas, entre ellas iOS, Android y Windows Phone, estas aplicaciones consiguen acceder a funciones del terminal como una cámara o a un acelerómetro [29]. Esta combinación permite la ejecución de la aplicación aunque no se disponga de conexión a Internet.

Por otro lado están las aplicaciones nativas, que son desarrolladas completamente en el lenguaje de programación asociado al sistema operativo, por lo que su instalación se restringe a aquel sistema para el que se ha diseñado. Permiten aprovechar al máximo el potencial del dispositivo, pudiendo utilizar todas sus funciones y escribiendo código adaptado al sistema operativo para lograr la máxima eficiencia y eficacia.

El uso de HTML5 tiene tanto ventajas como desventajas. En los puntos a favor de esta tecnología destaca la versatilidad. Permite fácilmente su exportación a distintas plataformas (como Android e iOS) compartiendo el código HTML (en el caso de las aplicaciones híbridas, varía la porción de código que no es HTML5 y que está ligada al sistema operativo) y abaratando en gran medida el coste de desarrollo, ya que esta tecnología tiene menor complejidad que un lenguaje de programación como Java.

En contra del uso de esta vía de desarrollo de aplicaciones, está su menor rendimiento respecto a una aplicación nativa, al no estar programada y pensada para adaptarse a un sistema operativo concreto. Tampoco poseen la misma calidad, variedad de gráficos, ni tienen acceso a todas las funcionalidades que ofrece un sistema operativo.

	Aplicaciones nativas	Aplicaciones híbridas	Aplicaciones web
Integración con el hardware y sistema operativo	Elevada	Media	Baja
Facilidad de desarrollo	Baja	Media	Alta
Funcionamiento sin conexión a Internet	Si	Si	No
Rendimiento	Elevado	Medio	Bajo
Compatibilidad entre sistemas operativos	No	No	Si
Calidad de gráficos	Elevada	Baja	Baja

Tabla 4 - Comparativa de aplicaciones nativas, híbridas y web

En la Tabla 4 se muestra una comparativa entre los tres tipos de aplicaciones existentes. La elección para desarrollar una aplicación dependerá de sus características y objetivos. Por ejemplo, si se busca la solución más económica y que sea compatible con el mayor número posible de dispositivos, la opción más adecuada sería una aplicación web. Por el contrario, si lo que se busca es una aplicación de calidad y en la que se utilicen recursos del sistema (como por ejemplo la cámara), la opción más indicada será una aplicación nativa.

### 2.6.6 ENTORNO DE DESARROLLO ANDROID SDK

El SDK de Android ofrece un conjunto de herramientas para llevar a cabo la implementación de una aplicación. Los elementos esenciales que lo forman son los siguientes:

- Eclipse: es un poderoso entorno de desarrollo de código abierto para múltiples lenguajes, donde el mayoritariamente utilizado es Java. Existen diferentes complementos para aumentar sus funcionalidades.
- ADT Plugin: es el complemento brindado por Google para Eclipse con el objetivo de adaptarlo al desarrollo de una aplicación Android. Contiene las herramientas necesarias para construir una interfaz gráfica, permite depurar las aplicaciones o generar un archivo ejecutable para ser instalado en un dispositivo o ser distribuida.
- SDK Tools: es un conjunto de herramientas para el desarrollo y depuración de aplicaciones Android.
- Android Emulator: es un emulador de dispositivos para ejecutar la aplicación en desarrollo. Permite configurar múltiples opciones del dispositivo tanto a nivel de hardware (procesador, memoria RAM...) como de opciones de software (la versión de Android).
- Documentación y ejemplos: este entorno integra abundantes recursos para facilitar el proceso de creación de una aplicación.



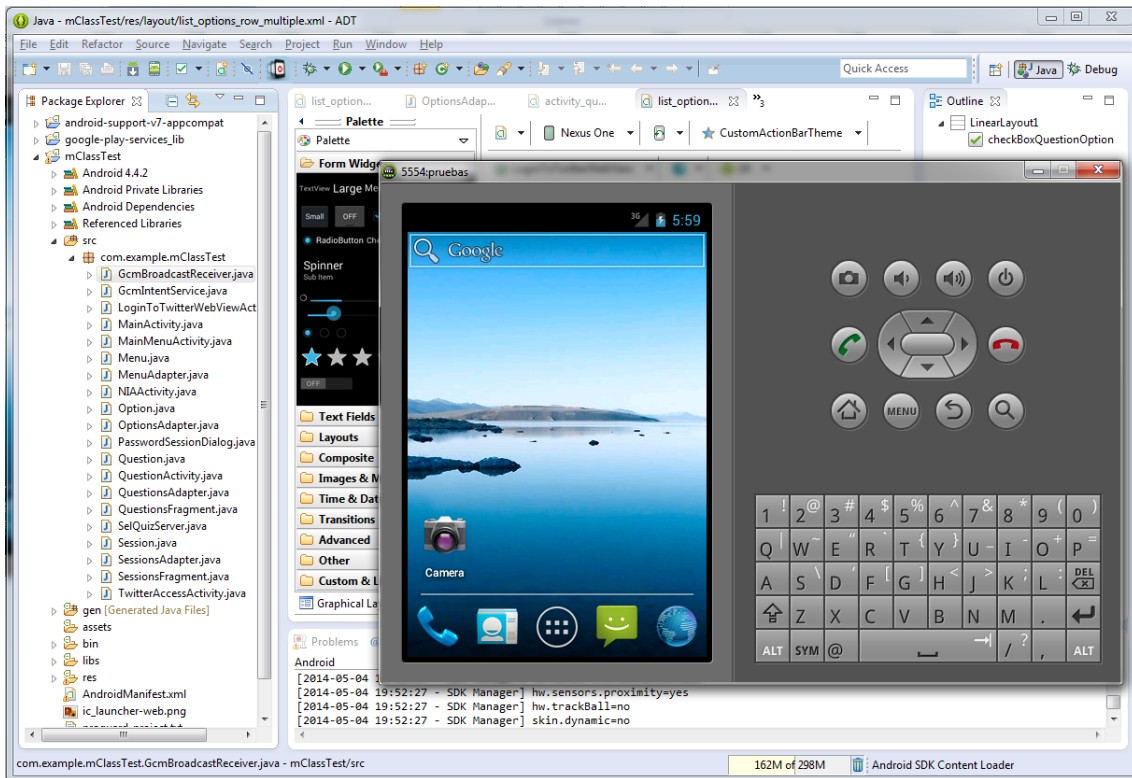


Figura 11 - Entorno de desarrollo Android SDK

## 2.7 PLAY FRAMEWORK

Es un marco de trabajo para el desarrollo de aplicaciones web. Su principal objetivo es facilitar este proceso. Trabaja en los lenguajes Java y Scala y permite crear una aplicación web sin trabajar en HTML, aunque el resultado estará totalmente integrado con dicha tecnología. Fue creado en el 2007 como un proyecto privado de Zenexity y se hizo público en el 2009 como software libre.

Entre las características que hacen de este sistema una importante opción para el desarrollo de aplicaciones web, se encuentra la abstracción que proporciona o la integración de servicios que ofrece, como es el de almacenamiento de datos.

## 3. ANÁLISIS DE LA SOLUCIÓN

En este apartado se procede a realizar una descripción de la solución adoptada. Se detallarán las distintas fases por las que ha ido evolucionando hasta llegar a la solución final. A continuación se realizará una especificación de casos de uso y requisitos. Finalmente, se aporta una matriz de trazabilidad entre los requisitos software y los casos de uso.

### 3.1 DESCRIPCIÓN DE LA SOLUCIÓN

Inicialmente, se elaboró una primera solución al problema planteado. Sin embargo, una vez se analizó ésta en detalle, se encontraron una serie de problemas. Con sucesivas propuestas, se llegó a una solución final viable. En los siguientes puntos se exponen los pasos dados hasta obtener la solución final.

#### 3.1.1 PROPUESTA INICIAL

La solución inicialmente planteada consistía en desarrollar únicamente una aplicación, para la plataforma Android, que conectara con Moodle para obtener los cuestionarios y que permitiera al alumno su realización durante una sesión lectiva. Se dotaría a la aplicación de una funcionalidad para controlar la presencia del alumno en el aula de la asignatura en cuestión.

La aplicación desarrollada realizaría una conexión directa con Moodle, sin ningún punto intermedio. Accedería al LMS para obtener los cuestionarios y, una vez contestados por el alumno en su dispositivo, este devolvería las respuestas correspondientes. En esta solución, la aplicación permitiría hacer el equivalente a que un alumno entre en la página web de Moodle y conteste desde ella al cuestionario planteado.

#### 3.1.2 PROBLEMÁTICAS Y CAMBIOS EN LA PROPUESTA INICIAL

Fruto de un trabajo de búsqueda de información para analizar el estado del arte sobre la plataforma Moodle, sus posibilidades, qué restricciones podrían afectar a la solución inicial, cómo se afrontaría una conexión para intercambiar datos y otros sistemas existentes relacionados con los mismos objetivos, se tuvo que modificar la propuesta inicial. En las siguientes líneas se explican cuáles fueron todos estos problemas encontrados y qué soluciones se escogieron.

En primer lugar, para acceder desde una aplicación externa a una plataforma como Moodle, ubicada en un servidor, se necesita un conjunto de servicios web. Este PLE implementa por defecto una serie de servicios para enviar y recibir ciertos datos desde sistemas externos conectados en red. Entre todas las funciones existentes, no existen las necesarias para manipular cuestionarios, primer motivo por el cual fue necesario modificar la solución inicial.

Para solucionar esto, era necesario incluir nuevas funciones en Moodle que permitieran manejar los cuestionarios mediante la creación de una extensión, que una vez insertada en un determinado directorio del servidor, será reconocida e instalada de forma automática.

Sin embargo, se detectó otro problema, relacionado con el envío de notificaciones. Para poder implementar este servicio, sería necesario que Moodle, al detectar la creación de un nuevo cuestionario o la apertura de la fecha para ser respondido por los alumnos, conectara con un servicio de notificaciones. Debido a que el envío de una notificación requiere del identificador (*token*) del dispositivo del alumno, obligaría a almacenar este en la base de datos del LMS. Si a esto se le suma el aumento de carga al servidor de Moodle que conllevaría gestionar una serie de llamadas al servicio de notificaciones desde la extensión, la solución se vuelve compleja e insegura. Por lo tanto, queda descartada.

El mejor modo a proceder para solventar los anteriores problemas comentados, sería establecer un servidor intermedio entre Moodle y la aplicación del alumno. Con ello se evitaría realizar cualquier modificación en la base de datos de Moodle y cualquier posible sobrecarga en su servidor debido a la inclusión de un servicio de notificaciones.

Por otro lado, se descubrió el conjunto de aplicaciones mSuite que posee el departamento miSEL de la Universidad. Tras una serie de conversaciones con su responsable Javier García Guzmán, se llegó a la conclusión de que una unión de mSuite con los objetivos de este proyecto no solo era viable, sino que permitiría aumentar la funcionalidad de un sistema ya en funcionamiento y a la vez solucionar los problemas planteados a la solución inicial anteriormente descritos. A continuación se describen en detalle las razones de esta unión:

- Unificación de aplicaciones: tener el máximo conjunto de funcionalidades en una única aplicación para el alumno, de forma que no tenga que instalarse varias aplicaciones en su dispositivo. Además, se favorece la publicidad de este sistema y por consiguiente un mayor alcance.
- Integración con Moodle: siendo un PLE ampliamente utilizado y el que rige la mayor parte de la actividad lectiva en línea de la Universidad Carlos III de Madrid, con este trabajo se da un importante paso en una integración total de Moodle con el conjunto de aplicaciones mSuite.
- Integración de aplicaciones: mSuite cuenta con aplicaciones ya desarrolladas para impartir clases o prepararlas.

- Infraestructura ya existente: existen varios servidores con bases de datos ya creadas, siendo reducidos los cambios necesarios para adaptar el sistema a los nuevos propósitos.

### 3.1.3 DESCRIPCIÓN DE LA SOLUCIÓN FINAL

La solución finalmente adoptada se concreta en tres puntos que marcarán los principales pilares de construcción del proyecto:

- Creación de una extensión para Moodle: se implementará una nueva extensión o *plug-in* que podrá ser instalado junto cualquier instalación de Moodle. Este complemento permitirá lo siguiente:
  - Recuperar un listado de estudiantes de un curso: para tener una relación entre los usuarios de la aplicación para su dispositivo móvil y los alumnos de la Universidad. De esta forma se podrán generar ficheros con las calificaciones y datos de los alumnos (como el NIA) que serán subidos por los profesores a Moodle.
  - Recuperar un listado de cursos de un profesor: se podrán obtener los cursos, con información sobre ellos, para poder acceder posteriormente a un curso concreto y descargarse sus cuestionarios.
  - Recuperar un listado de cuestionarios: se podrá recuperar información sobre los cuestionarios, con sus preguntas y respuestas, que tiene creados un profesor en un curso. Por un lado, se obtendrán datos globales del cuestionario, como por ejemplo la contraseña establecida para responder el cuestionario si la hubiera, las fechas de apertura y cierre del cuestionario o el título del cuestionario. Por otro lado, se obtendrán datos de las preguntas que contiene, como el texto de la pregunta, detalles sobre método de calificación o las respuestas posibles y correctas.
- Desarrollo de una aplicación para la plataforma Android. Esta sustituirá a la actual aplicación híbrida para la misma plataforma escrita en HTML5, denominada mClassTest. Debido a la incompatibilidad de lenguajes y a la adición de nuevas funcionalidades, no se podrá reutilizar código de la anterior. Las funcionalidades que cubrirá esta nueva aplicación serán las siguientes:
  - Las que ya tenía la aplicación mClassTest: iniciar sesión mediante Twitter, capacidad para suscribirse y cancelar la suscripción a una sesión, responder preguntas de respuesta simple y sincronizarse con el SELQuizServer.
  - Implementar un servicio de notificaciones para la recepción de avisos sobre nuevas preguntas o sesiones.
  - Posibilitar la respuesta de preguntas de respuesta múltiple.

- Solicitar contraseña para poder suscribirse a una sesión.
- Solicitar y almacenar el NIA del estudiante.
- Modificación del SELQuizServer. Este es el servidor de mSuite al que actualmente se conecta la aplicación mClassTest, tanto en su versión para Android (en HTML5) como en su versión para iOS y también la aplicación del profesor mTeachClass. Los cambios consistirán en ampliar su funcionalidad y su base de datos para admitir preguntas de respuesta múltiple, la incorporación del NIA en los datos del alumno en la base de datos, para así tener un identificador del alumno compatible con Moodle, y establecer un control por contraseña para poder suscribirse a una sesión.

Estos tres hitos no completan todo el trabajo necesario para hacer funcional el sistema global, dado que parte de él está siendo desarrollado paralelamente por otro alumno en su Trabajo Fin de Grado. La unión de estos dos proyectos hará posible tanto la recuperación de los cuestionarios de Moodle para incorporarlos a mSuite, su posterior respuesta por aplicaciones de las plataformas iOS y Android y la preparación de ficheros para incorporar manualmente a Moodle las calificaciones obtenidas por los alumnos. En los siguientes puntos se muestran, en líneas generales, las principales tareas que abordará el otro trabajo:

- Modificar la aplicación mTeachPlan (la que utiliza el profesor para preparar sus clases) para realizar una conexión con Moodle, mediante la extensión desarrollada, y así poder recuperar los cuestionarios que tenga creados el profesor en sus cursos.
- Modificar la aplicación mTeachPlan para generar los ficheros con las calificaciones de los alumnos para exportarlas a Moodle. Si se realizara este proceso mediante la extensión, esta debería realizar accesos a la base de datos del LMS para incorporar las calificaciones. La estructura de su base de datos es muy compleja, por lo que una modificación en ella podría causar inconsistencia en los datos, problemas de integridad o de sincronización, además de seguridad. Sin embargo, existe otra vía para incorporar estos datos de forma más sencilla y sin asumir los mencionados riesgos. Moodle permite la subida de ficheros, de forma manual por el profesor y en un determinado formato, con calificaciones de los alumnos.
- Modificar mTeachPlan para permitir crear sesiones protegidas por contraseña.
- Modificar la aplicación mClassTest para iOS de forma que admita preguntas de respuesta múltiple, incluya el acceso a sesiones protegidas y registre el NIA del alumno.
- Realizar las modificaciones necesarias en la aplicación mTeachClass para adaptarse a los cambios realizados en el servidor SELQuizServer.

Los ficheros generados con mTeachPlan consistirán en un listado de alumnos indicando la calificación obtenida por cada uno en un cuestionario. Debido a que en el fichero cada alumno debe estar identificado unívocamente, es necesario incluir en el sistema un dato que sea reconocible por Moodle a la hora de evaluar el listado. El mejor dato para este fin es el NIA del alumno, puesto que es un valor de pocos caracteres y a la vez conocido por los propios alumnos. Este es el motivo de introducir el NIA en el modelo de datos del servidor SELQuizServer y solicitarlo a los alumnos desde la aplicación mClassTest. Se podrá así, tener una relación en el sistema entre cuentas de Twitter y alumnos de la Universidad.

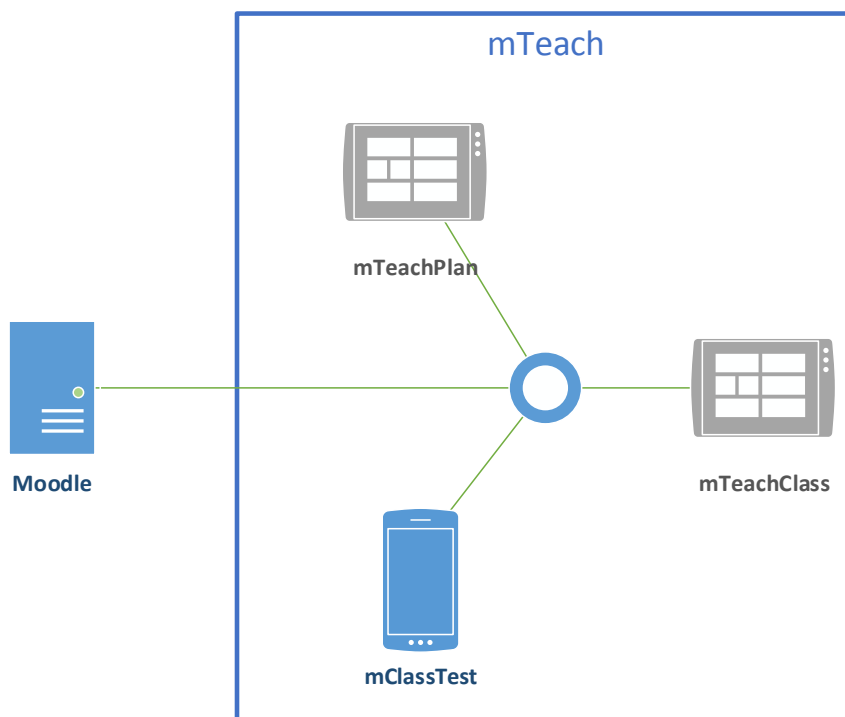


Figura 12 - Diagrama general de la solución final

En la Figura 12 se muestra un diagrama general de la solución adoptada, sin mostrar detalles propios de la arquitectura como los servidores, que se mostrarán en el apartado de Diseño. En gris se muestran las aplicaciones que forman parte de mSuite pero que no van a ser modificadas en este proyecto. En azul se han marcado los nodos en los que interviene este trabajo.

### 3.2 RESTRICCIONES DEL SISTEMA

En este punto se definen en líneas generales las restricciones del sistema, siendo en los requisitos donde se detallan minuciosamente.

- La extensión para Moodle únicamente servirá para exportar datos, evitando problemas derivados de una modificación de la base de datos de Moodle.
- Debido a que la Universidad cuenta con la versión 2.4 de esta plataforma, se construirá sobre esta para asegurar su correcto funcionamiento.

- El lenguaje utilizado en la implementación de la extensión para Moodle será PHP, debido a que es el establecido por la plataforma para desarrollar estas extensiones
- La aplicación para dispositivos móviles será desarrollada únicamente para la plataforma Android y para una versión que permita maximizar el número de dispositivos compatibles.
- La aplicación Android cubrirá todas las funciones que ya se cubrían anteriormente con la aplicación en HTML5 de mSuite, además de añadir nuevas funcionalidades.
- A la hora del desarrollo de la aplicación Android se seguirá su documentación oficial para utilizar los patrones de diseño y de implementación que propone.
- Se utilizará Android SDK como entorno de desarrollo de la aplicación.

### 3.3 ENTORNO OPERACIONAL

El entorno operacional se puede dividir en las tres partes diferenciadas de este proyecto:

- Entorno operacional de Moodle:
  - Necesitará de unos servidores con un sistema operativo como Linux (el más recomendado y utilizado), Apache, MySQL y PHP.
  - El hardware de los servidores donde se encuentre instalado el LMS dependerá del número de personas que se espera que puedan acceder a él simultáneamente. En su documentación oficial, Moodle establece unos requerimientos mínimos [30]:
    - 5GB mínimos de espacio de disco para el almacenamiento del LMS y materiales y la misma cantidad para un módulo para copias de seguridad.
    - 1GB de memoria por cada 20 estudiantes conectados a la vez a Moodle y haciendo uso de algún recurso.
- Entorno operacional sobre la aplicación Android:
  - La aplicación se podrá ejecutar en dispositivos Android con una versión superior a Android 2.3.3.
  - La pantalla del dispositivo donde se instale podrá ser de cualquier resolución, pudiendo ser una tableta o un teléfono inteligente.
  - El dispositivo deberá contar con conexión a Internet.
  - Para el desarrollo de la aplicación se utilizará el entorno de desarrollo Android SDK.
  - El dispositivo debe contar con al menos 7MB de espacio libre para la instalación de la aplicación y su ejecución.

- Entorno operacional del SELQuizServer:
  - Para realizar una implementación como la del SELQuizServer es necesario contar con un servidor con Java 5.0 o superior.
  - Para la compilación del servidor, lo cual se debe hacer cuando se han introducido cambios en su código, se necesita tener instalado Play Framework en su versión 2.1.1.

### 3.4 ESPECIFICACIÓN DE CASOS DE USO

Se procede a detallar los casos de uso del sistema, que servirán para descubrir y plasmar los comportamientos del sistema en las interacciones con los actores que se comunican con él.

Se han definido dos actores intervinientes en los casos de uso:

- Alumno: será el usuario de la aplicación Android. Utilizará la aplicación para suscribirse a sesiones y responder preguntas en un centro educativo donde opere la suite mSuite.
- Usuario de la extensión para Moodle: será aquella persona o sistema que haga llamadas a la extensión para Moodle. En este proyecto estas llamadas, desde la suite mSuite, no han sido implementadas (se lleva a cabo en otro trabajo) por lo que el actor final del mismo será distinto. Sin embargo, el modo a proceder será idéntico, manteniendo su validez los casos de uso planteados en este documento.

#### 3.4.1 DIAGRAMAS DE CASOS DE USO

Se diferencian dos diagramas que se corresponden con el desarrollo de la extensión para Moodle y la aplicación Android. Esta separación se debe a que en el presente proyecto no se ha llevado a cabo una integración de ambas partes en un único sistema.



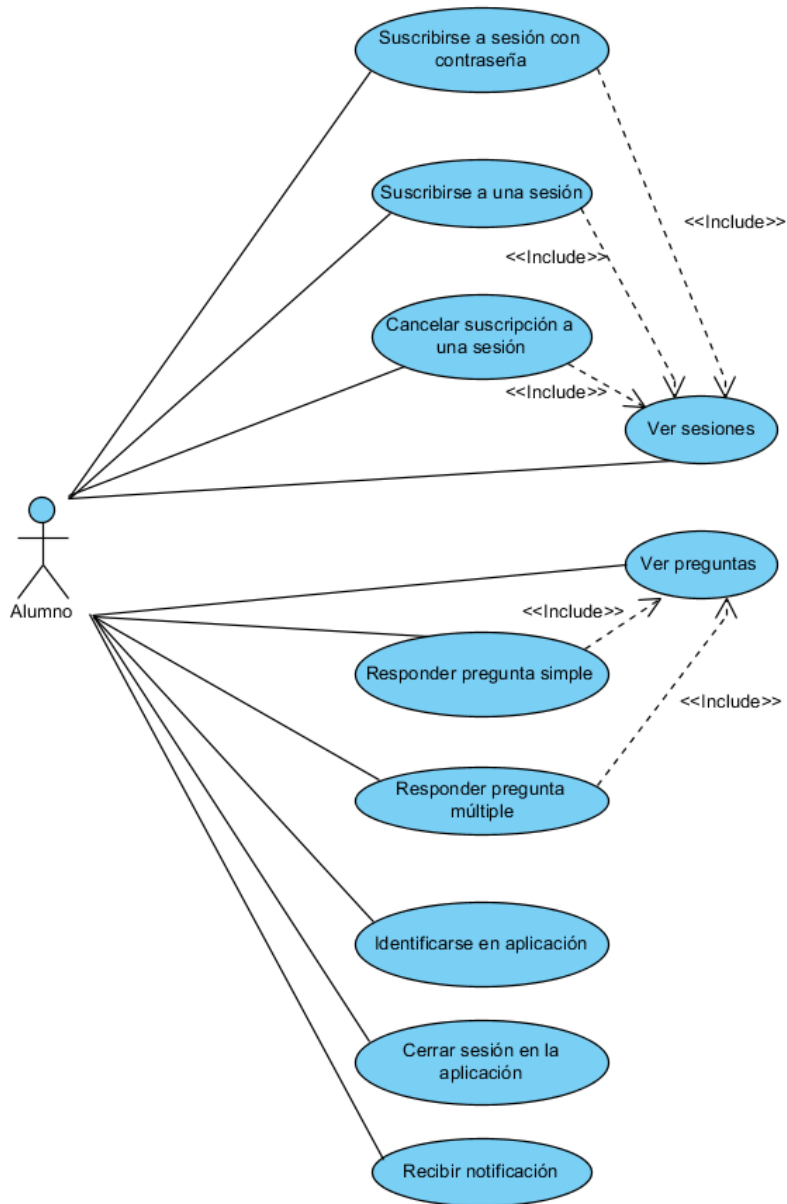


Figura 13 - Diagrama de casos de uso del alumno

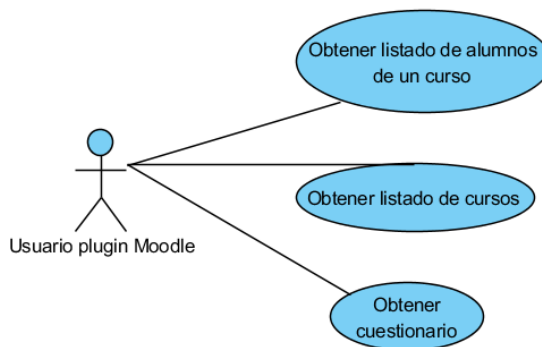


Figura 14 - Diagrama de casos de uso del usuario de la extensión de Moodle

### 3.4.2 DESCRIPCIÓN DE LOS ATRIBUTOS DE LOS CASOS DE USO

A continuación se enumeran y explican los campos de las tablas que definirán los casos de uso:

- **Código:** Identificación unívoca abreviada del caso de uso, se construye mediante CU seguido de un guión y de tres dígitos. Por ejemplo CU-001.
- **Nombre:** Identificación extendida del caso de uso.
- **Actor:** Conjunto de entidades que interactúan con el caso de uso. El caso de uso representa una funcionalidad demandada por un actor.
- **Descripción:** Se realiza una descripción básica de la funcionalidad o funcionalidades del caso de uso.
- **Precondiciones y efectos:** Se realiza una descripción de las condiciones que deben cumplirse para poder realizar una operación, y el estado en el que queda el sistema tras realizar una operación.
- **Curso de eventos:** Se realiza una descripción básica de las acciones que se ejecutarán paso a paso en el caso de uso.

Cuando en uno de estos campos no se informe, se rellenará con *N/A* (no aplica).

Estos campos se distribuirán en tablas como la que se proporciona de ejemplo (Tabla 5).

Caso de uso	
<b>Código</b>	
<b>Nombre</b>	
<b>Actor</b>	
<b>Descripción</b>	
<b>Precondiciones</b>	
<b>Curso de eventos</b>	
<b>Efectos</b>	

Tabla 5 - Tabla de ejemplo de especificación de caso de uso

### 3.4.3 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO

Caso de uso	
<b>Código</b>	CU-001
<b>Nombre</b>	Identificarse en aplicación Android
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno se identifica en la aplicación mediante su cuenta de Twitter y su NIA para poder ver las sesiones y preguntas.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno posee una cuenta de Twitter.</li> <li>• El alumno tiene un identificador en la Universidad (NIA).</li> <li>• El alumno no se ha identificado previamente en la aplicación o el alumno ha cerrado sesión [CU-009].</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la aplicación.</li> <li>2. La aplicación muestra una ventana para introducir los datos de inicio de sesión en Twitter, los cuales rellena el alumno.</li> <li>3. El alumno pulsa el botón de identificación con Twitter.</li> <li>4. La aplicación muestra una ventana para solicitar el NIA, que es introducido por el alumno.</li> <li>5. El alumno pulsa el botón de envío del NIA.</li> <li>6. La aplicación muestra la vista principal de la aplicación donde aparece la lista de sesiones indicando si el alumno está suscrito a alguna de ellas.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• El alumno puede ver una lista de sesiones disponibles (algunas protegidas con contraseña) a las que puede suscribirse.</li> </ul>

Tabla 6 - Caso de uso CU-001 - Identificarse en aplicación

Caso de uso	
<b>Código</b>	CU-002
<b>Nombre</b>	Ver sesiones
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno entra en la aplicación para ver un listado de todas las sesiones existentes.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la aplicación</li> <li>2. Se muestra una vista con la lista total de sesiones indicando si el alumno está suscrito a alguna de ellas, pudiendo suscribirse o cancelar la suscripción a cualquiera de ellas.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• Se muestra el listado de sesiones con información de las suscritas y no suscritas.</li> </ul>

Tabla 7 - Caso de uso CU-002 - Ver sesiones

Caso de uso	
<b>Código</b>	CU-003
<b>Nombre</b>	Ver preguntas
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno entra en la aplicación y accede a la vista de visualización de preguntas.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El alumno se ha identificado previamente en la aplicación [CU-001].</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>El alumno abre la aplicación</li> <li>Se muestra una vista con la lista total de sesiones indicando si el alumno está suscrito a alguna de ellas.</li> <li>El alumno selecciona la opción para ver las preguntas disponibles</li> <li>Se muestra una vista con las preguntas disponibles.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>La aplicación muestra una ventana con las preguntas disponibles, indicando si ya están o no contestadas.</li> </ul>

Tabla 8 - Caso de uso CU-003 - Ver preguntas

Caso de uso	
<b>Código</b>	CU-004
<b>Nombre</b>	Suscribirse a una sesión
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno se suscribe a una sesión para poder recibir notificaciones y visualizar las preguntas asociadas.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El alumno se ha identificado previamente en la aplicación [CU-001].</li> <li>Existe una sesión no suscrita por el alumno y no protegida con contraseña.</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>El alumno abre la aplicación.</li> <li>Se muestra una vista con la lista total de sesiones indicando si el alumno está suscrito a alguna de ellas, pudiendo suscribirse o cancelar la suscripción a cualquiera de ellas.</li> <li>El alumno selecciona una sesión no suscrita y no protegida por contraseña.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>El alumno queda suscrito a la sesión seleccionada pudiendo recibir y contestar preguntas asociadas a esa sesión.</li> <li>Si el alumno estuviera previamente suscrito a otra sesión, la aplicación automáticamente cancelará la suscripción con esa sesión.</li> </ul>

Tabla 9 - Caso de uso CU-004 - Suscribirse a una sesión

Caso de uso	
<b>Código</b>	CU-005
<b>Nombre</b>	Cancelar suscripción a una sesión
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno cancela la suscripción a una sesión no recibiendo más notificaciones nuevas ni pudiendo ver preguntas asociadas a esa sesión.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> <li>• Existe una sesión suscrita por el alumno.</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la aplicación.</li> <li>2. Se muestra una vista con la lista total de sesiones indicando si el alumno está suscrito a alguna de ellas, pudiendo suscribirse o cancelar la suscripción a cualquiera de ellas.</li> <li>3. El alumno selecciona una sesión suscrita.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• La suscripción del alumno a la sesión seleccionada queda cancelada.</li> </ul>

Tabla 10 - Caso de uso CU-005 - Cancelar suscripción a una sesión

Caso de uso	
<b>Código</b>	CU-006
<b>Nombre</b>	Suscribirse a una sesión con contraseña
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno se suscribe a una sesión protegida con contraseña para poder recibir notificaciones y visualizar las preguntas asociadas.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> <li>• Existe una sesión no suscrita por el alumno y protegida con contraseña.</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la aplicación.</li> <li>2. Se muestra una vista con la lista total de sesiones indicando si el alumno está suscrito a alguna de ellas, pudiendo suscribirse o cancelar la suscripción a cualquiera de ellas.</li> <li>3. El alumno selecciona una sesión no suscrita y protegida por contraseña.</li> <li>4. La aplicación muestra una vista para solicitar la contraseña con la cual está protegida la sesión, la cual introduce el alumno.</li> <li>5. El alumno pulsa el botón de envío de contraseña.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• El alumno queda suscrito a la sesión seleccionada pudiendo recibir y contestar preguntas asociadas a esa sesión.</li> <li>• Si el alumno estuviera previamente suscrito a otra sesión, la aplicación automáticamente cancelará la suscripción con esa sesión.</li> </ul>

Tabla 11 - Caso de uso CU-006 - Suscribirse a una sesión con contraseña

Caso de uso	
<b>Código</b>	CU-007
<b>Nombre</b>	Responder pregunta simple
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno selecciona una pregunta de respuesta simple para contestarla. Se abrirá una vista en la que el alumno elegirá una opción y pulsará el botón de envío.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> <li>• El alumno está suscrito a una sesión que tiene al menos una pregunta de respuesta simple sin responder.</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la vista que lista las preguntas disponibles [CU-003].</li> <li>2. El alumno selecciona una pregunta de respuesta simple que no esté marcada como respondida.</li> <li>3. La aplicación muestra en una vista una pregunta con sus respuestas posibles, de las cuales el alumno solo podrá seleccionar una.</li> <li>4. El alumno selecciona una respuesta.</li> <li>5. El alumno pulsa en el botón de enviar respuesta.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• La respuesta dada por el alumno es registrada por la aplicación.</li> <li>• Se muestra de nuevo la vista con el listado de preguntas, marcando como respondida la pregunta que el alumno acaba de responder.</li> </ul>

Tabla 12 - Caso de uso CU-007 - Responder pregunta simple

Caso de uso	
<b>Código</b>	CU-008
<b>Nombre</b>	Responder pregunta múltiple
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno selecciona una pregunta de respuesta múltiple para contestarla. Se abrirá una vista en la que el alumno elegirá una o varias opciones y pulsará el botón de envío.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> <li>• El alumno está suscrito a una sesión que tiene al menos una pregunta múltiple sin responder.</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la vista que lista las preguntas disponibles [CU-003].</li> <li>2. El alumno selecciona una pregunta que no esté marcada como respondida.</li> <li>3. La aplicación muestra en una vista una pregunta con sus respuestas posibles, de las cuales el alumno podrá seleccionar varias.</li> <li>4. El alumno selecciona una o varias respuestas.</li> <li>5. El alumno pulsa en el botón de enviar respuesta.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• La respuesta dada por el alumno es registrada por la aplicación.</li> <li>• Se muestra de nuevo la vista con el listado de preguntas, marcando como respondida la pregunta que el alumno acaba de responder.</li> </ul>

Tabla 13 - Caso de uso CU-008 - Responder pregunta múltiple

Caso de uso	
<b>Código</b>	CU-009
<b>Nombre</b>	Cerrar sesión en la aplicación
<b>Actor</b>	Alumno
<b>Descripción</b>	El alumno cierra sesión en la aplicación. Al abrir la aplicación de nuevo, se mostrará de nuevo la vista de identificación para realizar el caso de uso [CU-001].
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno se ha identificado previamente en la aplicación [CU-001].</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno abre la aplicación.</li> <li>2. El alumno pulsa el botón destinado a cerrar sesión en la aplicación.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• Se muestra la vista de inicio de la aplicación para que el alumno se identifique de nuevo [CU-001].</li> </ul>

Tabla 14 - Caso de uso CU-009 - Cerrar sesión en la aplicación

Caso de uso	
<b>Código</b>	CU-010
<b>Nombre</b>	Recibir notificación
<b>Actor</b>	Usuario de la extensión para Moodle
<b>Descripción</b>	El usuario recibe una notificación en su dispositivo avisándole de nuevas preguntas disponibles.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El alumno se ha identificado previamente en la aplicación [CU-001].</li> </ul>
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>El alumno recibe una notificación en su dispositivo con un mensaje informándole del motivo del aviso.</li> <li>El alumno pulsa sobre la notificación.</li> <li>La aplicación se inicia.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>La aplicación se abre para mostrar la lista de sesiones [CU-002].</li> </ul>

Tabla 15 - Caso de uso CU-010 - Recibir notificación

Caso de uso	
<b>Código</b>	CU-011
<b>Nombre</b>	Obtener listado de alumnos de un curso
<b>Actor</b>	Usuario de la extensión para Moodle
<b>Descripción</b>	El usuario hace una llamada al servidor de Moodle para obtener un listado de alumnos.
<b>Precondiciones</b>	N/A
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>El usuario prepara una llamada indicando la ruta del servidor donde se encuentra instalado Moodle, el identificador de la función que devuelve el listado de alumnos y el <i>token</i> del profesor que requiere los datos.</li> <li>El usuario realiza la llamada a la función.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>El usuario recibe un listado de alumnos con diversos datos de cada uno.</li> </ul>

Tabla 16 - Caso de uso CU-011 - Obtener listado de alumnos de un curso

Caso de uso	
<b>Código</b>	CU-012
<b>Nombre</b>	Obtener listado de cursos
<b>Actor</b>	Usuario de la extensión para Moodle
<b>Descripción</b>	El usuario hace una llamada al servidor de Moodle para obtener un listado de cursos sobre los que es profesor .
<b>Precondiciones</b>	N/A
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>El usuario prepara una llamada indicando la ruta del servidor donde se encuentra instalado Moodle, el identificador de la función que devuelve el listado de cursos del usuario en los que es profesor y el <i>token</i> del profesor que requiere los datos.</li> <li>El usuario realiza la llamada a la función.</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>El usuario recibe un listado de cursos con diversos datos de cada uno.</li> </ul>

Tabla 17 - Caso de uso CU-012 - Obtener listado de cursos



Caso de uso	
<b>Código</b>	CU-013
<b>Nombre</b>	Obtener cuestionario
<b>Actor</b>	Usuario de la extensión para Moodle
<b>Descripción</b>	El usuario hace una llamada al servidor de Moodle para obtener los cuestionarios previamente creados en la plataforma con información general del mismo y de las preguntas que incluye.
<b>Precondiciones</b>	N/A
<b>Curso de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario prepara una llamada indicando la ruta del servidor donde se encuentra instalado Moodle, el identificador de la función que devuelve los cuestionarios de un determinado curso, el identificador del curso y el <i>token</i> del profesor que requiere los datos.</li> <li>2. El usuario realiza la llamada a la función</li> </ol>
<b>Efectos</b>	<ul style="list-style-type: none"> <li>• El usuario recibe los cuestionarios de un determinado curso sobre el que posee el rol de profesor. La información recuperada contendrá información sobre las preguntas, respuestas posibles y respuesta correcta entre muchos otros parámetros.</li> </ul>

Tabla 18 - Caso de uso CU-013 - Obtener cuestionario

### 3.5 ESPECIFICACIÓN DE REQUISITOS

Con el fin de definir específicamente la solución para proceder con el diseño de la misma, se elabora un plan de requisitos. Estos estarán divididos en dos apartados: requisitos funcionales, que detallan las funcionalidades que debe cubrir el sistema desarrollado y requisitos no funcionales, que restringen las características o funciones del sistema.

#### 3.5.1 DESCRIPCIÓN DE LOS ATRIBUTOS DE LOS REQUISITOS

- **Código:** Identificación unívoca abreviada del requisito, se construye mediante el código del requisito seguido de un guión y de tres dígitos. Los requisitos serán divididos en funcionales y no funcionales y sus códigos son RF para los requisitos funcionales y RNF para los requisitos no funcionales. Por ejemplo RF-001.
- **Nombre:** Identificación extendida del requisito.
- **Descripción:** Descripción básica del requisito.
- **Fuente:** Indica a través de qué fuente ha sido identificado el requisito. Normalmente este valor se corresponderá con uno o varios códigos de los casos de uso o con otro requisito.
- **Necesidad:** Determina el grado de implementación del requisito. Los valores que puede tomar este atributo son los siguientes:
  - **Esencial:** El requisito tiene que ser implementado.
  - **Deseable:** Es preferible implementar el requisito, pero no es obligatorio.
  - **Opcional:** El requisito se podrá implementar, pero no es importante ni obligatorio.

- **Prioridad:** Define la importancia del requisito, de forma que permita definir el orden en el cual será incluido en el proceso de diseño y el orden de implementación. Los valores que puede tomar este atributo son los siguientes:
  - Alta: El requisito debe ser implementado en las fases iniciales del desarrollo.
  - Media: El requisito debe ser implementado una vez que hayan sido implementados los requisitos de prioridad alta.
  - Baja: El requisito debe ser implementado en las fases finales del desarrollo. Estos requisitos no influirán en el correcto funcionamiento del sistema.
- **Estabilidad:** Define la estabilidad del requisito durante la vida útil del software. Esto implica si el requisito podrá ser o no modificado durante el ciclo del vida. Los valores que puede tomar este atributo son los siguientes:
  - Estable: El requisito no puede variar durante el ciclo de vida del sistema.
  - Inestable: El requisito puede variar a lo largo de la ciclo de vida del sistema.
- **Verificabilidad:** Define el grado de verificabilidad de un requisito, es decir indica en qué grado es posible comprobar que el requisito se ha incorporado en el sistema desarrollado. Los valores que puede tomar este atributo son los siguientes:
  - Alta: Se puede verificar que el requisito ha sido implementado en el sistema. Este tipo de requisitos se corresponden con las funcionalidades básicas del sistema.
  - Media: Se puede verificar que el requisito ha sido implementado en el sistema pero requiere de una comprobación compleja o del código fuente del sistema.
  - Baja: Es difícil verificar si el requisito ha sido implementado en el sistema o en algunos casos no es posible.

Requisito software			
<b>Código</b>		<b>Fuente</b>	
<b>Nombre</b>			
<b>Descripción</b>			
<b>Necesidad</b>		<b>Prioridad</b>	
<b>Estabilidad</b>		<b>Verificabilidad</b>	

Tabla 19 - Tabla de ejemplo de especificación de requisito

### 3.5.2 ESPECIFICACIÓN DE REQUISITOS

#### 3.5.2.1 REQUISITOS FUNCIONALES

Requisito software funcional			
<b>Código</b>	RF-001	<b>Fuente</b>	CU-001
<b>Nombre</b>	Identificación mediante Twitter		
<b>Descripción</b>	El alumno se identificará en la aplicación mediante una cuenta de Twitter. El alumno introducirá su usuario y contraseña de la red social para comprobar que es dueño de la cuenta. Este proceso se realizará cuando el alumno abra por primera vez la aplicación, no volviendo a mostrar solicitar la identificación con Twitter, excepto si cierra sesión en la aplicación o la desinstala de su dispositivo.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 20 - Requisito RF-001 - Identificación mediante Twitter

Requisito software funcional			
<b>Código</b>	RF-002	<b>Fuente</b>	CU-001
<b>Nombre</b>	Solicitud de NIA		
<b>Descripción</b>	La aplicación solicitará al alumno, una vez se haya identificado mediante Twitter [RF-001], el NIA para guardar una relación en el servidor de la cuenta Twitter con el alumno de la Universidad al que pertenece.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 21 - Requisito RF-002 - Solicitud de NIA

Requisito software funcional			
<b>Código</b>	RF-003	<b>Fuente</b>	CU-002
<b>Nombre</b>	Lista sesiones		
<b>Descripción</b>	El alumno podrá ver un listado de sesiones. Este listado se mostrará en una vista de la aplicación Android habilitada para este fin. Para cada sesión se indicará si el alumno está suscrito o no a ella. Las sesiones se identificarán por un nombre, que será el mostrado para cada una en el listado.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 22 - Requisito RF-003 - Lista sesiones

Requisito software funcional			
<b>Código</b>	RF-004	<b>Fuente</b>	CU-003
<b>Nombre</b>	Lista preguntas		
<b>Descripción</b>	El alumno podrá ver un listado de preguntas. Este listado se mostrará en una vista de la aplicación Android habilitada para este fin. Para cada pregunta se indicará si el alumno la ha respondido ya o no. Las preguntas se identificarán por su enunciado, que será el mostrado para cada una en el listado.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 23 - Requisito RF-004 - Lista preguntas

Requisito software funcional			
<b>Código</b>	RF-005	<b>Fuente</b>	Analista
<b>Nombre</b>	Sincronización de la aplicación Android con servidor		
<b>Descripción</b>	<p>La aplicación Android se sincronizará con el servidor SELQuizServer, del que obtendrá los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Lista de sesiones, indicando cuales están protegidas con contraseña</li> <li>• Lista de preguntas, indicando cuales están ya respondidas por el alumno</li> <li>• Sesión suscrita por el alumno, si hubiera</li> <li>• Preguntas con sus respuestas posibles indicando si es de respuesta simple o de respuesta múltiple</li> </ul> <p>La aplicación enviará al servidor los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Usuario de la cuenta Twitter y NIA introducido por el alumno</li> <li>• Sesión a suscribir por el alumno</li> <li>• Sesión sobre la que cancelar la suscripción</li> <li>• Respuesta dada por un alumno a una pregunta</li> </ul>		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Inestable	<b>Verificabilidad</b>	Media

Tabla 24 - Requisito RF-005 - Sincronización de la aplicación Android con servidor

Requisito software funcional			
<b>Código</b>	RF-006	<b>Fuente</b>	CU-004 y CU-005
<b>Nombre</b>	Suscripción de sesiones		
<b>Descripción</b>	Los alumnos podrán suscribirse a sesiones para ver el listado de preguntas, responderlas y recibir notificaciones acerca de nuevas preguntas de la sesión suscrita. También podrán cancelar la suscripción a aquella sesión que tuvieran suscrita.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 25 - Requisito RF-006 - Suscripción de sesiones

Requisito software funcional			
<b>Código</b>	RF-007	<b>Fuente</b>	CU-006
<b>Nombre</b>	Sesiones protegidas con contraseña		
<b>Descripción</b>	Existirán sesiones protegidas con contraseña a las que el alumno no podrá suscribirse sin introducirla. En la vista en la que se listan las sesiones, aquellas protegidas con contraseña se distinguirán de las demás con un elemento indicativo de esta condición.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 26 - Requisito RF-007 - Sesiones protegidas con contraseña

Requisito software funcional			
<b>Código</b>	RF-008	<b>Fuente</b>	CU-007 y CU-008
<b>Nombre</b>	Responder pregunta		
<b>Descripción</b>	Los alumnos podrán responder preguntas desde la aplicación Android. Podrán ser de respuesta simple o de respuesta múltiple, informando al alumno de si se trata de un tipo u de otro. Se mostrará en una vista de la aplicación el enunciado de la pregunta, las opciones posibles y por último un botón para el envío de la respuesta.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 27 - Requisito RF-008 - Responder pregunta

Requisito software funcional			
<b>Código</b>	RF-009	<b>Fuente</b>	CU-009
<b>Nombre</b>	Cerrar sesión en aplicación		
<b>Descripción</b>	La aplicación permitirá al alumno borrar los datos almacenados y la identificación con Twitter para poder volver a identificarse [RF-001].		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Baja
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 28 - Requisito RF-009 - Cerrar sesión en la aplicación

Requisito software funcional			
<b>Código</b>	RF-010	<b>Fuente</b>	CU-010
<b>Nombre</b>	Servicio de notificaciones		
<b>Descripción</b>	La aplicación Android contendrá un servicio de notificaciones que permita avisar al alumno cuando hay nuevas sesiones disponibles o cuando tiene nuevas preguntas para responder.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 29 - Requisito RF-010 - Servicio de notificaciones

Requisito software funcional			
<b>Código</b>	RF-011	<b>Fuente</b>	Analista
<b>Nombre</b>	Menú en la aplicación		
<b>Descripción</b>	La aplicación tendrá un menú para elegir entre dos vistas, la del listado de sesiones y la del listado de preguntas, pudiendo alternar entre ambas.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 30 - Requisito RF-011 - Menú en la aplicación

Requisito software funcional			
<b>Código</b>	RF-012	<b>Fuente</b>	Analista
<b>Nombre</b>	Vista predeterminada		
<b>Descripción</b>	Al abrir la aplicación, siempre y cuando el alumno ya se haya identificado mediante Twitter, se mostrará automáticamente la vista del listado de sesiones.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 31 - Requisito RF-012 - Vista predeterminada

Requisito software funcional			
<b>Código</b>	RF-013	<b>Fuente</b>	Analista
<b>Nombre</b>	Intento de envío de preguntas sin proporcionar respuesta		
<b>Descripción</b>	En caso de que el alumno pulse el botón de envío de respuesta a una pregunta, sin haber seleccionado ninguna opción, la aplicación mostrará un mensaje avisándole de ello e imposibilitará el envío de la respuesta.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 32 - Requisito RF-013 - Intento de envío de preguntas sin proporcionar respuesta

Requisito software funcional			
<b>Código</b>	RF-014	<b>Fuente</b>	Analista
<b>Nombre</b>	Una única sesión suscrita simultáneamente		
<b>Descripción</b>	La aplicación permitirá suscribirse a una única sesión. En caso de que el alumno intente suscribirse a una sesión, teniendo otra ya suscrita, la suscripción a esta última se anulará.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Inestable	<b>Verificabilidad</b>	Alta

Tabla 33 - Requisito RF-014 - Una única sesión suscrita simultáneamente

Requisito software funcional			
<b>Código</b>	RF-015	<b>Fuente</b>	CU-011
<b>Nombre</b>	Obtención de un listado de estudiantes asociados a un curso de Moodle		
<b>Descripción</b>	La extensión para Moodle permitirá obtener un listado de estudiantes asociados a un curso. Será necesario proporcionar un <i>token</i> de un usuario de Moodle, del que se comprobará su validez, y un identificador del curso sobre el que se requieren los estudiantes. La información obtenida incluirá datos acerca de cada uno de los estudiantes incluyendo su nombre, apellidos, NIA y correo electrónico.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 34 - Requisito RF-015 - Obtención de un listado de estudiantes asociados a un curso de Moodle

Requisito software funcional			
<b>Código</b>	RF-016	<b>Fuente</b>	CU-012
<b>Nombre</b>	Obtención de un listado de cursos de Moodle		
<b>Descripción</b>	La extensión permitirá obtener un listado de cursos de Moodle. Será necesario proporcionar un <i>token</i> de un usuario de Moodle, del que se comprobará su validez. En el listado únicamente se incluirán cursos en los que el usuario (identificado por el <i>token</i> ) posee el rol de profesor. La información obtenida incluirá datos acerca de cada uno de los cursos incluyendo su nombre y los campos necesarios para recuperar posteriormente los estudiantes matriculados en el curso o los cuestionarios asociados.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 35 - Requisito RF-016 - Obtención de un listado de cursos de Moodle

Requisito software funcional			
<b>Código</b>	RF-017	<b>Fuente</b>	CU-013
<b>Nombre</b>	Obtención de cuestionarios de Moodle		
<b>Descripción</b>	La extensión para Moodle permitirá obtener cuestionarios. Será necesario proporcionar un <i>token</i> de un usuario de Moodle, del que se comprobará su validez y un identificador de un cuestionario. En la información devuelta por la extensión para Moodle se incluirá su nombre, contraseña definida para realizarlo, fechas de apertura y cierre, listado de preguntas y opciones de respuesta para cada una.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 36 - Requisito RF-017 - Obtención de cuestionarios de Moodle

Requisito software funcional			
<b>Código</b>	RF-018	<b>Fuente</b>	Analista
<b>Nombre</b>	Cambios en SELQuizServer		
<b>Descripción</b>	El servidor SELQuizServer se modificará en los siguientes aspectos: <ul style="list-style-type: none"> <li>• Admisión de preguntas de respuesta múltiple</li> <li>• Proteger la suscripción a sesiones mediante una contraseña</li> <li>• Inclusión de campo NIA en los estudiantes</li> </ul>		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Inestable	<b>Verificabilidad</b>	Media

Tabla 37 - Requisito RF-018 - Cambios en SELQuizServer

### 3.5.2.2 REQUISITOS NO FUNCIONALES

Requisito software no funcional			
<b>Código</b>	RNF-001	<b>Fuente</b>	RF-010
<b>Nombre</b>	Servicio de notificaciones con GCM		
<b>Descripción</b>	El servicio de notificaciones que implementará la aplicación utilizará la solución ofrecida por Google: GCM (Google Cloud Messaging).		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 38 - Requisito RNF-001 - Servicio de notificaciones con GCM

Requisito software no funcional			
<b>Código</b>	RNF-002	<b>Fuente</b>	RF-011
<b>Nombre</b>	Diseño menú en aplicación Android		
<b>Descripción</b>	El menú desarrollado en la aplicación será lateral de tipo DrawerLayout.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 39 - Requisito RNF-002 - Diseño menú en aplicación Android

Requisito software no funcional			
<b>Código</b>	RNF-003	<b>Fuente</b>	Analista
<b>Nombre</b>	Versión mínima de Android		
<b>Descripción</b>	La aplicación Android deberá ser compatible con cualquier dispositivo que tenga como mínimo la versión 2.3.3 del sistema operativo Android, correspondiente al nivel de API 10.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Inestable	<b>Verificabilidad</b>	Alta

Tabla 40 - Requisito RNF-003 - Versión mínima de Android



Requisito software no funcional			
<b>Código</b>	RNF-004	<b>Fuente</b>	Analista
<b>Nombre</b>	Normas de estilo, diseño y programación		
<b>Descripción</b>	Se seguirán las normas de estilo, diseño y programación definidas en la documentación oficial de Android.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 41 - Requisito RNF-004 - Normas de estilo, diseño y programación

Requisito software no funcional			
<b>Código</b>	RNF-005	<b>Fuente</b>	RF-003
<b>Nombre</b>	Uso de Switch en las sesiones		
<b>Descripción</b>	En la vista del listado de sesiones de la aplicación Android, cada sesión llevará asociado un Toogle Button de tipo Switch que indicará, con sus dos estados posibles (activado o desactivado), si el alumno tiene o no suscrita la sesión. Este Switch podrá pulsarse para permutar su estado.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 42 - Requisito RNF-005 - Uso de Switch en las sesiones

Requisito software funcional			
<b>Código</b>	RNF-006	<b>Fuente</b>	RF-008
<b>Nombre</b>	Un único intento de respuesta		
<b>Descripción</b>	Los alumnos podrán responder cada pregunta desde la aplicación Android una única vez.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 43 - Requisito RNF-006 - Un único intento de respuesta

Requisito software no funcional			
<b>Código</b>	RNF-007	<b>Fuente</b>	RF-014
<b>Nombre</b>	Único Switch activo		
<b>Descripción</b>	Cuando un alumno pulse en un Switch para activarlo, cualquier otro Switch que ya estuviera activado se posicionaría en su valor contrario.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 44 - Requisito RNF-007 - Único Switch activo

Requisito software no funcional			
<b>Código</b>	RNF-008	<b>Fuente</b>	RF-006
<b>Nombre</b>	Uso de CheckBox en el listado de preguntas		
<b>Descripción</b>	En la vista de la aplicación Android que muestra el listado de preguntas, cada pregunta tendrá asociado un CheckBox, que indicará, si está seleccionado, que la pregunta ya ha sido respondida. El alumno no podrá interactuar con el CheckBox para cambiar su estado.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 45 - Requisito RNF-008 - Uso de CheckBox en el listado de preguntas

Requisito software no funcional			
<b>Código</b>	RNF-009	<b>Fuente</b>	Analista
<b>Nombre</b>	Distintas resoluciones		
<b>Descripción</b>	La aplicación será compatible con todas las resoluciones existentes de dispositivos Android, ya sean propias de teléfonos inteligentes o de tabletas.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 46 - Requisito RNF-009 - Distintas resoluciones

Requisito software no funcional			
<b>Código</b>	RNF-010	<b>Fuente</b>	Analista
<b>Nombre</b>	Vista vertical		
<b>Descripción</b>	La aplicación Android únicamente se visualizará en vista vertical, imposibilitando su visualización en vista horizontal.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Alta

Tabla 47 - Requisito RNF-010 - Vista vertical

Requisito software no funcional			
<b>Código</b>	RNF-011	<b>Fuente</b>	Analista
<b>Nombre</b>	Actualización de contenido en la aplicación		
<b>Descripción</b>	Para actualizar el listado de sesiones o de preguntas en la aplicación se utilizará el estándar de arrastrar el dedo hacia abajo sobre la vista. Para ello, se utilizará la clase de Android SwipeRefreshLayout.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 48 - Requisito RNF-011 - Actualización de contenido en la aplicación

Requisito software no funcional			
<b>Código</b>	RNF-012	<b>Fuente</b>	Analista
<b>Nombre</b>	Protocolo de llamadas a la extensión de Moodle		
<b>Descripción</b>	Se utilizará el protocolo XML-RPC para la realización de llamadas a la extensión de Moodle, siendo necesaria su activación en la configuración de la plataforma por parte de su administrador.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 49 - Requisito RNF-012 - Protocolo de llamadas a la extensión para Moodle

Requisito software no funcional			
<b>Código</b>	RNF-013	<b>Fuente</b>	Analista
<b>Nombre</b>	Lenguaje de implementación de la extensión para Moodle		
<b>Descripción</b>	La extensión para Moodle se implementará en el lenguaje PHP en su versión 5.3.2.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 50 - Requisito RNF-013 - Lenguaje de implementación de la extensión para Moodle

Requisito software no funcional			
<b>Código</b>	RNF-014	<b>Fuente</b>	Analista
<b>Nombre</b>	Implementación de la extensión para Moodle en distintas llamadas		
<b>Descripción</b>	Se construirán diversas llamadas en la extensión para Moodle según los datos a obtener con el fin de favorecer la rapidez de las mismas y disminuir la cantidad de datos incluidos en cada una.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 51 - Requisito RNF-014 - Implementación de la extensión para Moodle en distintas llamadas

Requisito software no funcional			
<b>Código</b>	RNF-015	<b>Fuente</b>	Analista
<b>Nombre</b>	Control de roles		
<b>Descripción</b>	La extensión para Moodle se desarrollará maximizando la seguridad, comprobando que el usuario del que se proporciona el <i>token</i> posee los privilegios necesarios para acceder a los datos que requiera.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 52 - Requisito RNF-015 - Control de roles

Requisito software no funcional			
<b>Código</b>	RNF-016	<b>Fuente</b>	Analista
<b>Nombre</b>	Token para la extensión para Moodle		
<b>Descripción</b>	Para llamar a las nuevas funciones de Moodle será necesario poseer un <i>token</i> asociado a la extensión instalada.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Media
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	Media

Tabla 53 - Requisito RNF-016 - Token para la extensión para Moodle

### 3.5.2.3 TRAZABILIDAD ENTRE REQUISITOS Y CASOS DE USO

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013
RF-001	X												
RF-002	X												
RF-003		X											
RF-004			X										
RF-005													
RF-006				X	X								
RF-007						X							
RF-008							X	X					
RF-009									X				
RF-010										X			
RF-011													
RF-012													
RF-013													
RF-014													
RF-015											X		
RF-016												X	
RF-017													X
RF-018													

Tabla 54 - Matriz de trazabilidad entre requisitos y casos de uso

## 4. DISEÑO

Con vistas a afrontar la implementación de este proyecto, es necesario realizar un diseño del mismo. Se realizará partiendo de una representación de la totalidad del objeto que será construido y será refinado hasta proporcionar una guía para construir cada detalle [31].

### 4.1 DESCRIPCIÓN GENERAL DE LA ARQUITECTURA

En la Figura 15 se presenta un diagrama con la arquitectura general de mSuite, incluyendo todos los elementos y conexiones entre ellos que la forman. Esta arquitectura es la esperada una vez se haya finalizado la implementación que abarca este proyecto y la correspondiente al Trabajo Fin de Grado paralelo a este. En color azul oscuro se han señalado aquellos puntos de los que se ocupa el presente trabajo.

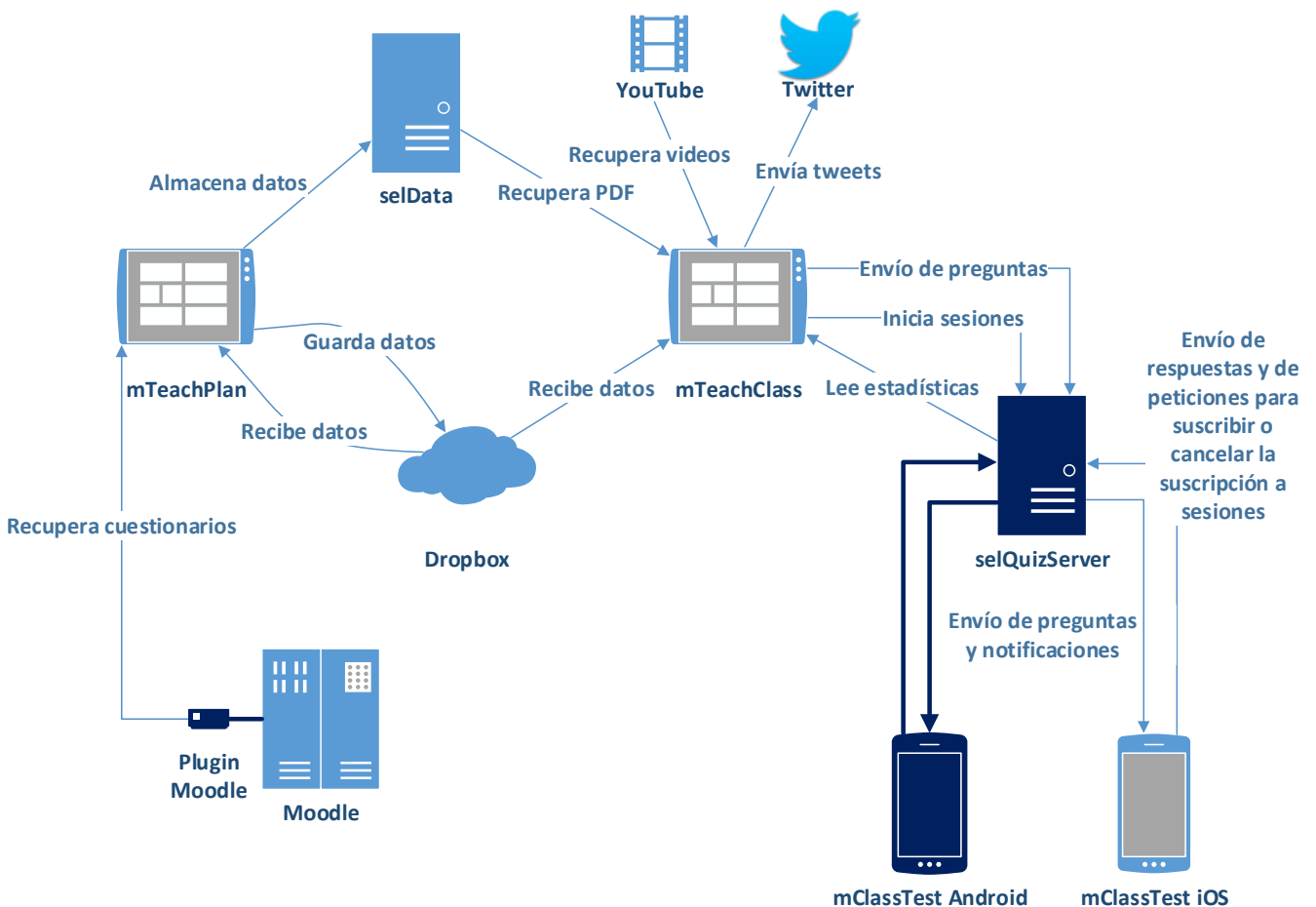


Figura 15 - Diagrama general de la arquitectura del sistema

mSuite se compone de un conjunto de aplicaciones (mTeachPlan, mTeachClass y mClassTest, cuya funcionalidad está descrita en el apartado de Estado del arte), de un conjunto de servidores y de accesos a distintos servicios externos. Con el fin de conocer el funcionamiento global del sistema, se describirán todos los elementos aunque no vayan a ser modificados en este trabajo.

La suite cuenta con dos servidores propios, denominados SelData y SELQuizServer. Además, también se utiliza Dropbox como servidor. Este último contiene el almacenamiento principal de datos. A él se conecta la aplicación mTeachPlan para guardar información de los cursos, sesiones, preguntas y *tweets*. La aplicación mTeachClass accederá a este servidor para utilizar esos datos en la impartición de las clases. Entre ambas aplicaciones, mTeachPlan y mTeachClass, también se encuentra SelData actuando de servidor. Contiene los ficheros PDF y PowerPoint que se utilizan para impartir las clases con mTeachClass. La aplicación mTeachClass, además de conectarse a Dropbox y al servidor SelData, también está enlazada con YouTube para la visualización de vídeos en las sesiones, con Twitter para el envío de *tweets* durante su transcurso y con el servidor SELQuizServer. Este último es el vínculo del sistema con las aplicaciones de los alumnos, mClassTest.

Este servidor almacena información de las sesiones, los alumnos y las preguntas y se encarga del cálculo de las estadísticas, permitiendo visualizarlas mediante páginas web alojadas en el mismo servidor. También suministra las preguntas a las aplicaciones de los alumnos y gestiona sus peticiones de suscripción o de cancelación de suscripción de sesiones. Por último, envía notificaciones a los dispositivos de los alumnos. La aplicación de impartición de clases, mTeachClass, se conecta a este servidor para incorporar los datos de sesiones y preguntas, dar de alta o de baja sesiones y lanzar nuevas preguntas para que puedan ser respondidas por los alumnos.

Con este trabajo, se incorporan dos nuevos elementos en el sistema. Por un lado, la extensión para Moodle permitirá una conexión de la suite mSuite con el LMS mediante llamadas HTTP. Por otro, se implementará una nueva aplicación mClassTest compatible con dispositivos Android, estableciendo una conexión con el servidor SELQuizServer al igual que la correspondiente para iOS. Este servidor será modificado en base a los nuevos requerimientos del sistema.

### 4.2 ALTERNATIVAS DE DISEÑO

Para realizar un análisis de las posibles alternativas a tomar en lo que a diseño se refiere, es necesario diferenciar la extensión para Moodle, las modificaciones en el SELQuizServer y la aplicación Android.

En el caso de las modificaciones en el SELQuizServer, no se puede realizar un análisis de distintas alternativas para el diseño. Esto se debe a que es un elemento ya desarrollado, en el que únicamente se van a realizar unas pequeñas modificaciones sin alterar la arquitectura del mismo.

La extensión para Moodle tampoco admite definir distintas alternativas de diseño. Una vez tomada la decisión (en el apartado de Análisis) sobre desarrollar esta extensión para obtener datos desde un sistema externo, no es posible plantear distintas alternativas para su diseño puesto que tiene un desarrollo sistemático, en el que únicamente se deben programar una serie de funciones.

En cuanto a la aplicación Android, sí se pueden evaluar distintas alternativas. Partiendo del objetivo de construir una aplicación para dispositivos móviles con sistema operativo Android, estas aplicaciones pueden seguir tres modelos diferentes, como se ha comentado en el apartado de Estado del arte. Se corresponden a tres tipos de aplicaciones: nativas, híbridas y web.

La aplicación mClassTest ya disponía de una versión híbrida para Android en HTML5. Sin embargo, en este proyecto se requiere implementar una nueva aplicación de calidad y con un elevado rendimiento que contenga las nuevas funcionalidades como la de responder preguntas de respuesta múltiple. Además, esta aplicación deberá contener un servicio de notificaciones.

En la comparativa de los diferentes tipos de aplicaciones (en la Tabla 4 del apartado de Estado del arte), se puede ver como el modelo más cercano a los objetivos es el de una aplicación nativa. Este modelo permitirá desarrollar una aplicación de calidad, con una buena integración con el hardware y el sistema operativo que permita utilizar recursos del sistema como son las notificaciones. Como desventajas, el tiempo de desarrollo será mayor y no será compatible con otros sistemas operativos de dispositivos móviles. Además, se quiere dar la misma experiencia de funcionamiento al usuario que utiliza Android que la que ya tiene el usuario de iOS con su aplicación nativa.

### 4.3 ARQUITECTURA DETALLADA DEL SISTEMA

Para la descripción de la arquitectura, se tomarán por separado los elementos a modificar o implementar: la extensión para Moodle, el servidor SELQuizServer y la aplicación mClassTest para Android. Esto se debe a que siguen distintos patrones de arquitectura y es necesaria una explicación individualizada.

#### 4.3.1 SELQUIZSERVER

La implementación de este servidor está basada en el marco de trabajo Play Framework. Este permite crear una aplicación web siguiendo el clásico patrón de arquitectura MVC (Modelo Vista Controlador). Mediante una implementación de clases en java, se define el modelo de datos y el controlador y mediante ficheros HTML se construyen las vistas del sistema (páginas web). Los métodos implementados en el controlador serán accesibles como un conjunto de funciones de unos servicios web.

Como ya se ha comentado en apartados anteriores, este servidor ya se encuentra implementado, pero es necesaria una modificación del modelo de datos y del controlador para adaptarlo a los nuevos requerimientos.

En la Figura 16 se muestra el diagrama de componentes correspondiente al servidor SELQuizServer.

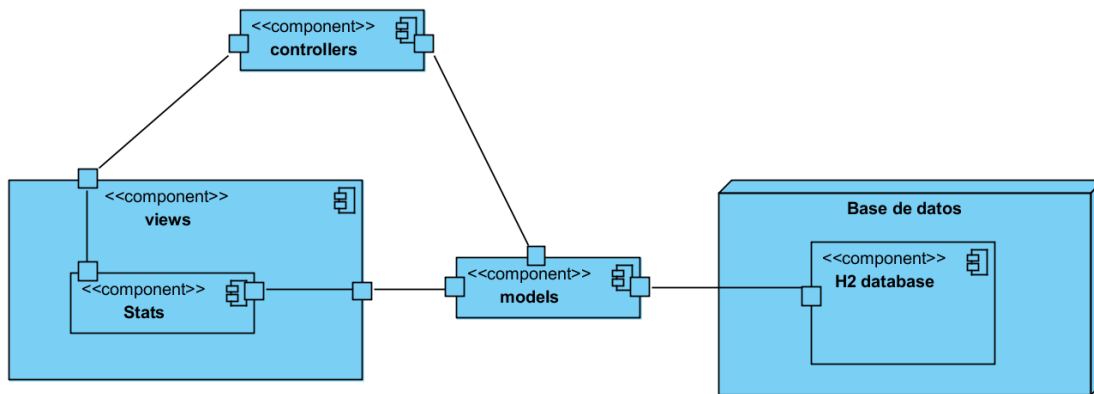


Figura 16 - Diagrama de componentes del servidor SELQuizServer

Las modificaciones se efectúan sobre los componentes *controllers* y *models*. A continuación se enuncian los cambios para cada uno de ellos:

#### 4.3.1.1 CONTROLLERS

En este componente se disponen cuatro clases:

- *QuestionController*: contiene las funciones necesarias para el manejo de preguntas y respuestas. Se modifican los siguientes puntos:
  - Implementación del método *createQuestionMult* para crear preguntas de respuesta múltiple.
  - Implementación del método *answerQuestionMult* para responder preguntas de respuesta múltiple.
- *Session Controller*: establece la gestión de sesiones. Se modifican los siguientes aspectos:
  - Modificación de la función *createSession*. Si en los parámetros de la función se encuentra informado el campo de contraseña, la sesión quedará protegida con ella. Sin esta contraseña, no se podrá suscribir por el alumno.
- *StatsController*: ejecuta vistas de la aplicación (ficheros HTML) y permite generar ficheros Excel con estadísticas. No se han introducido modificaciones en esta clase.



- *StudentController*: permite gestionar a los estudiantes y las suscripciones de estos a sesiones. Se producen los siguientes cambios:
  - Modificación de la función *checkin* para almacenar el NIA y el tipo de dispositivo (Android o iOS) en la base de datos.
  - Modificación de la función *subscribe* para comprobar que la sesión a suscribir no está protegida.
  - Implementación de la función *subscribeWithPassword*, permitiendo la suscripción a sesiones protegidas con contraseña si esta ha sido suministrada correctamente.
  - Modificación de la función *checkStudentInQuestion* para devolver una lista de respuestas si es una pregunta de respuesta múltiple.

### 4.3.1.2 MODELS

El modelo de SELQuizServer se compone de cuatro clases en lenguaje java. En el momento de compilación del servidor, serán analizadas para convertirlas automáticamente en un fichero en lenguaje SQL, que creará la estructura inicial de la base de datos, según el código desarrollado en ellas. En la Figura 17 se muestra un diagrama de la base de datos del servidor una vez completados los cambios propuestos en este documento.

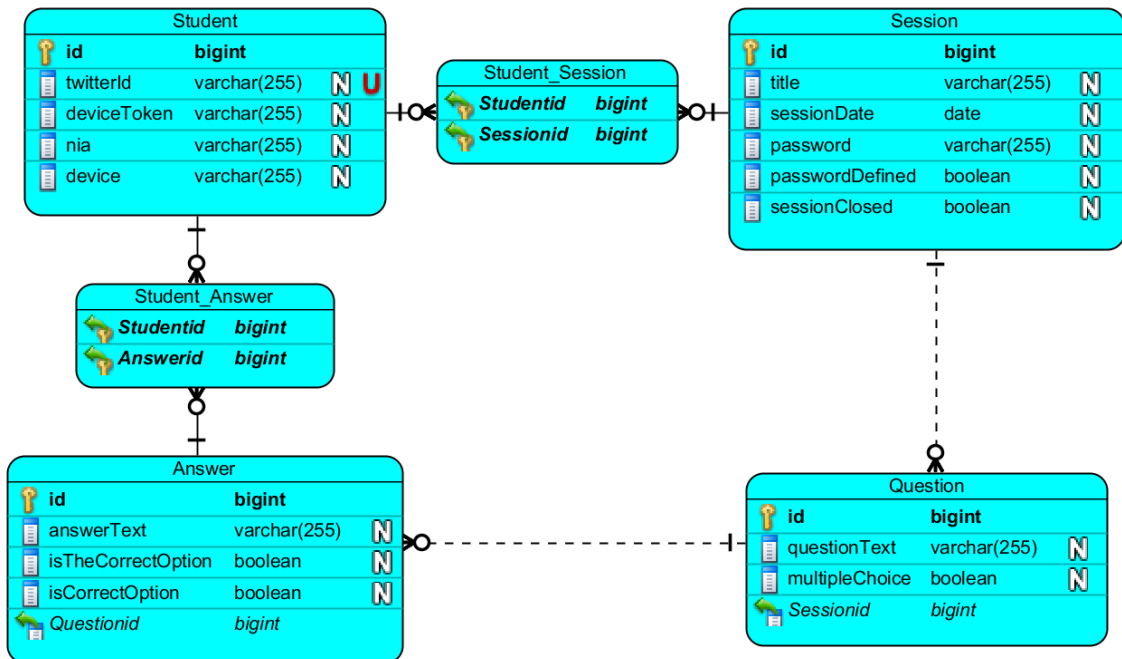


Figura 17 - Diagrama de la base de datos del SELQuizServer

Los cambios introducidos en el modelo son los siguientes:

- Student:
  - Inclusión del campo *nia*: permitirá almacenar un identificador del alumno en el servidor reconocible por Moodle. A la hora de generar ficheros con calificaciones de los alumnos, se utilizará este dato para identificar al alumno, de forma que incorporar el fichero al LMS, este asigne las calificaciones correctamente a los alumnos.
  - Inclusión del campo *device*: este campo informará del tipo de dispositivo que utiliza el alumno. Los dos valores admitidos serán “android” o “ios”. El motivo de introducir este dato, es para conocer a qué servicio concreto se debe llamar cuando se desea enviar una notificación, puesto que es diferente para cada plataforma: Apple Push Notification Service para el caso de iOS o Google Cloud Messaging para el caso de Android.
- Session:
  - Inclusión del campo *password*: contendrá la contraseña definida para poder suscribirse una sesión.
  - Inclusión del campo *passwordDefined*: indicará si una sesión concreta está protegida (tiene contraseña definida), caso en el que el campo *password* deberá estar informado.
- Answer:
  - Inclusión del campo *isCorrectOption*: este campo se utilizará en aquellas respuestas pertenecientes a una pregunta de respuesta múltiple, indicando que es una de las opciones que conforma la respuesta correcta.
- Question:
  - Inclusión del campo *multipleChoice*: indicará que una pregunta concreta es de respuesta simple o múltiple.

### 4.3.2 APLICACIÓN ANDROID

La aplicación para dispositivos móviles con sistema operativo Android también sigue un patrón de arquitectura MVC. Los componentes de vista y controlador serán los que centran la mayor parte de la implementación, disminuyendo en el modelo, que será muy reducida. Esto se debe a que la aplicación almacenará únicamente datos relativos a la identificación del alumno y al dispositivo (por el envío de notificaciones), obteniendo el resto de datos mediante llamadas al servidor SELQuizServer.

Se define un componente general para enmarcar a toda la aplicación mClassTest. Interiormente albergará otros cuatro componentes, los cuales representan módulos de funcionalidades diferenciadas.

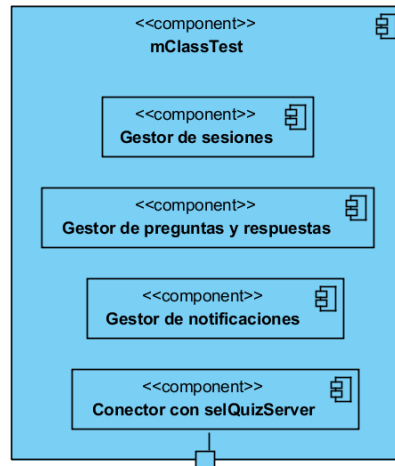


Figura 18 - Diagrama de componentes de la aplicación Android

A continuación se detalla cada componente de la aplicación por separado.

### 4.3.2.1 MCLASSTEST

Este componente engloba a todos los demás que forman parte de la aplicación. Incluye funcionalidad básica de la aplicación. Contiene las clases que se ejecutan al iniciar la aplicación, el registro en GCM, la identificación del alumno mediante Twitter y la identificación en el servidor SELQuizServer.

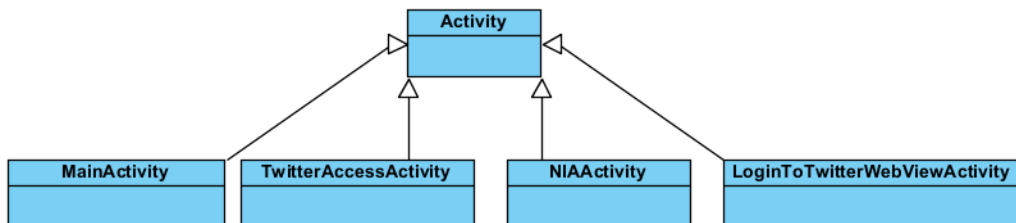


Figura 19 - Diagrama de clases 1 del componente mClassTest

El controlador de este componente (siguiendo el patrón MVC) reside en cuatro clases:

- MainActivity: clase asociada al layout *activity\_nia*. Es la actividad inicial de la aplicación, cada vez que el alumno la inicie, se ejecutará esta clase. Su funcionalidad incluye comprobar si la aplicación está registrada en GCM. Si no lo está, se procede a realizar el registro. Comprobará también que el dispositivo tiene instalados los Servicios de Google Play, los cuales resultan imprescindibles para que el servicio de notificaciones pueda operar correctamente. En caso de que el alumno no se haya identificado previamente con Twitter, se llamará a la clase TwitterAccessActivity. En caso contrario se cargará la vista principal de la aplicación (con el listado de sesiones).

- **TwitterAccessActivity**: tiene asociado el layout *activity\_twitter\_access*. Se ocupa de preparar un navegador web para mostrar la web de inicio de sesión de Twitter cuando se pulsa el botón existente en la mencionada vista.
- **NIAActivity**: esta clase está asociada al layout *activity\_nia*. Se encarga de recoger el NIA introducido por el alumno y registrarlo, junto con los datos de Twitter, en el SELQuizServer. No lo comprobará, puesto que en la versión actual del sistema no hay una relación entre el NIA de cada alumno y su cuenta de Twitter.
- **LoginToTwitterWebViewActivity**: asociada al layout *activity\_login\_to\_twitter*. Carga el navegador web para identificarse con Twitter y recoge los datos del usuario para enviarlos al SELQuizServer.

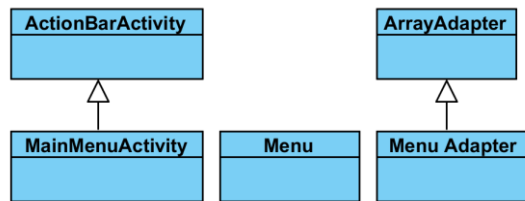


Figura 20 - Diagrama de clases 2 del componente *mClassTest*

Dentro del componente *mClassTest* también se incluyen las clases correspondientes a la vista principal de la aplicación con el menú lateral. Se construyen tres clases en este punto:

- **MainMenuActivity**: tiene asociado el layout *activity\_main\_menu*. En esta clase se describen las opciones del menú lateral que serán cargadas en la vista y se captura la opción elegida por el alumno, para realizar la acción correspondiente. Hereda de *ActionBarActivity*, que define una vista principal y un menú lateral, que queda oculto en el lado izquierdo, a menos que el alumno desee abrirlo (desplazando el dedo por la pantalla desde el borde izquierdo hacia el centro de la pantalla o con un botón situado en la barra superior). En la vista principal se carga una clase de tipo *Fragment*, que será la asociada a la vista de sesiones (*sessionsFragment*) o de preguntas (*questionsFragment*).
- **MenuAdapter**: es un *Adapter* para listar las opciones del menú. Aunque únicamente se van a mostrar tres opciones fijas, utilizar un *Adapter* facilita los posibles cambios futuros en la aplicación, pudiendo añadir nuevas opciones al menú sin modificar la vista e incluso crear diferentes opciones según se esté en una u otra vista.
- **Menu**: esta clase se utiliza por *MenuAdapter* para crear objetos por cada una de las opciones del menú (contiene el nombre y un identificador de cada opción).

### 4.3.2.2 GESTOR DE SESIONES

Este módulo será el encargado de gestionar la suscripción a sesiones, con una solicitud de contraseña para aquellas protegidas y de la visualización del listado. Se definen 4 clases:

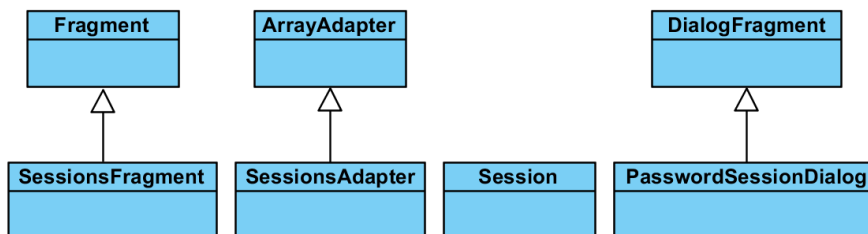


Figura 21 - Diagrama de clases del componente Gestor de sesiones

- SessionsFragment: clase asociada al layout *fragment\_sessions*. Contiene la funcionalidad necesaria para obtener el listado de sesiones disponibles del servidor y enviar las solicitudes de suscripción o de cancelación de suscripción de las distintas sesiones. Esta clase también se ocupa de lanzar un DialogFragment para solicitar al alumno la contraseña de una sesión protegida.
- SessionsAdapter: es el Adapter para crear el listado de sesiones. Toma los datos obtenidos de SELQuizServer y crea un listado en el que cada posición se compone del nombre de la sesión y un Switch. Cada una de estas posiciones se establece visualmente con el layout *list\_row*.
- Session: es la clase que utiliza SessionsAdapter para relacionar los datos de cada una de las sesiones (el identificador en SELQuizServer, el nombre o el estado del Switch).
- PasswordSessionDialog: DialogFragment personalizado para mostrarlo cuando el alumno intenta suscribirse a una sesión protegida. Muestra una ventana emergente que solicita la contraseña y le informa de si es correcta o no.

### 4.3.2.3 GESTOR DE PREGUNTAS Y RESPUESTAS

Este componente contiene todo lo relativo a la visualización del listado de preguntas y de las propias preguntas con respuestas, así como de la funcionalidad necesaria para capturar las respuestas del alumno y enviarlas al servidor.

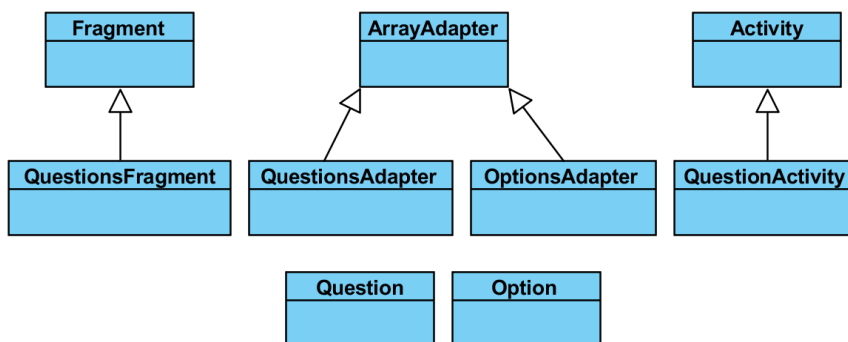


Figura 22 - Diagrama de clases del componente Gestor de preguntas y respuestas

- QuestionsFragment: clase asociada al layout *fragment\_questions*. Contiene la funcionalidad para recuperar el listado de preguntas y cargar una concreta cuando el alumno la selecciona.
- QuestionsAdapter: Adapter para mostrar las preguntas en una lista. Utiliza el layout *list\_question\_row* para cada uno de los elementos que compone la lista.
- Question: es la clase que utiliza *QuestionsAdapter* para crear diferentes objetos asociados a los elementos del listado (contiene todos los datos de la pregunta, con las respuestas incluidas).
- QuestionActivity: clase asociada al layout *activity\_question*. Se encarga de pasar a la vista los datos de una pregunta con sus opciones de respuesta y de enviar la respuesta de vuelta a *SELQuizServer*.
- OptionsAdapter: es el Adapter correspondiente al listado de opciones. Controla que el alumno pueda seleccionar una o varias opciones dependiendo de si la pregunta es de respuesta simple o de respuesta múltiple.
- Option: clase utilizada en *OptionsAdapter* para establecer los datos asociados a cada una de las respuestas.

#### 4.3.2.4 GESTOR DE NOTIFICACIONES

En este componente se incluye la funcionalidad necesaria para conectar a la aplicación con el servicio de notificaciones GCM. En la documentación oficial de Android se ofrece el código necesario para introducirlo en una aplicación desarrollada que quiera implementar los servicios mencionados. Únicamente hacen falta pequeñas modificaciones para adaptarlo a la aplicación en cuestión. Se incluyen dos clases en este punto:

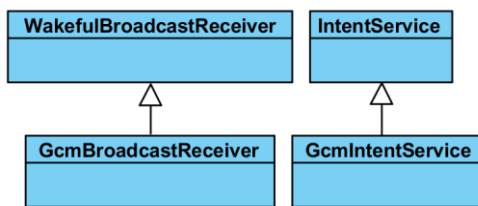


Figura 23 - Diagrama de clases del componente Gestor de notificaciones

- GcmBroadcastReceiver: inicia el servicio de recepciones en segundo plano. Cuando se reciba una, aunque la aplicación no esté iniciada, el dispositivo mostrará la notificación.
- GcmIntentService: es la especificación del servicio que se ejecutará en segundo plano. Cuando se recibe una notificación del servidor del GCM, toma el mensaje contenido en ella y configura, junto a una serie de parámetros, la notificación que aparecerá en la pantalla del dispositivo Android.

#### 4.3.2.5 CONECTOR CON SELQUIZSERVER

La aplicación mClassTest que se desarrolla en este proyecto se conectará con el servidor SELQuizServer para obtener información de las sesiones, suscribirse y cancelar la suscripción a las mismas, recibir preguntas y enviar respuestas. Este componente contiene todo lo relativo a la conexión entre la aplicación y el servidor SELQuizServer. Incluye una única clase, que contendrá una serie de métodos para realizar las diferentes llamadas posibles al SELQuizServer. Cualquier requerimiento que se quiera hacer al servidor, desde cualquier clase de la aplicación, se realizará llamando al correspondiente método de esta clase.

Cada uno de los métodos contendrá una llamada HTTP diferente. Recibirá por parámetros los datos que se deban enviar en la llamada y devolverá los datos remitidos por el servidor (en formato JSON). Al ser una tarea que puede tomar cierto tiempo, variable en función del estado del servidor y de la calidad de la conexión a Internet del dispositivo, las invocaciones a métodos de esta clase deben realizarse de forma asíncrona. La forma más sencilla de implementar estas llamadas será con la clase AsyncTask, que permite crear hilos de ejecución evitando bloquear el proceso principal de la aplicación.

#### 4.3.3 EXTENSIÓN PARA MOODLE

En la extensión para Moodle se implementan una serie de funciones que serán accesibles como servicios web, dejando al margen la configuración del protocolo en el servidor, del UDDI o del WSDL. En cambio, sí se ha decidido qué protocolo utilizar, puesto que es necesario para establecer el formato de parámetros y valores de retorno de las funciones. Se ha tomado XML-RPC, debido a que la simplicidad de este protocolo es suficiente para cubrir los requisitos y está profundamente documentado [32].

La estructura de una extensión básica forma un conjunto de directorios y ficheros en lenguaje PHP. El directorio principal contendrá:

- **externallib.php:** contiene la implementación de las distintas funciones. Cada función consta de dos métodos adicionales, uno para comprobar los datos recibidos por parámetros y otro para los devueltos.
- **version.php:** contiene un campo con la fecha de último cambio. Si este valor es modificado, Moodle lo detectará y dará automáticamente la posibilidad de actualizar la extensión. También contiene otros campos como la versión mínima del PLE para poder ser instalado o un campo para establecer la estabilidad de la extensión.
- Directorio **db:**
  - **services.php:** contiene una definición de los servicios y las funciones.

En la Figura 24 se muestra el diagrama de clases de la extensión para Moodle.

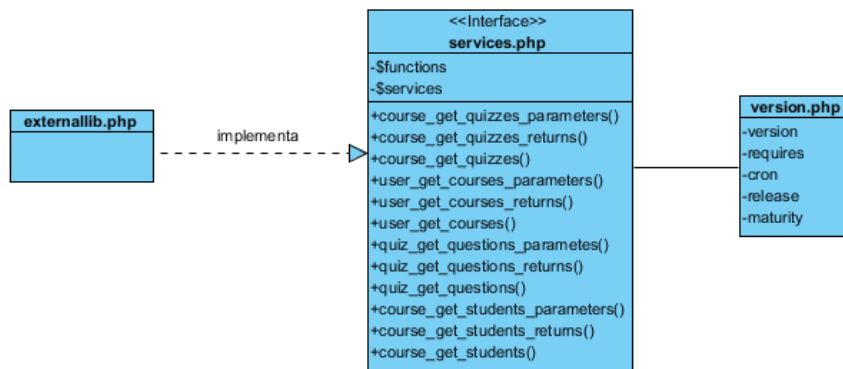


Figura 24 - Diagrama de clases de la extensión para Moodle

Como ya se ha comentado, cada función consta de otras dos funciones para comprobar el tipo de datos que se reciben por parámetro o que son devueltos. Se procede a definir cada función:

- **course\_get\_quizzes()**
  - Parámetros:
    - \$courseid: identificador del curso del que se quieren obtener los cuestionarios
  - Salida: información de cada cuestionario del curso pasado por parámetro. Contiene datos como su identificador, el nombre, fechas de apertura y cierre o la contraseña de realización.



- Descripción: Con esta función se obtiene información general sobre todos los cuestionarios que contiene un curso o asignatura concreta. Esta información no contiene las preguntas o información sobre la respuesta correcta.
- **user\_get\_courses()**
  - Parámetros: no recibe parámetros.
  - Salida: listado de cursos en los que está incluido el usuario que ha llamado a la función.
  - Descripción: el objetivo de esta función es obtener información acerca de todos los cursos en los que se incluye a un determinado profesor. Debido a que ya existe una función interna que realiza esta función (*enrol\_get\_my\_courses*), no es necesario realizar una consulta en base de datos, sino una llamada a la función comentada.
- **quiz\_get\_questions()**
  - Parámetros:
    - \$quizid: identificador del cuestionario del que se quieren obtener las preguntas.
  - Salida: información que contiene cada pregunta, sus posibles respuestas e información para la corrección.
  - Descripción: esta función permite obtener un cuestionario completo, con todas sus preguntas, las posibles respuestas (se admiten preguntas de tipo verdadero/falso o de respuesta múltiple) e información para la calificación.
- **course\_get\_students()**
  - Parámetros:
    - \$courseid: identificador del curso del que se quieren obtener los estudiantes
  - Salida: listado de estudiantes matriculados en el curso.
  - Descripción: con esta función se obtiene un listado de alumnos, con su correo y otra información. De esta forma se podrá establecer una asociación entre usuarios de la mSuite y de Aula Global.

## 4.4 FUNCIONAMIENTO DEL SISTEMA

En este apartado se muestran una serie de diagramas de secuencia. Estos permitirán describir el funcionamiento de la aplicación Android, mostrando las interacciones entre las clases descritas anteriormente, la interfaz gráfica y el alumno.

Las acciones producidas cuando se abre la aplicación se repiten en la mayoría de los diagramas generados. Se caracterizan por abrir la aplicación por parte del alumno, lo que desencadena que el dispositivo cargue la aplicación. El primer elemento a ejecutar en la aplicación es la clase MainActivity, que carga una vista inicial en la interfaz con el nombre de la aplicación. Durante la visualización de esta vista, la aplicación comprueba que el dispositivo tiene instalados los Servicios de Google Play, verifica si la aplicación está conectada a GCM y si el alumno ya se ha identificado en la aplicación.

Se han generado cuatro diagramas.

### 4.4.1 IDENTIFICACIÓN CON TWITTER

Se ha elaborado un diagrama (Figura 25) que muestra el proceso de inicio de la aplicación y posterior identificación con Twitter. Esto se realizará cada vez que el alumno abra por primera vez la aplicación recién instalada o cuando haya cerrado sesión.

Una vez cargada la vista de título de la aplicación (paso número 3 del diagrama), la aplicación comprobará si el alumno ya se ha identificado previamente. En caso negativo, la aplicación mostrará una vista con un botón para identificarse con Twitter. Cuando el alumno pulse el botón (paso número 6), se cargará un navegador web para que el alumno se identifique.

Cuando el alumno se haya identificado correctamente con su cuenta de Twitter, se mostrará una vista para solicitar el NIA al alumno (paso número 13). Una vez introducido, la aplicación procederá a registrar al alumno en SELQuizServer. Este, como respuesta, devolverá el listado de sesiones disponibles, que la aplicación mostrará en la vista principal (paso número 21).

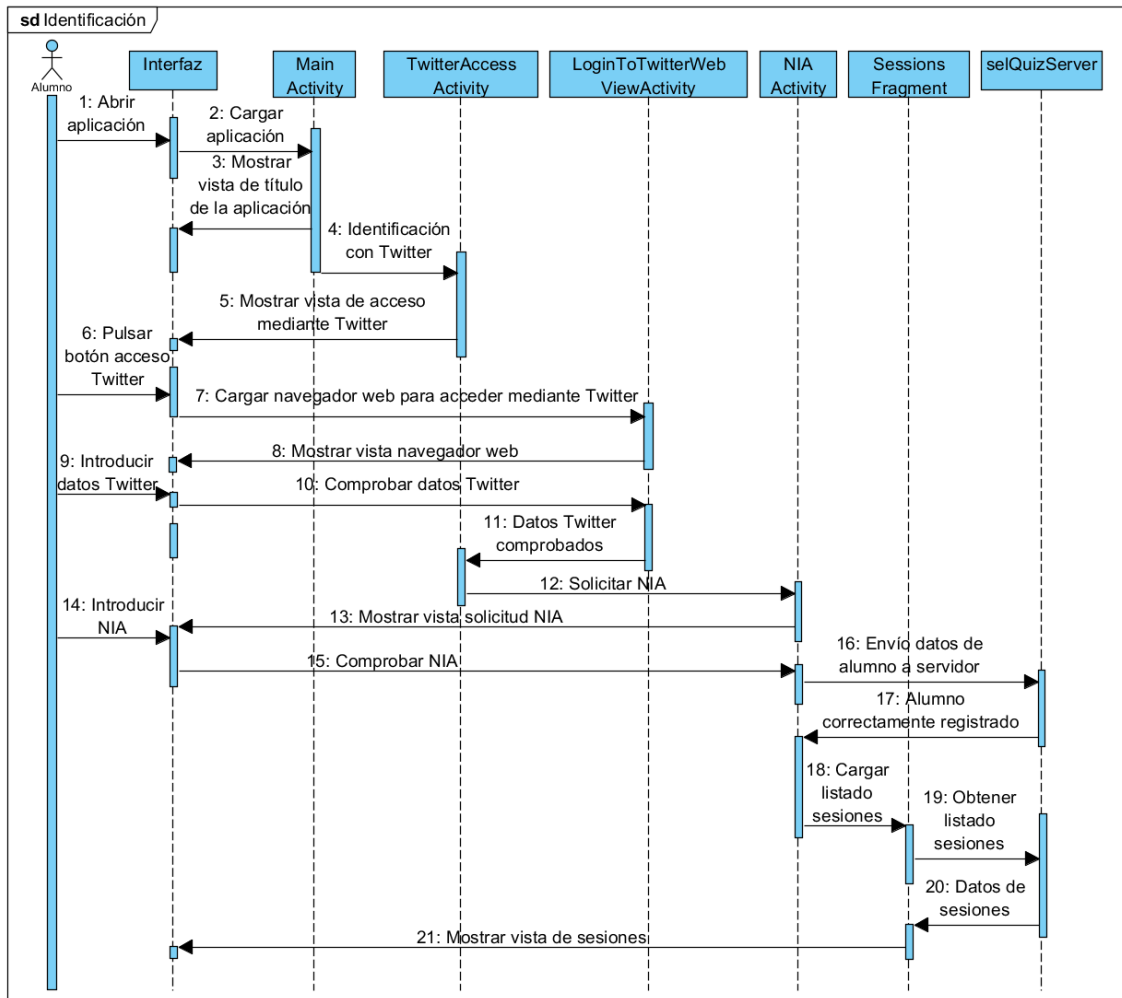


Figura 25 - Diagrama de secuencia 1 - Identificación con Twitter

#### 4.4.2 LISTAR SESIONES

En la Figura 26 se muestra un diagrama que contiene todas las interacciones de la aplicación en lo relativo a las sesiones, tanto su obtención del servidor como la suscripción y cancelación de suscripción a las mismas. En los pasos 1 al 8 se muestran las interacciones relativas a la apertura de la aplicación y visualización del listado de sesiones. A continuación se han dispuesto otros tres subdiagramas que muestran las posibles acciones a realizar sobre el listado de sesiones:

- Suscribir sesión: cuando el alumno pulsa sobre el Switch de una de las sesiones (no protegidas) para suscribirse, la aplicación realiza una llamada al SELQuizServer y se actualiza el estado del Switch para indicar que está suscrita. Si hubiera una sesión ya suscrita, se realizará también una llamada al servidor para cancelar esa suscripción.
- Cancelar suscripción a sesión: cuando el alumno selecciona una sesión para cancelar la suscripción, la aplicación hará la correspondiente llamada al servidor y actualizará la vista.

- Suscribirse sesión protegida: en caso de que la sesión a suscribir esté protegida por contraseña, la aplicación mostrará un Dialog (paso número 22) para solicitar al alumno la contraseña. Una vez introducida, se realizará la llamada correspondiente al servidor y se actualizará la vista.

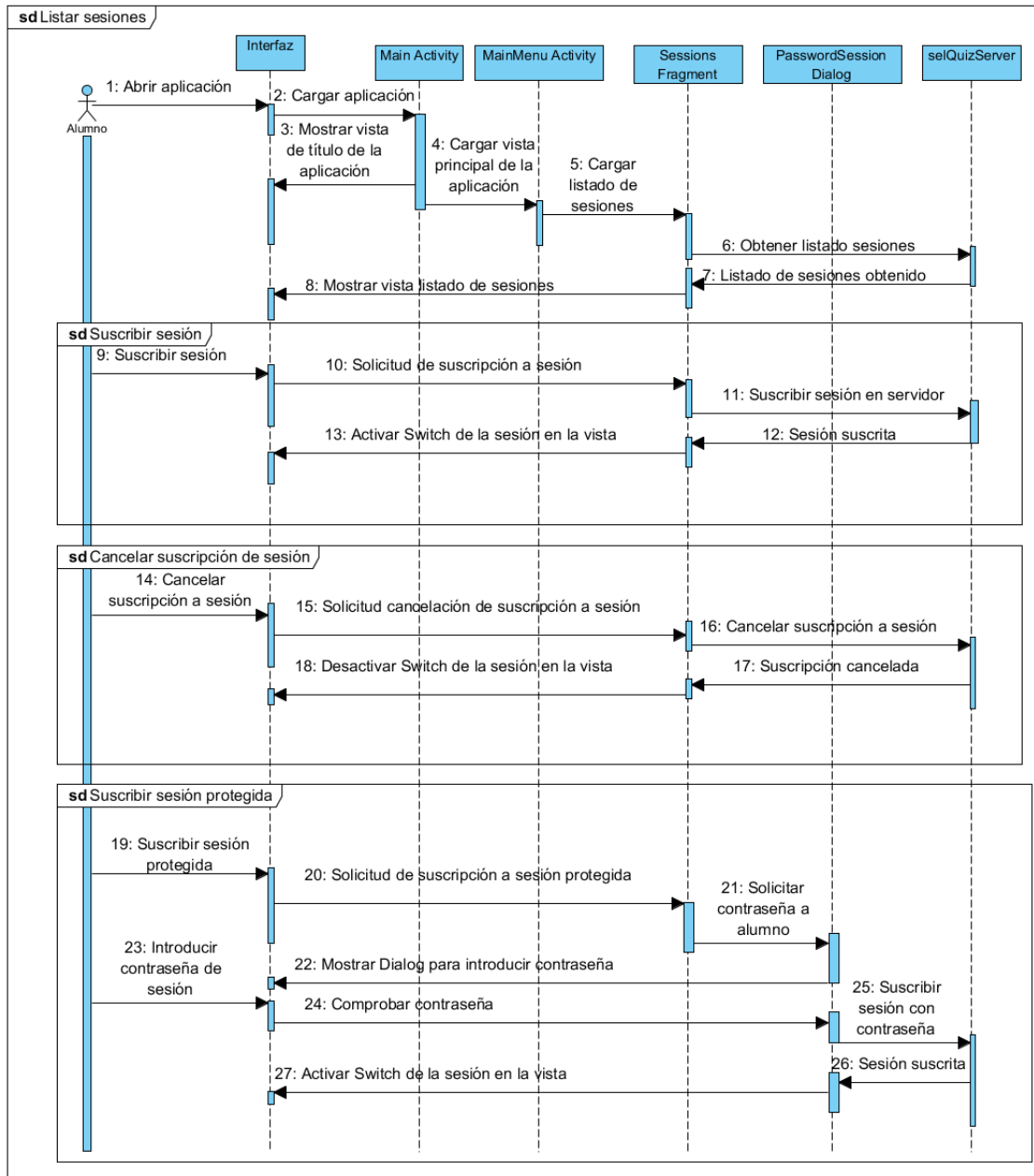


Figura 26 - Diagrama de secuencia 2 - Listar sesiones

### 4.4.3 LISTAR PREGUNTAS

La Figura 27 muestra el diagrama de secuencia correspondiente a las interacciones relacionadas con las preguntas, tanto para la visualización del listado como para la respuesta a ellas.

Los primeros pasos que se muestran en el diagrama son los comunes a los demás diagramas, correspondientes a la apertura de la aplicación y visualización del listado de sesiones. Una vez realizados estos primeros pasos, la aplicación mostrará la vista principal de la aplicación, en la que el alumno podrá seleccionar la opción relativa al listado de preguntas en el menú lateral (paso número 9). La aplicación realizará la llamada al servidor para obtener los datos y los mostrará en la interfaz (paso número 14).

También se incluye un subdiagrama que muestra las interacciones referentes a responder a una pregunta. El alumno seleccionará una pregunta (no respondida previamente), que se visualizará con sus posibles opciones de respuesta (paso número 17). El alumno seleccionará una o varias opciones, la aplicación registrará en el servidor la respuesta y mostrará de nuevo la vista del listado de preguntas (paso número 23).

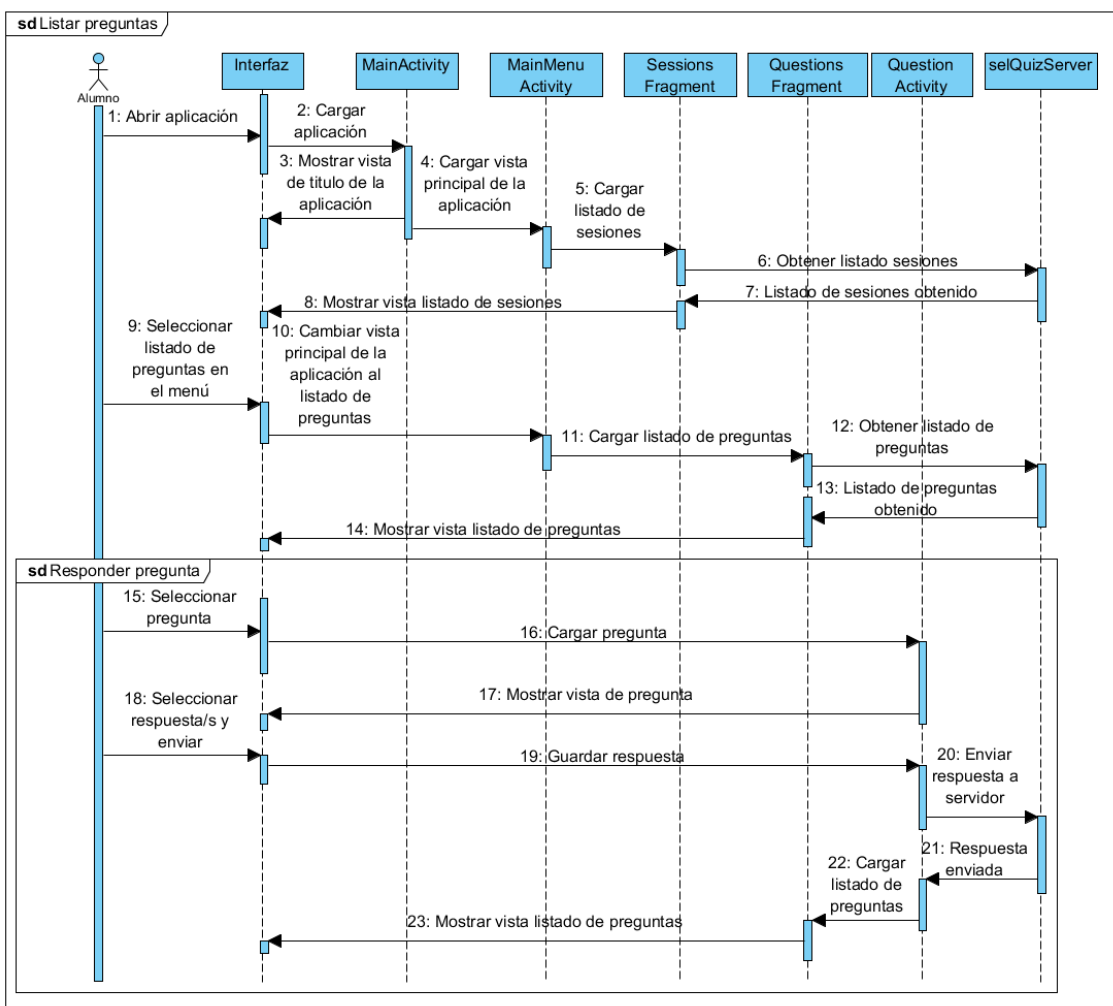


Figura 27 - Diagrama de secuencia 3 - Listar preguntas

#### 4.4.4 RECIBIR NOTIFICACIÓN

En la Figura 28 se muestra el diagrama con las interacciones de la aplicación Android en el proceso de recepción de una notificación. Al contrario que en los anteriores diagramas de secuencia, en este la interacción no la inicia el alumno. El servidor envía una notificación hasta la aplicación a través del servicio GCM. En el momento en que el servicio en segundo plano de la aplicación la detecte, mostrará el mensaje en la barra de notificaciones del dispositivo (paso número 2). Cuando el alumno pulse sobre la notificación, la aplicación se abrirá, siguiendo los mismos pasos que se han explicado anteriormente.

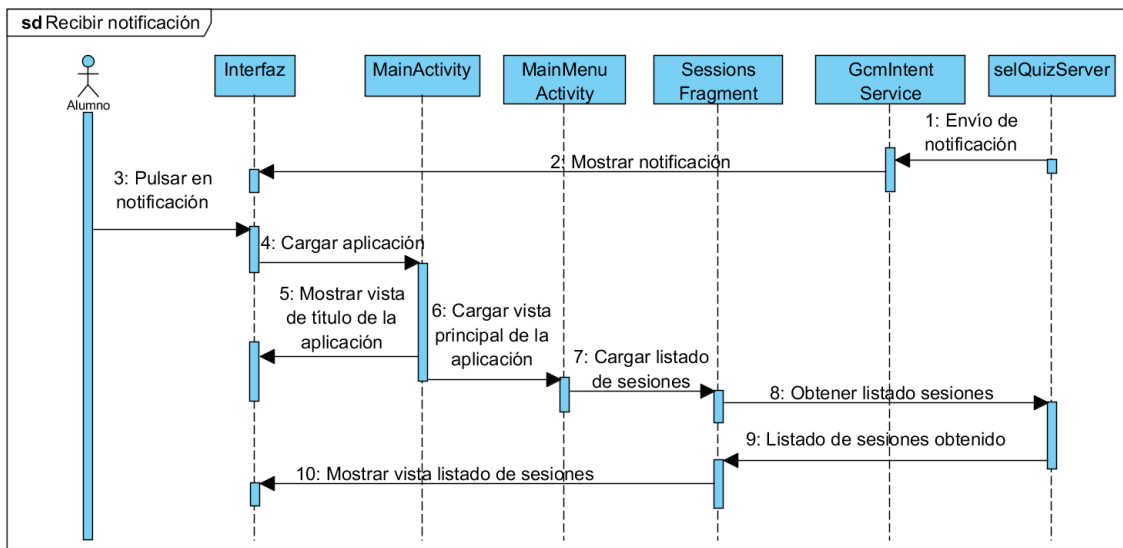
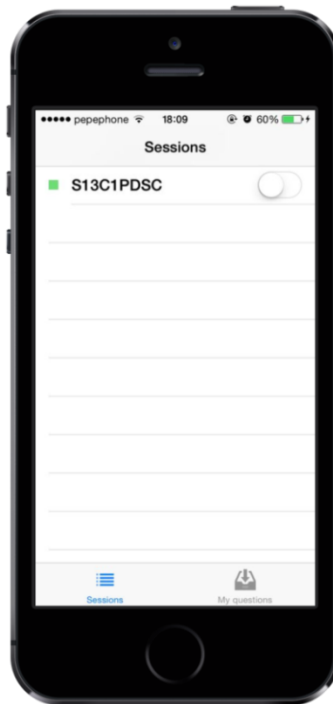


Figura 28 - Diagrama de secuencia 4 - Recibir notificación

#### 4.5 DISEÑO DE LA INTERFAZ GRÁFICA DE LA APLICACIÓN ANDROID

El diseño de la interfaz gráfica deberá ser lo más cercano posible al de la ya existente aplicación para iOS. De esta forma, se conseguirá un diseño unificado, haciendo que las aplicaciones sean muy similares estéticamente. Se evitarán así confusiones entre los alumnos que puedan utilizar la aplicación en distintos dispositivos. Además, esto debe ser compatibilizado con las normas de estilo que propone la documentación oficial de Android, que contiene ciertas diferencias con la de iOS.

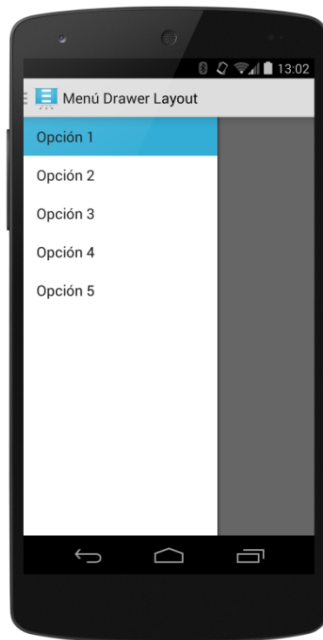
El diseño de la aplicación para iOS se caracteriza por su simplicidad, facilitando su uso sin conocimiento previo de la misma. En la Figura 29 se muestra la vista del listado de sesiones en la aplicación mClassTest para iOS. Como se puede apreciar, se caracteriza por un predominio de colores claros. En la imagen también podemos ver el menú que permite cambiar entre la vista del listado de sesiones y de preguntas. Se encuentra en la parte inferior de la pantalla, y es fijo, no se oculta en ningún momento.



**Figura 29 - Vista del listado de sesiones en la aplicación mClassTest para iOS**

La aplicación para Android que se desarrolla en este proyecto mostrará también un listado de sesiones, cada una de ellas acompañada de un interruptor para suscribirse o cancelar la suscripción a la sesión. El diseño del interruptor variará de un sistema operativo a otro. En el caso de iOS, se utiliza un estilo redondeado, mientras que en Android, los interruptores son de forma rectangular. El utilizar este estilo de interruptor en la aplicación, se debe a que es un diseño propio de la plataforma, utilizado por todas las aplicaciones desarrolladas para ella.

En el caso del menú inferior de la aplicación iOS, debe ser sustituido en la nueva aplicación para Android, ya que al igual que con el interruptor, no es propio de la plataforma Android. Por el contrario, se utilizará un DrawerLayout, como el que se muestra en la Figura 30, muy extendido en aplicaciones de este sistema. Consiste en un menú oculto, residente en el margen izquierdo de la pantalla. Para abrirlo, basta con arrastrar el dedo del lateral hacia el centro de la pantalla o pulsar sobre las tres barras paralelas que se muestran en el lado izquierdo de la barra superior. Contendrá las dos opciones que contemplaba el de la aplicación para iOS más una tercera opción para cerrar sesión en la aplicación.



**Figura 30 - Ejemplo de DrawerLayout en Android**

Entre el resto de vistas, se implementará una a modo de presentación con el nombre de la aplicación durante el arranque de la aplicación. Mientras tanto, se comprobará si el dispositivo tiene instalados los Servicios de Google Play y si la aplicación está registrada en GCM. En el resto de vistas se seguirá el mismo planteamiento que en las de la aplicación iOS. Sin embargo, es necesario introducir dos cambios. Por un lado se introduce una nueva vista para solicitar el NIA al alumno, una vez se haya identificado en Twitter. El segundo cambio consiste en la vista donde aparecerá una pregunta con sus posibles opciones de respuesta. Cuando se trate de preguntas de respuesta simple, se mostrará un Radio Button para cada una de las opciones (al igual que en la respectiva para iOS), mientras que si es de respuesta múltiple, las opciones irán acompañadas de un Checkbox, que indicará que se puede seleccionar más de una.

#### 4.5.1 PROTOTIPO DE LA APLICACIÓN ANDROID

Partiendo de las consideraciones realizadas en el punto anterior sobre el diseño de la interfaz gráfica de la aplicación Android, se ha realizado un prototipo de la misma. Con él se obtendrá una visión preliminar de las vistas de la aplicación, para así tener una idea clara del resultado visual que se quiere alcanzar con la implementación de la aplicación.

El prototipo ha sido creado con la herramienta Fluid UI<sup>5</sup>, que ofrece un entorno donde crear prototipos para distintos sistemas operativos y con multitud de elementos propios de cada uno (botones, menús, iconos...etc.). A continuación se muestran capturas de las diferentes vistas creadas en el prototipo.

---

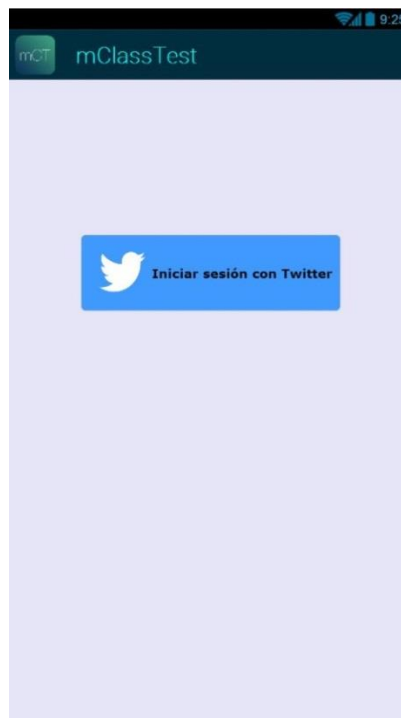
<sup>5</sup> <https://www.fluidui.com/>





**Figura 31 - Prototipo de la aplicación Android - Vista de presentación**

En la Figura 31 se puede apreciar la vista que se mostrará al iniciar la aplicación, a modo de presentación de la misma. No permite la interacción, desapareciendo pasados unos segundos.



**Figura 32 - Prototipo de la aplicación Android - Inicio de sesión con Twitter**

En la Figura 32 se muestra la vista que se visualizará cuando el alumno no se haya identificado previamente. Constará de un único botón para acceder con el navegador a una web de Twitter para realizar la identificación.

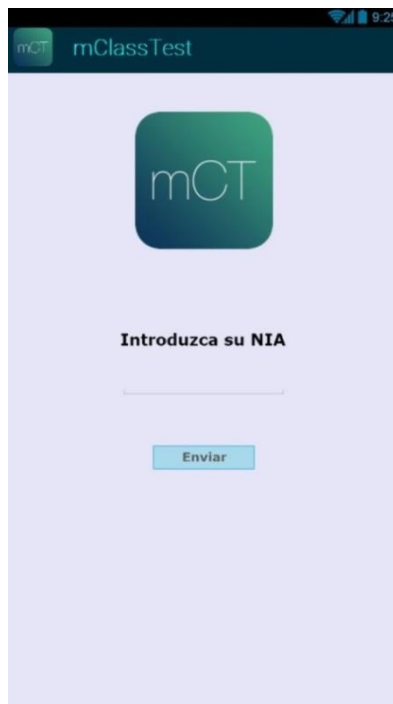


Figura 33 - Prototipo de la aplicación Android - Vista de introducción del NIA

En la Figura 33 se muestra la vista que solicita la introducción del NIA al alumno. Se mostrará una vez realizada la identificación en Twitter.



Figura 34 - Prototipo de la aplicación Android - Vista del listado de sesiones

La Figura 34 contiene la vista en la que se detalla el listado de sesiones, indicándose en cada una si está suscrita o no.

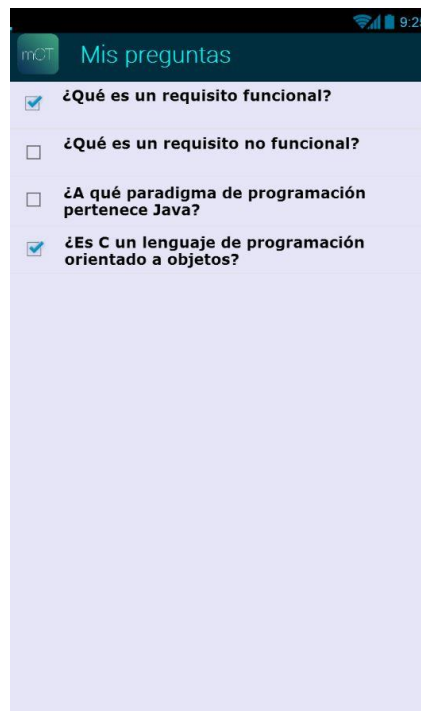


Figura 35 - Prototipo de la aplicación Android - Vista del listado de preguntas

En la Figura 35 se muestra la vista del listado de preguntas, donde se indica para cada una si ya ha sido respondida.

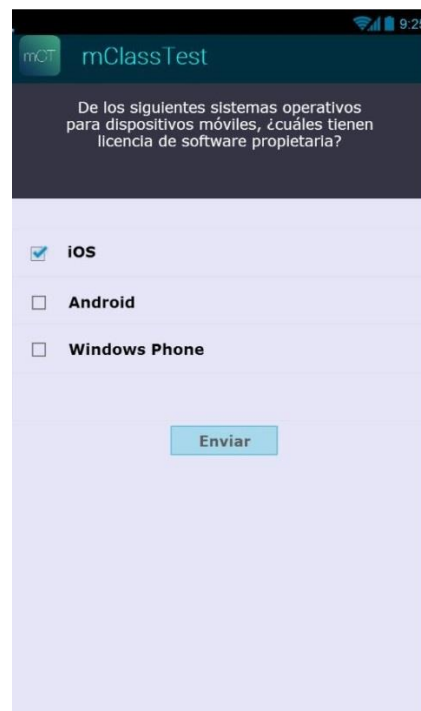
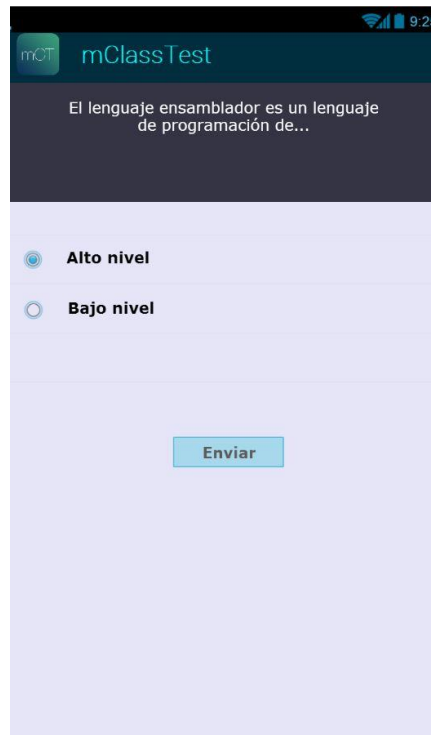


Figura 36 - Prototipo de la aplicación Android - Pregunta de respuesta múltiple

La Figura 36 muestra la vista destinada a dar respuesta a una pregunta de respuesta múltiple. En la parte superior se mostrará el enunciado de la pregunta, y debajo las posibles respuestas. Cada una irá acompañada de un Checkbox, que indicará si esa respuesta se ha seleccionado.



**Figura 37 - Prototipo de la aplicación Android - Pregunta de respuesta simple**

Para acabar de detallar las diferentes vistas del dispositivo, en la Figura 37 se muestra la propuesta para preguntas de respuesta simple. Cada posible respuesta está acompañada de un botón de radio para seleccionar una única respuesta.

# 5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

En este apartado se describe el entorno de desarrollo utilizado para la fase de implementación del proyecto, los problemas encontrados durante esa etapa y, por último, se elabora un plan de pruebas con el que comprobar el correcto funcionamiento del sistema.

## 5.1 IMPLEMENTACIÓN

La implementación varía en función del elemento del conjunto de aplicaciones mSuite a desarrollar. En el caso de la extensión para Moodle, tal y como se ha descrito en el apartado de Diseño, su implementación se basa en la construcción de unos ficheros en PHP con las nuevas funciones a incorporar a Moodle. Para esto no es necesario ningún entorno de desarrollo especial, sirviendo un simple editor de textos. Lo mismo sucede con el servidor SELQuizServer, donde se han realizado una serie de modificaciones en ficheros codificados en Java.

Para el desarrollo de la aplicación mClassTest para Android sí es necesario utilizar un entorno específico de desarrollo. El utilizado en este proyecto ha sido Android SDK, el cual provee de todas las herramientas necesarias para la implementación tanto interna como externa (interfaz gráfica). La parte de controlador de la aplicación se construye sobre lenguaje Java, mientras que las vistas se definen en ficheros XML.

En cuanto al equipo informático utilizado para el desarrollo del proyecto, este consta de las siguientes características:

- Procesador AMD Phenom II X6 1055T 2.80 GHz
- Memoria RAM de 8GB
- Sistema operativo Windows 7 Professional 64 bits Service Pack 1

### 5.1.1 PROBLEMAS ENCONTRADOS EN LA IMPLEMENTACIÓN

Durante la etapa de implementación, los problemas que más impacto han tenido son los relativos a la versión mínima funcional de Android. Para poder hacer compatible la aplicación con Android 2.3.3 (nivel 10 de API) se han tenido que adoptar ciertas consideraciones. Algunas de estas pasan por métodos de distintas clases del sistema que únicamente son válidos a partir de cierta versión, por lo que han debido ser reemplazados por otros que cumplieran la misma funcionalidad. Por otro lado, para poder usar el Switch de Android en cualquier versión, ha sido necesario introducir una librería externa denominada Android Switch Widget Backport que posee licencia Apache. El motivo de usar esta librería es que versiones de Android inferior a la de nivel de API 14 no permiten el uso de este elemento.

## 5.2 PRUEBAS DEL SISTEMA

Con vistas a comprobar el correcto funcionamiento del sistema y su adecuación a los requisitos establecidos, se plantea un plan de pruebas. Con él se realizará una evaluación completa del sistema, comprobando punto por punto que el resultado es el esperado. El plan consistirá en un conjunto de pruebas detalladas con una secuencia de pasos que deben realizarse para cumplir cierto objetivo, caso en el que se dará la prueba por válida.

### 5.2.1 ENTORNO DE PRUEBAS DEL SISTEMA

El equipo utilizado para la ejecución de las pruebas es el mismo que el descrito para la implementación.

Al igual que en los apartados anteriores, es necesario realizar una distinción en función del elemento a desarrollar. Para realizar pruebas con la extensión para Moodle ha sido necesario montar un servidor con una instalación de Moodle, similar a la de Aula Global. Debido a la alta carga computacional de este LMS, no ha sido posible encontrar ningún servidor público gratuito con las características suficientes para hacerlo funcionar, por lo que se ha realizado una instalación de un servidor local. Los elementos que conforman el entorno de pruebas para la extensión para Moodle son los siguientes:

- Servidor WAMP instalado con el programa EasyPHP 14.1
- Moodle 2.4.8

El servidor SELQuizServer también se ha instalado en un servidor local, pero sobre una máquina virtual con sistema operativo Ubuntu. De esta forma, permitirá operar al mismo tiempo el entorno Android SDK en Windows y el servidor SELQuizServer, pudiendo hacer llamadas al mismo desde la aplicación Android. La máquina virtual consta de las siguientes características:

- Máquina virtual VMware Workstation 9.0.0
- Sistema operativo instalado en la máquina virtual Ubuntu 12.04 LTS 64 bits
- Servidor Apache 2.2.22
- Play Framework 2.1.1
- Java JRE 7

Por último, para la aplicación Android, se realizarán pruebas con dispositivos de diferentes características. Debido a que la aplicación utiliza los Servicios de Google Play, no se pueden utilizar emuladores (no admiten los mencionados servicios), requiriendo de dispositivos físicos. Se han tomado cuatro terminales con diferentes características, abarcando un amplio margen de opciones facilitando la detección de posibles problemas:

- Nexus 5
  - Versión de Android: 4.4.2 KitKat
  - Pantalla: 4,95 pulgadas con resolución 1920 x 1080 px
  - RAM: 2GB
  - Procesador: Qualcomm Snapdragon 800 a 2.3GHz
- Nexus 7 (segunda generación)
  - Versión de Android: 4.4.2 KitKat
  - Pantalla: 7 pulgadas con resolución 1920 x 1200 px
  - RAM: 2GB
  - Procesador: Qualcomm Snapdragon S4 Pro a 1.5 GHz
- Sony Ericsson Xperia Neo
  - Versión de Android: 4.0.4 Ice Cream Sandwich
  - Pantalla: 3,7 pulgadas con resolución 854 x 480 px
  - RAM: 512 MB
  - Procesador: Qualcomm Snapdragon 1GHz Scorpion
- Orange Monte Carlo
  - Versión de Android: 2.3.5 Gingerbread
  - Pantalla: 4,3 pulgadas con 800 x 480 px
  - RAM: 512 MB
  - Procesador Qualcomm Snapdragon 800 MHz

### 5.2.2 DESCRIPCIÓN DE LOS ATRIBUTOS DE LAS PRUEBAS DEL SISTEMA

- Código: Identificación unívoca abreviada de la prueba. Se construye mediante las letras “PS” seguidas de un guion y de tres dígitos. Por ejemplo PS-001.
- Nombre: Identificación extendida de la prueba del sistema.
- Requisito Fuente: Requisitos que verifica la prueba.
- Descripción: Descripción básica de la prueba.
- Procedimiento de prueba: Secuencia de pasos que deben realizarse en la prueba.
- Resultado esperado: Condiciones que deben darse en el sistema para dar la prueba como superada.

Las pruebas se presentarán en tablas como la Tabla 55, suministrada a modo de ejemplo.

Especificación de prueba del sistema	
<b>Código</b>	<b>Requisito fuente</b>
<b>Nombre</b>	
<b>Descripción</b>	
<b>Procedimiento de prueba</b>	
<b>Resultado esperado</b>	

Tabla 55 - Tabla de ejemplo de especificación de prueba del sistema

### 5.2.3 ESPECIFICACIÓN DE LAS PRUEBAS DEL SISTEMA

Especificación de prueba del sistema	
<b>Código</b>	PS-001 <b>Requisito fuente</b> RF-001, RF-002, RF-005
<b>Nombre</b>	Identificación en Twitter y registro en SELQuizServer
<b>Descripción</b>	Se comprueba que al iniciar el alumno la aplicación sin haberse identificado previamente, el sistema le solicita los datos de Twitter, que una vez validados serán guardados en el servidor.
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación mClassTest para Android.</li> <li>2. Pulsar el botón para identificarse con Twitter.</li> <li>3. Cuando se haya cargado el navegador web, introducir los datos de una cuenta de Twitter y pulsar el botón para iniciar sesión.</li> <li>4. Introducir un número NIA cuando la aplicación lo solicite y pulsar en el botón de enviar.</li> </ol>
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Al abrir de nuevo la aplicación (sin haber cerrado sesión), aparece directamente la vista del listado de sesiones.</li> <li>• Un nuevo alumno ha quedado registrado en el servidor SELQuizServer con su NIA y el nombre de usuario en Twitter introducidos en el procedimiento de la prueba.</li> </ul>

Tabla 56 - Prueba del sistema PS-001 - Identificación en Twitter y registro en el SELQuizServer



Especificación de prueba del sistema			
<b>Código</b>	PS-002	<b>Requisito fuente</b>	RF-003, RF-005
<b>Nombre</b>	Recuperación y visualización de sesiones		
<b>Descripción</b>	En esta prueba se verifica que se obtienen las sesiones definidas en el servidor, las cuales se muestran correctamente en un listado en el que se muestran diferenciadas las protegidas por contraseña.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear un número determinado de sesiones en el servidor SELQuizServer. Debe ser un valor superior al número de sesiones que se pueden visualizar simultáneamente en el listado de sesiones y al menos una de ellas debe estar protegida por contraseña.</li> <li>2. Abrir la aplicación (en la que ya se está identificado y que por tanto mostrará la vista de sesiones).</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Los elementos visualizados en la vista de sesiones se ajustan a los incorporados previamente en el SELQuizServer, pudiendo arrastrar la vista para observar sesiones que no se muestran por falta de espacio.</li> <li>• Las sesiones definidas en el servidor como protegidas por contraseña quedarán diferenciadas en el listado.</li> <li>• En todas las sesiones, el Switch estará en la opción desactivado, para indicar que el alumno no está suscrito.</li> </ul>		

Tabla 57 - Prueba del sistema PS-002 - Recuperación y visualización de sesiones

Especificación de prueba del sistema			
<b>Código</b>	PS-003	<b>Requisito fuente</b>	RF-004, RF-005, RF-006
<b>Nombre</b>	Suscripción de sesión		
<b>Descripción</b>	Verificación que al suscribirse a una sesión en la aplicación, este proceso se registra en el servidor, de forma que se puedan obtener únicamente las preguntas asociadas a la sesión suscrita visualizándose correctamente.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear una sesión en el SELQuizServer.</li> <li>2. Crear un número de preguntas, superior al de máximas preguntas que se pueden visualizar en la vista simultáneamente y asociarlas a la sesión creada.</li> <li>3. Abrir la aplicación.</li> <li>4. Suscribirse a la sesión creada.</li> <li>5. Seleccionar la opción "Mis preguntas" del menú de opciones.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Los elementos visualizados en la vista de preguntas se corresponden a las preguntas establecidas en el servidor para la sesión suscrita.</li> <li>• La vista del listado de preguntas permite desplazarse para observar preguntas que no se muestran por falta de espacio.</li> <li>• Al suscribir la sesión, el Switch correspondiente debe quedar marcado como activo.</li> <li>• En el servidor se ha registrado la suscripción a la sesión.</li> </ul>		

Tabla 58 - Prueba del sistema PS-003 - Suscripción de sesión

Especificación de prueba del sistema			
<b>Código</b>	PS-004	<b>Requisito fuente</b>	RF-007
<b>Nombre</b>	Solicitud de contraseña para sesiones protegidas		
<b>Descripción</b>	Comprobación de que para poder suscribir una sesión protegida es necesario introducir la contraseña correcta.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear una sesión protegida en el servidor, definiendo una contraseña.</li> <li>2. Abrir la aplicación Android.</li> <li>3. Pulsar sobre el Switch de la sesión creada en el servidor para suscribirse.</li> <li>4. En el Dialog introducir una contraseña distinta a la definida en el servidor y pulsar el botón de “Aceptar”.</li> <li>5. Cuando la aplicación vuelva a solicitar la contraseña, introducir la correcta.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Al introducir la contraseña incorrecta, el sistema mostrará un mensaje avisando de tal efecto, solicitando una nueva contraseña.</li> <li>• Cuando se introduzca la contraseña correcta, el sistema mostrará la vista de sesiones con el Switch de la sesión suscrita activado.</li> </ul>		

Tabla 59 - Prueba del sistema PS-004 - Solicitud de contraseña para sesiones protegidas

Especificación de prueba del sistema			
<b>Código</b>	PS-005	<b>Requisito fuente</b>	RF-008
<b>Nombre</b>	Respuesta a preguntas		
<b>Descripción</b>	Comprobación de que la aplicación Android permite responder preguntas simples y de respuesta múltiple creadas en el servidor y que este registra correctamente la respuesta dada por el alumno.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear una sesión en el servidor.</li> <li>2. Crear una pregunta de respuesta simple y asociarla a la sesión creada.</li> <li>3. Crear una pregunta de respuesta múltiple y asociarla a la sesión creada.</li> <li>4. Abrir la aplicación.</li> <li>5. Suscribirse a la sesión creada.</li> <li>6. Ir a la vista de preguntas.</li> <li>7. Pulsar sobre la pregunta de respuesta simple.</li> <li>8. Seleccionar una respuesta y pulsar el botón de enviar.</li> <li>9. Pulsar sobre la pregunta de respuesta múltiple.</li> <li>10. Seleccionar dos respuestas y pulsar el botón de enviar.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• En la vista de la pregunta simple, únicamente se podrá seleccionar una respuesta simultáneamente.</li> <li>• En la vista de la pregunta múltiple, se podrán seleccionar varias respuestas simultáneamente.</li> <li>• En el servidor han quedado registradas correctamente las respuestas dadas en la aplicación para cada pregunta.</li> </ul>		

Tabla 60 - Prueba del sistema PS-005 - Respuesta a preguntas

Especificación de prueba del sistema			
<b>Código</b>	PS-006	<b>Requisito fuente</b>	RF-009
<b>Nombre</b>	Cierre de sesión		
<b>Descripción</b>	Verificación de que la aplicación da la opción de cerrar sesión, de forma que al abrir de nuevo la aplicación pida identificarse de nuevo.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación (estando ya identificado en ella).</li> <li>2. Abrir el menú y pulsar la opción "Cerrar sesión".</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• La aplicación se reiniciará y volverá a solicitar identificarse con Twitter.</li> </ul>		

Tabla 61 - Prueba del sistema PS-006 - Cierre de sesión

Especificación de prueba del sistema			
<b>Código</b>	PS-007	<b>Requisito fuente</b>	RF-010
<b>Nombre</b>	Recepción de notificaciones		
<b>Descripción</b>	Comprobación del correcto funcionamiento del servicio de notificaciones de la aplicación Android.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Identificarse con Twitter en la aplicación.</li> <li>2. Enviar notificación con un mensaje al GCM con el <i>token</i> del dispositivo sobre el que se está realizando la prueba.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• El dispositivo mostrará en la barra de notificación el mensaje enviado al GCM.</li> </ul>		

Tabla 62 - Prueba del sistema PS-007 - Recepción de notificaciones

Especificación de prueba del sistema			
<b>Código</b>	PS-008	<b>Requisito fuente</b>	RF-011
<b>Nombre</b>	Intercambio de vistas de preguntas y sesiones		
<b>Descripción</b>	Comprobación de que el menú de la aplicación permite intercambiar entre la vista del listado de sesiones y el de preguntas.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación (estando ya identificado en ella).</li> <li>2. Abrir el menú y seleccionar la opción para ver el listado de preguntas.</li> <li>3. Abrir el menú y seleccionar la opción para ver el listado de sesiones.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Al pulsar sobre una de las opciones del menú relativa a las preguntas o a las sesiones, la vista principal de la aplicación mostrará la vista asociada.</li> </ul>		

Tabla 63 - Prueba del sistema PS-008 - Intercambio de vistas de preguntas y sesiones

Especificación de prueba del sistema			
<b>Código</b>	PS-009	<b>Requisito fuente</b>	RF-012
<b>Nombre</b>	Vista predeterminada		
<b>Descripción</b>	Comprobación de que la vista predeterminada de la aplicación al abrirla es la del listado de sesiones.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación (estando ya identificado en ella).</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• La aplicación mostrará la vista del listado de sesiones.</li> </ul>		

Tabla 64 - Prueba del sistema PS-009 - Vista predeterminada

Especificación de prueba del sistema			
<b>Código</b>	PS-010	<b>Requisito fuente</b>	RF-013
<b>Nombre</b>	Imposibilidad de enviar respuesta vacía		
<b>Descripción</b>	Comprobación de que la aplicación no permite enviar la respuesta a una pregunta sin haber seleccionado ninguna opción de respuesta		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear una sesión en el servidor.</li> <li>2. Crear una pregunta de respuesta simple y asociarla a la sesión creada.</li> <li>3. Crear una pregunta de respuesta múltiple y asociarla a la sesión creada.</li> <li>4. Abrir la aplicación.</li> <li>5. Suscribirse a la sesión creada.</li> <li>6. Ir a la vista de preguntas.</li> <li>7. Seleccionar la pregunta de respuesta simple creada.</li> <li>8. Sin seleccionar ninguna opción pulsar sobre el botón de "Enviar".</li> <li>9. Repetir los pasos 6, 7, 8 con la pregunta de respuesta múltiple.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Al pulsar sobre el botón enviar, sin haber seleccionado ninguna respuesta, la aplicación mostrará un error, permaneciendo en la vista de la pregunta con sus posibles respuestas.</li> </ul>		

Tabla 65 - Prueba del sistema PS-010 - Imposibilidad de enviar respuesta vacía

Especificación de prueba del sistema			
<b>Código</b>	PS-011	<b>Requisito fuente</b>	RF-014
<b>Nombre</b>	Suscripción única		
<b>Descripción</b>	Comprobación de que la aplicación Android permite suscribir una única sesión simultáneamente.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear dos sesiones en el servidor.</li> <li>2. Abrir la aplicación.</li> <li>3. Suscribirse a una sesión creada.</li> <li>4. Suscribirse a la otra sesión creada.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Al suscribir de nuevo una sesión, se activará el Switch de esta sesión y se desactivará el de la suscrita anterior.</li> <li>• En el servidor se registra la cancelación de suscripción a la primera sesión y la suscripción a la segunda.</li> </ul>		

Tabla 66 - Prueba del sistema PS-011 - Suscripción única

Especificación de prueba del sistema			
<b>Código</b>	PS-012	<b>Requisito fuente</b>	RF-015
<b>Nombre</b>	Obtención listado de estudiantes de Moodle		
<b>Descripción</b>	Comprobación de que la extensión para Moodle devuelve el listado de estudiantes de un curso cuando se realiza la llamada a la función correspondiente.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear un curso en Moodle.</li> <li>2. Matricular dos alumnos en el curso creado.</li> <li>3. Crear un profesor en el curso creado.</li> <li>4. Realizar una llamada HTTP a la función que devuelve el listado de alumnos con el <i>token</i> del profesor.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• El servidor donde se aloje Moodle devuelve un listado de estudiantes con datos de cada uno:                             <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Apellidos</li> <li>○ NIA</li> <li>○ Correo electrónico</li> </ul> </li> </ul>		

Tabla 67 - Prueba del sistema PS-012 - Obtención de listado de estudiantes de Moodle

Especificación de prueba del sistema			
<b>Código</b>	PS-013	<b>Requisito fuente</b>	RF-016
<b>Nombre</b>	Obtención listado de cursos de Moodle		
<b>Descripción</b>	Comprobación de que la extensión para Moodle devuelve el listado de cursos cuando se realiza la llamada a la función correspondiente		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear dos cursos en la plataforma.</li> <li>2. Crear un profesor asociado a los dos cursos.</li> <li>3. Realizar una llamada HTTP a la función que devuelve el listado de cursos con el <i>token</i> del profesor.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• El servidor de Moodle devuelve un listado que contiene información de los dos cursos creados, que incluye:                             <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Identificador del curso dentro de la plataforma para realizar el resto de llamadas sobre cada curso</li> </ul> </li> </ul>		

Tabla 68 - Prueba del sistema PS-013 - Obtención de listado de cursos de Moodle

Especificación de prueba del sistema			
<b>Código</b>	PS-014	<b>Requisito fuente</b>	RF-017
<b>Nombre</b>	Obtención listado de cuestionarios de Moodle		
<b>Descripción</b>	Comprobación de que la extensión para Moodle devuelve el listado de cuestionarios de un curso cuando se realiza la llamada a la función correspondiente.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear un curso en la plataforma.</li> <li>2. Crear un profesor y asociarlo al curso creado.</li> <li>3. Crear dos cuestionarios dentro del curso creado.</li> <li>4. Realizar una llamada HTTP a la función que devuelve el listado de cursos con el <i>token</i> del profesor.</li> <li>5. Realizar una llamada HTTP a la función que devuelve el listado de cuestionarios con el <i>token</i> del profesor y el identificador del curso obtenido con la llamada del paso 4.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• El servidor de Moodle devuelve un listado que contiene información de los dos cuestionarios creados, que incluye: <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Contraseña</li> <li>○ Fechas de apertura y cierre</li> <li>○ Identificadores de las preguntas que forman parte del cuestionario</li> </ul> </li> </ul>		

Tabla 69 - Prueba del sistema PS-014 - Obtención de listado de cuestionarios de Moodle

Especificación de prueba del sistema			
<b>Código</b>	PS-015	<b>Requisito fuente</b>	RF-018
<b>Nombre</b>	Obtención listado de preguntas y respuestas de Moodle		
<b>Descripción</b>	Comprobación de que la extensión para Moodle devuelve el listado de preguntas y posibles respuestas de un cuestionario cuando se realiza la llamada a la función correspondiente.		
<b>Procedimiento de prueba</b>	<ol style="list-style-type: none"> <li>1. Crear un curso en la plataforma.</li> <li>2. Crear un profesor y asociarlo al curso creado.</li> <li>3. Crear un cuestionario dentro del curso creado.</li> <li>4. Crear dos preguntas en el cuestionario.</li> <li>5. Realizar una llamada HTTP a la función que devuelve el listado de cursos con el <i>token</i> del profesor.</li> <li>6. Realizar una llamada HTTP a la función que devuelve el listado de cuestionarios con el <i>token</i> del profesor y el identificador del curso obtenido con la llamada del paso 5.</li> <li>7. Realizar una llamada HTTP a la función que devuelve el listado de preguntas con el <i>token</i> del profesor y el identificador del cuestionario obtenido con la llamada del paso 6.</li> </ol>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• El servidor de Moodle devuelve un listado que contiene información de las preguntas del cuestionario creado: <ul style="list-style-type: none"> <li>○ Texto de la pregunta</li> <li>○ Tipo de pregunta (respuesta simple o múltiple)</li> <li>○ Respuestas posibles</li> </ul> </li> </ul>		

Tabla 70 - Prueba del sistema PS-015 - Obtención de listado de preguntas y respuestas de Moodle

5.2.4 TRAZABILIDAD ENTRE PRUEBAS DEL SISTEMA Y REQUISITOS

	RF-001	RF-002	RF-003	RF-004	RF-005	RF-006	RF-007	RF-008	RF-009	RF-010	RF-011	RF-012	RF-013	RF-014	RF-015	RF-016	RF-017	RF-018
PS-001	X	X			X													
PS-002			X		X													
PS-003				X	X	X												
PS-004							X											
PS-005								X										
PS-006									X									
PS-007										X								
PS-008											X							
PS-009												X						
PS-010													X					
PS-011														X				
PS-012															X			
PS-013																X		
PS-014																	X	
PS-015																		X

Tabla 71 - Matriz de trazabilidad entre pruebas del sistema y requisitos

## 6. GESTIÓN DEL PROYECTO

En este apartado se realiza una descripción del método de trabajo seguido durante las diferentes etapas del proyecto, con una explicación de los objetivos de cada una. A continuación, se muestra la planificación planteada para estas etapas y, por último, se establece un presupuesto para el proyecto.

### 6.1 METODOLOGÍA DE DESARROLLO

Para la elaboración del presente proyecto se ha seguido un modelo de desarrollo, o ciclo de vida en cascada realimentado, que es una variación del ciclo de vida clásico. En este último, se definen una serie de etapas sucesivas que se desarrollan de forma lineal, una tras otra sin posibilidad de retroceder a una anterior. Es evidente que esto plantea graves inconvenientes, puesto que se debe completar cada etapa de forma muy exhaustiva evitando imprevistos en las fases sucesivas, algo muy difícil de lograr en la práctica.

Por el contrario, el ciclo de vida clásico realimentado establece un flujo de información de cada etapa a las anteriores, de forma que es posible realizar ajustes en una etapa previa a la que se esté desarrollando. Este modelo se ajusta fielmente a la naturaleza de este proyecto por su carácter lineal. Partiendo de la definición de unos objetivos y, a través de una serie de etapas bien definidas, termina con la obtención de un producto final. Al añadir realimentación, se perfilan las etapas anteriores a la que se esté ejecutando, para así obtener un producto de mayor calidad.

A continuación, en la Figura 38, se muestran las diferentes etapas ordenadas del ciclo de vida en cascada. Las flechas continuas reflejan el flujo y orden de las distintas etapas, mientras que las flechas discontinuas marcan la realimentación.

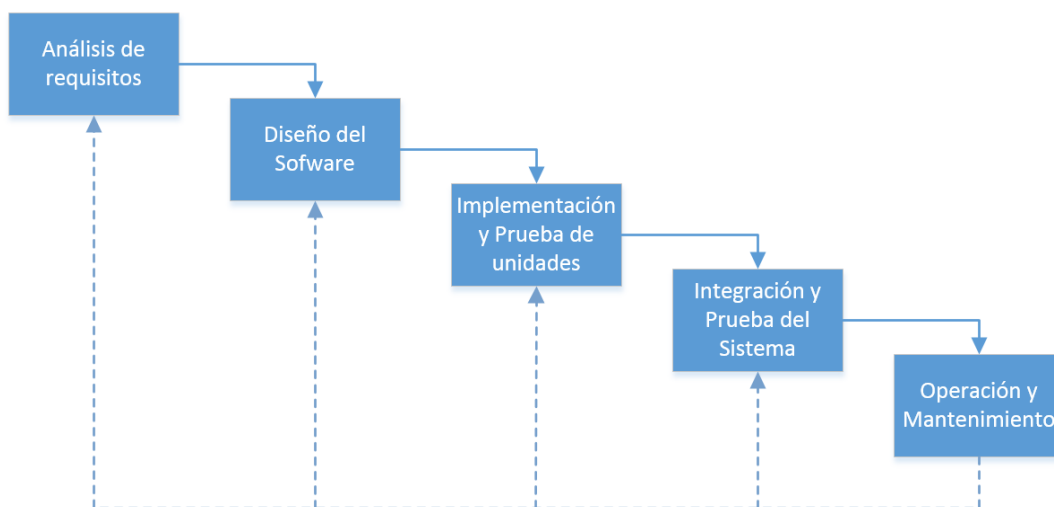


Figura 38 - Modelo en cascada realimentado



Una de las problemáticas más conocidas del ciclo de vida en cascada es su dificultad de implantación en equipos de varias personas. Sin embargo, en el presente proyecto, la mayor parte de la carga de trabajo reside en una única persona, por lo que este problema se reduce. Por otro lado, se requiere del establecimiento de unos requisitos muy detallados y ajustados al resultado final, lo cual se cumple en este proyecto. Desde el principio se tuvo una idea concisa de la solución final deseada. A esto se suma, que el producto final únicamente podrá ser mostrado al cliente en las últimas etapas [31].

Para entender en mayor profundidad este modelo de desarrollo, se describe cada una de las etapas:

- **Análisis de requisitos:** en esta etapa se establecen los objetivos del proyecto, de forma que se parta de una idea clara de la motivación del proyecto. En las tareas que se incluyen en esta primera fase se incluye una descripción de casos de uso, de requisitos funcionales y no funcionales y una evaluación de las distintas alternativas para afrontar el proyecto.
- **Diseño del software:** en esta etapa se profundiza en la descripción del sistema a desarrollar, con una descripción de la arquitectura en que se enmarcará. Además, el sistema se divide en diferentes elementos para facilitar la implementación. En este punto se utilizan diversas herramientas que ayudarán a especificar el sistema y describir su comportamiento, como diagramas de componentes, diagramas de clases, modelo de datos o diagramas de secuencia.
- **Implementación y Prueba de unidades:** es la etapa en la que se construye el sistema según el diseño realizado. Cuanto mayor sea el detalle y calidad del diseño, más rápida y mecánica será esta etapa. Además, se realizan pequeñas pruebas para comprobar el correcto funcionamiento de los diferentes módulos que se construyen.
- **Integración y pruebas del sistema:** se integran los diferentes subsistemas implementados en un único sistema. En esta etapa se realizan las pruebas del sistema, con las que se verifica el funcionamiento del sistema en su conjunto y su adecuación a los requerimientos inicialmente planteados.
- **Operación y mantenimiento:** en esta última etapa se realiza un mantenimiento del software, para solucionar los posibles errores que se puedan dar con el funcionamiento prolongado del sistema. A pesar de ser la última fase, y de que ya se ha obtenido la solución, esta etapa es de suma importancia y en la que recaen la mayor parte de los gastos de un proyecto. Es tal la importancia y el coste que supone, que puede suponer hasta el 70% de los recursos de una organización de software [33].

## 6.2 PLANIFICACIÓN DEL PROYECTO

En la Figura 39 se muestra un diagrama con la planificación realizada para las diferentes etapas del proyecto. Se ha construido con la herramienta GanttProject<sup>6</sup>, un programa destinado a la gestión de proyectos software con un funcionamiento muy simple y sencillo.

En la planificación se han incluido las etapas propias del ciclo de vida en cascada, al que se le han añadido dos etapas adicionales. Estas fases son:

- Planteamiento: en esta etapa se analiza el problema planteado, y se definen unos objetivos preliminares. Se toma una idea del producto que se quiere construir.
- Documentación del proyecto: en esta nueva etapa, situada en último lugar, se realiza la memoria del proyecto y se prepara su defensa.

Debido al modelo de desarrollo elegido (ciclo de vida en cascada) y a su carácter lineal, en el que se han establecido una serie de etapas consecutivas no superpuestas, no se definen tareas críticas. El desarrollo del proyecto está sujeto a la ejecución de todas las etapas, pasando a la siguiente cuando se haya completado la anterior y no siendo posible omitir ninguna.

---

<sup>6</sup> <http://www.ganttproject.biz/>

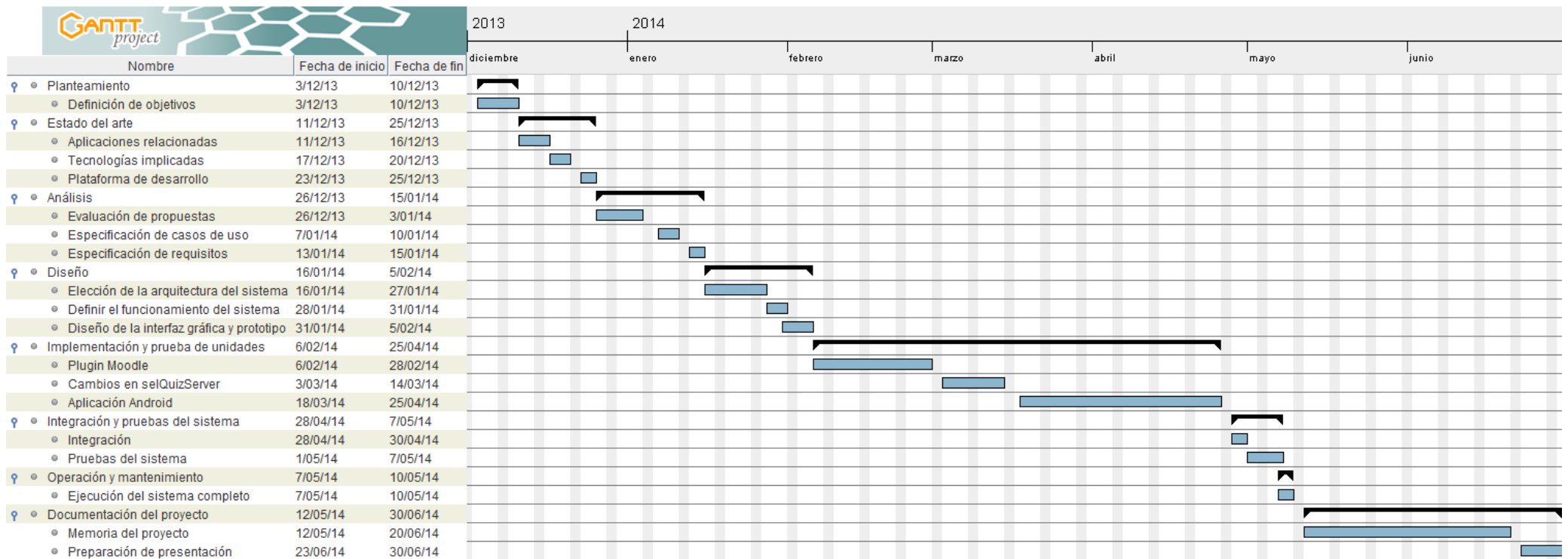


Figura 39 - Diagrama Gantt de la planificación del proyecto

## 6.3 PRESUPUESTO DEL PROYECTO

A continuación se elabora un presupuesto del proyecto, teniendo en cuenta los recursos tanto humanos como materiales.

### 6.3.1 COSTES DE RECURSOS HUMANOS

En la Tabla 72 se disponen los costes derivados de los recursos humanos. Los salarios se ajustan a los datos publicados en el BOE número 174 del 21 de julio de 2012. Al coste calculado según el número de meses trabajados, se ha añadido el porcentaje de Seguridad Social que paga la empresa. Según las Bases y tipos de cotización del 2014, publicadas en la página web del Ministerio de Empleo y Seguridad Social, se describen los siguientes porcentajes de costes de Seguridad Social para la empresa:

- Contingencias comunes: 23,60%
- Desempleo por contrato duración determinado Tiempo Completo: 6,70%
- Fondo de Garantía Salarial: 0,20%
- Formación Profesional: 0,60%
- Tarifa para la cotización por accidentes de trabajo y enfermedades profesionales, extraída del BOE número 311 de 29 de Diciembre de 2006, para la categoría de Programación, consultoría y otras actividades relacionadas con la informática: 1,35%.

En total, estos cuatro conceptos suman un 32,45%.

Costes relativos a los recursos humanos				
Cargo	Sueldo base	Número de meses	Coste total	Coste total con Seguridad Social (coste total * 1,325)
Jefe de proyecto	1.895,17€	1	1.895,17€	2.511,10€
Analista	1.536,07€	1,5	2.304,10€	3.052,93€
Técnico de pruebas	1.193,24€	0,5	596,62€	790,52€
Programador	1.193,24€	2,63	3.138,22€	4.158,14€
			<b>TOTAL</b>	10.512,69€

Tabla 72 - Costes de los recursos humanos del proyecto

### 6.3.2 COSTES DE MATERIAL

Para el cálculo del total de gastos asociados a los recursos materiales del proyecto, se ha realizado un cálculo por separado de cada elemento, teniendo en cuenta el número de meses dedicados al proyecto, el período de amortización y el porcentaje de dedicación al proyecto.

El cálculo se basa en la siguiente fórmula:

$$\text{Coste} = \text{Precio} * \text{Porcentaje de dedicación al proyecto} * \frac{\text{Número de meses de dedicación}}{\text{Período de amortización}}$$

En la Tabla 73 se muestra el coste para cada uno de los materiales y la suma total que genera.

Costes relativos a los materiales					
Material	Precio	Número de meses de dedicación	Período de amortización (meses)	Porcentaje de dedicación al proyecto	Coste
PC AMD Phenom II 1055T	950€	7	48	50%	69,27€
Windows 7 Professional	125€	7	48	50%	9,11€
Office	125,10€	2	48	60%	10,94€
Nexus 5	349€	4	20	30%	4,36€
Sony Ericsson Neo	99€	1	20	30%	1,48€
Orange Monte Carlo	80€	1	20	30%	1,2€
				<b>TOTAL</b>	<b>96,36€</b>

Tabla 73 - Costes de los recursos materiales del proyecto

### 6.3.3 PRESUPUESTO TOTAL DEL PROYECTO

Finalmente, se establece el presupuesto total, que integra los costes de los recursos humanos y materiales, costes indirectos y beneficio industrial:

	Coste
Costes de recursos humanos	10.512,69€
Costes de recursos materiales	96,36€
Costes indirectos (10%)	1.060,91
Beneficio industrial (6%)	636,54
<b>TOTAL</b>	<b>12.943,04€</b>

El presupuesto del proyecto que describe el presente documento asciende a DOCE MIL NOVECIENTOS CUARENTA Y TRES MIL EUROS CON CUATRO CÉNTIMOS (12.943,04€).

# 7. CONCLUSIONES Y TRABAJOS FUTUROS

## 7.1 CONCLUSIONES

Llegado este punto del documento, es conveniente realizar unos comentarios finales sobre el trabajo realizado. Tal y como se comentaba en los primeros apartados, la idea inicial sufrió modificaciones hasta plantear una definitiva consistente y viable. Desde el objetivo de crear una aplicación que permitiera responder directamente los cuestionarios de la plataforma Moodle hasta finalmente enmarcar el trabajo en un sistema de aplicaciones, ya existente, del que se iba a ampliar su funcionalidad. Esto último ha generado grandes ventajas al desarrollo de este proyecto, puesto que no se ha partido desde cero, siendo posible entablar comunicación directa con sus desarrolladores. Además, es un sistema que actualmente se encuentra en funcionamiento, asegurando la puesta en práctica del trabajo realizado.

Debido a la importante carga de trabajo de las modificaciones en mSuite para hacer posible una comunicación con Moodle y la adaptación de todo el sistema, fue necesario dividir la tarea en dos proyectos. Por este motivo, todavía no es posible usar el sistema al completo con las nuevas funcionalidades, como por ejemplo, recibir cuestionarios de Moodle. El proyecto paralelo a este se encuentra actualmente en desarrollo, y se espera su presentación para el mes de septiembre, por lo que es posible que para el próximo curso académico sea plenamente funcional.

Este proyecto se ha centrado en:

- La creación de una aplicación nativa para Android que permita responder preguntas de respuesta múltiple, acceder a sesiones protegidas y solicite el NIA al alumno. También se ha establecido, tal y como se especificaba en los objetivos del trabajo definidos en el primer capítulo, un control para vigilar la presencia en clase del alumno. Este control consiste introducir una contraseña en la aplicación para poder suscribir una sesión. La clave será dada por el profesor dentro del aula, por lo que será necesario estar presente para conocerla.
- La modificación del servidor al que se conecta la aplicación de los alumnos, mClassTest.
- La creación de una extensión para Moodle que permita recuperar cuestionarios a un sistema externo.

Por su parte, en el otro proyecto, se modificará la aplicación mTeachPlan para conectarse con Moodle mediante la extensión desarrollada y recuperar así los cuestionarios, se modificará la aplicación actual para iOS de forma que admita preguntas de respuesta múltiple y se modificará la aplicación mTeachClass para adaptarse a los cambios realizados en el SELQuizServer en este TFG.

Una vez integrados los dos proyectos, se obtendrá un sistema que ayudará a los profesores a impartir sus clases y de un coste mínimo, reducido al mantenimiento de los servidores donde se aloja el sistema y al de los dispositivos que utilicen los profesores. Los dispositivos de los alumnos no se contabilizan para este coste, puesto que utilizarán sus teléfonos inteligentes propios. Por tanto, es un sistema del que su viabilidad económica está garantizada.

En cuanto al trabajo realizado, este ha sido muy diverso, implicando utilizar diversas tecnologías. Esto ha incrementado de forma considerable el número de horas empleadas en el proyecto, ya que era necesario un período de aprendizaje sobre cada nueva tecnología para poder utilizarla. No obstante, lejos de ser una desventaja, ha sido un gran aliciente del trabajo, puesto que ha permitido aprender a usar herramientas desconocidas que de seguro, serán muy útiles en el futuro, tanto en el mundo académico como en el laboral. Una de las tecnologías utilizadas ha sido Android, uno de los sistemas operativos para dispositivos móviles de mayor uso mundial. Esto hace que tener una formación sobre el desarrollo de aplicaciones para este sistema sea de gran utilidad, abriendo un gran número de puertas en el mercado laboral.

Finalmente, se debe remarcar la importancia de proyectos como el presente en la actualidad. La integración de la tecnología en la enseñanza abre un mundo de posibilidades, siempre con el objetivo de mejorar la labor de la docencia.

### 7.2 TRABAJOS FUTUROS

La unión del presente trabajo con el realizado paralelamente conllevará obtener un sistema con múltiples utilidades y totalmente funcional. Sin embargo, mSuite tiene un carácter muy dinámico, posibilitando la adopción de nuevas funcionalidades con gran facilidad. Además, ciertas partes de las ya existentes tienen margen de mejora, por lo que es posible perfeccionar el sistema. El trabajo futuro que se puede desarrollar en la plataforma se resume en los siguientes puntos:

- Aumentar la seguridad: las llamadas al SELQuizServer deberían enviarse mediante un protocolo seguro en el que queden encriptadas. Actualmente, se puede realizar una llamada al servidor sin ninguna clave, lo que puede permitir que terceras personas realicen modificaciones si conocen las direcciones y llamadas correspondientes. Otra solución, en vez de establecer un protocolo, sería definir contraseñas para todos los usuarios de mSuite, de forma que sea necesario introducir la clave de cada uno para poder realizar una llamada.
- Aumentar los sistemas para los que existen las aplicaciones de la suite. Actualmente, las aplicaciones mTeachPlan y mTeachClass se encuentran disponibles para iOS y Android. Debido al auge de nuevos sistemas operativos, en el futuro se deberían implementar todas las aplicaciones para nuevos sistemas, como por ejemplo Windows Phone, el cual tiene una cuota de mercado importante.

- Admitir nuevos tipos de preguntas de Moodle: este LMS posibilita la incorporación de preguntas que incluyen imágenes o preguntas de asociar palabras entre sí, que podrían resultar muy útiles en la plataforma.
- Crear una web para acceder al sistema y poder responder preguntas: si un alumno no dispone de un teléfono inteligente iOS o Android, debe facilitarse otra opción para que pueda responder a las preguntas planteadas.
- Mayor integración con Moodle: se puede llevar a cabo una mayor unión entre mSuite y Moodle permitiendo, por ejemplo, que los profesores puedan obtener presentaciones o documentos desde Moodle para utilizarlos en la impartición de clases con mTeachClass.
- Posibilidad de que las calificaciones de los test se actualicen directamente en Moodle: se evitaría así que el profesor tenga que subir manualmente un fichero con las calificaciones al LMS.
- Posibilidad de iniciar sesión con otras plataformas (Facebook, Google...) o incluso sin registrarse en ninguna de ellas: para hacer posible el uso de la aplicación a cualquier alumno, se podrían incluir métodos de identificación con otras plataformas.
- Comprobar la veracidad del NIA que ha introducido el alumno en su aplicación: de esta forma se evitaría que un alumno pudiera introducir un identificador inválido o de otro alumno.



## 8. BIBLIOGRAFÍA

1. Huneycutt, Timothy. National Math and Science Initiative. [En línea] 8 de Octubre de 2013. [Citado el: 7 de Mayo de 2014.] <https://nms.org/Blog/TabId/58/PostId/188/technology-in-the-classroom-the-benefits-of-blended-learning.aspx>.
2. Epicor Software Corporation. [En línea] [Citado el: 16 de Marzo de 2014.] <http://www.epicor.com/lac/Solutions/Pages/Serviceoriented.aspx>.
3. Accenture, Centro de Alto Rendimiento de. *Arquitectura Orientada a Servicios (SOA). Cómo reformular la Arquitectura Corporativa para alcanzar el alto rendimiento*. 2008. [http://www.kybele.etsii.urjc.es/docencia/IS\\_LADE/2012-2013/Material/CAR%20Accenture%20-%20SOA\[1\].pdf](http://www.kybele.etsii.urjc.es/docencia/IS_LADE/2012-2013/Material/CAR%20Accenture%20-%20SOA[1].pdf).
4. msaffirio. *¿Qué son los Web Services?* [En línea] 05 de Febrero de 2006. [Citado el: 16 de Marzo de 2014.] <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>.
5. Apuntes de Programación. *Web Service: Definición, utilización y estructura del WSDL*. [En línea] Enero de 2012. [Citado el: 16 de Marzo de 2014.] <http://programacion.jias.es/2012/01/web-service-definicion-utilizacion-estructura-del-wsdl/>.
6. Aula Virtual. [En línea] Noviembre de 2011. [Citado el: 30 de Abril de 2014.] <http://aula.virtual.ucv.cl/wordpress/importancia-tic-en-la-edu-sup/>.
7. Aula Virtual. [En línea] Abril de 2012. [Citado el: 30 de Abril de 2014.] <http://aula.virtual.ucv.cl/wordpress/entornos-personales-de-aprendizaje-como-sistemas-de-gestion/>.
8. Ministerior de Educación y Ciencia. Cnice. Serie Informers. *Uso de estándares aplicados a TIC en educación*. [En línea] [Citado el: 30 de Abril de 2014.] <http://ares.cnice.mec.es/informes/16/contenido/47.htm>.
9. Capterra. *Top LMS Software*. [En línea] Enero de 2014. [Citado el: 1 de Mayo de 2014.] <http://www.capterra.com/learning-management-system-software/?utf8=%E2%9C%93&v=2#infographic>.
10. Área de Tecnologías para la Docencia. *Universidad de Zaragoza*. [En línea] [Citado el: 1 de Mayo de 2014.] [http://add.unizar.es/add/area/index.php?option=com\\_content&view=article&id=37:blackboard-learning-system-ce8&catid=35&Itemid=27](http://add.unizar.es/add/area/index.php?option=com_content&view=article&id=37:blackboard-learning-system-ce8&catid=35&Itemid=27).
11. *escuelatic*. [En línea] [Citado el: 1 de Mayo de 2014.] <http://www.escuelatic.es/2011/03/02/edmodo-plataforma-social-educativa/>.

12. escuela20. [En línea] 15 de Junio de 2014. [http://www.escuela20.com/edmodo-plataforma-comunicacion/articulos-y-actualidad/15-usos-de-edmodo--y-muchomas\\_3161\\_42\\_4665\\_0\\_1\\_in.html](http://www.escuela20.com/edmodo-plataforma-comunicacion/articulos-y-actualidad/15-usos-de-edmodo--y-muchomas_3161_42_4665_0_1_in.html).
13. Moodle. *Mobile Moodle FAQ*. [En línea] 8 de Junio de 2013. [Citado el: 16 de Marzo de 2014.] [http://docs.moodle.org/24/en/Mobile\\_Moodle\\_FAQ](http://docs.moodle.org/24/en/Mobile_Moodle_FAQ).
14. Moodle. *Mobile app*. [En línea] 2 de Octubre de 2013. [Citado el: 16 de Marzo de 2014.] [http://docs.moodle.org/24/en/Mobile\\_app](http://docs.moodle.org/24/en/Mobile_app).
15. moodleTouch. [En línea] [Citado el: 16 de Marzo de 2014.] <http://www.pragmasql.com/moodletouch/Default.aspx>.
16. Thibault, Joseph. Moodle News. [En línea] 10 de Mayo de 2010. [Citado el: 16 de Marzo de 2014.] <http://www.moodlenews.com/2010/more-info-on-mpage-the-free-mobile-moodle-app/>.
17. Moodle. *Mobile app*. [En línea] [Citado el: 16 de Marzo de 2014.] [http://docs.moodle.org/22/en/Mobile\\_app](http://docs.moodle.org/22/en/Mobile_app).
18. Google Code. *Moodbile*. [En línea] 1 de Octubre de 2010. [Citado el: 16 de Marzo de 2014.] <https://code.google.com/p/moodbile/downloads/detail?name=Moodbile%20Manual%20de%20Usuario.pdf>.
19. Moodbile. [En línea] [Citado el: 30 de Abril de 2014.] <http://www.moodbile.org/>.
20. Moodle News. [En línea] 30 de Abril de 2014. <http://www.moodlenews.com/2011/mbot-review-an-android-mobile-moodle-app-android-mobile/>.
21. Moodle. [En línea] [Citado el: 30 de Abril de 2014.] <https://moodle.org/mod/forum/discuss.php?d=245849>.
22. Moodle Mobile FAQ. [En línea] [Citado el: 30 de Abril de 2014.] [http://docs.moodle.org/21/en/Mobile\\_Moodle\\_FAQ](http://docs.moodle.org/21/en/Mobile_Moodle_FAQ).
23. Moodle. Web services Roadmap. [En línea] [Citado el: 15 de Mayo de 2014.] [http://docs.moodle.org/dev/Web\\_services\\_Roadmap](http://docs.moodle.org/dev/Web_services_Roadmap).
24. Moodle. Web services API. [En línea] 20 de Febrero de 2013. [Citado el: 15 de Marzo de 2014.] [http://docs.moodle.org/dev/Web\\_services\\_API](http://docs.moodle.org/dev/Web_services_API).
25. Centro de Alto Rendimiento de Accenture, Ametic. Persuadir al cliente. [En línea] [Citado el: 1 de Mayo de 2014.] <http://www.ametic.es/DescargarDocumento.aspx?idd=5257>.
26. Kantar Worldpanel. [En línea] Diciembre de 2013. [Citado el: 1 de Mayo de 2014.] [http://www.kantarworldpanel.com/dwl.php?sn=news\\_downloads&id=399](http://www.kantarworldpanel.com/dwl.php?sn=news_downloads&id=399).
27. Gironés, Jesús Tomás. *El gran libro de Android*. Barcelona : Marcombo, 2012. 978-84-267-1850-1.

28. Android Developers. [En línea] [Citado el: 19 de Junio de 2014.] [https://developer.android.com/about/dashboards/index.html?utm\\_source=ausdroid.net](https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net).

29. ICT S.A.S. *Ingeniería y Consultoría Tecnológica*. [En línea] 28 de Marzo de 2013. [Citado el: 16 de Marzo de 2014.] <http://www.ictsas.com/web/index.php/blog/45-java-vs-html5>.

30. Moodle. Instalar Moodle . [En línea] [Citado el: 11 de Mayo de 2014.] [http://docs.moodle.org/24/en/Installing\\_Moodle](http://docs.moodle.org/24/en/Installing_Moodle).

31. Pressman, Roger S. *Ingeniería del Software*. Sexta edición. s.l. : McGraw-Hill. págs. 119-121. 970-10-5473-3.

32. php.net. [En línea] [Citado el: 30 de Mayo de 2014.] <http://www.php.net/manual/es/book.xmlrpc.php>.

33. Pressman, Roger S. *Ingeniería del software. Un enfoque práctico*. Séptima edición. México, D.F : McGraw-Hill, 2010. pág. 656. 978-607-15-0314-5.

# Anexos

# Anexo A: Manual de usuario de la aplicación mClassTest para Android

## REQUISITOS PARA LA INSTALACIÓN

---

Para instalar mClassTest para Android, será necesario poseer un dispositivo con dicho sistema operativo y cumpliendo los siguientes requisitos:

- Versión mínima de Android 2.3.3
- Servicios de Google Play instalados
- 512MB de memoria RAM
- Procesador de un núcleo a 800 MHz
- 7MB de espacio libre

Los puntos anteriores reflejan unos requisitos en los que se asegura la funcionalidad correcta de la aplicación. Sin embargo, no se descarta que pueda funcionar con características inferiores de memoria RAM o procesador.

## INSTALACIÓN DE LA APLICACIÓN

---

La aplicación mClassTest podrá ser descargada e instalada desde la tienda oficial de aplicaciones de Android, Google Play Store (buscando por el nombre de la aplicación). En las próximas semanas será incorporada a dicha tienda. La instalación no ofrece ninguna dificultad y en pocos segundos será funcional la aplicación.

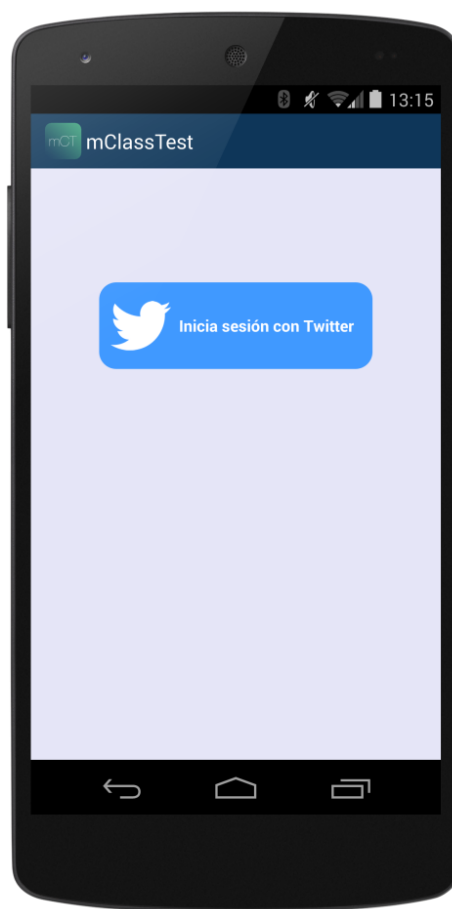
## USO DE LA APLICACIÓN

Nada más abrir la aplicación, se mostrará una vista de presentación (Figura 40). Internamente, la aplicación comprobará si el dispositivo tiene instalados los Servicios de Google Play. En caso negativo, se mostrará un mensaje para ayudar a instalarlos.

Después de la vista de presentación, se mostrará otra para invitar al usuario a identificarse con Twitter (Figura 41).



**Figura 40 - Vista de presentación de mClassTest para Android**



**Figura 41 - Vista de solicitud de inicio de sesión con Twitter de mClassTest para Android**

Cuando se pulse el botón para iniciar sesión con Twitter, la aplicación cargará una página web (Figura 42), donde se deben introducir los datos de una cuenta válida de la red social. También se posibilita registrarse, en caso de que el alumno no lo esté previamente.

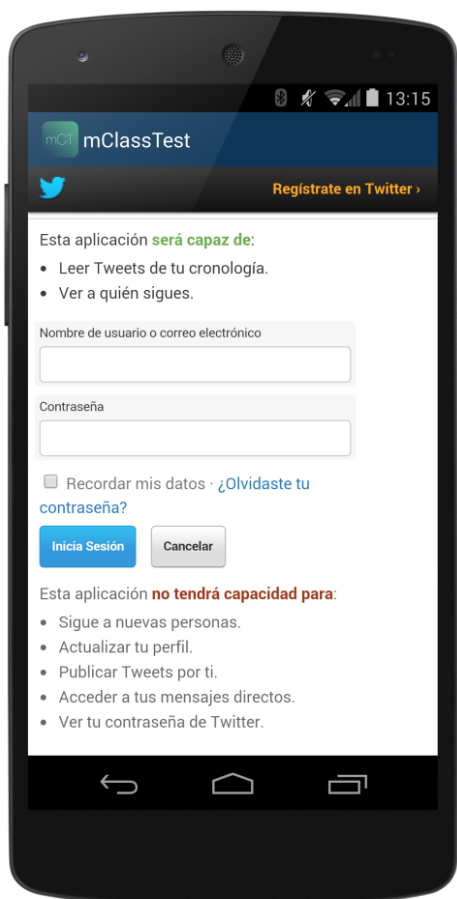


Figura 42 - Vista inicio de sesión con Twitter en mClassTest para Android



Figura 43 - Vista de solicitud de NIA en mClassTest para Android

Una vez introducidos los datos para iniciar sesión con Twitter y de que se haya pulsado el botón “Iniciar Sesión”, la aplicación mostrará una vista para solicitar el NIA al alumno. Únicamente habrá que introducir el dato mencionado en el cuadro de texto y pulsar en el botón “Enviar”. A continuación se mostrará la vista que contiene el listado de sesiones disponibles (Figura 44).

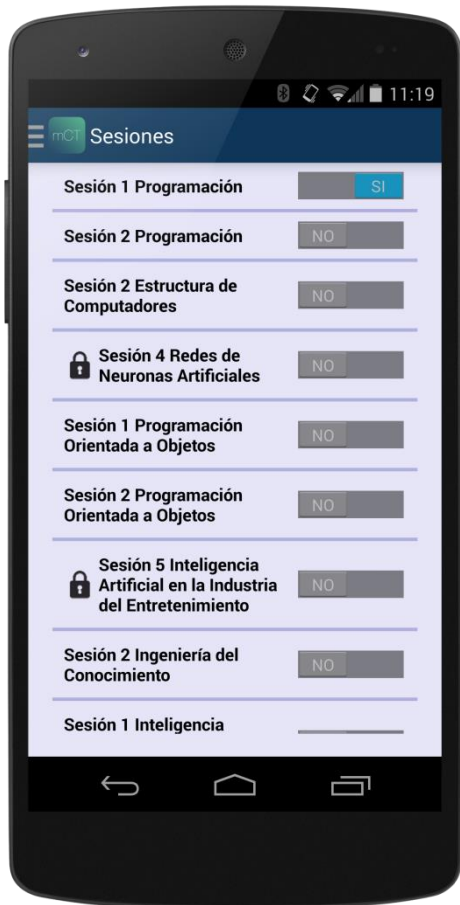


Figura 44 - Vista del listado de sesiones en mClassTest para Android



Figura 45 - Cuadro de diálogo en sesiones protegidas en mClassTest para Android

En la vista del listado de sesiones se podrán pulsar los Switches que acompañan al nombre de cada sesión para suscribir o cancelar la suscripción. Cuando se suscriba una de ellas, estando previamente suscrito a otra, la suscripción se cancelara a esta última, desactivándose el Switch asociado.

La imagen de un pequeño candado que acompaña a algunas sesiones indica que esa sesión está protegida por contraseña. Para poder suscribirla y, por consiguiente, ver las preguntas de esa sesión, será necesario introducir una contraseña. Cuando se pulse en el interruptor de una sesión de este tipo, se mostrará un cuadro de diálogo como el de la Figura 45, en el que se deberá introducir la contraseña correcta y pulsa en el botón "Aceptar".



mClassTest dispone de un menú donde acceder a visualizar un listado con las preguntas planteadas para la sesión suscrita o para volver al de las sesiones. Este menú se puede abrir pulsando en el lateral izquierdo de la barra superior que contiene el nombre de la aplicación (donde se encuentra el logotipo de la misma) o arrastrando el dedo desde el borde izquierdo de la aplicación hacia el centro de la pantalla.

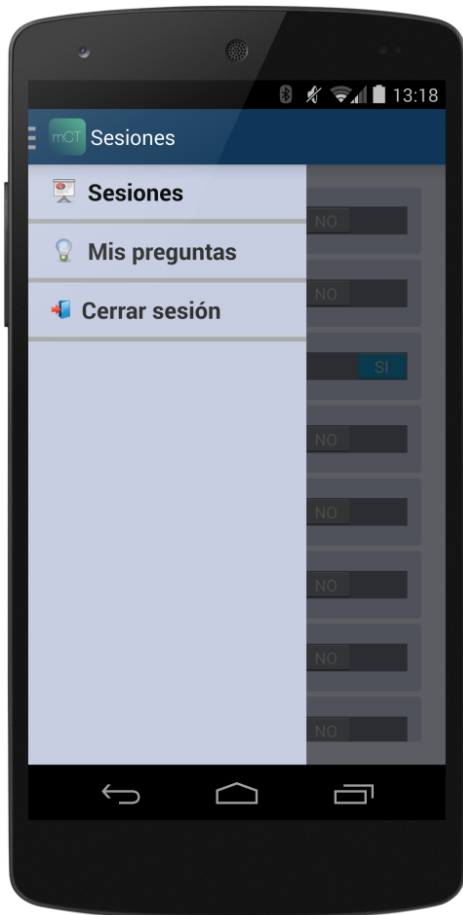


Figura 46 - Vista del menú en mClassTest para Android

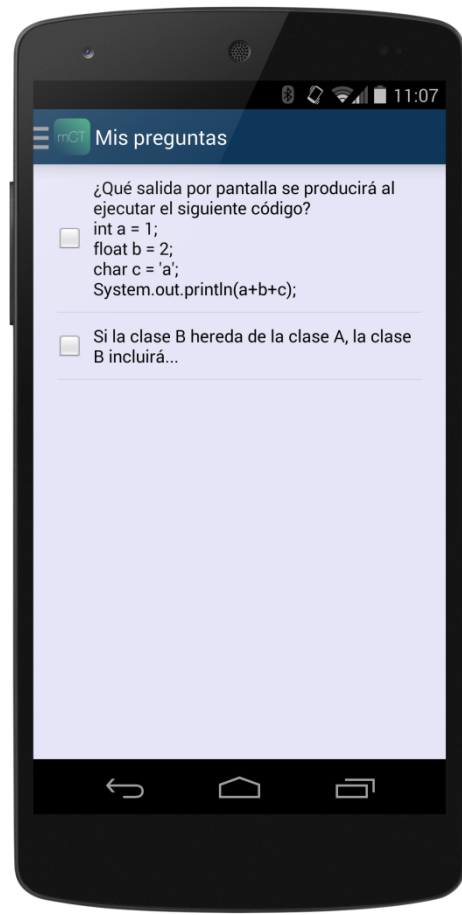


Figura 47 - Vista del listado de preguntas en mClassTest para Android

En el menú también se muestra una tercera opción, “Cerrar sesión”. Al pulsar sobre ella, la aplicación borrará los datos de identificación y se reiniciará, de forma que se puede volver a realizar el proceso de identificación.

Cuando se pulse en la opción de “Mis preguntas” en el menú, se mostrará una vista similar a la de la Figura 47. Contiene el listado de preguntas de la sesión suscrita. Para aquellas que ya se han contestado, se mostrará una casilla de verificación rellena a la izquierda. Las preguntas podrán ser contestadas una única vez, por lo que únicamente se podrá acceder a aquellas con la casilla de verificación vacía.

Si se desea actualizar el listado de sesiones o de preguntas, no es necesario cerrar y volver a abrir la aplicación. Bastará con arrastrar el dedo hacia abajo (estando en las primeras posiciones del listado) y automáticamente se actualizará el listado correspondiente.

En caso de que se pulse sobre una de las preguntas no contestadas, se mostrará una vista similar a la mostrada en la Figura 48 o de la Figura 49, dependiendo de si se trata de una pregunta de respuesta simple o múltiple respectivamente.

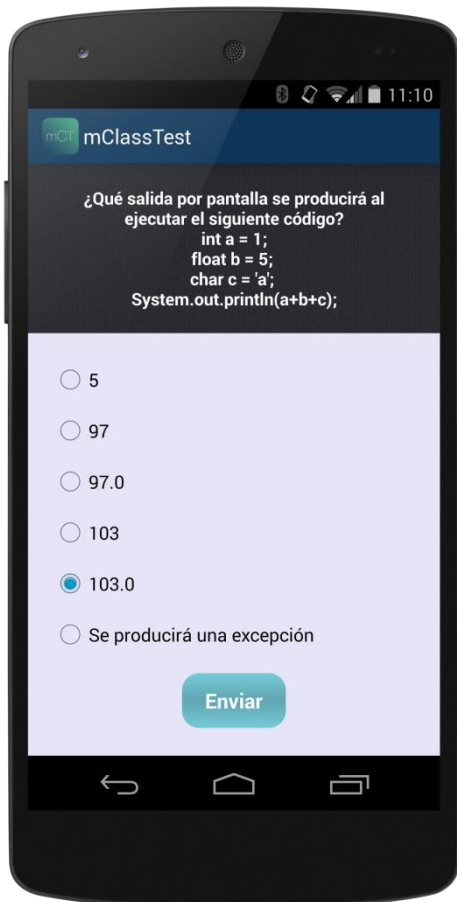


Figura 48 - Vista de pregunta de respuesta simple en mClassTest para Android

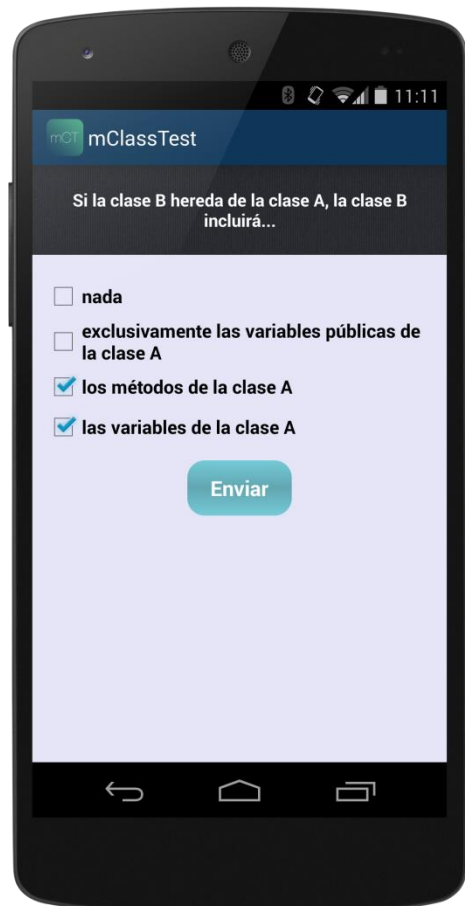


Figura 49 - Vista de pregunta de respuesta múltiple en mClassTest para Android

Una vez elegida la respuesta, se debe pulsar en el botón "Enviar" para guardar la respuesta. La información informará si la respuesta ha sido registrada correctamente o, por el contrario, si ha sucedido un problema y no se ha podido registrar.

# Anexo B: Manual de instalación y uso de la extensión para Moodle

En este anexo se explica cómo instalar y usar la extensión desarrollada en Moodle. Ha sido implementada y probada en la versión 2.4.8 de Moodle, no descartando su correcto funcionamiento en versiones superiores.

## INSTALACIÓN DE LA EXTENSIÓN

---

La extensión está formada por un único fichero comprimido, que con unos sencillos pasos puede ser instalado en Moodle. Lo primero de todo, es descomprimir el fichero para obtener el directorio *quizzes\_plugin*. Este debe ser copiado en el servidor donde se encuentre instalado Moodle, concretamente en el directorio *'/moodle/local/'*. Una vez copiado es necesario acceder al sistema, por la web de acceso, usando una cuenta de administrador. Al identificarse, el sistema detectará automáticamente el nuevo *plug-in* y mostrará una ventana como la de la siguiente imagen para permitir instalarlo:

## Comprobación de 'plugins'

Esta página muestra las extensiones (plugins) que pueden requerir su atención durante la actualización. Los elementos resaltados incluyen nuevas extensiones (plugins) que están a punto de ser instalados, los que van a ser actualizados y las extensiones anteriores que ahora faltan. Los módulos externos (add-ons) también se destacan. Se recomienda que compruebe si hay versiones más recientes de los módulos externos disponibles y actualice su código fuente antes de continuar con esta actualización de Moodle.

[Compruebe actualizaciones disponibles](#)

**Número de extensiones (plugins) que requieren atención durante esta actualización: 1**

[Mostrar la lista completa de extensiones \(plugins\) instalados](#)

Nombre de la extensión	Directorio	Origen	Versión actual	Nueva versión	Requiere	Estado
<b>Extensiones (plugins) locales</b>						
[pluginname,local_quizzes_plugin]	/local/quizzes_plugin	Módulo externo		2014060702	Moodle 2010112400	Para instalarse

[Recargar](#)

[Actualizar base de datos Moodle ahora](#)

**Figura 50 - Reconocimiento de la nueva extensión para Moodle**

En caso de que el sistema no detecte la extensión automáticamente, el administrador deberá seleccionar la opción de *Administración del sitio* en el menú de ajustes y después pinchar sobre la opción *Notificaciones*. Se abrirá una ventana como la de la Figura 50 informando si hay una nueva extensión para ser instalada. A continuación, en la ventana se debe pulsar en el botón inferior “Actualizar base de datos Moodle ahora” y el sistema realizará la instalación de la extensión.

Para que la extensión sea completamente funcional, es necesario realizar unos ajustes en el sistema.

### **Habilitar los servicios web**

De nuevo en el menú de ajustes, se debe ir a *Administración del sitio -> Características avanzadas*, donde se mostrarán una serie de funcionalidades de Moodle, informando si se encuentran habilitadas. Se debe activar, si no lo estuviera previamente, la opción *Habilitar servicios web*.

### Activar el protocolo del servicio web

Para que la nueva extensión sea accesible, es necesario activar el protocolo que utiliza, denominado XML-RPC. En el menú de ajustes se debe navegar por las opciones *Administración del sitio* -> *Extensiones* -> *Servicios Web* -> *Administrar protocolos*, llegando a una página donde se muestran los distintos protocolos admitidos. Se deberá habilitar el mencionado, pulsando sobre el icono con forma de ojo de la columna *Habilitar* en la fila correspondiente, mostrándose como en la Figura 51.

[Página Principal](#) ▾ [Administración del sitio](#) ▾ [Extensiones](#) ▾ [Servicios Web](#) ▾ [Administrar protocolos](#) [Activar la edición de bloques](#)

## Administrar protocolos

### Protocolos activos de servicio web

Protocolo	Versión	Habilitar	Desinstalar	Configuración
Protocolo AMF	2012112900	✖	<a href="#">Desinstalar</a>	
Protocolo REST	2012112900	✖	<a href="#">Desinstalar</a>	
Protocolo SOAP	2012112900	✖	<a href="#">Desinstalar</a>	
Protocolo XML-RPC	2012112900	👁	<a href="#">Desinstalar</a>	

Por razones de seguridad, solo se habilitarán los protocolos que vayan a ser utilizados.

[Documentación de servicios web](#)  
enablewsdocumentation

Valor por defecto: No  
 Habilitar la auto-generación de documentación de servicios web. Un usuario puede tener acceso a su propia documentación en su página de claves de seguridad [Más detalles](#). Solo se muestra la documentación de los protocolos habilitados.

[Guardar cambios](#)

Figura 51 - Activación de protocolo en Moodle

### Gestión de tokens

En las llamadas a la extensión será necesario identificarse mediante un *token*, que es una cadena de caracteres alfanuméricos que sirve para identificar unívocamente a un usuario de Moodle. Aquellos profesores que vayan a utilizar dispositivos que hagan llamadas a la extensión, necesitaran uno. Desde el menú de ajustes, en *Administración del sitio* -> *Extensiones* -> *Servicios Web* -> *Administrar tokens*, se podrán dar de alta a usuarios de la plataforma en los diferentes servicios web que haya instalados. En el momento en que se incluya a un usuario en servicio Web, se le asociará un *token* que será con el que pueda llamar a las diferentes funciones para recuperar los cuestionarios.

Los profesores, que son los que podrán obtener datos de Moodle mediante la extensión, no tendrán por qué conocer su *token*. Realizando otra llamada, ya implementada, en la que se incluya el nombre de usuario y la contraseña de la plataforma y el “nombre corto” del servicio concreto, se podrá recuperar este valor.

Debido a un fallo en la plataforma, que no se ha corregido hasta la versión 2.7, es necesario introducir manualmente este nombre especial para la extensión en la base de datos. Para ello, se debe acceder a la herramienta phpMyAdmin para manipular la base de datos. En la tabla *mdl\_external\_services* (las tres primeras letras pueden variar) y buscar la fila cuyo valor en la columna es *quizzes\_plugin*. En esa misma fila, en la columna *shortname*, se debe pinchar con doble clic para modificar su valor e introducir el deseado, como por ejemplo “quizzesPlugin”.

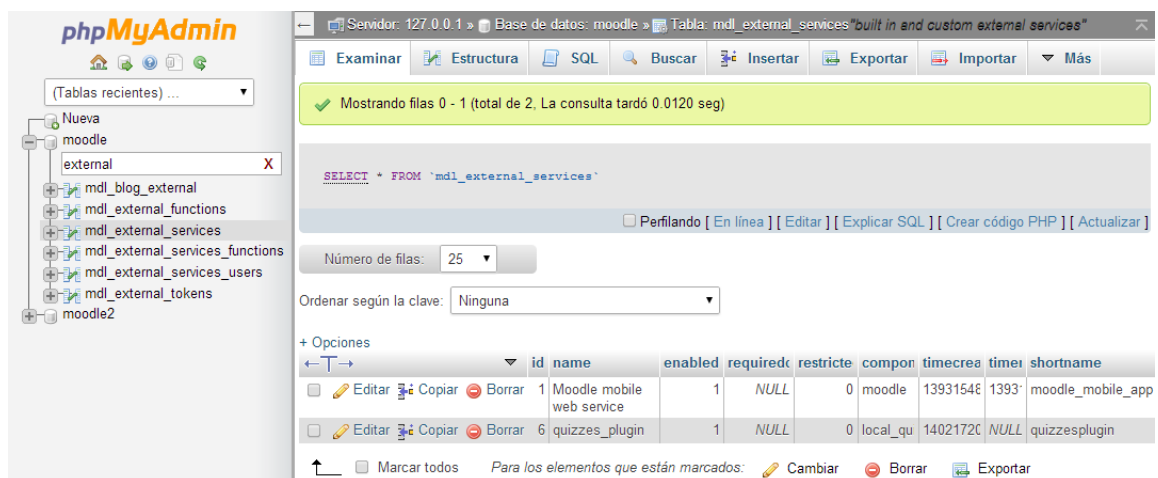


Figura 52 - Modificación del nombre corto de la extensión en la base de datos Moodle

## USO DE LA EXTENSIÓN

La extensión desarrollada está pensada para ser utilizada por una aplicación o programa, y no directamente por una persona. La aplicación que lo use, solicitará al profesor que esté utilizándola su nombre de usuario y su contraseña. Sin embargo, para realizar las llamadas, es necesario indicar el *token* asociado a la extensión que contenga las nuevas funciones. Llamando a la siguiente dirección web, se podrá obtener el *token* como respuesta:

DIRECCIONMOODLE/login/token.php?username=USUARIO&password=CONTRASEÑA&service=SERVICESHORTNAME

En la ruta indicada, será necesario modificar el dato DIRECCIONMOODLE por la dirección real del servidor donde esté instalado Moodle. Los campos USUARIO y CONTRASEÑA también deberán ser reemplazados por los datos de acceso del profesor y el dato SERVICESHORTNAME por el introducido manualmente en la base de datos (según se explica en la instalación de la extensión).

A continuación, se describen las llamadas que se pueden realizar, todas bajo el protocolo XML-RPC. Estas llamadas se harán a la siguiente dirección web: *'dirección del servidor/webservice/xmlrpc/server.php'*, a la que se añade el parámetro *'wstoken'* con el valor del *token* obtenido con la llamada anteriormente descrita. Dependiendo de la función requerida, se deberá incluir en la llamada el nombre de la función y los atributos necesarios. Además, algunas de las funciones requieren de los datos obtenidos con otras, por lo que será necesario ejecutarlas en cierto orden.

Dando un ejemplo de la llamada en un lenguaje concreto, por ejemplo PHP, con el método *xmlrpc\_encode\_request* se indica el método y parámetros de la llamada a realizar, en su primer y segundo parámetro respectivamente. Las cuatro funciones disponibles con los posibles parámetros son:

- **local\_quizzes\_plugin\_user\_get\_courses**
  - Esta función no recibe parámetros, por lo que se introducirá el valor *null* en el segundo parámetro de la función *xmlrpc\_encode\_request*.
- **local\_quizzes\_plugin\_course\_get\_quizzes**
  - Parámetro: Id del curso sobre el que obtener los cuestionarios. Se obtendrá de la llamada anterior y será numérico.
- **local\_quizzes\_plugin\_quiz\_get\_questions**
  - Parámetro: id del cuestionario del que se quieren obtener las preguntas con sus posibles respuestas. Se obtendrá de la llamada anterior y será un valor numérico.
- **local\_quizzes\_plugin\_course\_get\_students**
  - Parámetro: id del curso del que se quieren obtener los estudiantes. Para obtenerlo, se deberá realizar previamente la llamada *local\_quizzes\_plugin\_user\_get\_courses*.

El posible flujo de llamadas se puede visualizar de forma más sencilla en el siguiente diagrama:

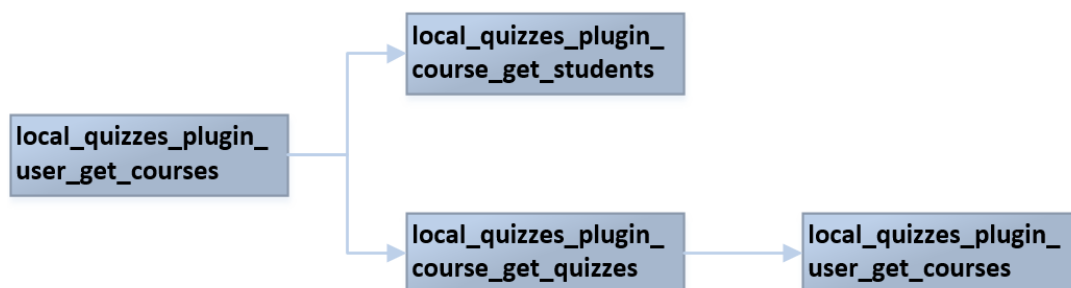


Figura 53 - Orden de las llamadas hacia la extensión para Moodle

# Anexo C: Instalación del servidor SELQuizServer

En este anexo se detalla cómo actualizar el servidor SELQuizServer para incluir las funcionalidades desarrolladas en este proyecto así como realizar una instalación desde el principio del mismo. Lo primero a tener en cuenta, es disponer de un servidor. En este proyecto se ha utilizado Apache, instalado en Ubuntu 12.04 LTS de 64 bits, sin embargo también puede ser instalado en Lighttpd entre otros.

## INSTALACIÓN DE PLAY FRAMEWORK

---

El servidor SELQuizServer está desarrollado con Play Framework. Cuando se desee compilar el servidor, se deberá disponer de esta herramienta. La versión utilizada ha sido la 2.1.1 y puede descargarse de la página web oficial<sup>7</sup>. Una vez descargado, debe descomprimirse. No es necesaria ninguna instalación, pero si incluirlo en la variable de entorno PATH. Para ello, se debe ejecutar el siguiente comando:

```
export PATH=RUTA/play-2.1.1:$PATH
```

*RUTA* deberá indicar el directorio donde se haya descomprimido el fichero que se ha descargado.

## INSTALACIÓN DEL SERVIDOR

---

En este apartado del anexo se explica cómo realizar una instalación del SELQuizServer y hacerlo operativo en un servidor Apache. Si se va a realizar una actualización del mismo, este paso puede ser obviado.

---

<sup>7</sup> <http://www.playframework.com>



El procedimiento es muy simple. Se ha generado un fichero comprimido, que contiene todos los archivos necesarios, incluidas las modificaciones realizadas sobre la versión actual.

Dentro de este fichero comprimido se incluye un directorio denominado *SELQuizServer* que a su vez incluye otro llamado *www*. El contenido de esta carpeta debe ser copiado al directorio raíz del servidor donde vaya a instalarse, que en el caso de Ubuntu será */var/www*. A continuación se deberá actualizar el servidor con las modificaciones realizadas.

### ACTUALIZACIÓN DEL SERVIDOR

---

Tanto si se está realizando una instalación nueva como si se va actualizar el servidor existente, es necesario compilar e introducir un fichero con todos los cambios en el servidor.

Para ello, en otro directorio del servidor o en caso de no ser posible, en otro ordenador, se debe copiar el contenido del directorio *SELQuizServer/carpeta compilación*, incluido en el fichero comprimido. En este directorio es donde se va a realizar la compilación del servidor. Para ello es necesario ejecutar dos comandos en una terminal dentro del directorio mencionado:

```
play clean-all  
play dist
```

Una vez se complete la compilación, deberá moverse el fichero situado en un subdirectorio de la carpeta donde se ha compilado, concretamente */target/scala-2.10/selquizserver\_2.10-1.0-SNAPSHOT.jar* a la carpeta */var/www/lib*.

Una vez copiado, el servidor puede iniciarse.

### INICIAR SERVIDOR

---

Dentro de una terminal, situándose en el directorio */var/www*, deberá ejecutarse el siguiente comando:

```
./start -Dconfig.resource=production.conf -Dhttp.port=9001 &
```

Donde el puerto (en el comando se utilizado el 9001) puede ser modificado según se desee.

En caso de que se vuelva a compilar el servidor y se mueva el fichero *selquizserver\_2.10-1.0-SNAPSHOT.jar* a la carpeta del servidor, este deberá ser detenido. Para ello, se debe ejecutar en una terminal, situándose en el servidor */var/www*:

```
sudo rm RUNNING_PID  
sudo netstat -putona
```

Se listará una serie de procesos que se están ejecutando en el sistema. Uno de ellos se denominará *java*. Este debe ser detenido, para ello, se debe anotar el número PID del proceso indicado y ejecutar el siguiente comando:

```
sudo kill -9 PID
```

Donde *PID* debe ser igual al valor obtenido anteriormente para el proceso *java*.

A continuación se deberá volver a copiar a la carpeta del servidor el fichero compilado y volver a iniciar el servidor con el primer comando indicado en este subapartado.

# Anexo D: Project abstract

## D.1 INTRODUCTION

Looking back at the last decades in education, it can be seen that an external element has been introduced progressively: technology. The benefits of this element are numerous [1]. Technology provides a broad spectrum of possibilities to conduct education. For example, the projector is widely used in class, particularly in university environment. Professors use projector as a tool to give their classes, but must a good use of this technology, away of distractions, must be made, becoming a tool which helps professors to display diverse contents.

Another example of technology entry in educational environment is online educational platforms, which are widespread in universities. This is the case of Aula Global, the system used in Universidad Carlos III de Madrid. Thanks to this platform, professors deliver resources to students, send them messages or even assess them. Nowadays, in most colleges there are systems as these to improve education.

During the class, professors may need to get feedback from students to assess them, to conduct a survey or to perform other tasks. The most frequently option chosen to get this information, has been to use a paper where students write the required information and then is handed over to the professor. Obviously, this process takes a lot of time and information cannot be analyzed in real time.

One way to solve this problem, could be to use the online education platform of the university. With it, the information could be collected using questionnaires or surveys, so that professors could receive information almost immediately. Furthermore, it adds the possibility to have the information already analyzed by a computer. As an example, professors could see a hit rate of students who have answered a question. Even an automatic correction of questions can be carried out and then scores can be added, also automatically, to student's grades.

However, this solution generates some problems. Not all students have a laptop in the classroom to access the educational platform website. Even when the class is given in a computer room, not always there are enough computers to be individually used. All these problems make impossible to use computers to establish communications in class between students and professors.

At the same time that technology has been introduced in classrooms, new types of devices have become very important in recent years among students: tablets and smartphones. The last element is the most widespread and a large majority of students always carry one. Smartphones have experienced a vertiginous developing in last seven years. Devices that only allowed making calls and sending messages have become small computers that allow doing a wide range of tasks through thousands of available applications.

The extended use of smartphones and tablets opens a world of possibilities due to the high potential of these devices. Internet connection of these devices, that permits to surf the web and to use different applications, gives the ability to interact electronically with students. In this way, the problem of getting feedback during the class can be resolved using of these devices, avoiding the mentioned problems related to the use of computers in class. Applications can be developed for these devices allowing getting information quickly and in a well-organized format.

## D.2 PROJECT MOTIVATION

This project tries to strengthen the bond between technology and education in order to improve its development and quality. This union allows obtaining support tools to the professor in order to get better teaching process, to keep the students attention, to improve the learning process of students and to create participatory and interactive classes.

Smartphones, carried by most students, can be established as the core to build new tools and services. The high potential of these devices permits developing applications allowing easy and quick interaction with students. In addition, smartphones make possible to access to new resources, such as social networks. These may help teaching process, increasing the communication between all the people involved.

In the previous section some problems not covered by the technology usually used in teaching have been described, leaving a long way to go. Once mentioned the possibilities of smartphones, it is reasonable to combine them to get an improvement in current methods. Within these limits the problems of the online educational platform are found. For example, there are some resources of it that can only be used from a web browser on a computer. Smartphones can increase functionality and accessibility of the system.

From the student perspective, the use of these mentioned devices is a very common practice. This would make possible to increase the attention in classroom to achieve a largest involvement and attendance. Students will be using a familiar technology, but also a new one, because it has been hardly used in class as another tool in the teaching process.

Interest in projects like this is increased due to the wide scope of possibilities and improvements that can be achieved in a future. At present it is a field that has barely been taken advantage of its benefits.

The main objective of this work is to allow answering questionnaires, created by professors in the online educational platform of the university (Moodle), from an application for mobile devices. Questionnaires can be obtained from Moodle to be managed by an external system. From this system, the questions of the questionnaires are going to be sent to the students and they will be received on their mobile devices with Android operating system. Students will be able to answer the questions from their application and the answers will be sent to the half-way system between Moodle and the Android application.

Below a definition of several milestones, which together make up the development of this project, can be found:

- To develop an application for Android devices to be used by students to answer the questions set out by professors. This application will support single and multiple-choice questions. To know when new questions are posted, the application will implement a notification service to display messages on the student devices. The application will connect with a server to download questions, receive notifications and send answers.
- To implement a complement for Moodle platform: given this system does not allow the access of external services to obtain questionnaires and information related to them, it is necessary to include a complement to extend its functionality. With it, other systems or applications will access Moodle to get questionnaires created by professors but also other necessary data, such as a list of courses of the professor or a list of students enrolled in those courses.
- To set up a simple control to monitor the presence in the classroom of the students when they answer a questionnaire, in order to prevent students from answering questions outside the classroom.

## D.3 ABSTRACT

As explained in the introduction, this project deals with developing an Android application connected to Moodle platform to answer the questionnaires created by professors. This system is outlined within Learning Management Systems.

These systems are used in academic environments to manage the educational process and include four basic functionalities. The first of these is to allow managing all the information involved in the teaching process in a single system. The second is to allow managing courses and students groups, providing different contents to each group. The next functionality that includes any LMS is to enable a communication between the members of the university community, especially between professors and students. The last functionality is to give tools to the professors to receive works or to make tests in order to assess the students.

There are many LMS and some of them are very widespread; such are the cases of Blackboard, Edmodo or Moodle. This last system is currently operating at Universidad Carlos III de Madrid. Moodle is a robust open source software with a large number of options. This system has to be installed on a server and will host a webpage so that students and professors can access the system. Moodle divides students in different courses and in each one, professors can add different resources, such as forums or questionnaires, which is the element this work is focused on.

The most common way to Access Moodle in order to reply questionnaires is through the webpage. However, as explained previously, answering these questionnaires from mobile devices facilitates this task during the class. There are diverse applications or clients able to connect to this platform and to carry out multiple tasks such as to download contents or to see varied information. Despite this and after an analysis of each one, it is concluded that none allows answering questionnaires.

On the other hand, the way to connect an application with Moodle has been analyzed. Using the web services (which allow a system to be accessible from another system in the same network) both can be connected. This LMS has a default web service with several functions to get diverse data from an external service. However, among these functions, none can be used for retrieving questionnaires.

To extend the Moodle web services in order to create new functions, the system allows installing plug-ins on which several new functions are encoded to be added to the system. With the new developed plug-in, another system or application can access Moodle to retrieve the questionnaires.

When studying different applications that could accomplish the objectives of this work, it was discovered an applications set of the University with similar purposes. Its name is mSuite and has been developed by the research group miSEL which belongs to the Software Engineering Lab of the Universidad Carlos III de Madrid.

mSuite has as main objective to provide an integrated system in order to help the professors to give classes, with tools to plan classes, to teach with the help of social networks sending students information in real time, and to assess them. However, this system does not connect with Moodle so it is impossible to retrieve questionnaires from it.

This system is composed of different applications. One of these is mTeachPlan, which helps professors to plan their classes. In the application, courses and sessions are created. Inside sessions, the professor can write questions and tweets to have ready them and send them directly to students during the classes. Currently, mTeachPlan is available for iOS and Android devices.

There is another application also used only by professors and it is called mTeachClass. It is used during the classes to display slides or videos. This application allows posting tweets automatically as the professor advances the slides or throwing questions to the students. As it happens in mTeachClass, this application is available for iOS and Android devices.

The third and last application of mSuite is mClassTest. It is used by students to reply questions thrown by the professor. There are versions for Android and iOS devices; however, the first is not a native application because has been developed in HTML5, so it is very restricted to include new functionalities and to improve its performance. The application is very easy to use; students can see a list with all available sessions, to which the student can subscribe. If a student subscribes one, he can see the questions associated to the session in order to reply them. In the case of iOS application, it implements a notifications service to notify the student when there are new questions that can be replied.

In addition to the described applications, there are also other different elements in the mSuite infrastructure: two own servers and the tools offered by Dropbox. The professor applications, mTeachPlan and mTeachClass are connected together through an own server and Dropbox.

mTeachClass, in addition to be connected to SelData and Dropbox, is also connected to Youtube to display videos in the sessions, to Twitter to send tweets and to SELQuizServer. This last is the bond between system and student applications, mClassTest.

This server has been developed with Play Framework, which gives tools to develop web applications. Its main objective is to make easier this process. It works with Java and Scala programming languages and allows building a web application without using HTML, but the result will be compatible with this technology. Among its advantages it is the abstraction or the services integration (as data storage) that it provides.

This server stores information about sessions, students and questions and calculates the statistics, which can be seen from webpages hosted on the same server. SELQuizServer also supplies questions to the students' applications, manages their requests of subscribing or cancellation of subscription of sessions and sends notifications to student's devices. The mTeachClass application connects to this server to upload data about sessions and questions, to activate or deactivate sessions and to send questions in order to be answered by the students.

The initial idea of this Project consisted on implementing an Android application connected directly to Moodle and to use it to answer the questionnaires, but this generated diverse problems. As previously commented, through a plug-in, an external system could access Moodle to retrieve data such as questionnaires. Nevertheless, to establish a notifications service would require setting up an intermediate system between Android application and the LMS.

mSuite provides a system with most of the utilities required. It allows sending notifications and has two applications for professors in order, among other things, to see the statistics of the class results. Therefore, an adaption of this system would allow to achieve the objectives, even exceeding expectations. These are some of the points which summarize this adaptation:

- To develop a native application for Android that allows to subscribe and to unsubscribe sessions, which connects to Twitter to identify the student and which allows answering multiple-choice questions (actual system only allows to manage single-choice questions). This application must implement a notification service to receive messages when new questions can be answered, to manage protected sessions and to save the student NIA.
- To implement a plug-in for Moodle to allow retrieving lists of courses, students and the questionnaires of these courses. mSuite will connect to Moodle through this plug-in.
- To modify the SELQuizServer to include multiple-choice questions and to store identification number NIA of the students to allow to upload the grades to the server.

The above points show the work performed on this Project. However, they do not complete the whole adaptation formulated of mSuite that, due to the high workload required, has been divided into two projects: the present one and another which is being performed simultaneously.

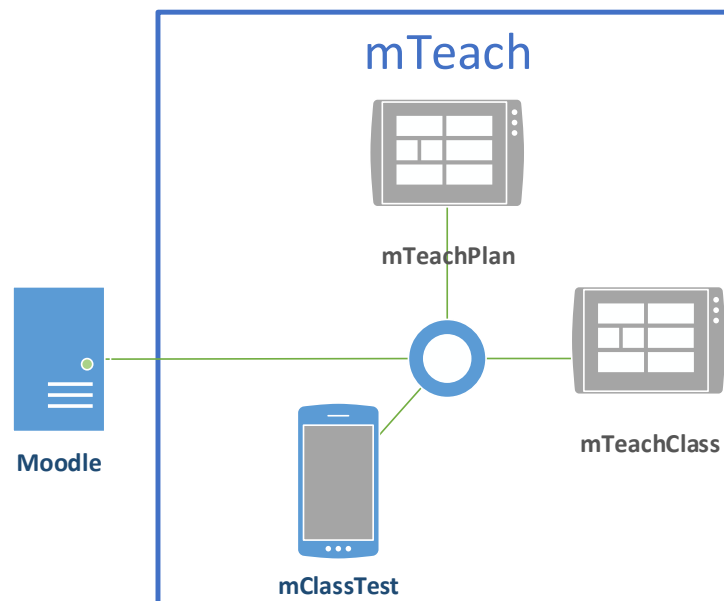


Figure 54 - General diagram of mSuite

In Figure 1, it is shown a simple diagram which allows to get a general idea of the system. There is a connection between Moodle and mTeachPlan applications. The professor, through this application, is going to retrieve the questionnaires created in the LMS to include the questions in the system. Those questions will be grouped in one or many sessions. The students, from their mClassTest application, will subscribe to one session when the class associated is started in order to see the questions proposed by the professor (according the professor sends them). Afterwards, the system calculates the grades and generates a file so the professor can upload them manually to Moodle. However, this last point is performed in the parallel project, and not in this.



Following, the work done is described in detail, describing each of the elements involved.

Regarding the Android application, it has been developed attempting to maximize supported devices. To that purpose, 2.3.3 has been defined as minimum required version of Android in order to run the application due a 98% of devices has and equal or newer version. This application, unlike the previous developed in HTML5 one, will implement a notification service. Specifically, it will use Google Cloud Messaging, the officially supplied system to send messages to Android devices.

mClassTest also will establish a control security by using passwords to protected sessions. When a student tries to subscribe a protected session, the application will request a password. If a student introduces a wrong password, the application will impede to subscribe the session and as a result, the session questions will not be accessible.

As regards to Moodle plug-in, it has four defined functions, each one associated with a different call. These calls allow retrieving a courses list of the professor who makes the call, a list of students of a specific course, the questionnaires created in a course and lastly the questions and possible answers of a questionnaire.

The plug-in just allows retrieving data. To upload grades to the LMS, mSuite would have to generate a file with the grades in order to be manually uploaded to Moodle by the professor. This is due to that, if several calls were developed in order to receive data and be stored in the system, the database would have to be modified. Making this modification would generate too many risks in the system.

About changes in SELQuizServer, these will allow the system to admit multiple-choice questions and save student NIA. These changes also include implementing new functions that allow creating password-protected sessions and multiple-choice questions. Other functions of the system have to be modified to register token of Android devices.

To facilitate Android Application development and to be clear about design and solve some issues about the implementation, a prototype was built with Fluid tool, which displays the different views expected of the application.

As for the implementation, it has been necessary to use a special development environment, specifically Android SDK, which provides a complete set of tools to develop applications for this system. For the Moodle plug-in or the changes in SELQuizServer, it has not been necessary a specific environment. Some problems were found in the implementation stage about Android minimum version, but they could be solved.

After the implementation, diverse tests have been performed in the system, according to a previously defined plan. With it, all system functionalities and their relation with requirements have been checked. At this stage the different tasks that can be performed with the Android application, the calls to SELQuizServer and the connection between Android application and the server are verified.

Once described the work done, it is advisable to comment some issues about project management.

It has followed a Waterfall model with feedback. This model defines a series of consecutive stages which are developed linearly. Thanks to the feedback, an information flow of each stage with all previous ones is established, so it is possible to perform adjustments in a previous phase to the currently being developed. This model fits accurately to this project due its linear nature.

The stages of this model start with one of Analysis of requirements in order to formally establish the objectives, followed by another of Software design in which the solution is defined in depth and divided in different components. The next stage is Implementation and Unit testing, where the components are implemented and tested. The following is Integration and system testing, where the different components are integrated and the complete system is tested. The last stage is called Operation and maintenance and is very important because in this phase the system operation in an extended period is reviewed.

In this project, two new stages have been added in order to plan it. One of them has been called Identifying of problem, where the problem is analyzed and some preliminary objectives are defined. The other is Project documentation, which includes the development of this document.

## D.4 CONCLUSIONS

At this point, some aspects of the carried out work must be commented. As explained in first sections, the main idea was modified to finally define one consistent and feasible. Initial target of developing an application to answer directly the Moodle questionnaires was modified to work on extending the functionality of an existing application system. This has produced large benefits for the development of the project, because the work has not started from scratch and it has been possible to speak directly with the developers of the system. Besides, the system is currently working so the implementation of the work is assured.

Because of the heavy workload needed from the modifications in the suite of applications mSuite to make possible a connection with Moodle and therefore an adaptation of several elements of the system, it was decided to divide the work into two parts. For this reason, is not yet possible to use the system with all new functionalities, such to get questionnaires of Moodle. The other project is being developed and is going to be submitted on September, so in the next academic year the system would be fully operational.

This Project has focused on developing a native Android application which allows answering multiple-choice questions, to access protected sessions, to request NIA to students, to modify the server that connects student's applications and to develop a plug-in for Moodle to retrieve questionnaires. In the other project, mTeachPlan is going to be modified to connect to Moodle through the plug-in, the iOS mClassTest application will admit multiple-choice questions and mTeachClass according changes will be performed in SELQuizServer.

Integrating both systems will allow to get a system that will help professors to give their classes at a minimum cost, just the produced for server maintenance and the costs of professor's devices. The smartphones of the students are not included in these costs given that they will use their own devices. Therefore, the viability of the system is guaranteed.

Regarding the work done, it has been very diverse implying the use of several technologies. This has increased strongly the total hours worked in the project so that it was necessary a learning period for each new technology to know how to use it. However, far of being a disadvantage, it has been a great incentive because it has allowed learning how to use new tools that will be very useful in the future, either in academic or business world. One of the technologies used has been Android that is one of the most extended operating systems for mobile devices in the world. This makes the training in applications development using this system very useful since it opens many doors in workplace.

To wrap up, the current importance of projects like this must be emphasized. Integration of technology in education opens a wide of possibilities, always with the main target of improving the education.

## D.5 FUTURE WORK

Joining together this project and the work simultaneously performed, will allow to obtain a system with multiple use possibilities and completely operative. However, mSuite has a very dynamic nature that makes it possible to add new functionalities easily. In addition, some already existing elements of the system can be improved so it is possible to improve the system. The future work that can be developed in this platform is summed up by the following points:

- To increase system security: calls to server should be sent through a secure protocol in order to encrypt data. At present, a call can be made without any key. This allows to someone not authorized to make modifications in the system if he knows the server address and the functions definitions. Other solution, instead of using a secure protocol, would be to establish a password for each mSuite user, making necessary to have one in order to make calls to the system.

- To increase systems for which there are applications developed of the mTeach system. mTeachPlan and mTeachClass are currently available only for iOS and Android. It would be advisable to develop these applications for new operating systems, such Windows Phone, which has a market share worthy of consideration.
- To add new question types: Moodle has multiple kinds of questions that include questions with pictures or questions where several words must be associated to each other, which could be very useful in mSuite.
- To create a webpage that allows accessing to system and answering questions: if a student has not an iOS or Android smartphone, there must be another way to students to answer questions.
- Greater integration with Moodle: the union between mSuite and Moodle can be improved, for example, to allow professors to obtain documents from Moodle to use them in their classes with mTeachClass.
- To enable that the grades are automatically updated to Moodle: currently, professors have to manually update the grades to the LMS.
- To enable signing in with other platforms (Facebook, Google...etc.) or even without none: adding options in order all students can use the application.
- Check NIA introduced by students: it should be checked to prevent the introduction of wrong data.

# GLOSARIO DE TÉRMINOS

- Framework: conjunto de herramientas para el desarrollo de software.
- Entorno de desarrollo: conjunto de herramientas de programación que facilitan ayudas al desarrollador.
- Cliente: aplicación informática destinada al intercambio de información con un servidor.
- LMS: sistema de gestión de aprendizaje, del inglés *Learning Management System*.
- SDK: kit de desarrollo de software, del inglés *software development kit*.
- PLE: Entorno Personal de Aprendizaje, del inglés *Personal Learning Environment*.
- TIC: tecnologías de la información y la comunicación.
- SOA: arquitectura orientada a servicios, del inglés *Service Oriented Architecture*.
- NDK: kit de desarrollo nativo, del inglés *Native Development Kit*.
- GCM: Google Cloud Messaging.