

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



USO DE TELÉFONOS INTELIGENTES PARA CONTESTAR PREGUNTAS EN CLASES PRESENCIALES

Grado en Ingeniería Informática

TRABAJO DE FIN DE GRADO

Curso 2014-2015

Autor:

Sergio Andrés Cabrera Concha

Tutor:

Ángel García Olaya

Agradecimientos

Quiero dedicar este proyecto a mi familia, y en especial a mis padres. Gracias a su esfuerzo, entrega y apoyo incondicional he logrado llegar a donde estoy, tanto a nivel académico como personal. Gracias por confiar en mí siempre.

Quiero dar las gracias a mi tutor, Ángel García Olaya, por su apoyo, su guía y su paciencia conmigo durante la realización de este proyecto. Tampoco puede faltar mi compañero Miguel Ángel Rozalén Lozano, por su colaboración y ayuda.

A todos los compañeros con los que he tratado durante mi etapa en la Universidad. Cada uno de ellos me ha aportado algo nuevo y han hecho más llevadero el esfuerzo diario. Por la motivación y las horas de estudio que nos han ayudado a mejorar en muchas ocasiones tanto individual como colectivamente.

Y a todas aquellas personas que han colaborado en mi desarrollo intelectual y en mi formación en la Universidad durante todos estos años.

Tabla de contenido

Agradecimientos	1
1. Introducción	9
1.1 Contexto.....	9
1.2 Descripción del problema	9
1.3 Objetivos	10
1.4 Estructura del documento.....	11
2. Estado del Arte	12
2.1 Sistemas Operativos Móviles	12
2.1.1 ANDROID	12
2.1.2 iOS.....	14
2.1.4 WINDOWS PHONE	14
2.1.5 Estudio de mercado sistemas operativos móviles	15
2.2 IDE de desarrollo	16
2.2.1 Android Studio	16
2.2.2 Eclipse	16
2.3 Comunicaciones.....	17
2.3.1 Wi-Fi	17
2.3.2 Comunicación por red móvil.....	17
2.3.3 Bluetooth	18
2.4 Modelos de Arquitectura.....	19
2.4.1 Modelo Cliente-Servidor.....	19
2.4.5 Modelo red entre pares (P2P)	20
2.5 Restricciones y definición de roles.....	20
2.5.1 Definición de roles	20
2.5.2 Restricciones del cliente	21
2.6 Estudio de soluciones.....	21
2.6.1 Selección de soluciones.....	21
2.6.2 Alternativas de implementación	24
2.7 Alternativas existentes.....	26
2.7.1 Socrative.....	26
2.7.2 Blackboard Learn.....	26
3. Análisis	27
3.1 Diagrama Casos de Uso	27
3.2 Especificación de los casos de uso	28
3.3 Requisitos software	32

3.2.1	Requisitos funcionales	33
3.2.2	Requisitos no funcionales	43
3.2.3	Matriz de trazabilidad	50
4.	Diseño	51
4.1	Definición de la arquitectura del sistema	51
4.2	Determinación de subsistemas de análisis	52
4.2.1	Interfaz gráfica servidor.....	52
4.2.2	Actividades aplicación móvil	53
4.2	Diagrama de clases	55
4.2.1	Clases de la interfaz del servidor	55
4.2.2	Aplicación Android	61
4.3	Diseño de interfaces del servidor	77
4.3.1	Diseño de la vista principal	77
4.3.2	Diseño de configuración de test	78
4.3.3	Diseño de vista de test.....	78
4.3.4	Diseño de estadísticas	79
4.4	Diseño de interfaces de la aplicación móvil	80
4.4.1	Diseño de identificación	80
4.4.2	Diseño de espera.....	80
4.4.3	Diseño de contraseña	81
4.4.4	Diseño de test.....	81
4.4.5	Diseño finalizar	82
5.	Implementación y pruebas	83
5.1	Especificación del entorno de desarrollo	83
5.2	Entorno de pruebas del sistema	84
5.3	Pruebas del sistema	85
5.3.1	Especificación de las pruebas del sistema	85
5.3.2	Pruebas del Sistema	85
6.	Gestión del proyecto.....	92
6.1	Metodología	92
6.1.1	Modelo de Prototipo.....	92
6.2	Planificación del proyecto	93
6.2	Cálculo de Costes.....	94
6.3.1	Personal	94
6.3.2	Material físico.....	95
6.3.3	Software	96

6.3.4 Coste total.....	96
7. Conclusiones y futuras líneas de mejora	97
7.1 Conclusiones.....	97
7.2 Líneas futuras de mejora	97
8. Bibliografía.....	99
Anexo A – Manual de Usuario TestUC3M.....	101
Requisitos para la instalación	101
Instalación de la aplicación.....	101
Uso de la aplicación	101
Caso 1. Test con respuesta única.....	105
Caso 2. Test con múltiples respuestas	109
Finalizando el test	113
Anexo B – Manual de Usuario TestUC3M (Escritorio).....	114
Requisitos para la instalación	114
Instalación de la aplicación.....	114
Uso del programa.....	114
Vista principal	114
Importar ficheros de alumnos o test.....	115
Configurar test	116
Iniciar test	119
Notas	121
Estadísticas	122

Índice de Figuras

Figura 1 - Android	12
Figura 2 - Capas de Android	13
Figura 3 - iOS	14
Figura 4 - Windows Phone.....	14
Figura 5 - Grafico cuota de mercado.....	15
Figura 6 - Tabla de cuota de mercado	15
Figura 7 - Android Studio	16
Figura 8 - Eclipse.....	16
Figura 9 - WiFi	17
Figura 10 – Evolución de la red móvil [20]	18
Figura 11 - Bluetooth	18
Figura 12 - Modelo Cliente-Servidor	19
Figura 13 - Modelo P2P	20
Figura 14 – Versiones de Android [24].....	22
Figura 15 - Red local con router inalámbrico.....	24
Figura 16 - Red local con router USB portátil.....	24
Figura 17 - Red local por red interna uc3m	25
Figura 18 - Casos de uso Alumno.....	27
Figura 19 - Casos de Uso Profesor.....	27
Figura 20 - Esquema de arquitectura del sistema	51
Figura 21 - Esquema de subsistemas de interfaz del servidor	52
Figura 22 - Subsistemas de actividades TestUC3M.....	53
Figura 23 - Diagrama de Clases Interfaz del Servidor.....	55
Figura 24 - Diagrama de clase ServerInterface.....	56
Figura 25 - Diagrama de clase MainViewController	57
Figura 26 - Diagrama de clase SettingsViewController	59
Figura 27 - Diagrama de clase TestViewController	60
Figura 28 - Diagrama de clase StatisticsController	61
Figura 29 - Diagrama de clase Singleton	61
Figura 30 - Diagrama de clase LoginActivity	63
Figura 31 - Diagrama de clase WaitActivity.....	64
Figura 32 - Diagrama de clase PasswordActivity	66
Figura 33 - Diagrama de clase TestActivity	67
Figura 34 - Diagrama de clase TestHideActivity.....	71
Figura 35 - Diagrama de clase de TestMultiActivity	72
Figura 36 - Diagrama de clase TestHideMultiActivity	75
Figura 37 - Diagrama de clase FinishedActivity	76
Figura 38 - Diseño vista principal servidor	77
Figura 39 - Diseño vista configuración de test	78
Figura 40 - Diseño vista de test del servidor	78
Figura 41 - Diseño vista estadísticas del servidor	79
Figura 42 - Diseño de vista de identificación.....	80
Figura 43 - Diseño de vista de espera	80
Figura 44 - Diseño de vista de contraseña.....	81
Figura 45 - Diseño de vista de test.....	81
Figura 46 - Diseño de vista de finalización.....	82

Figura 47 - Ciclo de vida Modelo de Prototipo.....	92
Figura 48 - Diagrama de Gantt de planificación del proyecto	93
Figura 49 - Ventana de inicio de TestUC3M	101
Figura 50 - Introducción de datos para la conexión.....	102
Figura 51 - Notificaciones de errores de conexión	102
Figura 52 - Ventana de conexión realizada y espera	103
Figura 53 - Ventana de introducción de contraseña.....	103
Figura 54 - Notificación de contraseña incorrecta	104
Figura 55 - Vista general de test de respuesta única	105
Figura 56 - Selección de respuesta.....	106
Figura 57 - Vista de botón finalizar test de respuesta única	106
Figura 58 - Vista de tiempo límite.....	107
Figura 59 - Vista de envío de respuesta única, pregunta a pregunta	108
Figura 60 - Vista general de test de respuesta múltiple.....	109
Figura 61 - Marcar una o más opciones.....	110
Figura 62 - Vista de botón finalizar test de respuesta múltiple	110
Figura 63 - Vista de tiempo límite en test de respuesta múltiple	111
Figura 64 - Vista de envío de respuesta múltiple, pregunta a pregunta.....	112
Figura 65 - Vista de finalización de test	113
Figura 66 - Vista principal servidor.....	114
Figura 67 - Opción de importar	115
Figura 68 - Ventana de importación	115
Figura 69 - Aviso de selección de test para configurar.....	116
Figura 70 - Ventana de configuración de test.....	116
Figura 71 - Selección tipo de test.....	117
Figura 72 - Selección de tiempo de respuesta	117
Figura 73 - Selección de modificación de respuesta	117
Figura 74 - Selección de penalización	117
Figura 75 - Selección de tipo de visualización.....	118
Figura 76 - Campo de introducción de contraseña.....	118
Figura 77 - Vista de test configurado	118
Figura 78 - Aviso de inicio de test	119
Figura 79 - Vista general de navegación de test	119
Figura 80 - Vista de botón Acabar.....	120
Figura 81 - Ventana de confirmación de finalización de test	120
Figura 82 - Opción de Notas	121
Figura 83 - Ventana de exportación de notas	121
Figura 84 - Ventana de introducción de datos de correo electrónico.....	121
Figura 85 - Opción de Estadísticas	122
Figura 86 - Aviso de estadística inexistente	122
Figura 87 - Ventana de gráfico de barras.....	122

Índice de Tablas

Tabla 1 - Ejemplo tabla casos de uso	28
Tabla 2 - Caso de Uso 1	28
Tabla 3 – Caso de Uso 2	29
Tabla 4 - Caso de Uso 3	29
Tabla 5 - Caso de Uso 4	29
Tabla 6 - Caso de Uso 5	30
Tabla 7 - Caso de Uso 6	30
Tabla 8 - Caso de Uso 7	31
Tabla 9 - Caso de Uso 8	31
Tabla 10 - Caso de Uso 9	32
Tabla 11 - Ejemplo de requisito	32
Tabla 12 - Requisito Funcional 1	33
Tabla 13 - Requisito Funcional 2	33
Tabla 14 - Requisito Funcional 3	34
Tabla 15 - Requisito Funcional 4	34
Tabla 16 - Requisito Funcional 5	34
Tabla 17 - Requisito Funcional 6	35
Tabla 18 - Requisito Funcional 7	35
Tabla 19 - Requisito Funcional 8	35
Tabla 20 - Requisito Funcional 9	36
Tabla 21 - Requisito Funcional 10.....	37
Tabla 22 - Requisito Funcional 11.....	37
Tabla 23 - Requisito Funcional 12.....	37
Tabla 24 - Requisito Funcional 13.....	38
Tabla 25 - Requisito Funcional 14.....	38
Tabla 26 - Requisito Funcional 15.....	38
Tabla 27 - Requisito Funcional 16.....	39
Tabla 28 - Requisito Funcional 17.....	39
Tabla 29 - Requisito Funcional 18.....	39
Tabla 30 - Requisito Funcional 19.....	40
Tabla 31 - Requisito Funcional 20.....	40
Tabla 32 - Requisito Funcional 21.....	40
Tabla 33 - Requisito Funcional 22.....	41
Tabla 34 - Requisito Funcional 23.....	41
Tabla 35 - Requisito Funcional 24.....	41
Tabla 36 - Requisito Funcional 25.....	42
Tabla 37 - Requisito Funcional 26.....	42
Tabla 38 - Requisito Funcional 27.....	42
Tabla 39 - Requisito No Funcional 1	43
Tabla 40 - Requisito No Funcional 2	43
Tabla 41 - Requisito No Funcional 3	44
Tabla 42 - Requisito No Funcional 4	44
Tabla 43 - Requisito No Funcional 5	44
Tabla 44 - Requisito No Funcional 6	45
Tabla 45 - Requisito No Funcional 7	45
Tabla 46 - Requisito No Funcional 8	46

Tabla 47 - Requisito No Funcional 9	46
Tabla 48 - Requisito No Funcional 10	47
Tabla 49 - Requisito No Funcional 11	47
Tabla 50 - Requisito No Funcional 12	47
Tabla 51 - Requisito No Funcional 13	48
Tabla 52 - Requisito No Funcional 14	48
Tabla 53 - Requisito No Funcional 15	48
Tabla 54 - Requisito No Funcional 16	49
Tabla 55 - Requisito No Funcional 17	49
Tabla 56 - Requisito No Funcional 18	49
Tabla 57 - Matriz de trazabilidad requisitos	50
Tabla 58 - Tabla descripción subsistema Menú Superior	52
Tabla 59 - Tabla descripción subsistema Tabla de Alumnos	53
Tabla 60 - Tabla descripción subsistema Tabla de Test	53
Tabla 61 - Tabla descripción subsistema Botón de Inicio de Test	53
Tabla 62 - Tabla descripción subsistema Vista de Test	53
Tabla 63 - Tabla descripción subsistema Login	54
Tabla 64 - Tabla descripción subsistema Password	54
Tabla 65 - Tabla descripción subsistema Test	54
Tabla 66 - Tabla descripción subsistema Finalizar	54
Tabla 67 - Ejemplo tabla Prueba de Sistema	85
Tabla 68 - Prueba del Sistema 1	85
Tabla 69 - Prueba del Sistema 2	86
Tabla 70 - Prueba del Sistema 3	86
Tabla 71 - Prueba del Sistema 4	87
Tabla 72 - Prueba del Sistema 5	87
Tabla 73 - Prueba del Sistema 6	88
Tabla 74 - Prueba del Sistema 7	88
Tabla 75 - Prueba del Sistema 8	89
Tabla 76 - Prueba del Sistema 9	89
Tabla 77 - Prueba del Sistema 10	90
Tabla 78 - Prueba del Sistema 11	90
Tabla 79 - Prueba del Sistema 12	91
Tabla 80 - Prueba del Sistema 13	91
Tabla 81 - Tabla de Coste de Personal	94
Tabla 82 - Tabla de Costes de Material Físico	95
Tabla 83 - Tabla de Costes de Software	96
Tabla 84 - Tabla de Costes Finales	96

1. Introducción

En esta sección se detalla el contexto del trabajo a realizar. Se expondrá de manera concisa el problema planteado y los objetivos que se buscan alcanzar al final del proyecto, así como un resumen de la estructura de este documento.

1.1 Contexto

El mundo de la enseñanza siempre ha sido un entorno tradicional hasta hace poco. Con el gran salto tecnológico que se ha producido en los últimos años, sería incomprensible no sacar partido de ello para potenciar aún más los recursos docentes. Estos recursos suponen por un lado mejorar la experiencia del alumno y facilitar el trabajo de los docentes. El uso de pizarras electrónicas, proyectores para presentaciones o la entrega de trabajos de forma digital son algunos ejemplos de cómo aprovechar estos recursos.

Por otro lado, el ámbito de la tecnología móvil está experimentando un crecimiento y popularidad sin precedentes a lo largo de los últimos años, llegando casi a superar en uso a los dispositivos más tradicionales [1]. Los teléfonos inteligentes o “smartphones” se han estandarizado como el dispositivo móvil por excelencia, otorgando al usuario una versatilidad en sus funciones gracias a sus constantes mejoras en potencia y accesibilidad. Al disponer de más capacidades surgen nuevas oportunidades como el desarrollo de aplicaciones, que cuentan con un gran mercado hoy en día. Según datos recientes, los usuarios de Android tienen a su disposición aproximadamente 1,5 millones de aplicaciones para descargar, mientras que los usuarios de iOS contarían con unos nada despreciables 1,4 millones[2]. Y los números no hacen más que aumentar con el paso del tiempo.

No es de extrañar pues que con su demostrada popularidad, se quiera incorporar este dispositivo como una herramienta complementaria para la enseñanza. Entre sus bazas se cuenta su gran popularidad, su portabilidad y su facilidad de manejo.

1.2 Descripción del problema

En el ámbito educacional, las pruebas que evalúan los conocimientos del alumnado al final o durante el periodo de evaluación son conocidas como exámenes.

La realización de estos exámenes puede realizarse de diferentes maneras, por escrito, en ordenador, de manera oral, etc. El problema que nos ocupa se centra en la realización presencial de un examen tipo test.

En la universidad los exámenes tipo test suelen ser frecuentes para llevar un seguimiento del desempeño en clase de manera rápida y sencilla. Las alternativas disponibles por el momento son algo rudimentarias. Se cuenta con la opción de realizarlo en papel, por escrito. Sin embargo esto conlleva un proceso lento de despliegue y de corrección por parte del profesor.

Se cuenta con la opción de realizarlo en ordenadores. Esto beneficia a ambas partes implicadas, favoreciendo su realización, despliegue y corrección, pero es una opción que no siempre es viable, ya que las pruebas se pretenden realizar en sesiones magistrales.

Por último, se puede realizar gracias a un software controlador instalado en el ordenador del profesor, recogiendo las respuestas a través de un sistema hardware compuesto por un receptor y unos mandos a distancia que leen las respuestas del alumno. Es la manera más automática hasta la fecha pero algo engorrosa pues precisa de componentes externos y su despliegue presenta alguna incomodidad.

Con este proyecto se busca abordar este problema y simplificar la realización de los test mediante la creación de software para dispositivos móviles que use el alumnado y para el control del profesor.

1.3 Objetivos

El objetivo principal es proporcionar un sistema funcional para la realización de test mediante el uso de un dispositivo móvil y un programa instalado en el ordenador del profesor.

El sistema se compondrá de:

- Una aplicación móvil, desarrollada para el sistema operativo Android, que proporcionará al alumno una interfaz para la visualización y realización de los test.
- Un programa de escritorio, compatible con los sistemas operativos actuales (Windows, Linux, Mac Os), que permitirá realizar un seguimiento del test, así como de la recolección y calificación de las respuestas enviadas por el alumno. Funcionará como un servidor al que se conectarán los dispositivos.

Los test se obtendrán de la herramienta Aula Global de la Universidad, implementado sobre el sistema **Moodle**[3]. Estos test se podrán importar de forma automática gracias al programa de escritorio.

Se busca dar soporte a una base de usuarios de manera eficiente, empleando en el proceso tecnología orientada específicamente a las comunicaciones cliente-servidor y de aplicaciones móviles. Se tendrán en cuenta aspectos de seguridad en las comunicaciones e intercambio de datos, así como el control de los alumnos que puedan realizar la prueba, pues un requisito indispensable es estar físicamente en clase para poder hacerla.

Se tienen en cuenta otros objetivos derivados de este proyecto:

- El presente trabajo trata de unificar conocimientos adquiridos a lo largo de la carrera, tanto en el ámbito de tecnologías de interfaces y comunicaciones como orientadas a la programación para dispositivos móviles. Se busca poner en práctica todo lo aprendido a lo largo de la formación del alumno.

- Se busca desarrollar una aplicación útil e intuitiva que se pueda usar regularmente una vez terminada, y que sirva como apoyo a la docencia.

1.4 Estructura del documento

El presente documento se divide en diferentes secciones, las cuales se presentan resumidas a continuación:

- **Introducción:** Engloba la parte introductoria del trabajo, el contexto del trabajo, la descripción del problema, los objetivos que se buscan completar, así como la estructuración del documento.
- **Estado del Arte:** Engloba el estudio de las tecnologías a usar y de las aplicaciones existentes en el mercado, si las hubiera, así como un estudio de las posibles soluciones y alternativas para solucionar el problema planteado.
- **Análisis:** Engloba toda la información recolectada durante la fase de análisis del proyecto. En esta sección se encuentran aspectos como los requisitos de la aplicación, los casos de uso, o las gestiones del proyecto en cuanto a planificación y entorno.
- **Diseño:** Engloba los aspectos de diseño de diagramas de clases para entender el funcionamiento del sistema. Incluye prototipos del diseño de interfaces de la aplicación.
- **Implementación y pruebas:** Engloba los aspectos más técnicos del documento. Se describe el entorno de implementación y las pruebas realizadas.
- **Gestión del proyecto:** Engloba los aspectos de gestión y de planificación del proyecto acordes a la metodología empleada.
- **Costes del proyecto:** Engloba los aspectos financieros para el cálculo monetario del proyecto en total.
- **Conclusiones:** Engloba las ideas finales obtenidas tras la realización del proyecto, así como las posibles líneas de mejoras para el futuro.
- **Bibliografía:** Engloba la bibliografía empleada durante la redacción de este documento.
- **Anexo A:** Manual de usuario para la aplicación móvil.
- **Anexo B:** Manual de usuario para la aplicación de escritorio.

2. Estado del Arte

En esta sección se analizarán las alternativas de tipo software y metodologías a considerar para la realización del proyecto, con una breve descripción de las características de cada una de ellas y argumentando la elección realizada en cada caso.

Se analizarán las posibles soluciones a tener en cuenta, valorando sus características y justificando la elección de la solución elegida.

En caso de haber aplicaciones existentes que compartan características con el proyecto a desarrollar, se tendrán en cuenta.

2.1 Sistemas Operativos Móviles

Se presentan los principales sistemas operativos móviles que dominan el mercado y sobre los que sería conveniente desarrollar la aplicación.

2.1.1 ANDROID

Android [4] es un sistema operativo para dispositivos móviles, basado en un kernel de Linux, desarrollado por la empresa Google. El proyecto es de carácter “open source”, es decir, de código abierto. Su interfaz gráfica permite al usuario usar gestos táctiles para navegar por ella.



Figura 1 - Android

En 2003, Andy Rubin, Rich Miner, Nick Sears y Chris White fundaron la Android, Inc. Su propósito inicial fue el de desarrollar un sistema operativo para dispositivos móviles, concretamente para cámaras de fotos digitales. Cuando vieron que la cuota de mercado no era muy grande, decidieron orientar su desarrollo a móviles, para crear un sistema operativo que rivalizase con los sistemas operativos Symbian o Windows Mobile. En 2005 la empresa es adquirida por Google, la cual buscaba entrar en el negocio de los terminales móviles. El sistema operativo propiamente dicho no sería presentado hasta 2007.

Las principales características que destacaban de Android eran su flexibilidad, su capacidad de actualización y su versatilidad. Gracias a estas virtudes goza de mucha popularidad, no solo entre los dispositivos móviles. Se puede encontrar instalado en dispositivos como consolas, relojes, TVs o cámaras, por mencionar algunos ejemplos.

Las actualizaciones importantes del sistema operativo corren a cargo del equipo de desarrolladores de Google. Las actualizaciones se lanzan en intervalos de seis a nueve meses de separación. La versión más reciente es la 5.x “Lollipop”.

Estas actualizaciones se realizan teniendo en mente la gama insignia de Google, denominada Nexus. Una gama de dispositivos manufacturados por fabricantes externos con la colaboración de Google. Para el resto de dispositivos del mercado es necesario realizar ajustes y modificaciones, pues las diferencias a nivel de hardware son numerosas entre fabricantes y modelos.

El lenguaje de programación sobre el que está basado Android es **Apache Harmony** [5]. Este lenguaje es una reinterpretación legal de las librerías de clases de Java, desarrollada por la

Apache Software Foundation [6]. El código corre sobre una máquina virtual denominada **Dalvik** [7].

Su arquitectura está formada por cuatro capas. En el primer nivel se encuentra la capa de aplicaciones. En el segundo nivel se encuentra el entorno de aplicaciones al que acceden las aplicaciones para usar funciones del terminal. En el tercer nivel se encuentran las librerías nativas y el Runtime de Android, donde se ejecuta la máquina virtual Dalvik. Por último se encuentra el núcleo o kernel que es el que interactúa directamente con el hardware.



Figura 2 - Capas de Android

2.1.2 iOS

iOS [8] es un sistema operativo para dispositivos móviles creado y desarrollado por Apple Inc.

Es un software propietario, está diseñado específicamente para hardware de Apple, tales como iPhone, iPad o iPod touch.



Figura 3 - iOS

Fue presentado al mundo en 2007, durante la Macworld Conference & Expo. Su lanzamiento se produjo en Junio de ese mismo año. Su nombre en aquel entonces era iPhone OS, pues fue presentado junto al primer modelo de iPhone. Su estructura está basada en el sistema operativo de escritorio **OS X** [9], que a su vez está basado en Unix. Su desarrollo parte de la idea de portar las características del sistema operativo OS X a un dispositivo móvil como el iPod. Se produjo entonces una competición entre los equipos encargados del desarrollo del sistema operativo y de iPod para encontrar una solución al problema. Scott Forstall y su equipo se llevarían la victoria gracias a la consecución del iPhone OS.

Las actualizaciones importantes del sistema se lanzan de manera anual. La última versión del sistema es la 8.x. En 2010, a partir de la versión 4.0 el sistema pasó a denominarse como iOS.

Sus principales características eran:

- Llevar la experiencia de un sistema de escritorio a un dispositivo móvil.
- El uso de aplicaciones parecidas a las aplicaciones de escritorio.
- Diseño y usabilidad de una interfaz gráfica controlada por gestos táctiles.

En un primer momento, iOS no soportaba la instalación de aplicaciones externas, ni se planeaba dar soporte a su desarrollo, más allá de aplicaciones basadas en web.

Más adelante cambiaron de idea y liberaron el primer SDK para el desarrollo de aplicaciones nativas. El desarrollo de las aplicaciones se basa en el lenguaje **Objective-C** [10]. Desde entonces la popularidad de su servicio de aplicaciones creció exponencialmente y es uno de los principales referentes como contenedor de apps de carácter comercial.

2.1.4 WINDOWS PHONE

Windows Phone [11] es un sistema operativo de dispositivos móviles desarrollado por Microsoft.

Surge como sucesor de Windows Mobile, un sistema operativo diseñado para dispositivos móviles de corte empresarial como el

Pocket PC. La idea original consistía en actualizar y renovar Windows Mobile de forma significativa. Tras sufrir varios retrasos en el proyecto, se canceló.



Figura 4 - Windows Phone

Su desarrollo como sistema móvil independiente comienza en 2008. Su lanzamiento se produjo en 2010, bajo el nombre de Windows Phone 7.

Su objetivo es el de conquistar cuota de mercado en el negocio de dispositivos móviles, ofreciendo una alternativa a los sistemas ya instaurados como Android o iOS.

2.1.5 Estudio de mercado sistemas operativos móviles

Se estudia la cuota de mercado que poseen los tres principales sistemas operativos previamente analizados.

El estudio está realizado por la empresa de análisis de telecomunicaciones IDC (**International Data Corporation**)[12].

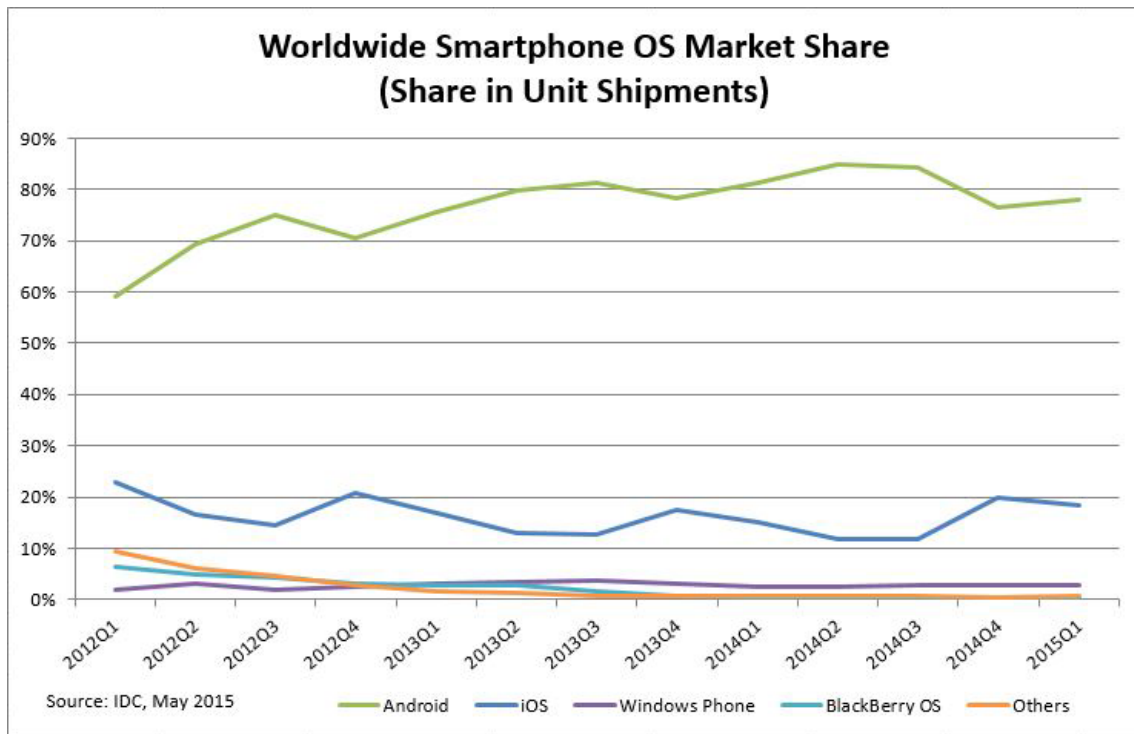


Figura 5 - Grafico cuota de mercado

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Figura 6 - Tabla de cuota de mercado

Los datos abarcan el primer trimestre de 2015 [13]. Como se observa en el gráfico y más detalladamente en la tabla, Android es el sistema con la cuota de mercado más grande, llegando al 78%. Su siguiente competidor es iOS con una cuota del 18,3%.

2.2 IDE de desarrollo

El entorno de desarrollo juega un papel importante en la implementación, pues se encarga de unificar las herramientas de desarrollo para facilitar su realización. El acrónimo **IDE** viene del inglés “Integrated Development Environment”, entorno de desarrollo integrado.

2.2.1 Android Studio

Android Studio [14] es un entorno de desarrollo gratuito y diseñado específicamente para el desarrollo en Android.

Tiene relativamente poco tiempo de vida, fue anunciado por primera vez en el año 2013, en la conferencia Google I/O. Google se encarga de su desarrollo.



Figura 7 - Android Studio

Tras una fase en beta, su primera versión estable saldría a finales de 2014.

Entre sus características se encuentran:

- Un editor de interfaz gráfico con posibilidad de ver el resultado mientras se va configurando.
- Su estructura se basa y da soporte a **Gradle**[15]. Gradle es una herramienta para automatizar la construcción y despliegue del proyecto, ayudando en tareas de compilación, pruebas y empaquetado.
- Editor inteligente diseñado para facilitar la codificación y el análisis del código.
- Plantillas para la creación de actividades y proyectos.
- Emulador de dispositivos Android.

2.2.2 Eclipse

Eclipse [16] es un entorno de desarrollo gratuito y open source. Su objetivo es el facilitar el desarrollo de aplicaciones de variada índole. No se centra en un único lenguaje de programación, mediante el uso de complementos se puede adaptar a las necesidades que se requieran. Entre otros, soporta Ada, ABAP, C/C++, COBOL, Fortran, Haskell, JavaScript, Java, Lasso, Natural, Perl, PHP, Python, Ruby, Groovy o Erlang.



Figura 8 - Eclipse

Eclipse comenzó como un proyecto de Smart Canadá, para desarrollar un IDE basado en Java. Su última versión es Eclipse Luna. Incorpora soporte para la última versión de Java, la número 8.

Está fundamentado como un espacio de trabajo, el cual se puede personalizar mediante el uso de entornos y complementos. Entre sus entornos hay de todo tipo, como el Eclipse Java Development Tools, para Java y Scala, el Eclipse CDT para lenguajes como C/C++ o el Eclipse PDT, para PHP.

Para el proyecto se usaría la extensión **ADT (Android Development Tools)** [17], que permite la creación de proyectos Android, así como de las librerías y emuladores que se requieran para realizar pruebas o ejecutar aplicaciones.

2.3 Comunicaciones

En este apartado veremos algunas alternativas de comunicación entre dispositivos. Al tratarse de dispositivos móviles, solo tendremos en cuenta métodos de comunicación inalámbrica.

2.3.1 Wi-Fi

Wi-Fi es un sistema de comunicación local inalámbrico entre dispositivos. Las redes inalámbricas locales establecidas a través de Wi-Fi son conocidas como WLAN, basadas en el estándar IEEE 802.11 [18].



Figura 9 - WiFi

Su historia se remonta a un proyecto de la Universidad de Hawaii llamado ALOHAnet. Este sistema fue la base de lo que vendría después, comunicando islas de Hawaii mediante ondas UHF inalámbricas.

El estándar 802.11 engloba una serie de especificaciones para el control de acceso de medios (MAC) y la capa física (PHY) para la implementación de una red local inalámbrica. La primera versión del protocolo 802.11 fue lanzada en 1997, proporcionando velocidades de conexión de hasta 2Mbps/s. En la actualidad el estándar ha ido evolucionando para dar soporte a velocidades mayores de transmisión de datos, así como ampliar el rango de acción.

2.3.2 Comunicación por red móvil

Las comunicaciones a través de la red móvil son específicas de dispositivos móviles, valiéndose de las antenas y cobertura de las compañías telefónicas [19].

La primera generación de comunicaciones se centraba en mejorar la calidad de transmisión de voz, las tecnologías empleadas se conocerían como AMPS (Advanced Mobile Phone Systems). Eran sistemas de comunicación analógicos, pues entonces no primaba tanto el contenido digital.

El siguiente paso en la evolución de las comunicaciones fueron las redes GSM y CDMA, que serían conocidas como 2G. A diferencia de las redes analógicas, estas eran digitales y se centraban ya no solo en mejorar la calidad de transmisión de voz. En esta etapa el uso de red móvil empezaría a extenderse rápidamente y, a medida que evolucionaban los terminales, su uso empezaba a ser más frecuente. Cobraban más importancia otros elementos que ganarían en popularidad como los mensajes de texto, mensajes multimedia o un primer acercamiento a la navegación por Internet a través del móvil, ya sea navegando o con descargas de contenido como tonos de llamada.

A medida que evolucionaban los terminales era evidente que había que mejorar la red para dar soporte a un mayor número de usuarios y de datos. Mejorar las tasas de transmisión de datos y la calidad de las llamadas de voz. Entran en juego elementos como el envío de emails, videos, videoconferencias, navegadores de internet funcionales o el streaming de

música/videos entre otros. Todo ello necesita una infraestructura superior a lo que 2G podía ofrecer. La siguiente generación de comunicaciones sería conocida como 3G. Gracias al cambio en el modo de transferencias de datos mediante el envío de paquetes, se logran alcanzar velocidades de transmisión cercanas a las vistas en comunicaciones de red más propias del hogar, llegando hasta 14Mbit/s.

En la actualidad llegamos a la cuarta generación de telecomunicaciones móviles, 4G. Su propósito es soportar el incremento aún mayor de usuarios y de consumo de datos gracias a la popularidad de los smartphones. Los estándares actuales más representativos son WiMAX, en Estados Unidos, y LTE en Europa. Con estos estándares se logra llegar a una tasa de descarga de hasta 100Mbit/s.



Figura 10 – Evolución de la red móvil [20]

2.3.3 Bluetooth

Bluetooth [21] es un estándar de tecnología inalámbrica para el intercambio de datos en distancias cortas usando ondas de radio UHF. La comunicación engloba dispositivos móviles y fijos mediante el uso de adaptadores o módulos integrados. Fue inventado en 1994 por la empresa de telecomunicaciones Ericsson.



Figura 11 - Bluetooth

Permite crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales. La mayoría de los dispositivos actuales poseen esta tecnología de serie, pues es eficiente y rápida.

2.4 Modelos de Arquitectura

Se presentan las alternativas de modelos de arquitectura a tener en cuenta.

2.4.1 Modelo Cliente-Servidor

El modelo de **cliente-servidor** [22] se basa en un modelo de aplicación distribuida que distribuye la carga de tareas entre los que proporcionan el servicio, llamados servidores, y los que realizan las peticiones de servicio, llamados clientes.

El servidor ejecuta la mayor parte de la carga en la mayoría de los casos, compartiendo recursos con los clientes conectados. Los clientes se encargan de recibir y/o enviar datos al servidor.

El intercambio de mensajes se produce mediante el patrón de petición-respuesta. El cliente envía una petición y el servidor le devuelve la respuesta. Para que la conexión y el intercambio de mensajes entre los terminales funcionen, deben compartir una serie de reglas, definidas por el protocolo de comunicación.

El servidor debe ser capaz de atender las peticiones de 1 a n clientes al mismo tiempo. Siempre y cuando se disponga de un terminal con unas características y especificaciones adecuadas a la carga de trabajo. Para organizar los servicios se recurre a herramientas como la multitarea y la ejecución de hilos que trabajen de forma paralela atendiendo a cada cliente.

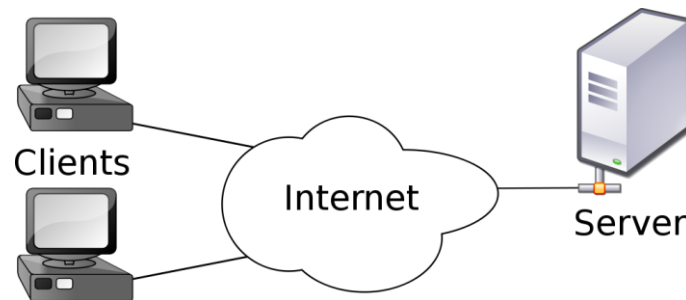


Figura 12 - Modelo Cliente-Servidor

2.4.5 Modelo red entre pares (P2P)

El modelo de comunicaciones llamado **peer-to-peer (P2P)** [23] es una arquitectura de aplicación distribuida que divide la carga de trabajo y de tareas entre los participantes o pares. Cada participante tiene los mismos privilegios y la misma importancia dentro de la red.

Los participantes actúan como proveedores y consumidores a la vez. Ceden parte de sus recursos a los demás participantes. Estos recursos se traducen en intercambio de datos, ancho de banda o capacidad de computación.

El modelo tradicional prescinde de la figura del servidor central que se encarga de las peticiones. Para el problema planteado se podría considerar un sistema híbrido con un servidor central que almacena cierta información y responda a peticiones de los participantes. Estos seguirían compartiendo recursos, aunque en menor medida.

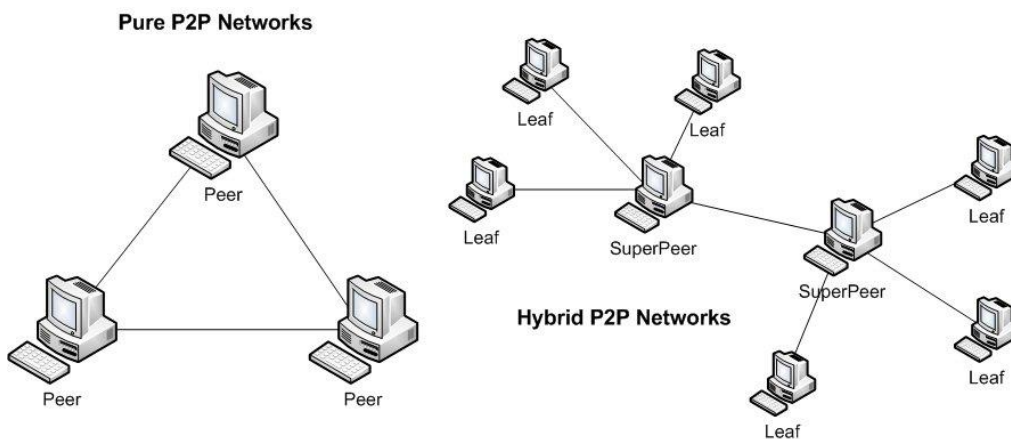


Figura 13 - Modelo P2P

2.5 Restricciones y definición de roles

2.5.1 Definición de roles

En este apartado se definen los roles de los participantes en el proyecto.

Cliente: Ángel García Olaya. Es el cliente principal del proyecto. Valorará la realización del mismo a lo largo del desarrollo, así como proporcionará soporte para la realización del mismo mediante reuniones bisemanales para intercambiar impresiones.

Equipo de desarrollo: El equipo de desarrollo está formado por una sola persona, Sergio Cabrera Concha.

Usuarios: Los principales usuarios del sistema serán los profesores de la Universidad que decidan utilizarlo, y los alumnos que realizarán los test a través de la aplicación.

2.5.2 Restricciones del cliente

Las restricciones impuestas para este proyecto son las siguientes:

- La plataforma de desarrollo de la aplicación será Android.
- El alumno debe estar presente en clase para realizar el test.
- No se podrán usar los test directamente desde la aplicación de Moodle integrada en la universidad.

2.6 Estudio de soluciones

En este apartado se expondrán las alternativas al problema propuesto que se han tenido en cuenta, así como la solución elegida y su justificación.

2.6.1 Selección de soluciones

Tras analizar el problema planteado, y teniendo en cuenta las tecnologías disponibles, se eligen las siguientes alternativas:

2.6.1.1 Arquitectura Cliente-servidor

Se decide utilizar la estructura cliente servidor, por ser la que más se adapta a las necesidades expuestas y por su implementación.

Las justificaciones de la elección son las siguientes:

- **Diversidad de clientes.** La idea global del sistema es que los alumnos podrán realizar los test a través de una aplicación en su dispositivo móvil. Como hemos visto antes, existen tres sistemas operativos que se reparten el grueso de usuarios. Este proyecto engloba únicamente los aspectos de la aplicación relevantes para Android. No podemos ignorar que puede haber alumnos con otros sistemas operativos en clase, por lo que es necesario usar una arquitectura que pueda dar soporte al cliente, independientemente del que sea. Con esta arquitectura y la definición de un protocolo de comunicaciones neutro se consigue este objetivo.
- **Carga de trabajo.** Es más sencillo delegar la mayor parte de la carga de trabajo al servidor, pues se ejecutara en una maquina más potente y con más recursos que los clientes. Los clientes en sí solo deben preocuparse de interpretar el test y de enviar las respuestas.

2.6.1.2 Comunicación por Wi-Fi

Se decide utilizar la comunicación a través de red Wi-Fi.

Las justificaciones de la elección son las siguientes:

- **Facilidad de comunicación.** Prácticamente todos los dispositivos móviles actuales que pueden ejecutar la aplicación cuentan con un módulo Wi-Fi para la conexión a Internet. Mediante el uso de un protocolo TCP/IP la conexión se realizará de manera segura y fiable.
- **Facilidad de implementación.** La conexión podrá ser de forma local a través de una red montada en el aula o a través de la red interna de la Universidad, lo que facilitaría el aspecto de implementación y aseguraría que los alumnos tengan que estar en clase para realizar la conexión.

2.6.1.3 Sistema Operativo Android

Se decide desarrollar la aplicación para el sistema operativo Android. La justificación viene dada por un requisito explícito del cliente para el desarrollo de la aplicación. Como elección unilateral por parte del alumno, se decide dar soporte a sistemas a partir de la versión 2.3.3, ya que supone abarcar la totalidad prácticamente del mercado actual de usuarios. Podemos ver los porcentajes en la siguiente tabla y gráfico:

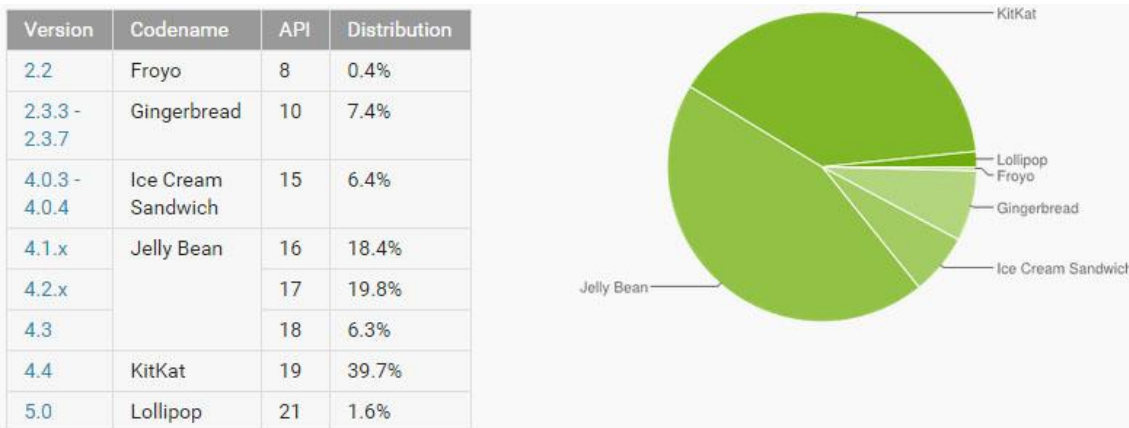


Figura 14 – Versiones de Android [24]

2.6.1.4 Lenguaje de programación Java

Se decide utilizar el lenguaje de programación **Java** [25] para la implementación del servidor.

Las justificaciones son las siguientes:

- **Portabilidad del programa.** Gracias a la programación en Java es posible exportar el programa a cualquier sistema operativo actual.
- **Facilidad de implementación.** Se valora la familiarización y la experiencia que se ha adquirido a lo largo de la carrera con el lenguaje. Esto supone menos tiempo de aprendizaje y de implementación. Además, las librerías de Java disponibles facilitan la comunicación y la creación de un protocolo. Este lenguaje también sirve para el desarrollo de la aplicación móvil y para su integración con el servidor de manera sencilla.
- **Diseño de interfaz.** Mediante el uso de un complemento, se puede realizar la interfaz del programa gracias al lenguaje de **JavaFX** [26]: un lenguaje basado en Java para la realización de la interfaz gráfica y su integración con las funciones del servidor.

2.6.1.5 IDE de programación Android Studio y Eclipse

Se decide utilizar el entorno de desarrollo de Android Studio, para la aplicación móvil.

Las justificaciones son las siguientes:

- **IDE específico de Android.** Es un IDE desarrollado por Google para el desarrollo específico de aplicaciones Android. Sus herramientas, librerías y emuladores son bastante completos y son de gran utilidad para el desarrollo.
- **IDE versátil para Java.** Eclipse es un IDE que principalmente se usa para el desarrollo en Java, y mediante el uso de complementos permite realizar muchas funciones que facilitan el trabajo. Por ejemplo, podemos integrar el desarrollo de la interfaz gracias al complemento de JavaFX y de **SceneBuilder** [27], un programa para el diseño de interfaces de manera visual.

2.6.2 Alternativas de implementación

Con las soluciones elegidas en el apartado anterior, se contemplan varias formas de llevar a cabo la implementación física del sistema. Se valorarán las opciones y se elegirá una justificando su elección.

2.6.2.1 Red local mediante router inalámbrico

En este escenario el profesor crea una red local mediante el uso de un router inalámbrico conectado al PC del profesor, al que se conectarán los alumnos para establecer la comunicación con el servidor. El sistema implica el que el profesor disponga de un router que tiene que conectar al PC para poder crear el entorno de red.

La ventaja de este sistema es que la red está a total disposición del profesor y puede configurarla a su gusto mediante la configuración interna del router.



Figura 15 - Red local con router inalámbrico

2.6.2.2 Red local a través de router inalámbrico portátil

Este caso es parecido al escenario anterior, pero usando un adaptador WiFi ya sea por USB o una tarjeta PCI para el ordenador.

Por comodidad lo más conveniente sería usar un adaptador USB, pues es menos engorroso de transportar.

La desventaja como con el router inalámbrico radica en tener que configurar el dispositivo y tener que llevarlo a todas partes. La principal ventaja es el control de la red creada sin restricciones.



Figura 16 - Red local con router USB portátil

2.6.2.3 Red interna de la Universidad

En este escenario, se usa la red interna de la universidad. Los ordenadores de las aulas pueden ser usados, al estar conectados mediante cable, o se puede usar un portátil personal conectándolo de forma inalámbrica a la red. El alumno puede del mismo modo conectarse a la red mediante WiFi. La principal ventaja es la comodidad de simplemente ejecutar el programa en el PC de las aulas o en un portátil.

La posible desventaja observada es que algunas aulas cuentan con una configuración de red muy estricta y no se pueda realizar la conexión. En este caso la única opción viable sería usar un ordenador portátil, aunque esto implicaría que cualquiera que conozca los datos del servidor pueda conectarse sin necesidad de estar presente en clase.



Figura 17 - Red local por red interna uc3m

2.6.2.4 Alternativa preferida

Tras hablarlo en una reunión con el cliente se decide dar preferencia al sistema que se vale de la red interna de la universidad, por la comodidad de su implementación y de su puesta en marcha.

Las demás alternativas cuentan con algún inconveniente añadido como por ejemplo el tener que desplazarse con un router y tener que montar la red, perdiendo algo de tiempo de clase, cosa que se quiere reducir al mínimo.

A pesar de estos inconvenientes, son alternativas viables que pueden servir como plan de reserva en caso de que la solución preferida falle o surja algún imprevisto, como por ejemplo la caída de la red de la universidad.

2.7 Alternativas existentes

Aquí se estudian las alternativas existentes en el mercado que satisfagan la necesidad que busca cubrir este proyecto. Nos encontramos con dos alternativas, **Socrative** y **Blackboard Learn**.

2.7.1 Socrative

Socrative [28] es una herramienta educativa gratuita para comprobar el conocimiento de los alumnos mediante test rápidos en tiempo real. Es necesario registrarse para poder hacer uso de sus funciones. La plataforma consta de dos aplicaciones, una para el profesor y otra para los alumnos. Ambas aplicaciones son compatibles con los sistemas operativos móviles Android, iOS y Windows Phone. Es posible incluso usar la plataforma a través del portal web Socrative.

Su funcionamiento es sencillo. El profesor puede crear test de diferentes tipos y guardarlos. Cuando quiera iniciar un test simplemente debe crear una sala definida por una clave y esperar a que los alumnos se conecten a la sala antes de iniciar el test.

Los alumnos simplemente se conectan a la sala creada por el profesor y esperan a que inicie el test. Una vez hayan respondido todas las preguntas finalizan el test.

El profesor puede observar en el apartado de resultados el desempeño de los estudiantes en tiempo real. Cuando termina el test, puede exportar las notas de los alumnos en diferentes formatos o enviarlas a un correo electrónico.

Tras analizar su funcionamiento, se concluye que esta plataforma cubre prácticamente todas las necesidades del problema a tratar.

Sin embargo, dado que es una plataforma externa a la universidad, no da garantías de que el alumno esté en clase para la realización del test. Si conoce el código de sala, el alumno puede conectarse desde cualquier lugar y realizar el test. Otro inconveniente es que no es compatible nativamente con los test exportados de la plataforma Moodle.

2.7.2 Blackboard Learn

Blackboard Learn [29] es una plataforma de aprendizaje online a disposición de alumnos y profesores. A través de ella pueden compartir recursos como exámenes, documentación, notas o realizar anuncios y apuntes. Varias universidades del mundo han adoptado esta plataforma [http://www.wilmu.edu/onlinelearning/blackboard_mobile.aspx]

Dentro del ecosistema Blackboard existe la aplicación **Blackboard Mobile Learn**. Esta aplicación se utiliza para realizar test entre otras funciones. Los profesores crean los test a través de la plataforma Blackboard Learn y los comparten. Los alumnos pueden descargárselos y realizarlos en cualquier momento.

Esta es la principal diferencia con el objetivo de este proyecto. Los test creados en Blackboard Learn no son de carácter presencial.

3. Análisis

En este apartado se tratarán aquellos datos relacionados con el análisis del sistema a desarrollar. Se determinarán principalmente los casos de uso y los requisitos de la aplicación, clasificándolos según su tipo.

3.1 Diagrama Casos de Uso

En esta sección veremos los posibles casos de uso que se pueden dar en la aplicación. Al contar con un único tipo de usuario, se cuenta con dos actores, que serán:

- **Alumno:** Es el usuario de la aplicación móvil.
- **Profesor:** Es la persona que gestiona los test desde el programa de escritorio.

Los casos de uso de cada actor se representan en los siguientes diagramas de caso de uso:

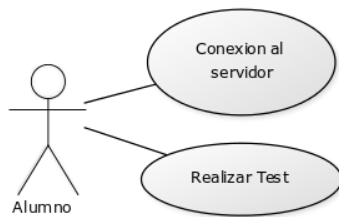


Figura 18 - Casos de uso Alumno

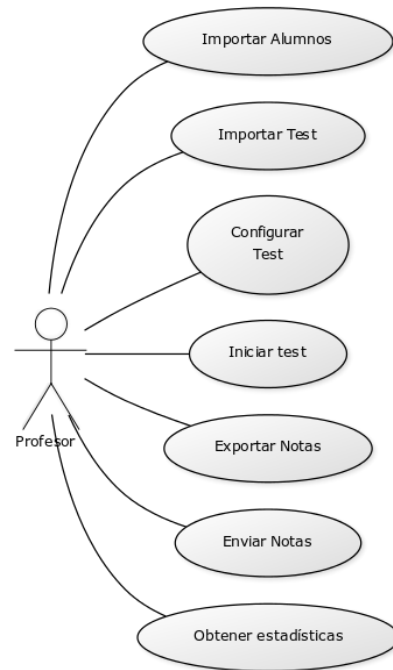


Figura 19 - Casos de Uso Profesor

3.2 Especificación de los casos de uso

Se detallan en esta sección los casos de uso presentados anteriormente de manera más detallada y específica.

La estructura de presentación es la siguiente:

Nombre	
ID	CU - XXX
Actores	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario básico	

Tabla 1 - Ejemplo tabla casos de uso

- **Nombre:** nombre del caso de uso.
- **ID:** identificador único para cada caso de uso con el formato CU - XXX, donde XXX es un número de tres cifras.
- **Actores:** los actores a los que pertenece el caso de uso.
- **Objetivo:** la finalidad del caso de uso.
- **Precondiciones:** condiciones anteriores necesarias.
- **Postcondiciones:** condiciones posteriores necesarias.
- **Escenario básico:** acciones que se realizan en el caso de uso.

Se definen los siguientes casos de uso:

Nombre	Conexión/Identificación
ID	CU - 001
Actores	Alumno
Objetivo	Establecer la conexión con el servidor mediante la identificación del alumno.
Precondiciones	<ul style="list-style-type: none"> ● Alumno perteneciente a la clase. Figura en la lista de alumnos importada por el profesor. ● Alumno posee un numero identificativo llamado NIA. ● Es la primera vez que accede a la aplicación o se ha cerrado la sesión.
Postcondiciones	El alumno accede a la aplicación y puede empezar a hacer uso de la misma.
Escenario básico	<ol style="list-style-type: none"> 1. Acceder a la aplicación móvil. 2. Introducir los datos requeridos para la identificación. 3. La aplicación avanza y muestra la siguiente pantalla de bienvenida.

Tabla 2 - Caso de Uso 1

Nombre	Realizar Test
ID	CU - 002
Actores	Alumno
Objetivo	El alumno podrá navegar por el test y responderlo de manera satisfactoria.
Precondiciones	<ul style="list-style-type: none"> • Alumno identificado en la aplicación. • Que el profesor configure y lance un test desde el servidor.
Postcondiciones	El alumno puede ver las preguntas del test y responder de acuerdo a las restricciones aplicadas al test.
Escenario básico	<ol style="list-style-type: none"> 1. Acceder a la aplicación móvil. 2. Identificarse en la aplicación. 3. Visualizar las preguntas y tener la capacidad de responderlas.

Tabla 3 – Caso de Uso 2

Nombre	Importar Alumnos
ID	CU - 003
Actores	Profesor
Objetivo	El profesor importará una lista de alumnos con éxito visualizándolos en la lista de alumnos desconectados.
Precondiciones	<ul style="list-style-type: none"> • Lista con alumnos en formato compatible con el servidor. • Ejecutar el programa de escritorio.
Postcondiciones	La lista de alumnos es importada y se visualiza en la interfaz del servidor.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio. 2. Seleccionar la opción “Importar Alumnos” 3. Seleccionar el fichero de alumnos correspondiente. 4. Hacer clic en el botón “Abrir” 5. Ver la lista de alumnos en la interfaz del programa.

Tabla 4 - Caso de Uso 3

Nombre	Importar test
ID	CU - 004
Actores	Profesor
Objetivo	El profesor podrá importar un test con éxito visualizándolo en la lista de test de la interfaz del programa.
Precondiciones	<ul style="list-style-type: none"> • Contar con un test en formato compatible con el programa. • Ejecutar el programa de escritorio.
Postcondiciones	El test se importa y se visualiza en la lista de test disponibles.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio 2. Seleccionar la opción “Importar Test” 3. Seleccionar el fichero de test correspondiente. 4. Hacer clic en el botón “Abrir” 5. Ver el test importado en la lista de test disponibles.

Tabla 5 - Caso de Uso 4

Nombre	Configurar test
ID	CU - 005
Actores	Profesor
Objetivo	El profesor podrá configurar el test seleccionado con las diferentes opciones disponibles.
Precondiciones	<ul style="list-style-type: none"> • Ejecutar el programa de escritorio. • Tener un test previamente importado en el programa. • Seleccionar un test de la lista.
Postcondiciones	El test queda configurado y listo para poder ser iniciado.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio. 2. Importar un test o seleccionar uno ya importado. 3. Seleccionar la opción "Configurar" 4. Seleccionar las opciones que se quieran configurar. 5. Hacer clic en "Aceptar" para continuar. 6. El test se muestra como configurado en la interfaz.

Tabla 6 - Caso de Uso 5

Nombre	Iniciar test
ID	CU - 006
Actores	Profesor
Objetivo	El profesor puede iniciar un test, previamente configurado, cuando lo decida.
Precondiciones	<ul style="list-style-type: none"> • Ejecutar el programa de escritorio. • Tener un test configurado en el programa. • Seleccionar un test de la lista.
Postcondiciones	El test se inicia y el alumno puede acceder a él a través de la aplicación móvil. El profesor podrá ver las preguntas del test si así está definido.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio 2. Tener un test configurado, o configurar uno en caso contrario. 3. Seleccionar el test. 4. Seleccionar la opción "Iniciar Test". 5. El test se envía a los alumnos y estos pueden visualizarlo y responder.

Tabla 7 - Caso de Uso 6

Nombre	Exportar notas
ID	CU - 007
Actores	Profesor
Objetivo	Exportar las notas de los alumnos tras realizar un test en un archivo.
Precondiciones	<ul style="list-style-type: none"> • Ejecutar el programa de escritorio. • Iniciar un test. • El alumno o alumnos responden a las preguntas del test. • El profesor da por finalizado el test.
Postcondiciones	El programa genera un archivo con las notas de cada alumno.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio. 2. Iniciar un test. 3. Esperar a que finalicen los alumnos y finalizar el test. 4. Seleccionar la opción “Notas” y dentro del menú, “Exportar”. 5. Seleccionar la localización donde guardar el archivo y cambiar el nombre si se desea. 6. El archivo se crea en la localización seleccionada y se puede abrir y leer sin problemas.

Tabla 8 - Caso de Uso 7

Nombre	Enviar notas
ID	CU - 008
Actores	Profesor
Objetivo	Enviar el archivo de notas de los alumnos por correo electrónico, a la dirección que se especifique.
Precondiciones	<ul style="list-style-type: none"> • Ejecutar el programa de escritorio. • Iniciar un test. • El alumno o alumnos responden a las preguntas del test. • El profesor da por finalizado el test.
Postcondiciones	El archivo de notas se recibe en la cuenta de correo elegida con éxito.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio. 2. Iniciar un test. 3. Esperar a que finalicen los alumnos y finalizar el test. 4. Seleccionar dentro del menú “Notas”, “Enviar por Correo” 5. Rellenar el formulario de credenciales para el envío. 6. Hacer clic en “Enviar”. 7. Las notas se envían al correo configurado.

Tabla 9 - Caso de Uso 8

Nombre	Estadísticas
ID	CU - 009
Actores	Profesor
Objetivo	Obtener estadísticas de notas de los alumnos relacionadas con el test que se ha realizado.
Precondiciones	<ul style="list-style-type: none"> • Ejecutar el programa de escritorio. • Iniciar un test. • El alumno o alumnos realizan el test. • El profesor da por finalizado el test.
Postcondiciones	Se muestra en la interfaz un gráfico con la estadística de notas de los alumnos.
Escenario básico	<ol style="list-style-type: none"> 1. Ejecutar el programa de escritorio. 2. Iniciar un test 3. Esperar a que finalicen los alumnos y finalizarlo. 4. Entrar en “Ver Gráfico” dentro del menú “Estadísticas”. 5. Se muestra el grafico estadístico de notas obtenidas.

Tabla 10 - Caso de Uso 9

3.3 Requisitos software

Basados en los casos de uso definidos, se detallan los requisitos de software que nos ayudan a entender y desglosar a más bajo nivel las funcionalidades del sistema.

Los requisitos se describirán con el siguiente formato:

ID	RY - XXX
Título	Título.
Descripción	Descripción.
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 11 - Ejemplo de requisito

- **ID:** identificador del requisito con el formato RSY - XXX donde XXX es un número de tres cifras y donde Y puede tener los valores F (de Funcional) o NF (de no funcional).
- **Título:** nombre corto y descriptivo del requisito.
- **Descripción:** descripción larga del requisito.
- **Prioridad:** ordena los requisitos según la necesidad de ser desarrollados antes que otros requisitos o módulos del sistema.
 - Alta: requisito cuya prioridad es máxima.
 - Media: requisito de prioridad media.
 - Baja: requisitos de menor prioridad.

- **Estabilidad:** establece si el requisito está sujeto o no a modificaciones.
 - Estable: baja probabilidad de ser alterado.
 - No estable: alta probabilidad de ser alterado.
- **Necesidad:** indica la importancia de que un requisito se cumpla.
 - Esencial: es obligatorio su cumplimiento.
 - Deseable: es recomendable su cumplimiento.
 - Opcional: no es de obligado cumplimiento.

3.2.1 Requisitos funcionales

Se presentan a continuación los requisitos software funcionales del sistema. Se dividen en requisitos funcionales de servidor y de aplicación móvil.

3.2.1.1 Requisitos funcionales del servidor

ID	RSF - 001
Título	Conexión
Descripción	El servidor podrá atender peticiones de conexión por parte de los alumnos, y los alumnos podrán conectarse a la aplicación de manera eficiente mediante el uso de un protocolo definido para ello.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 12 - Requisito Funcional 1

ID	RSF - 002
Título	Importar test
Descripción	Mediante el botón de "Importar Test" se podrá importar un test o una serie de test existentes al programa mediante un intérprete de ficheros XHTML. El programa de escritorio podrá importar test guardados en un directorio específico de forma automática al iniciarse.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 13 - Requisito Funcional 2

ID	RSF - 003
Título	Obtener listado de alumnos del curso
Descripción	El programa de escritorio permitirá al profesor importar un listado de alumnos asociados a la clase y turno deseados para facilitar el control de los alumnos que puedan realizar el test.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 14 - Requisito Funcional 3

ID	RSF - 004
Título	Envío de correo electrónico
Descripción	Los resultados de los test se enviarán por correo electrónico al profesor al finalizar la prueba. El correo lo determinará el profesor de forma manual.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 15 - Requisito Funcional 4

ID	RSF - 005
Título	Estadísticas
Descripción	El programa de escritorio podrá recopilar los datos de los test realizados para la realización de estadísticas definidas por el profesor.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 16 - Requisito Funcional 5

ID	RSF - 006
Título	Exportar notas
Descripción	El programa de escritorio podrá exportar las notas de los alumnos en un archivo.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 17 - Requisito Funcional 6

ID	RSF - 007
Título	Configurar test
Descripción	El profesor podrá configurar el test mediante la herramienta del menú "Configurar".
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 18 - Requisito Funcional 7

ID	RSF - 008
Título	Inicio test
Descripción	El profesor podrá dar inicio al test mediante el uso del botón "Iniciar test". El profesor podrá iniciar el test en el momento que crea conveniente.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 19 - Requisito Funcional 8

ID	RSF - 009
Título	Configuración de test
Descripción	<p>Al crear el test el profesor podrá configurar el test de acuerdo a las opciones que le brinda el programa de escritorio.</p> <p>Los tipos de respuestas de los test que se podrán definir al crear el test serán los siguientes:</p> <ul style="list-style-type: none"> • Respuesta única • Respuesta múltiple <p>Dentro de esta variedad de respuestas, el profesor podrá determinar también el número de opciones que quiera dar al alumno.</p> <p>Así mismo el profesor podrá determinar si:</p> <ul style="list-style-type: none"> • El test contará con una contraseña para su acceso. • El alumno puede ver la pregunta. • El alumno no puede ver la pregunta. • Las preguntas tendrán un tiempo determinado para su contestación. • Se permite modificar la respuesta. • Si las preguntas se presentarán en un orden aleatorio u ordenadas. • La penalización por pregunta fallada.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 20 - Requisito Funcional 9

ID	RSF - 010
Título	Contraseña test
Descripción	Los test podrán tener una contraseña para el acceso a los mismos, de manera opcional.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 21 - Requisito Funcional 10

ID	RSF - 011
Título	Visibilidad de las preguntas
Descripción	El profesor podrá decidir si solo el alumno podrá ver el enunciado en su móvil.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 22 - Requisito Funcional 11

ID	RSF - 012
Título	Tiempo test
Descripción	El profesor podrá elegir si establece un tiempo determinado para la realización de cada pregunta del test.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 23 - Requisito Funcional 12

ID	RSF - 013
Título	Interrumpir test
Descripción	El profesor podrá finalizar el test en cualquier momento que crea oportuno.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 24 - Requisito Funcional 13

ID	RSF - 014
Título	Diseño de interfaz de servidor
Descripción	La interfaz del servidor tendrá un diseño sencillo y efectivo, con una gran usabilidad para el profesor.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 25 - Requisito Funcional 14

3.2.1.2 Requisitos funcionales de la aplicación móvil

ID	RSF - 015
Título	Identificación en la aplicación
Descripción	El alumno debe identificarse para acceder a las funciones de la aplicación.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 26 - Requisito Funcional 15

ID	RSF - 016
Título	Obtener test
Descripción	La aplicación móvil recibirá las preguntas del test cuando el profesor inicie el test.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 27 - Requisito Funcional 16

ID	RSF - 017
Título	Ver test
Descripción	El alumno podrá ver el test a realizar en clase en la aplicación.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 28 - Requisito Funcional 17

ID	RSF - 018
Título	Acceder a test
Descripción	El alumno podrá acceder al test de manera automática si el profesor da inicio al mismo sin restricciones. Si estuviese protegido por contraseña, el alumno deberá introducirla antes de poder acceder.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 29 - Requisito Funcional 18

ID	RSF - 019
Título	Visualización de las preguntas
Descripción	El alumno podrá visualizar las preguntas del test siempre.
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 30 - Requisito Funcional 19

ID	RSF - 020
Título	Espacio de respuestas
Descripción	El alumno podrá visualizar el espacio reservado las posibles soluciones de las preguntas.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 31 - Requisito Funcional 20

ID	RSF - 021
Título	Enviar respuesta
Descripción	El alumno podrá enviar la respuesta elegida al servidor mediante el uso de un botón táctil para su envío.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 32 - Requisito Funcional 21

ID	RSF - 022
Título	Control de respuesta
Descripción	La aplicación controlará a través del código el tipo de respuesta que podrá dar el alumno.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 33 - Requisito Funcional 22

ID	RSF - 023
Título	Opciones de respuesta
Descripción	La aplicación podrá identificar todas las posibles combinaciones de respuestas posibles, de acuerdo a la configuración que haya montado el profesor.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 34 - Requisito Funcional 23

ID	RSF - 024
Título	Marcar respuesta
Descripción	En respuestas de seleccionar una o más opciones, el alumno podrá seleccionar la que desee mediante la marcación de la casilla correspondiente.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 35 - Requisito Funcional 24

ID	RSF - 025
Título	Seguridad en la comunicación
Descripción	La comunicación entre el dispositivo móvil y el servidor deberá realizarse de manera segura, a través de un protocolo seguro.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 36 - Requisito Funcional 25

ID	RSF - 026
Título	Notificaciones
Descripción	La aplicación móvil dispondrá de un sistema de notificaciones que avisará al alumno de cualquier evento que se produzca y sea relevante.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 37 - Requisito Funcional 26

ID	RSF - 027
Título	Diseño de aplicación
Descripción	La aplicación contará con elementos de diseño para facilitar la usabilidad y manejo de la misma.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 38 - Requisito Funcional 27

3.2.2 Requisitos no funcionales

Se presentan a continuación los requisitos software de interfaz del sistema. Se dividen en requisitos del servidor y requisitos de la aplicación móvil:

3.2.2.1 Requisitos no funcionales del servidor

ID	RSNF - 001
Título	Normas de estilo, diseño y programación Java.
Descripción	Se respetaran las normas de estilo establecidas por la documentación oficial de Java.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 39 - Requisito No Funcional 1

ID	RSNF - 002
Título	Portabilidad del programa de escritorio
Descripción	El programa de escritorio será compatible con los sistemas operativos Windows, Linux y Mac.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 40 - Requisito No Funcional 2

ID	RSNF - 003
Título	Disponibilidad de aplicación
Descripción	El sistema en conjunto solo estará disponible en clase, siendo el profesor el encargado de decidir cuándo se establecerá la comunicación.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 41 - Requisito No Funcional 3

ID	RSNF - 004
Título	Conexión local
Descripción	La aplicación se conectará de manera local al servidor mediante el uso de un puerto específico del ordenador en el que esté instalado.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 42 - Requisito No Funcional 4

ID	RSNF - 005
Título	Conexión mediante socket
Descripción	La comunicación mediante la aplicación y el servidor se realizara a través de un socket TCP/IP.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 43 - Requisito No Funcional 5

ID	RSNF - 006
Título	Interfaz del programa de escritorio
Descripción	<p>La interfaz del programa constará de las siguientes secciones destacadas:</p> <ul style="list-style-type: none"> • Alumnos y su estado de conexión. • Test importados. <p>Contará también con los siguientes elementos secundarios:</p> <ul style="list-style-type: none"> • Botón de importar test • Botón de importar alumnos • Botón para configurar un test • Botón para iniciar un test • Botón para acceder al envío de notas o exportar notas • Botón para acceder a las estadísticas.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 44 - Requisito No Funcional 6

ID	RSNF - 007
Título	Desconexión
Descripción	<p>La desconexión entre la aplicación y el servidor podrá producirse de la siguiente manera:</p> <ul style="list-style-type: none"> • Una vez finalizado el test el programa se encarga de finalizar la conexión. • El profesor podrá cortar la comunicación cerrando el test. • Se produce una desconexión por parte de una de las dos partes implicadas, ya sea el alumno cerrando la aplicación o desactivando el WiFi del dispositivo, o el profesor cerrando el servidor.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 45 - Requisito No Funcional 7

3.2.2.2 Requisitos no funcionales de la aplicación móvil

ID	RSNF - 008
Título	Vista de identificación
Descripción	Vista principal la primera vez que se arranque la aplicación. Contendrá los siguientes objetos: <ul style="list-style-type: none"> • Campo de texto para introducir el NIA. • Campo de texto para introducir la IP del servidor. • Campo de texto para introducir el puerto del servidor. • Botón para establecer la conexión en base a los campos antes mencionados.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 46 - Requisito No Funcional 8

ID	RSNF - 009
Título	Vista de test
Descripción	Vista con el nombre del test a realizar.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 47 - Requisito No Funcional 9

ID	RSNF - 010
Título	Vista de pregunta
Descripción	Vista de la pregunta del test, en caso de ser posible.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 48 - Requisito No Funcional 10

ID	RSNF - 011
Título	Compatibilidad Android
Descripción	Compatibilidad con sistemas Android desde la versión 2.3.3 en adelante.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 49 - Requisito No Funcional 11

ID	RSNF - 012
Título	Normas de estilo, diseño y programación Android
Descripción	Se respetarán las normas de estilo establecidas por la documentación oficial de Android.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 50 - Requisito No Funcional 12

ID	RSNF - 013
Título	Resolución de pantalla
Descripción	La aplicación tendrá en cuenta las resoluciones más usadas en dispositivos Android para adaptarse a ellas. Tamaños de pantalla 3,5"; 4,5" y 5".
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 51 - Requisito No Funcional 13

ID	RSNF - 014
Título	Permisos de la aplicación
Descripción	La aplicación solo requerirá de los permisos necesarios e indispensables para su funcionamiento sin extralimitarse.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 52 - Requisito No Funcional 14

ID	RSNF - 015
Título	Vista de tiempo
Descripción	Se visualizará un contador con el tiempo restante para cada pregunta en caso de que así lo haya determinado el profesor.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 53 - Requisito No Funcional 15

ID	RSNF - 016
Título	Usabilidad de inicio de test
Descripción	El programa tendrá una usabilidad buena a través de su interfaz para la gestión del test.
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 54 - Requisito No Funcional 16

ID	RSNF - 017
Título	Vista de finalización
Descripción	La aplicación mostrara una pantalla cuando se finalice el test para informar al usuario que ha terminado con éxito.
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input checked="" type="checkbox"/> No estable
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional

Tabla 55 - Requisito No Funcional 17

ID	RSNF - 018
Título	Documentación
Descripción	La documentación del proyecto seguirá el estándar de métrica V3 [30]
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> No estable
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Tabla 56 - Requisito No Funcional 18

3.2.3 Matriz de trazabilidad

Se presenta la matriz de trazabilidad entre los casos de uso y los requisitos funcionales:

	CU - 001	CU - 002	CU - 003	CU - 004	CU - 005	CU - 006	CU - 007	CU - 008	CU - 009
RSF - 001	X								
RSF - 002				X					
RSF - 003			X						
RSF - 004								X	
RSF - 005									X
RSF - 006							X		
RSF - 007					X				
RSF - 008						X			
RSF - 009					X				
RSF - 010					X				
RSF - 011					X				
RSF - 012					X				
RSF - 013			X						
RSF - 014						X			
RSF - 015	X								
RSF - 016		X							
RSF - 017		X							
RSF - 018		X							
RSF - 019		X							
RSF - 020		X							
RSF - 021		X							
RSF - 022		X							
RSF - 023		X							
RSF - 024		X							
RSF - 025		X							
RSF - 026		X							
RSF - 027		X							

Tabla 57 - Matriz de trazabilidad requisitos

4. Diseño

Tras realizar el análisis de requisitos del sistema, se procede al diseño del mismo teniendo en cuenta lo estudiado anteriormente. Esta sección incluye una visión general del modelo a seguir para la estructuración de la aplicación, así como una vista por componentes del sistema y su funcionamiento. Se presentan los prototipos preliminares de la interfaz a tener en cuenta para la implementación posterior.

4.1 Definición de la arquitectura del sistema

Se presenta la arquitectura en detalle del sistema para su posterior análisis de diseño.

La arquitectura elegida es la de cliente-servidor. El servidor es el programa que se ejecuta en el PC o portátil del profesor, conectado a la red de la Universidad Carlos III de Madrid.

El profesor importa los test y los configura a su gusto antes de iniciarlos. Los alumnos se conectarán al servidor a través de la identificación pertinente, y esperarán a que el profesor inicie el test para poder responderlo.

Se puede ver la relación en la siguiente figura:

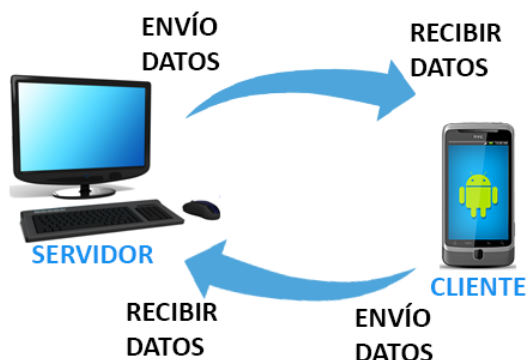


Figura 20 - Esquema de arquitectura del sistema

Identificamos dos sistemas principales, que son el **servidor** y el **cliente**. Como se observa en la imagen, su comunicación es bilateral enviando y recibiendo datos según sea oportuno.

En la realización del programa del servidor, este proyecto abarca el desarrollo de la interfaz y algunas funcionalidades de la misma, la parte referente a comunicaciones se ha realizado en otro proyecto.

El protocolo de comunicación y funcionamiento en sí de la conexión y la interpretación de los test se ha reutilizado de la implementación realizada por otro alumno, por lo que no se detallarán en este documento su funcionamiento o clases.

4.2 Determinación de subsistemas de análisis

Se trata de descomponer el sistema de información en subsistemas más pequeños que faciliten el proceso de diseño.

En esta tarea se determinarán los diferentes subsistemas por los que está formado cada sistema. Primero se mostrará un diagrama de interacción entre los distintos subsistemas identificados y después se procederá a analizarlos.

4.2.1 Interfaz gráfica servidor

Se muestra el esquema del subsistema de la interfaz gráfica:

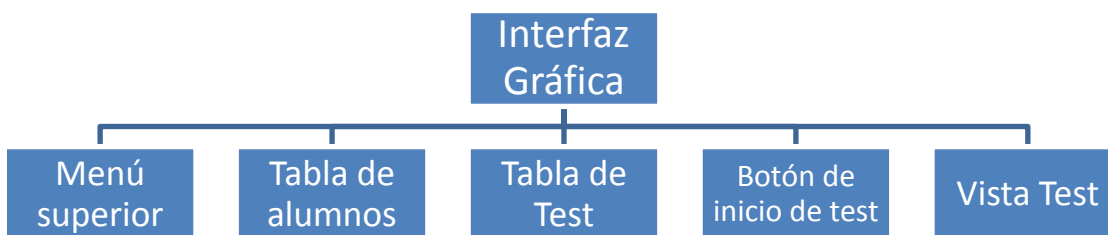


Figura 21 - Esquema de subsistemas de interfaz del servidor

El subsistema de **interfaz gráfica** se encargará de la representación gráfica de la interfaz como su nombre indica, englobando todas las funciones relacionadas con la misma.

Módulo	Menú superior
Propósito	Ofrecer al usuario un menú con las principales funciones del programa.
Funciones	<ul style="list-style-type: none"> • Opción de importar alumnos o test. • Gestión de importación de ficheros de test o alumnos. • Opción de configurar el test elegido. • Gestión de configuración del test seleccionado. • Opción de exportar las notas o de enviarlas por correo. • Gestión de exportación y envío de notas. • Opción de ver las estadísticas del ultimo test realizado. • Gestión de representación de estadísticas de notas. • Opción de Ayuda/Información acerca del programa. • Gestión de la ayuda o información del programa.

Tabla 58 - Tabla descripción subsistema Menú Superior

Módulo	Tabla de alumnos
Propósito	Ofrecer al usuario una fuente de información gráfica acerca de los alumnos conectados y no conectados en tiempo real.
Funciones	<ul style="list-style-type: none"> • Representar a los alumnos conectados y no conectados a través de su NIA. • Gestionar las tablas en tiempo real a través de listas de alumnos.

Tabla 59 - Tabla descripción subsistema Tabla de Alumnos

Módulo	Tabla de Test
Propósito	Ofrecer al usuario una fuente de información gráfica acerca de los test que se han importado a la aplicación.
Funciones	<ul style="list-style-type: none"> • Representar los test por su nombre en una lista, con la posibilidad de seleccionarlos.

Tabla 60 - Tabla descripción subsistema Tabla de Test

Módulo	Botón de Inicio de Test
Propósito	Ofrecer al usuario un botón para iniciar el test.
Funciones	<ul style="list-style-type: none"> • Iniciar el test si ya está configurado, o configurarlo si no lo estaba.

Tabla 61 - Tabla descripción subsistema Botón de Inicio de Test

Módulo	Vista de Test
Propósito	Ofrecer al usuario una fuente de información gráfica acerca del test que se está ejecutando en ese momento.
Funciones	<ul style="list-style-type: none"> • Representar las preguntas del test de forma visual para poder avanzar con las preguntas. • Interpretar el test para su visualización. • Representar en una tabla los alumnos que aun no han contestado a las preguntas o pregunta. • Gestionar los alumnos que no han respondido mediante el uso de una lista.

Tabla 62 - Tabla descripción subsistema Vista de Test

4.2.2 Actividades aplicación móvil

Los subsistemas de la aplicación móvil se pueden dividir en actividades:

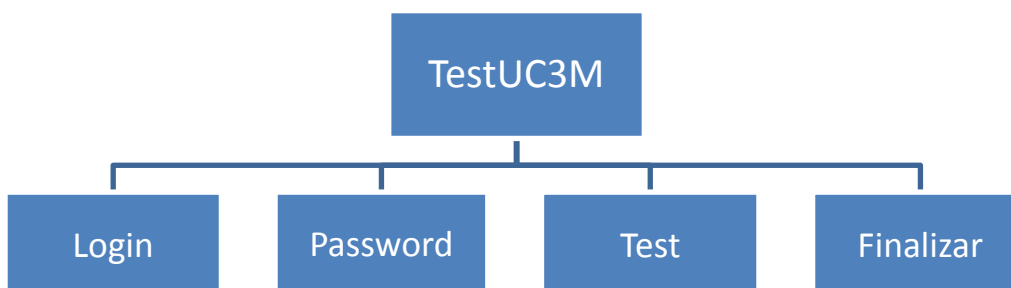


Figura 22 - Subsistemas de actividades TestUC3M

Las principales características de los subsistemas son las siguientes:

Módulo	Login
Propósito	Ofrecer al usuario una interfaz a través de la cual se identificará con el servidor y establecerá la conexión.
Funciones	<ul style="list-style-type: none"> • Proporcionar interfaz de identificación. • Establecer la conexión con el servidor mediante la verificación de los datos introducidos. • Encargarse de las posibles excepciones producidas por algún error de conexión o de identificación.

Tabla 63 - Tabla descripción subsistema Login

Módulo	Password
Propósito	Ofrecer al usuario una interfaz a través de la cual pueda introducir la contraseña del test y acceder al mismo.
Funciones	<ul style="list-style-type: none"> • Proporcionar interfaz para la introducción de la contraseña • Verificar si la contraseña introducida es correcta. • Acceder al test seleccionado si la contraseña es correcta.

Tabla 64 - Tabla descripción subsistema Password

Módulo	Test
Propósito	Ofrecer al usuario una interfaz a través de la cual pueda visualizar, navegar y realizar el test.
Funciones	<ul style="list-style-type: none"> • Proporcionar una interfaz para la visualización y realización del test, independientemente del tipo que sea. • Recibir e interpretar el test para su visualización. • Navegar entre preguntas si es posible. • Seleccionar opciones en cada pregunta, acorde al tipo de test. • Recolectar y enviar las respuestas de vuelta al servidor. • Finalizar el test.

Tabla 65 - Tabla descripción subsistema Test

Módulo	Finalizar
Propósito	Ofrecer al usuario una interfaz para finalizar la aplicación tras la realización de un test.
Funciones	<ul style="list-style-type: none"> • Proporcionar una interfaz para la finalización de la aplicación. • Terminar la conexión con el servidor. • Liberar recursos y terminar actividades abiertas.

Tabla 66 - Tabla descripción subsistema Finalizar

4.2 Diagrama de clases

En este apartado se presentan los diagramas de clases de programación de cada sistema.

4.2.1 Clases de la interfaz del servidor

Se presenta el diagrama general de clases en relación a la interfaz del servidor:

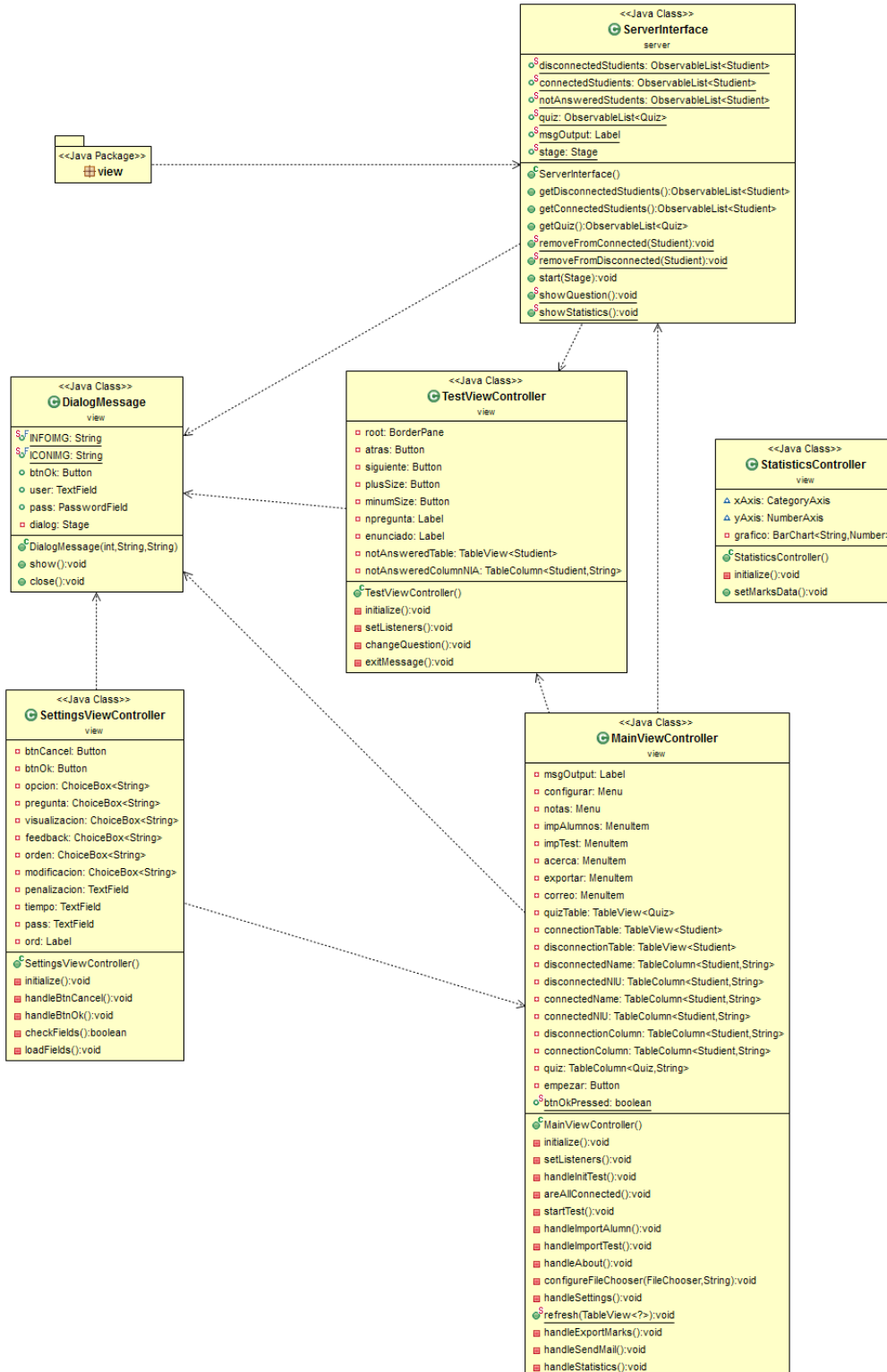


Figura 23 - Diagrama de Clases Interfaz del Servidor

Todas estas clases, excepto **ServerInterface**, se encuentran dentro del paquete **View**, pues son clases específicas de la interfaz. Se procede a analizar en detalle las clases.

4.2.1.1 ServerInterface

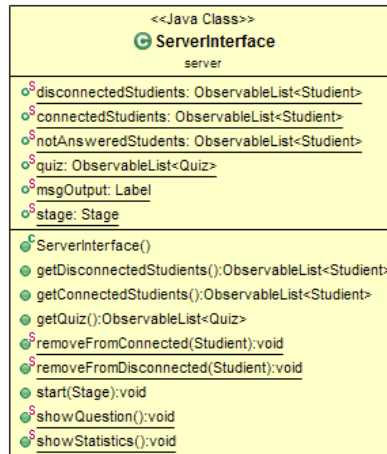


Figura 24 - Diagrama de clase ServerInterface

Esta clase se encarga de poner en marcha la interfaz. Sus atributos incluyen:

- **disconnectedStudents.** Es una lista de tipo ObservableList, que contiene objetos de tipo Student, en la que se almacenan los alumnos que no están conectados en ese momento.
- **connectedStudents.** Es una lista de tipo ObservableList, que contiene objetos de tipo Student, en la que se almacenan los alumnos que están conectados en ese momento.
- **notAnsweredStudent.** Es una lista de tipo ObservableList, que contiene objetos de tipo Student, en la que se almacenan los alumnos que aún no han contestado a una pregunta o al test por completo.
- **quiz.** Es una lista de tipo ObservableList, que contiene objetos de tipo Quiz, en la que se almacenan los test importados.
- **msgOutput.** Es una variable de tipo Label para la representación por pantalla de un mensaje. En este caso la IP del servidor y el puerto.
- **stage.** Variable de tipo Stage. Su función es la de crear un escenario para la interfaz.

Sus métodos son:

- **ServerInterface.** Se trata de un constructor para inicializar algunas variables como las listas de alumnos.
- **getDisconnectedStudents, getConnectedStudents, getQuiz.** Son métodos para la recuperación de las listas previamente vistas.

- **removeFromConnected, removeFromDisconnected.** Son métodos para la eliminación de un alumno en concreto de las listas de conectados y no conectados. El parámetro que reciben es el alumno a eliminar.
- **start.** Es el método que se encarga de iniciar la interfaz. Es el método principal de la clase, se la llama por primera vez cuando se ejecuta la aplicación.
- **showQuestion, showStatistics.** El primer método se encarga de mostrar la vista de test con las preguntas en una nueva ventana. La segunda se encarga de mostrar la pantalla de estadísticas en una ventana nueva.

4.2.1.2 MainViewController

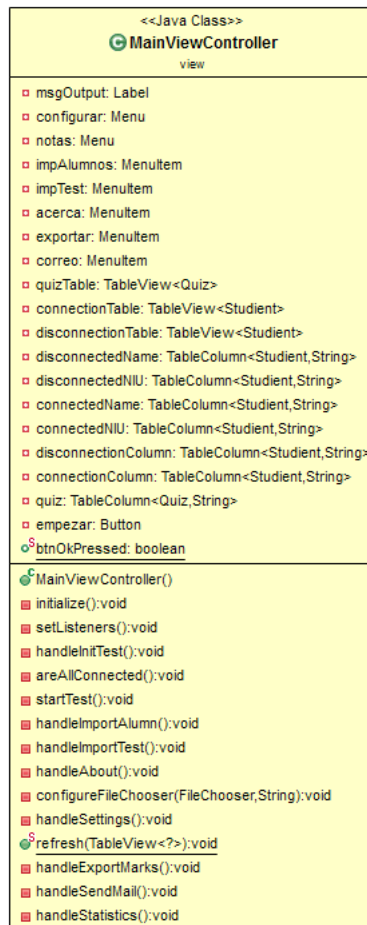


Figura 25 - Diagrama de clase MainViewController

Esta clase es una clase de tipo controlador para la vista principal de la interfaz. Se encarga de gestionar las posibles acciones que puede realizar el usuario desde la vista principal.

El atributo **msgOutput** es una etiqueta para mostrar información del servidor. Los objetos de tipo Menuitem **configurar**, **notas**, **impAlumnos**, **impTest**, **acerca**, **exportar** y **correo** son elementos del menú superior, cada uno encargado de una función asociada a un método que veremos más adelante.

Los atributos de tipo TableView **quizTable**, **connectionTable** y **disconnectionTable** son las tablas de test y de alumnos respectivamente. Sus columnas vienen identificadas por las variables **disconnectedName**, **disconnectedNIU**, **connectedName**, **connectedNIU**, **disconnectionColumn**, **connectionColumn** y **quiz**. El botón **Empezar** es el botón que se usa para iniciar un test.

Los métodos son los siguientes:

- **initialize**. Es el método inicial que se ejecuta cuando se abre la vista de la interfaz. Se usa para inicializar variables como los títulos de las tablas o los datos de las mismas.
- **setListeners**. Es el método encargado de implementar los métodos de escucha para las listas de alumnos y para los elementos del menú.
- **handleInitTest**. Este método se encarga de gestionar el inicio del test a través del botón **Empezar**, mediante comprobaciones de si está configurado o si se ha seleccionado un test de la lista.
- **areAllConnected**. Este método se ocupa de comprobar si todos los alumnos están conectados antes de iniciar el test, y de no ser así avisa al usuario mediante una ventana emergente.
- **startTest**. Este método se encarga de iniciar el test llamando al método `showQuestion`, además de dar la señal al servidor para que empiece con el envío del test al alumno.
- **handleImportAlumn**, **handleImportTest**, **configureFileChooser**. Los dos primeros como su nombre sugiere son los métodos encargados de la importación de alumnos y de test. El método **configureFileChooser** es un método auxiliar que se encarga de abrir una ventana de directorio para que el alumno pueda seleccionar el fichero de forma cómoda.
- **handleAbout**. Este método gestiona la vista del menú Ayuda, con información del programa.
- **handleSettings**. Este método gestiona la vista de configuración de un test. Se encarga de las comprobaciones pertinentes como tener un test un elegido de la lista y acto seguido abre una ventana nueva para configurar el test.
- **handleExportMarks**. Este método se encarga de exportar las notas de un test una vez terminado y guardarlo en un archivo en formato .xls o .txt.
- **handleSendMail**. Este método se encarga del envío de las notas de un test por correo electrónico.
- **handleStatistics**. Este método se encarga de iniciar la vista de estadísticas, siempre y cuando haya notas que se puedan usar.

4.2.1.3 SettingsViewController

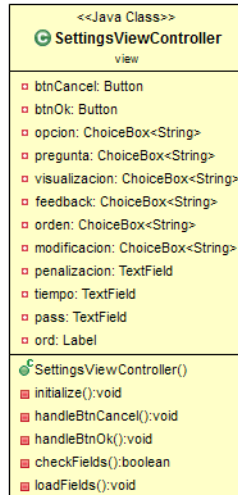


Figura 26 - Diagrama de clase `SettingsViewController`

Esta clase es una clase de tipo controlador para la vista de configuración de un test. Se encarga de gestionar las posibles configuraciones disponibles para un test.

Sus variables son partes de la interfaz, tiene botones **btnCancel** para cancelar la selección y salir de la ventana, **btnOk** para guardar los cambios y salir. Tiene varias opciones de tipo `ChoiceBox` para seleccionar como **opción**, **pregunta**, **visualización**, **feedback**, **orden** y **modificación**. Cuenta con campos de texto como **penalización**, **tiempo** y **pass**, para la penalización por pregunta, el tiempo por pregunta o la contraseña para el test respectivamente. Por último, **ord** es una etiqueta que se mostrará cuando sea posible para determinar el orden de las preguntas.

Los métodos son los siguientes:

- **initialize**. Este método se ejecuta cuando se inicia la vista de configuración. Se encarga de inicializar algunos valores. Si el test ya tiene una configuración anteriormente guardada, se cargan las opciones que ya tiene. Si no, se muestran las opciones por defecto.
- **loadFields**. Este método se encarga de recuperar la configuración que ya tenía un test para su representación en la interfaz.
- **handleBtnCancel**, **handleBtnOk**. Estos métodos se encargan de gestionar las acciones cuando se pulsa Cancelar o Aceptar. Si se cancela se vuelve a la interfaz principal cerrando la vista de configuración. Si se acepta, se llama al método para **checkFields** para configurar el protocolo del test.
- **checkFields**. Este método se encarga de recolectar las opciones que haya seleccionado el usuario y de guardarlas en el protocolo del test seleccionado.

4.2.1.4 TestViewController

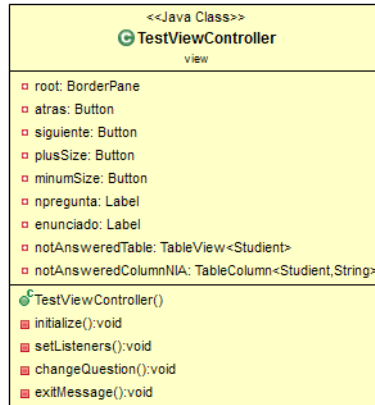


Figura 27 - Diagrama de clase TestViewController

Esta clase es una clase de tipo controlador para la vista de preguntas y navegación del test. Se encarga de mostrar las preguntas del test, de mostrar la tabla de alumnos que aún no han contestado, de avanzar a través de las preguntas y de terminar el test si así lo desea el usuario.

Los atributos **atrás**, **siguiente** son botones para la navegación por las preguntas. Los botones **plusSize** y **minumSize** son botones para aumentar o disminuir el tamaño del texto de la pregunta. La etiqueta **nPregunta** muestra el número de pregunta en la que estamos, la etiqueta **enunciado** muestra el enunciado de la pregunta. La tabla **notAnsweredTable** y su columna **notAnsweredColumnNIA** muestran la lista de alumnos que no han contestado.

Los métodos son los siguientes:

- **initialize.** Este método se ejecuta cuando se inicia la vista de test. Se encarga de inicializar los valores de la tabla de alumnos y de llamar a los métodos de gestión **setListeners** y **changeQuestion**.
- **setListeners.** Este método se encarga de gestionar las acciones cuando se seleccione algún elemento de la ventana, como los botones de navegación o los de ajuste de tamaño de texto.
- **changeQuestion.** Este método se encarga de gestionar la navegación por preguntas. Si se pulsa siguiente o atrás se encarga de cambiar el enunciado y el número de pregunta.
- **exitMessage.** Este método se encarga de realizar una comprobación antes de finalizar el test. Abre una ventana emergente preguntando al usuario si realmente quiere finalizar el test.

4.2.1.5 StatisticsController

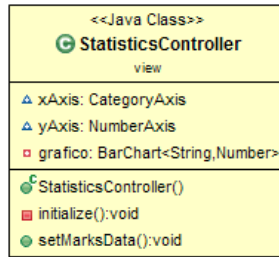


Figura 28 - Diagrama de clase StatisticsController

Esta clase es una clase de tipo controlador para la vista de estadísticas. Se encarga de mostrar en una ventana nueva el grafico con las notas del último test.

Su atributo principal es **gráfico**, una variable de tipo BarChart, es decir, un gráfico de barras. Está compuesto por las otras dos variables, **xAxis** e **yAxis**, que son los ejes del gráfico.

Sus métodos son:

- **initialize**. Es el método que se ejecuta al iniciar la vista de estadísticas. Se encarga de inicializar la gráfica con los datos pertinentes.
- **setMarksData**. Este método se encarga de asignar los datos de las notas en el gráfico.

4.2.2 Aplicación Android

Para la aplicación Android se detallan los diagramas de clase por separado, pues su jerarquía es independiente y se dividen en actividades. Cada actividad sirve una funcionalidad del sistema.

Como apunte, el protocolo de comunicación establece el siguiente patrón para la interpretación de las respuestas:

- Si la opción no ha sido marcada por el alumno, se representa con un 0.
- Si la opción u opciones se han marcado, se representa con un 1.

4.2.2.1 Singleton

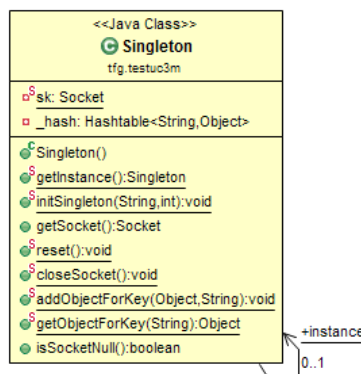


Figura 29 - Diagrama de clase Singleton

Singleton es una clase auxiliar para la gestión de la conexión en la aplicación.

Su uso principal es el de mantener la conexión abierta mientras se itera entre actividades. Se usa para la lectura y envío de datos. Su uso secundario es el de almacenar el test para poder pasarlo entre actividades.

Sus atributos son:

- **sk**. Es un atributo de tipo Socket. Sirve para establecer un canal de comunicación con el servidor.
- **_hash**. Es un atributo de tipo Hashtable. Su función es la de almacenar un objeto. Como la conexión, este atributo se usa para guardar y recoger el test leído por la aplicación.
- **instance**. Es un objeto de tipo Singleton que será el que perdure estático para su posterior uso.

Sus métodos son:

- **Singleton**. Es el constructor por defecto de la clase.
- **getInstance**. Este método devuelve la instancia de Singleton creada, o crea una nueva si no existe.
- **initSingleton**. Este método recibe como parámetros la IP y el puerto del servidor para realizar la conexión. Instancia el socket sk y establece la conexión.
- **getSocket**. Este método devuelve el socket sk.
- **reset**. Este método se encarga de limpiar todas las variables, eliminando su valor.
- **closeSocket**. Este método se encarga de cerrar el socket sk.
- **addObjectForKey**. Este método se encarga de almacenar un objeto en el atributo _hash e identificarlo con un nombre que se recibe por parámetro junto al objeto.
- **getObjectForKey**. Este método se encarga de recuperar el objeto dentro del atributo _hash a través de su nombre.
- **isSocketNull**. Este método comprueba si el socket sk no está instanciado.

4.2.2.2 LoginActivity

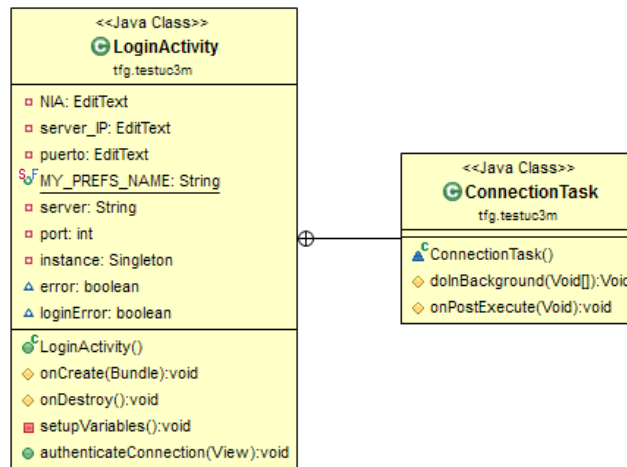


Figura 30 - Diagrama de clase LoginActivity

LoginActivity es la actividad principal que se ejecuta cuando se inicia la aplicación.

Sus atributos son:

- **NIA.** Es un atributo de tipo EditText. Sirve para acceder al elemento de la interfaz que recoge el NIA por pantalla.
- **server_IP.** Es un atributo de tipo EditText. Sirve para acceder al elemento de la interfaz que recoge la IP del servidor por pantalla.
- **puerto.** Es un atributo de tipo EditText. Sirve para acceder al elemento de la interfaz que recoge el puerto del servidor por pantalla.
- **MY_PREFS_NAME.** Es un atributo de tipo String. Sirve para almacenar el nombre del archivo de preferencias del teléfono específico de la aplicación.
- **server.** Es un atributo de tipo String. Sirve para almacenar la dirección IP del servidor y usarlo para la conexión.
- **port.** Es un atributo de tipo int. Sirve para almacenar el puerto del servidor y usarlo para la conexión.
- **instance.** Es un atributo de tipo Singleton. Sirve para instanciar la conexión con el servidor, mediante el uso de un socket TCP.
- **error/loginError.** Son atributos de tipo boolean. Su uso se limita a la gestión y captura de excepciones.

Los métodos de la clase son:

- **onCreate**. Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se carga la interfaz asociada a esta clase y se llama al método **setupVariables**.
- **setupVariables**. Este método se encarga de iniciar las variables para la recogida de datos, así como de comprobar si existe el archivo de preferencias para cargar la última dirección IP, puerto y NIA usados con éxito.
- **authenticateConnection**. Este método se encarga de realizar las comprobaciones previas al inicio de conexión. Es invocado cuando se pulsa el botón Conectar. Recoge y comprueba que se hayan introducido todos los campos, o que el NIA tenga un valor correcto. Si se dan las circunstancias esperadas, se procede a realizar la conexión.

La conexión debe realizarse en una clase auxiliar denominada **ConnectionTask** de tipo AsyncTask, pues es una acción concurrente que no puede realizarse en primer plano.

Los métodos de la clase **ConnectionTask** son los siguientes:

- **doInBackground**. Este método es el principal método de la clase. Se encarga de realizar las tareas en segundo plano. Este método se encarga de realizar la conexión con el servidor a través de la clase **Singleton**. Una vez establecida se envían los datos del NIA para comprobar si es correcto. Se recoge la respuesta.
- **onPostExecute**. Este método se ejecuta cuando se termina de ejecutar **doInBackground**. Aquí se controlan las excepciones que se hayan podido generar en la conexión. Si la conexión se realiza sin problemas, se guarda la información en el archivo de preferencias y se llama a la siguiente actividad.

4.2.2.3 WaitActivity

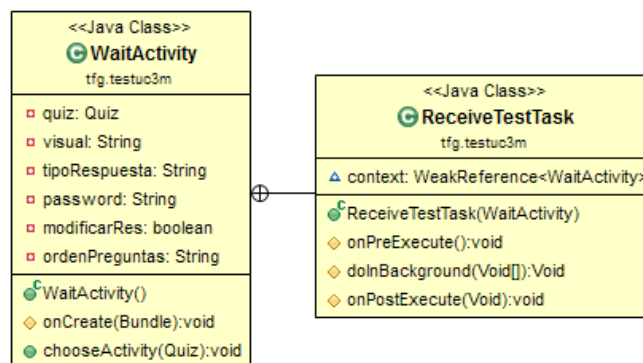


Figura 31 - Diagrama de clase WaitActivity

WaitActivity es la siguiente actividad que se ejecuta cuando se establece la conexión con éxito. Su función es la de vista intermedia entre la conexión y la realización del test. Se encarga de recibir el test y de decidir a qué actividad se va después en función a las características del test.

Sus atributos son:

- **quiz**. Es un atributo de tipo Quiz. El tipo Quiz pertenece a una clase creada específicamente para la interpretación del test y sus opciones. Aquí almacenaremos temporalmente el test recibido del servidor
- **visual, tipoRespuesta, password, ordenPreguntas**. Son atributos de tipo String. Se usan para recoger el tipo de visualización, respuesta del test, orden de las preguntas y de recoger la contraseña del test si la tuviera. Se usa para la decisión del camino a tomar en base a ellas.
- **modificarRes**. Es un atributo de tipo boolean. Se usa para recoger y gestionar el test en base a si se puede modificar la respuesta o no.

Sus métodos son:

- **onCreate**. Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se carga la interfaz asociada a esta clase. Aquí se llama a la clase de tipo AsyncTask **ReceiveTestTask**.
- **chooseActivity**. Este método se encarga de decidir qué actividad cargar a continuación. Recibe por parámetro el test leído. En base a atributos como **password** o **tipoRespuesta** se saltará una actividad u otra. Antes de eso se almacena el test leído en la instancia de **Singleton** para su posterior uso.

La clase auxiliar para la conexión y lectura del test es **ReceiveTestTask**. Sus métodos son los siguientes:

- **onPreExecute**. Este método se ejecuta antes de realizar la tarea principal. En este método instanciamos el test.
- **doInBackground**. Este método es el principal método de la clase. Se encarga de realizar las tareas en segundo plano. En este caso, se recupera el socket creado previamente en **Singleton**, y gracias a ello se recibe el test del servidor almacenándolo en el atributo **quiz**.
- **onPostExecute**. Este método se ejecuta cuando se termina de ejecutar **doInBackground**. Aquí se controlan las excepciones que se hayan podido generar al recibir el test. Principalmente si no se ha leído nada. Si el test se ha leído con éxito se llama al método **chooseActivity** para continuar.

4.2.2.4 PasswordActivity

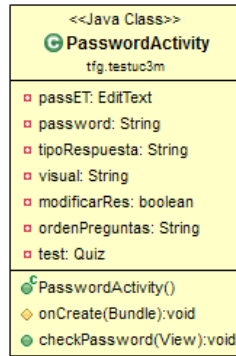


Figura 32 - Diagrama de clase PasswordActivity

PasswordActivity es la actividad que gestiona el control de la contraseña para acceder al test. Se accede a esta actividad solo si el test leído tiene contraseña.

Sus atributos son:

- **passET**. Es un atributo de tipo EditText. Se usa para recoger la contraseña introducida por teclado por el usuario.
- **password**. Es un atributo de tipo String. Se usa para almacenar la contraseña introducida y poder compararla con la contraseña correcta.
- **visual, tipoRespuesta, password, ordenPreguntas**. Son atributos de tipo String. Se usan para recoger el tipo de visualización, respuesta del test, orden de las preguntas y de recoger la contraseña del test si la tuviera. Se usa para la decisión del camino a tomar en base a ellas.
- **modificarRes**. Es un atributo de tipo boolean. Se usa para recoger y gestionar el test en base a si se puede modificar la respuesta o no.
- **test**. Es un atributo de tipo Quiz. El tipo Quiz pertenece a una clase creada específicamente para la interpretación del test y sus opciones. Aquí almacenaremos temporalmente el test recibido del servidor

Sus métodos son:

- **onCreate**. Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se carga la interfaz asociada a esta clase y de inicializar las variables de recogida de datos y de tipo de configuración del test. Se recoge el test guardado en la instancia de **Singleton**.
- **checkPassword**. Este método se encarga de comprobar la contraseña introducida por el usuario, y continuar a la siguiente actividad. Realiza la comprobación de si el usuario no ha introducido nada o si la contraseña no es correcta. En caso de ser correcta, decide a que actividad continuar en base a atributos como. Antes de eso se almacena el test de nuevo en la instancia de **Singleton** para su posterior uso.

4.2.2.5 TestActivity

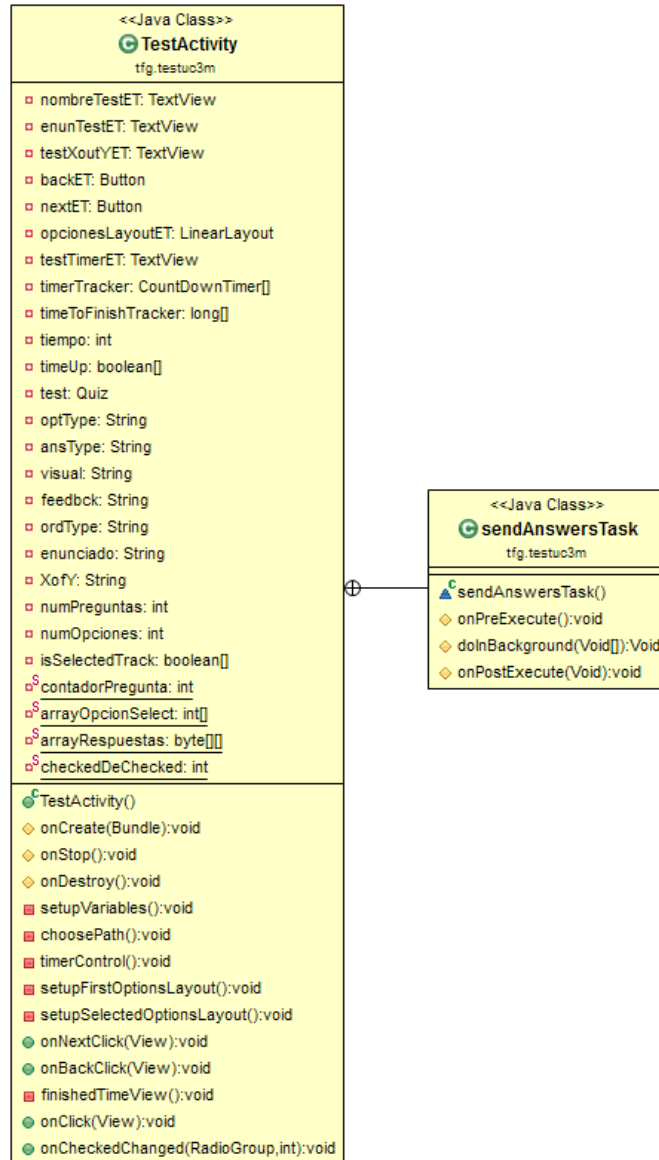


Figura 33 - Diagrama de clase TestActivity

TestActivity es la actividad que se ejecuta cuando el test es de tipo respuesta única. Esta clase controla el test si es posible modificar la respuesta.

Sus atributos son:

- **nombreTestET**, **enunTestET**, **testXoutYET**. Son atributos de tipo TextView y sirven para la representación de texto como el nombre del test, el enunciado o el número de pregunta en el que se está respectivamente.
- **backET**, **nextET**. Son atributos de tipo Button. Son los botones que permiten la navegación entre preguntas.
- **opcionesLayoutET**. Es un atributo de tipo LinearLayout. Se usa como espacio o marco donde irán las opciones de cada pregunta.
- **testTimerET**. Es un atributo de tipo TextView. Muestra el tiempo restante en cada pregunta. Solo será visible si el test cuenta con tiempo.

- **timerTracker**. Es un array de tipo CountdownTimer. Se usa para almacenar los contadores de cada pregunta y poder llevar un control de los mismos.
- **timeToFinishTracker**. Es un array de tipo long. Se usa para el control de tiempo restante de cada pregunta.
- **tiempo**. Es un atributo de tipo int. Se usa para almacenar el tiempo establecido para cada pregunta en la configuración del test.
- **timeUp**. Es un atributo de tipo boolean. Se usa para determinar si el tiempo en una pregunta ya ha terminado y actuar en consecuencia.
- **test**. Es un atributo de tipo Quiz. El tipo Quiz pertenece a una clase creada específicamente para la interpretación del test y sus opciones. Aquí almacenaremos temporalmente el test recibido del servidor
- **optType, ansType, visual, feedback, ordType**. Son atributos de tipo String. Se usan para almacenar la configuración del test. Almacenan el tipo de opción, el tipo de respuesta, el tipo de visualización, el tipo de feedback y el tipo de orden respectivamente.
- **enunciado**. Es un atributo de tipo String. Se usa para almacenar el enunciado del test.
- **XofY**. Es un atributo de tipo String. Se usa para representar en la interfaz el número de pregunta en el que está el usuario del número total.
- **numPreguntas, numOpciones**. Son dos atributos de tipo int. Se usan para almacenar el número de preguntas del test y el número de opciones de una pregunta respectivamente.
- **isSelectedTrack**. Es un array de tipo boolean. Se usa para controlar las preguntas en las que el usuario ya ha seleccionado alguna respuesta.
- **contadorPregunta**. Es un atributo de tipo int. Se usa como contador de pregunta para poder llevar un control en la navegación de la pregunta a tratar según se desplace el usuario.
- **arrayOpcionSelect**. Es un array de tipo int. Se usa para almacenar la selección del usuario en cada pregunta.
- **arrayRespuestas**. Es un array doble de tipo byte. Se usa para almacenar las respuestas elegidas y enviarlas al servidor.
- **checkedDeChecked**. Es un atributo de tipo int. Se usa como variable auxiliar para la función de desmarcar una pregunta si ya estaba marcada antes.

Sus métodos son:

- **onCreate**. Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se carga la interfaz asociada a esta clase. Se recoge el test guardado en la instancia de **Singleton**. Se llaman en orden consecutivo a los métodos **setupVariables** y **choosePath**.
- **setupVariables**. Este método se encarga de iniciar las variables para la recogida de datos y de inicializar las variables de configuración de test y de control de preguntas.
- **choosePath**. Este método se encarga de seleccionar qué camino tomar en base a unas condiciones. Es llamado cada vez que se cambia de pregunta. Su función principal es la de gestionar cómo será la vista de la pregunta a mostrar. Si el tiempo existiese y se hubiese acabado, si ya se ha seleccionado una opción o si no se ha seleccionado ninguna, son las condiciones para decidir. Lleva a cabo tareas secundarias como

eliminar las opciones para cargar las nuevas en cada iteración que se le llama o gestionar el texto de los botones de navegación en relación a la pregunta en la que se encuentre.

- **finishedTimeView.** Este método se encarga de gestionar la vista cuando el tiempo de una pregunta ha terminado. Se muestra el enunciado y un mensaje indicando que el tiempo está agotado. Se muestra en lugar de las opciones la respuesta que se ha tomado como final.
- **setupFirstOptionsLayout.** Este método se encarga de gestionar la vista cuando la pregunta a la que se accede no se ha modificado previamente. Se muestra por pantalla la información relevante como el nombre del test, el número de pregunta, el enunciado y se generan las opciones mediante el uso de un `RadioGroup`. El atributo `RadioGroup` engloba una serie de `RadioButton`, botones seleccionables. Esto asegura que solo uno pueda estar marcado a la vez, pues es una propiedad inherente a un `RadioGroup`. Si el test tiene un tiempo límite, dentro de este método se comprueba y se inicia el contador mediante el uso de un atributo de tipo `CountDownTimer`. Como su nombre indica, este atributo está diseñado para crear cuentas atrás.
- **setupSelectedOptionsLayout.** Este método se encarga de gestionar la vista cuando la pregunta a la que se accede ya ha sido modificada previamente. Su funcionamiento es similar al de **setupFirstOptionsLayout**, mostrando la misma información. La diferencia en este caso es que al generar las opciones, marca la que se haya seleccionado antes. En este caso usa el atributo `arrayOpcionSelect` para saber cuál es la opción seleccionada. Si la pregunta tiene tiempo asignado, se llama al método **timerControl** para gestionarlo.
- **timerControl.** Este método gestiona los tiempos de las preguntas. Mediante el uso del array `timeToFinishTracker` guarda los tiempos restantes de cada pregunta cuando se itera entre ellas. Cuando se vuelve a la pregunta se asigna el tiempo restante de nuevo. Se encarga también de avanzar hacia adelante automáticamente si el tiempo de una pregunta se acaba
- **onNextClick.** Este método se encarga de gestionar las funciones que se ponen en marcha cuando el usuario pulsa el botón Siguiente en la interfaz. Al avanzar aumenta el contador de preguntas `contadorPregunta`. Controla que si la pregunta en la que esté es la última, se llame a la clase de envío de las respuestas para finalizar el test. Si no lo es, se carga la siguiente pregunta llamando al método **choosePath**. Antes de cambiar de pregunta, para el tiempo del contador actual si hubiese tiempo.
- **onBackClick.** Este método se encarga de gestionar las funciones que se ponen en marcha cuando el usuario pulsa el botón Atrás en la interfaz. Al retroceder disminuye el contador de preguntas `contadorPregunta`. Carga la siguiente pregunta llamando al método **choosePath**. Antes de cambiar de pregunta, para el tiempo del contador actual si hubiese tiempo.

- **onClick**. Este método se ejecuta cuando el usuario toca una opción de la pregunta. Se encarga de gestionar que una opción se desmarque cuando el usuario toque sobre ella y ya ha sido seleccionada previamente.
- **onCheckedChanged**. Este método se ejecuta cuando el usuario marca una opción. Se encarga de guardar la selección realizada en el array **arrayOpcionSelect**, tanto si se marca como si se desmarca una opción.

La clase auxiliar para el envío de las respuestas es **SendAnswersTask**. Estos son sus métodos:

- **onPreExecute**. Este método se ejecuta antes de realizar la tarea principal. En este caso antes de enviar las respuestas, se pasan las respuestas contenidas en **arrayOpcionSelect** al array de bytes a enviar **arrayRespuestas**. La asignación se realiza de acuerdo al protocolo de comunicación, almacenando un 1 en la opción elegida y un 0 en las demás.
- **doInBackground**. Este método es el principal método de la clase. Se encarga de realizar las tareas en segundo plano. En este caso, se recupera el socket creado previamente en Singleton, y se envía el array de bytes con la respuesta.
- **onPostExecute**. Este método se ejecuta cuando se termina de ejecutar **doInBackground**. Una vez se han enviado las respuestas, se procede a la última actividad.

4.2.2.6 TestHideActivity

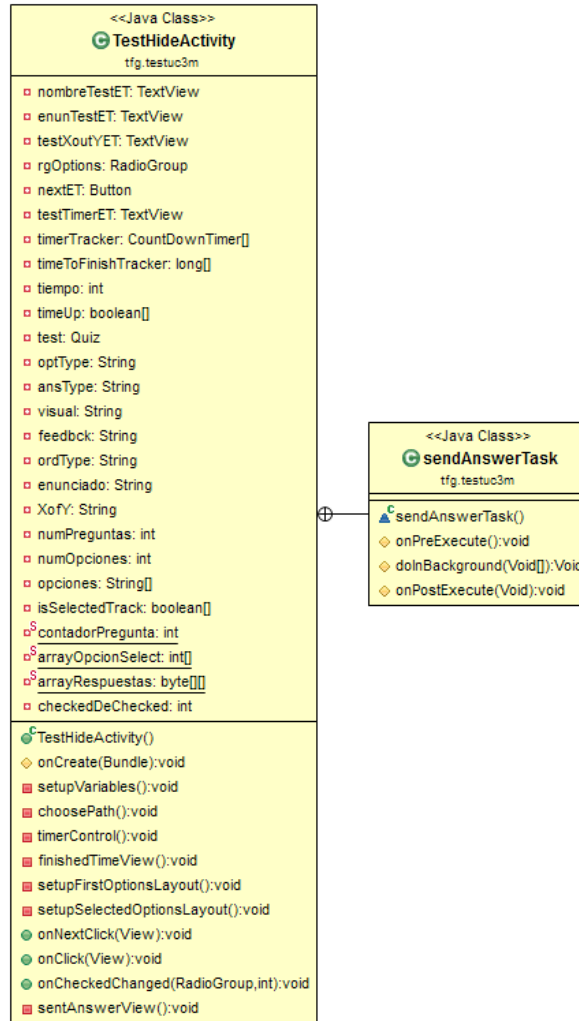


Figura 34 - Diagrama de clase TestHideActivity

TestHideActivity es una actividad muy similar a **TestActivity** pero con ligeras modificaciones para el caso en el que no se pueda modificar la respuesta y se envíen las respuestas de una en una, siguiendo el ritmo que marque el profesor.

Para no repetir de nuevo todos los atributos y métodos que ya se vieron anteriormente, se detallaran solo las diferencias significativas.

La principal diferencia a nivel de interfaz es que solo se dispone de un botón en la parte superior que es Enviar. Al pulsarlo se enviará la respuesta y se esperará a que el profesor continúe en su ordenador para mostrar la siguiente.

- **sentAnswerView**. Este método gestiona la vista después de enviar una pregunta. Cuando el usuario envía una pregunta con éxito, la vista cambia y muestra por pantalla un mensaje confirmando el envío de la respuesta.
- **onNextClick**. Se encarga de llamar a la clase de envío de preguntas cada vez que se le llama. Llama al método **sentAnswerView**.

La clase auxiliar para el envío de las respuestas en **SendAnswerTask**. Sus métodos son:

- **onPreExecute**. Este método se ejecuta antes de realizar la tarea principal. En este caso antes de enviar la respuesta, se pasa el contenido de **arrayOpcionSelect** al array de bytes a enviar **arrayRespuestas**. La asignación se realiza de acuerdo al protocolo de comunicación, almacenando un 1 en la opción elegida y un 0 en las demás.
- **doInBackground**. Este método es el principal método de la clase. Se encarga de realizar las tareas en segundo plano. En este caso, se recupera el socket creado previamente en Singleton, y se envían las respuestas almacenadas en el array de bytes. Una vez enviado, se mantiene a la espera de recibir la confirmación para continuar por parte del servidor.
- **onPostExecute**. Este método se ejecuta cuando se termina de ejecutar **doInBackground**. Una vez se han enviado las respuestas, se aumenta el atributo **contadorPregunta** para avanzar y se llama al método **choosePath** para cargar la siguiente pregunta. Si la pregunta respondida es la última, se procede a la actividad final.

4.2.2.7 TestMultiActivity

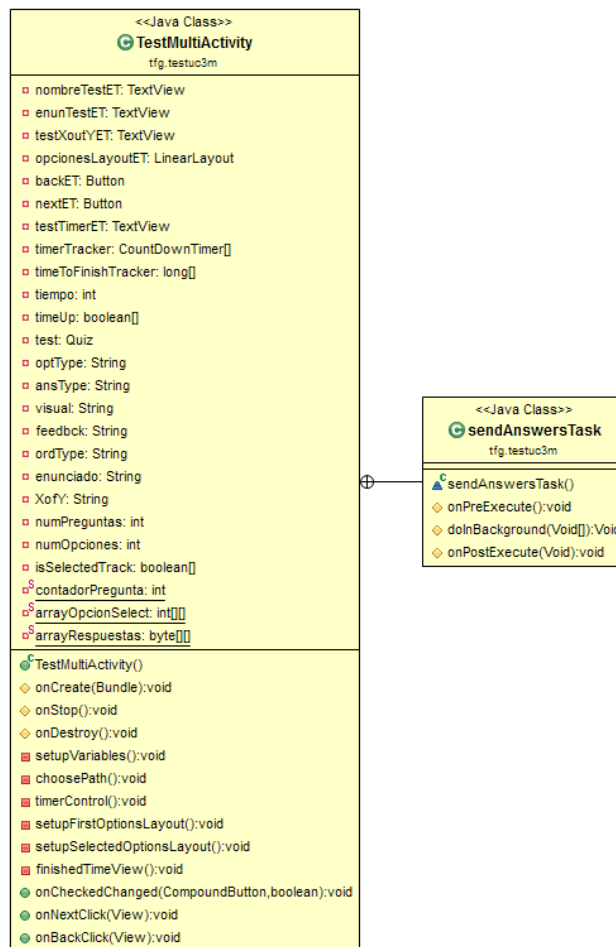


Figura 35 - Diagrama de clase de TestMultiActivity

TestMultiActivity es la actividad que se ejecuta cuando el test es de tipo respuesta múltiple. Esta clase controla el test si es posible modificar la respuesta.

Sus atributos son comunes a los de las clases previamente vistas, por lo que no se repetirán sus descripciones. Se cambia el tipo de dos atributos:

- **arrayOpcionSelect.** Es un array doble de tipo int. Se usa para almacenar la selección o selecciones del usuario en cada pregunta.
- **arrayRespuestas.** Es un array doble de tipo byte. Se usa para almacenar las respuestas elegidas y enviarlas al servidor.

Sus métodos son:

- **onCreate.** Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se carga la interfaz asociada a esta clase. Se recoge el test guardado en la instancia de **Singleton**. Se llama en orden consecutivo a los métodos **setupVariables** y **choosePath**.
- **setupVariables.** Este método se encarga de iniciar las variables para la recogida de datos y de inicializar las variables de configuración de test y de control de preguntas.
- **choosePath.** Este método se encarga de seleccionar qué camino tomar en base a unas condiciones. Es llamado cada vez que se cambia de pregunta. Su función principal es la de gestionar cómo será la vista de la pregunta a mostrar. Si el tiempo existiese y se hubiese acabado, si ya se ha seleccionado una opción o si no se ha seleccionado ninguna, son las condiciones para decidir. Lleva a cabo tareas secundarias como eliminar las opciones para cargar las nuevas en cada iteración que se le llama o gestionar el texto de los botones de navegación en relación a la pregunta en la que se encuentre.
- **finishedTimeView.** Este método se encarga de gestionar la vista cuando el tiempo de una pregunta ha terminado. Se muestra el enunciado y un mensaje indicando que el tiempo está agotado. Se muestra en lugar de las opciones la respuesta que se ha tomado como final.
- **setupFirstOptionsLayout.** Este método se encarga de gestionar la vista cuando la pregunta a la que se accede no se ha modificado previamente. Se muestran por pantalla la información relevante como el nombre del test, el número de pregunta, el enunciado y se generan las opciones mediante el uso de **CheckBox**. El atributo **CheckBox** representa una casilla seleccionable y deseleccionable. No tienen restricciones en este caso y pueden ser marcadas todas o ninguna. Si el test tiene un tiempo límite, dentro de este método se comprueba y se inicia el contador mediante el uso de un atributo de tipo **CountDownTimer**. Como su nombre indica, este atributo está diseñado para crear cuentas atrás.

- **setupSelectedOptionsLayout.** Este método se encarga de gestionar la vista cuando la pregunta a la que se accede ya ha sido modificada previamente. Su funcionamiento es similar al de **setupFirstOptionsLayout**, mostrando la misma información. La diferencia en este caso es que al generar las opciones, marca la que se haya seleccionado antes. En este caso usa el atributo **arrayOpcionSelect** para saber cuál o cuáles son las opciones seleccionadas. Si la pregunta tiene tiempo asignado, se llama al método **timerControl** para gestionarlo.
- **timerControl.** Este método gestiona los tiempos de las preguntas. Mediante el uso del array **timeToFinishTracker** guarda los tiempos restantes de cada pregunta cuando se itera entre ellas. Cuando se vuelve a la pregunta se asigna el tiempo restante de nuevo. Se encarga también de avanzar hacia adelante automáticamente si el tiempo de una pregunta se acaba
- **onNextClick.** Este método se encarga de gestionar las funciones que se ponen en marcha cuando el usuario pulsa el botón Siguiente en la interfaz. Al avanzar aumenta el contador de preguntas **contadorPregunta**. Controla que si la pregunta en la que esta es la última, se llame a la clase de envío de las respuestas para finalizar el test. Si no lo es, se carga la siguiente pregunta llamando al método **choosePath**. Antes de cambiar de pregunta, para el tiempo del contador actual si hubiese tiempo.
- **onBackClick.** Este método se encarga de gestionar las funciones que se ponen en marcha cuando el usuario pulsa el botón Atrás en la interfaz. Al retroceder disminuye el contador de preguntas **contadorPregunta**. Carga la siguiente pregunta llamando al método **choosePath**. Antes de cambiar de pregunta, para el tiempo del contador actual si hubiese tiempo.
- **onClick.** Este método se ejecuta cuando el usuario toca una opción de la pregunta. Se encarga de gestionar que una opción se desmarque cuando el usuario toque sobre ella y ya ha sido seleccionada previamente.
- **onCheckedChanged.** Este método se ejecuta cuando el usuario marca una opción. Se encarga de guardar la selección realizada en el array **arrayOpcionSelect**, tanto si se marca como si se desmarca una opción.

La clase auxiliar para el envío de las respuestas es **SendAnswersTask**. Esta clase ya se vio en detalle en la actividad **ActivityTest**.

4.2.2.8 TestHideMultiActivity

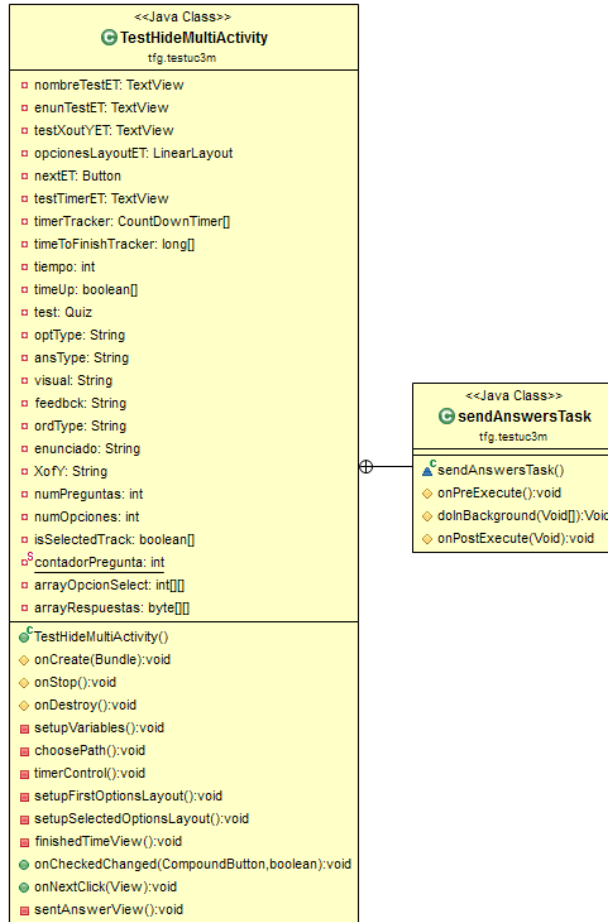


Figura 36 - Diagrama de clase TestHideMultiActivity

TestHideMultiActivity es una actividad muy similar a **TestMultiActivity** pero con ligeras modificaciones para el caso en el que no se pueda modificar la respuesta y se envíen las respuestas de una en una, siguiendo el ritmo que marque el profesor.

Para no repetir de nuevo todos los atributos y métodos que ya se vieron anteriormente, se detallaran solo las diferencias significativas.

La principal diferencia a nivel de interfaz es que solo se dispone de un botón en la parte superior que es Enviar. Al pulsarlo se enviará la respuesta y se esperará a que el profesor continúe en su ordenador para mostrar la siguiente.

- **sentAnswerView**. Este método gestiona la vista después de enviar una pregunta. Cuando el usuario envía una pregunta con éxito, la vista cambia y muestra por pantalla un mensaje confirmando el envío de la respuesta.
- **onNextClick**. Se encarga de llamar a la clase de envío de preguntas cada vez que se le llama. Llama al método **sentAnswerView**.

La clase auxiliar para el envío de las respuestas es **SendAnswerTask**. Esta clase ya se vio en detalle en la actividad **TestHideActivity**.

4.2.2.9 FinishedActivity

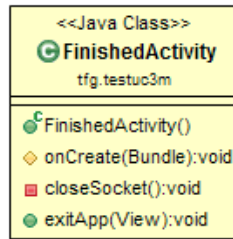


Figura 37 - Diagrama de clase FinishedActivity

FinishedActivity es la última actividad a la que se llega tras realizar el test.

Su función es informar al usuario de que se ha completado el test con éxito y el de cerrar la comunicación y liberar recursos.

Sus métodos son:

- **onCreate**. Este método es específico de las actividades. Como su nombre indica, se ejecuta en primer lugar al iniciar la actividad. En este método se llama a la función **closeSocket**.
- **closeSocket**. Este método se encarga de cerrar los buffers de envío y recepción de datos y de cerrar el socket, así como de eliminar la instancia de Singleton para liberar recursos.
- **exitApp**. Este método se ejecuta cuando el usuario pulsa el botón Salir en la interfaz. Su función es la de cerrar todas las actividades que estén abiertas aún mediante la llamada a la clase **LoginActivity** con un Intent con el flag "FLAG_ACTIVITY_CLEAR_TOP". Esta señal o flag indica al sistema que finalice las actividades que estén aún en ejecución. Al llegar a la clase **LoginActivity**, se recibe el intent y se cierra esta actividad finalizando el ciclo de vida de la aplicación.

4.3 Diseño de interfaces del servidor

En este apartado se presentan los diseños de las diferentes interfaces que componen el servidor, con una breve descripción y una imagen ilustrativa.

4.3.1 Diseño de la vista principal

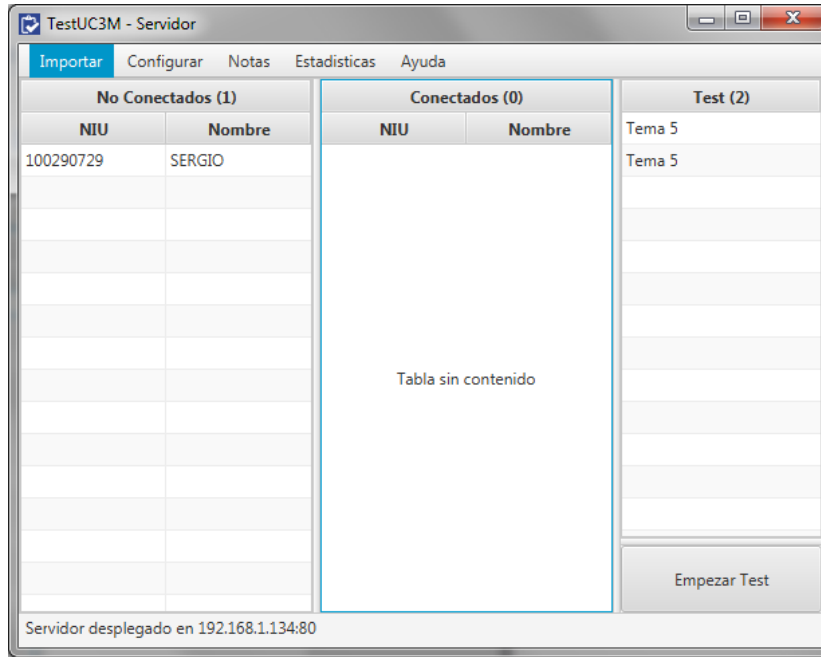


Figura 38 - Diseño vista principal servidor

Esta es la vista principal que se ve cuando se ejecuta el servidor.

En la parte superior se encuentra el menú con las opciones de Importar, Configurar, Nota, Estadísticas y Ayuda.

En medio tenemos tres partes diferenciadas:

- A la izquierda, la tabla de los alumnos que no están conectados con el servidor.
- En medio, la tabla con los alumnos que están actualmente conectados al servidor.
- A la derecha, la lista de test que están importados en el servidor. Debajo está el botón para iniciar el test.

En la parte inferior de la venta, se ve un mensaje con información del servidor.

4.3.2 Diseño de configuración de test

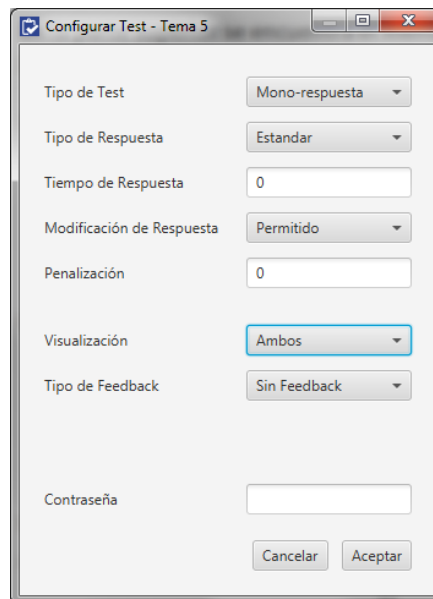


Figura 39 - Diseño vista configuración de test

Esta ventana emergente se muestra cuando se accede a la opción de Configurar en el menú o antes de iniciar un test que no esté configurado.

Cuenta con las diferentes opciones para la configuración del test, con menús desplegables para las opciones y campos de texto para introducir valores.

En la parte más baja tiene dos botones para aceptar los cambios realizados o para cancelar y salir de la ventana.

4.3.3 Diseño de vista de test

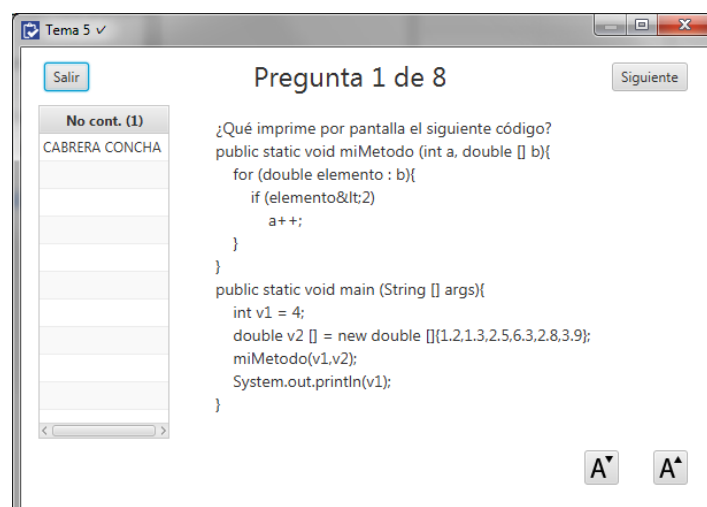


Figura 40 - Diseño vista de test del servidor

Esta ventana emergente se muestra cuando se inicia el test.

En la parte superior están los botones Atrás y Siguiente para la navegación entre preguntas. En medio se muestra la pregunta en la que estamos del número total.

En la parte izquierda está la tabla de alumnos que aún no han contestado el test o la pregunta.

En el centro está el enunciado del test. En la parte de abajo a la derecha están los botones que permiten aumentar o disminuir el tamaño del texto.

4.3.4 Diseño de estadísticas

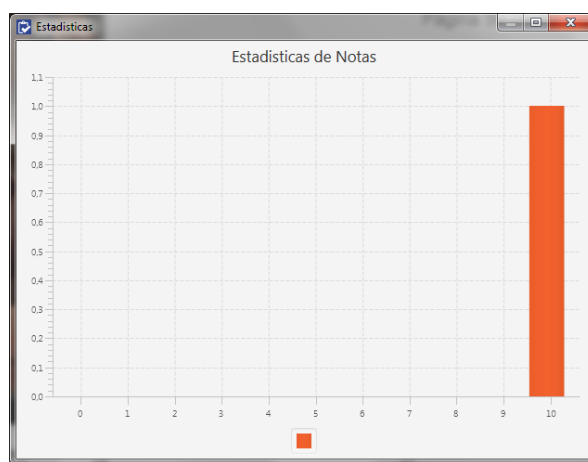


Figura 41 - Diseño vista estadísticas del servidor

La ventana emergente de estadística solo estará disponible cuando el test se haya realizado.

Se presenta en la ventana el título del gráfico, junto al gráfico de barras que representa el número de alumnos con la nota obtenida en el test del 0 al 10.

4.4 Diseño de interfaces de la aplicación móvil

En este apartado se presentan los diseños de las diferentes pantallas que componen la aplicación móvil, con una breve descripción y una imagen ilustrativa.

4.4.1 Diseño de identificación



Figura 42 - Diseño de vista de identificación

La pantalla principal de la aplicación es la pantalla de identificación para la conexión. Como se ve en la imagen, se compone de los campos de NIA, IP del servidor y puerto a rellenar para la conexión. El botón para realizar la conexión está situado justo en el centro.

4.4.2 Diseño de espera



Figura 43 - Diseño de vista de espera

La pantalla de espera aparece una vez realizada la conexión. Esta pantalla informa, como se ve en la imagen, de que la aplicación está esperando que el profesor inicie el test para continuar.

4.4.3 Diseño de contraseña



Figura 44 - Diseño de vista de contraseña

La pantalla de contraseña muestra el campo para la introducción de la contraseña y el botón para comprobar si es correcta.

4.4.4 Diseño de test

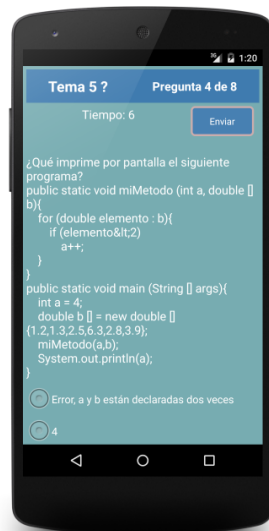


Figura 45 - Diseño de vista de test

La pantalla de test muestra en la parte superior el nombre del test y la pregunta en la que se encuentra del total. En esta opción tenemos el tiempo que resta para contestar el test (campo opcional) y el botón de enviar la respuesta. Si el test pudiese ser modificado, tendría un botón para navegar hacia atrás que en este caso no se muestra.

En medio podemos ver el enunciado de la pregunta, y más abajo las opciones disponibles para contestar.

4.4.5 Diseño finalizar



Figura 46 - Diseño de vista de finalización

Esta pantalla se muestra al finalizar el test. Se muestra un mensaje que informa al usuario de que el test ha finalizado y un botón para salir de la aplicación.

5. Implementación y pruebas

Una vez recopilados todos los datos relevantes en la fase de análisis y diseño, se procede con la implementación del sistema. En este apartado se tratarán los aspectos destacados de la implementación del proyecto y de las pruebas de sistema realizadas.

5.1 Especificación del entorno de desarrollo

El objetivo de esta tarea es definir el entorno necesario para la implementación de los componentes del sistema.

La especificación del entorno se realizará según los siguientes conceptos:

- **Entorno tecnológico:**
 - Procesador: Intel Core i5-3570k CPU 3,40 GHz.
 - Memoria 8 GB RAM.
 - Espacio libre disco duro: 400 GB de espacio libre.
 - Tarjeta gráfica: NVIDIA GeForce GTX 660 de 2 GB.
 - Teclado/ratón: Logitech.
 - Monitor: Dell U2312HM de 23 pulgadas.

- **Herramientas de construcción:**
 - **Android Studio:** Es un programa informático compuesto por un conjunto de herramientas de programación y de emulación orientadas al desarrollo de aplicaciones para el sistema Android.
 - **Eclipse Luna:** Es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar el proyecto.

 - **Windows:** Familia de sistemas operativos desarrollados por Microsoft. Tiene una fácil interfaz gráfica, así como el soporte multitarea; gráficos de pantalla e impresora independientes del dispositivo, multitarea cooperativa entre las aplicaciones e interfaz gráfica con menús desplegados, ventana en cascada y soporte para ratón.

- **Lenguajes de desarrollo:**
 - **Java**
 - **XML**

- **Restricciones técnicas del entorno:** Funcionamiento en terminales Android a partir de la versión 2.3.3.
Versión de Java JRE 8.x instalada en el ordenador para la ejecución del servidor.

- **Planificación de capacidades previstas:** se ha determinado el espacio en disco necesario que se va a asignar a cada parte del proyecto.

5.2 Entorno de pruebas del sistema

El equipo utilizado para la ejecución de las pruebas es el mismo que el descrito para la implementación. Este equipo se encargará de la ejecución del servidor, así como de las pruebas con clientes generados a través de unas clases de prueba.

Para la aplicación Android, se realizarán pruebas con dispositivos de diferentes características. Se tiene especialmente en cuenta que funcione en un dispositivo con la versión mínima requerida (2.3.3) y una con la más reciente (5.1.1). También se ha tenido en cuenta dispositivos de rango intermedio como 4.2.2 o 4.4.2, ya que son versiones con una gran cuota de usuarios.

Las pruebas se han realizado en dispositivos físicos y emuladores, por comodidad y limitación en la disponibilidad de dispositivos para pruebas. El dispositivo físico utilizado ha sido:

- **Nexus 7**
 - Versión de Android 4.2.2 JellyBean
 - Pantalla de 7 pulgadas con resolución 800x1280 px
 - RAM: 1GB
 - Procesador: Cortex A9 Quad-core a 1.2 GHz
 - Memoria interna: 16GB

Los emuladores utilizados han sido:

- **Nexus S**
 - Versión de Android 2.3.3 Gingerbread
 - Pantalla de 4 pulgadas con resolución 480x800
 - RAM: 512MB
 - Procesador: ARM Cortex A8 a 1GHz
- **Nexus 5**
 - Versión de Android: 5.1.1 Lollipop
 - Pantalla de 4,95 pulgadas con resolución 1920x1080 px
 - RAM: 2GB
 - Procesador: Qualcomm Snapdragon 800 a 2.3GHz
- **Nexus 5**
 - Versión de Android 4.4.2 KitKat
 - Pantalla de 4,95 pulgadas con resolución 1920x1080 px
 - RAM: 2GB
 - Procesador: Qualcomm Snapdragon 800 a 2.3GHz

Los procesadores de los emuladores son emulados gracias al hardware del ordenador y a librerías específicas para ello. Se muestran los procesadores reales para dar una estimación acotada del desempeño real de la simulación en cada caso.

5.3 Pruebas del sistema

5.3.1 Especificación de las pruebas del sistema

Se presenta la estructura para la representación de las pruebas del sistema.

- **Código:** Código para la identificación de la prueba realizada. 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.
- **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.

Especificación de Prueba del Sistema	
Código	PS-000
Nombre	
Descripción	
Procedimiento de Prueba	
Resultado Esperado	

Tabla 67 - Ejemplo tabla Prueba de Sistema

5.3.2 Pruebas del Sistema

Las pruebas realizadas del programa servidor se centran específicamente en la parte de la interfaz del sistema, su representación y funcionamiento. Algunas operaciones como las de importar test o la interpretación de los test no se tienen en cuenta por no haber participado en su elaboración.

Especificación de Prueba del Sistema	
Código	PS-001
Nombre	Interfaz principal
Descripción	Se comprueba que la interfaz principal se visualice de forma correcta con todas las opciones.
Procedimiento de Prueba	1. Ejecutar la aplicación TestUC3M servidor
Resultado Esperado	<ul style="list-style-type: none"> ● Se inicia la interfaz del programa con las diferentes opciones y tablas accesibles y funcionando

Tabla 68 - Prueba del Sistema 1

Especificación de Prueba del Sistema	
Código	PS-002
Nombre	Configuración
Descripción	Se comprueba que la vista de configuración se carga cuando se selecciona un test y que permite configurarlo mediante las opciones disponibles.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M servidor 2. Seleccionar un test 3. Seleccionar la opción Configurar
Resultado Esperado	<ul style="list-style-type: none"> • Se abre la ventana de configuración, permitiendo configurar las opciones y guardando la configuración una vez terminado

Tabla 69 - Prueba del Sistema 2

Especificación de Prueba del Sistema	
Código	PS-003
Nombre	Vista de Test
Descripción	Se comprueba que al iniciar el test, se carguen las preguntas y se puedan ver por pantalla, a la vez que se puede iterar entre ellas. Se comprueba que la tabla de alumnos que aún no han respondido cargue los alumnos que no han respondido y que los elimine si ya han contestado en tiempo real. Por último se quiere comprobar que al finalizar el test se cierre todo correctamente y se vuelva a la interfaz.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M servidor 2. Seleccionar un test 3. Configurar un test mediante la opción Configurar 4. Esperar a que uno o varios alumnos se conecten al servidor 5. Iniciar el test pulsando en el botón Empezar Test 6. El alumno responderá el test y lo enviará al servidor
Resultado Esperado	<ul style="list-style-type: none"> • Se lanza la ventana de la vista de test • Se puede navegar entre las preguntas viendo el enunciado de cada una de ellas • Se ve la tabla con los alumnos que aún no han contestado • Cuando el alumno termine de realizar el test, su apellido se eliminará de la tabla • Cuando se llegue a la última pregunta se podrá finalizar el test con éxito • Si se cierra la ventana, el test también se da por finalizado

Tabla 70 - Prueba del Sistema 3

Especificación de Prueba del Sistema	
Código	PS-004
Nombre	Estadísticas
Descripción	Se comprueba que se muestren las estadísticas del último test realizado. Se realizará la prueba con una nota fija para comprobar que realmente se representan bien los resultados.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M servidor 2. Seleccionar un test 3. Configurar un test mediante la opción Configurar 4. Esperar a que uno o varios alumnos se conecten al servidor 5. Iniciar el test pulsando en el botón Empezar Test 6. El alumno responderá el test y lo enviará al servidor 7. Finalizar el test 8. Seleccionar la opción Estadísticas
Resultado Esperado	<ul style="list-style-type: none"> • Se abre una ventana con las estadísticas • Se comprueba que el grafico contiene los datos esperados

Tabla 71 - Prueba del Sistema 4

Especificación de Prueba del Sistema	
Código	PS-005
Nombre	Identificación y conexión con el servidor
Descripción	Se comprueba que se realice la conexión con el servidor usando los datos de NIA, dirección IP del servidor y el puerto. Se comprueban excepciones generadas por fallo en la conexión o por una identificación incorrecta.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar
Resultado Esperado	<ul style="list-style-type: none"> • Se establece la conexión con el servidor. Se recibe una notificación informando de que se ha establecido la conexión y se lanza la vista de espera al inicio del test. • Si no se introduce algún campo se notifica al usuario que debe rellenar todos los campos. • Si el NIA introducido no es correcto se notifica al usuario que ese NIA no es válido.

Tabla 72 - Prueba del Sistema 5

Especificación de Prueba del Sistema	
Código	PS-006
Nombre	Lectura de test
Descripción	Se comprueba que el test se recibe y se lee correctamente para su realización.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test sin contraseña para la prueba
Resultado Esperado	<ul style="list-style-type: none"> • El test se recibe y se interpreta de manera correcta. Se pasa a la vista de test donde se puede navegar por todas las preguntas.

Tabla 73 - Prueba del Sistema 6

Especificación de Prueba del Sistema	
Código	PS-007
Nombre	Control de contraseña
Descripción	Se comprueba que la aplicación gestione bien la contraseña de un test. Se comprueba que se introduzca una contraseña y se verifica su validez.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test con contraseña para la prueba
Resultado Esperado	<ul style="list-style-type: none"> • Se abre la vista de contraseña, donde se pide al usuario que introduzca la contraseña asociada a ese test. • Si la contraseña introducida es correcta se pasa a la vista de test. • Si la contraseña es incorrecta, se avisa por notificación al usuario y puede volver a intentarlo. Si no ha introducido ninguna contraseña también se le notificará

Tabla 74 - Prueba del Sistema 7

Especificación de Prueba del Sistema	
Código	PS-008
Nombre	Navegación de Test
Descripción	Se comprueba que se puede navegar por las preguntas de un test y modificar las opciones, independientemente de si son test de respuesta única o múltiple.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test de respuesta única o de respuesta múltiple
Resultado Esperado	<ul style="list-style-type: none"> • Se puede navegar por las preguntas y marcar y desmarcar las opciones según el tipo de test. Si es de respuesta única, solo una única respuesta o ninguna. Si es de respuesta múltiple, una, más de una o ninguna • Si se marca una opción, se mantiene guardada si se abandona esa pregunta y se retorna después.

Tabla 75 - Prueba del Sistema 8

Especificación de Prueba del Sistema	
Código	PS-009
Nombre	Respuesta a test
Descripción	Se comprueba que se envían las respuestas del test al servidor con éxito.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test de respuesta única o de respuesta múltiple 6. El usuario responde las preguntas y las envía mediante el botón Finalizar
Resultado Esperado	<ul style="list-style-type: none"> • Las respuestas se envían al servidor y este las recibe sin problemas, calculando la nota de cada alumno.

Tabla 76 - Prueba del Sistema 9

Especificación de Prueba del Sistema	
Código	PS-010
Nombre	Tiempo en test
Descripción	Se comprueba que los temporizadores de respuesta funcionen bien, sin importar el tipo de test.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test sin importar el tipo con un tiempo para responder determinado
Resultado Esperado	<ul style="list-style-type: none"> • Al navegar entre las preguntas se observa el temporizador en pantalla con el tiempo restante de cada pregunta. • Al finalizar el tiempo no se permite responder esa pregunta y se pasa a la siguiente. Se toma la última opción elegida como respuesta final. • Si el tiempo de una pregunta ha finalizado se muestra en pantalla un mensaje indicándolo.

Tabla 77 - Prueba del Sistema 10

Especificación de Prueba del Sistema	
Código	PS-011
Nombre	Respuesta pregunta a pregunta
Descripción	Se comprueba que la aplicación permite contestar un test pregunta por pregunta siendo el profesor el que controle cuando se avanza de pregunta a través del servidor.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test con restricción de modificación de respuesta.
Resultado Esperado	<ul style="list-style-type: none"> • Las preguntas se muestran en orden, y se debe contestar a la pregunta mediante el botón Enviar antes de poder ver la siguiente. • Una vez enviada la respuesta, se muestra un mensaje en pantalla indicando que se ha enviado con éxito. La aplicación se mantiene a la espera de la siguiente. • Cuando el profesor avance a la siguiente pregunta, se mostrará por pantalla para su contestación.

Tabla 78 - Prueba del Sistema 11

Especificación de Prueba del Sistema	
Código	PS-012
Nombre	Orden aleatorio
Descripción	Se comprueba que el test muestre las preguntas en orden aleatorio. Esta característica es determinada por el profesor al configurar el test.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil y en emulador, para realizar la prueba en ambos dispositivos. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test con restricción de orden de preguntas aleatorio.
Resultado Esperado	<ul style="list-style-type: none"> • Las preguntas se muestran en un orden aleatorio al original. • Se compara entre ambos dispositivos y el orden original para verificar que el orden es distinto. • Al recibir las respuestas, se calcula la nota de manera correcta.

Tabla 79 - Prueba del Sistema 12

Especificación de Prueba del Sistema	
Código	PS-013
Nombre	Finalizar aplicación
Descripción	Se comprueba que una vez terminado el test, se muestra la vista de finalización para salir de la aplicación.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ejecutar la aplicación TestUC3M en el dispositivo móvil y en emulador, para realizar la prueba en ambos dispositivos. 2. Introducir los datos de conexión (NIA, IP y puerto) 3. Pulsar el botón Conectar 4. Establecer conexión con el servidor 5. El profesor inicia un test. 6. El usuario responde a todas las preguntas y envía las respuestas.
Resultado Esperado	<ul style="list-style-type: none"> • Se muestra la vista de finalización con un mensaje informando de que se ha realizado el test con éxito y un botón para salir de la aplicación. • Al pulsar el botón se cierran todas las actividades y la aplicación.

Tabla 80 - Prueba del Sistema 13

6. Gestión del proyecto

En esta sección se tratarán los aspectos de gestión y planificación del proyecto. Se detallarán los aspectos importantes como la metodología seguida, las fases de desarrollo en la planificación y un cálculo de costes total.

6.1 Metodología

En este apartado se presenta la metodología elegida para la realización del proyecto.

6.1.1 Modelo de Prototipo

El modelo de trabajo a seguir durante el desarrollo es el **modelo de prototipo** [31]. Este tipo de desarrollo consiste en realizar prototipos funcionales de diversas partes del proyecto para enseñar la funcionalidad principal del producto al cliente y realizar modificaciones o arreglos en base al feedback recibido en reuniones. Las reuniones fueron mantenidas en plazos de 15 días aproximadamente con el cliente en su despacho de la universidad.

Los ciclos de vida de este modelo se representan en el siguiente esquema.

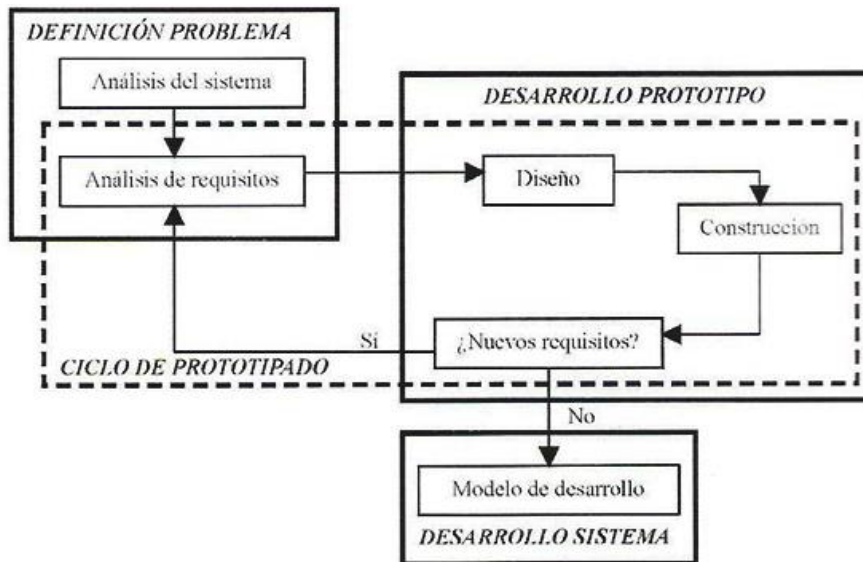


Figura 47 - Ciclo de vida Modelo de Prototipo

Hay tres etapas diferenciadas:

- **Definición del problema.** En esta etapa se define el problema a tratar y se realiza un análisis del mismo. Esto incluye el análisis de casos de uso y de requisitos.
- **Desarrollo del prototipo.** Una vez establecidos los requisitos y los casos de uso, se procede a realizar el prototipo. En esta fase entra en juego el diseño y la implementación en sí del prototipo, en base a los requisitos analizados.
- **Desarrollo del sistema.** Si el prototipo generado satisface las necesidades previstas, se procede a incorporarlo de manera permanente al sistema completo. En caso de que surjan nuevos requisitos o haya que modificar algún aspecto que no se haya tenido en cuenta antes, volvemos a la fase de análisis de requisitos.

6.2 Planificación del proyecto

En este apartado se presenta la planificación del proyecto, mediante un **diagrama de Gantt** [32].

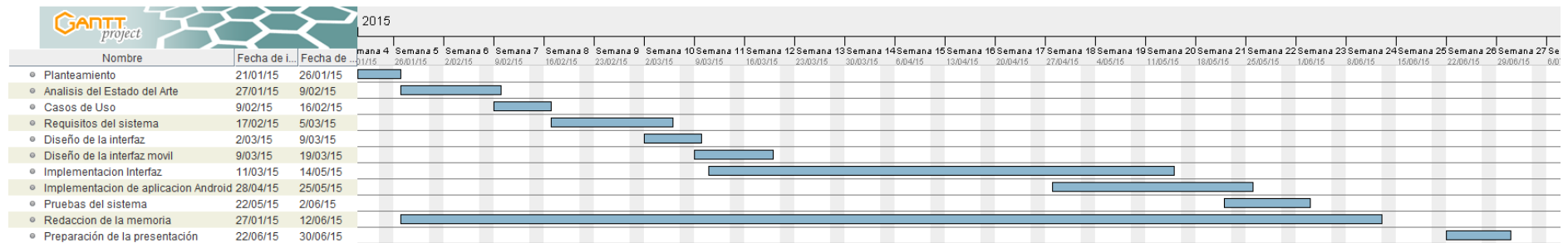


Figura 48 - Diagrama de Gantt de planificación del proyecto

El esquema muestra las diferentes tareas y sus plazos de desarrollo. Algunas se solapan pues se ha ido realizando el desarrollo de manera gradual e intercalando tareas entre sí. Para el posterior cálculo de costes, no se tienen en cuenta días festivos y fines de semana.

6.2 Cálculo de Costes

En esta sección se detalla el cálculo de costes tanto de personal como de materiales utilizado durante la realización de este proyecto.

6.3.1 Personal

El personal, como se ha comentado en el apartado de planificación, se compone del equipo de desarrollo, que en este caso está formado únicamente por el alumno.

El alumno puede desempeñar diversas funciones de programador, analista o diseñador según sea conveniente, en cada fase del proyecto.

Se muestra a continuación un resumen de los costes totales del personal derivados de la realización del proyecto.

Los cálculos de horas se realizan suponiendo una media de 4 horas por jornada de trabajo.

El cálculo según el rol desempeñado se efectúa de la siguiente manera. Se tienen en cuenta ofertas de trabajo actuales en la comunidad de Madrid para los diferentes puestos.

Por ejemplo el rol de Analista, el sueldo medio bruto al año es de 30.000€. Se calcula en base a las horas facturables al año de un trabajador medio, que son unas 1.080 horas tarifables. El coste por hora sería de unos 27,77€.

COSTE DE PERSONAL			
ROL	Horas dedicadas	Coste/hora (€/h)	Coste total (€)
Jefe de Proyecto	68 h	37,04 €/h	2.518,72 €
Diseñador	65 h	25,92 €/h	1.684,8 €
Analista	110 h	27,77 €/h	2.444,2 €
Programador Android y Java	276 h	25,77 €/h	7.112,52 €
TOTAL			13.760,24 €

Tabla 81 - Tabla de Coste de Personal

6.3.2 Material físico

Se tiene en cuenta para el cómputo del coste final el coste de los equipos hardware utilizados durante el proyecto.

A continuación se muestra un resumen de los costes totales de materiales físicos desglosados. Para el cálculo del coste se aplica la siguiente fórmula:

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

Descripción	Cantidad	Coste por unidad	Periodo de amortización (meses)	Meses de dedicación	Porcentaje de dedicación al proyecto	Coste total
Ordenador (Intel Core i5-3570k CPU 3,40 GHz, 8 GB de RAM, NVIDIA GeForce GTX 660 de 2 GB, HDD de 500 GB)	1	780,36 €	36	5	50%	54,19€
Monitor Dell U2312HM	1	99,00 €	36	5	50%	6,87€
Teclado Logitech K400	1	21,99 €	36	5	60%	1,83€
Ratón Logitech M90	1	9,99 €	36	5	60%	0,83€
Nexus 7	1	56,00€	36	5	40%	3,11€
TOTAL						66,83€

Tabla 82 - Tabla de Costes de Material Físico

6.3.3 Software

Se tiene en cuenta el coste del software utilizado durante el proyecto en caso de tenerlo. Se usa la fórmula vista anteriormente para el cálculo de costes.

Descripción	Cantidad	Coste por unidad	Periodo de amortización (meses)	Meses de dedicación	Porcentaje de dedicación al proyecto	Coste total
Licencia Microsoft Office Hogar y Pequeña Empresa 2010	1	120,48 €	36	5	70%	11,71€
Licencia Windows 7 Professional	1	259,00 €	36	5	50%	17,99€
TOTAL						29,70€

Tabla 83 - Tabla de Costes de Software

El resto de programas usados son gratuitos y por lo tanto no se reflejan en los costes.

6.3.4 Coste total

En esta sección para finalizar se suman los costes calculados anteriormente y se obtiene el coste total del proyecto.

DESCRIPCIÓN	COSTE TOTAL
Coste personal	13.760,24 €
Coste material	66,83 €
Coste software	29,70 €
Subtotal	13.856,77€
IVA (21%)	2.909,92 €
TOTAL	16.766,69 €

Tabla 84 - Tabla de Costes Finales

Se concluye que el coste total del proyecto asciende a DIECISÉIS MIL SETECIENTOS SESENTA Y SEIS EUROS CON SESENTA Y NUEVE CÉNTIMOS (16.766,69€).

7. Conclusiones y futuras líneas de mejora

En esta sección se presentarán las conclusiones derivadas del desarrollo e implementación del proyecto, así como futuras líneas de mejora para la aplicación.

7.1 Conclusiones

Los continuos cambios en nuestra sociedad llevan hacia la necesidad de potenciar y modernizar las técnicas educativas con la ayuda de las nuevas tecnologías. La comunidad docente debe adoptar nuevas herramientas de creación de contenidos electrónicos que van ganando popularidad entre los jóvenes estudiantes. El principal objetivo de este proyecto viene a suplir esta carencia, proporcionando una aplicación móvil para la realización de test presenciales y apta para cualquier tipo usuario, independientemente de su nivel de conocimientos técnicos previos.

El trabajo realizado ha sido de diversa índole, implicando utilizar diversas tecnologías y algunas nuevas como las herramientas para la creación de interfaz en Java. Esto ha supuesto horas de investigación y aprendizaje, al igual que en el desarrollo Android pues no se estaba familiarizado del todo con el lenguaje. Sin duda un aspecto favorable fue que el desarrollo en Java facilitaba el terreno para el aprendizaje, al ser un lenguaje ampliamente visto en la universidad.

Se buscaba obtener experiencia en la realización de un proyecto más independiente a lo visto en la carrera, solucionando además un problema que pueda aportar valor a la universidad. Lo cual resulta aún más gratificante pues es devolverle algo labrado gracias a los conocimientos adquiridos a lo largo de todos estos años de carrera.

Por último, en lo personal, este proyecto ha supuesto un desafío pues ha requerido superar distintos problemas en el camino y dedicarle muchas horas de trabajo, resultando al final en una experiencia enriquecedora y gratificante tanto a nivel personal como a nivel profesional.

7.2 Líneas futuras de mejora

Existe margen de mejora para la aplicación, y es algo característico de este tipo de modelo de negocios orientado a móviles. Los terminales y los sistemas operativos avanzan cada año proporcionando nuevos servicios que pueden ser integrados en las aplicaciones ya existentes mediante actualizaciones.

Se proponen una serie de sugerencias para mejorar la aplicación a largo plazo:

- **Seguridad en el intercambio de mensajes.** La seguridad del intercambio de mensajes podría verse reforzada mediante la encriptación de los mensajes. Esto aseguraría la integridad de los mensajes y evitaría algunos riesgos de ataques como **man-in-the-middle** [33]. En esta amenaza, el atacante se introduce en medio del servidor y del cliente accediendo a la información que se transmite entre ellos.

- **Versión para tablet.** Aunque es compatible con tabletas que cumplan el mínimo de la versión elegida, la interfaz no está diseñada para las dimensiones de pantalla de las tabletas de hoy en día. Se podría detectar que se está navegando en una tableta y ajustar la interfaz acorde al tamaño de pantalla para que fuese más fácil de leer y de interactuar.
- **Mejora de la usabilidad y diseño.** Las interfaces aunque efectivas siempre tienen margen de mejora, así como la usabilidad de las mismas. Esto vendría sujeto al uso de librerías más avanzadas de diseño para la elaboración de gráficos más atractivos.
- **Más funcionalidades.** Tanto el programa servidor como la aplicación pueden incorporar un mayor número de funcionalidades que faciliten aún más el trabajo del docente. Si surgen nuevos requisitos de tipos de test, se podrían actualizar las dos aplicaciones para incluirlos.

8. Bibliografía

- [1] "Mobile Marketing Statistics 2015", Danyl Bosomworth [Online] <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [2] "Number of apps available in leading app stores as of May 2015" [Online] <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>
- [3] "Moodle" [Online] https://docs.moodle.org/all/es/Acerca_de_Moodle
- [4] "Android" [Online] <http://source.android.com/source/index.html>
- [5] "Apache Harmony" [Online] <https://harmony.apache.org/index.html>
- [6] "Apache Software Foundation" [Online] <http://www.apache.org/foundation/>
- [7] "Dalvik" [Online] <https://source.android.com/devices/tech/dalvik/>
- [8] "iOS" [Online] <https://www.apple.com/ios/what-is/>
- [9] "OS X" [Online] <https://www.apple.com/es/osx/what-is/>
- [10] "Objective-C" [Online] <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
- [11] "Windows Phone" [Online] <http://www.windowsphone.com/es-es/features>
- [12] "IDC" [Online] <http://www.idc.com/about/about.jsp>
- [13] "Smartphone OS Market Share, Q1 2015" [Online] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [14] "Android Studio" [Online] <https://developer.android.com/tools/studio/index.html>
- [15] "Gradle" [Online] <https://docs.gradle.org/current/userguide/overview.html>
- [16] "Eclipse" [Online] <https://eclipse.org/org/#about>
- [17] "Android Development Tools" [Online] <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [18] "IEEE 802.11" [Online] <http://standards.ieee.org/about/get/802/802.11.html>
- [19] Sharma, Pankaj. "Evolution of Mobile Wireless Communication Networks-1G to 5G as well as Future Prospective of Next Generation Communication Network". International Journal of Computer Science and Mobile Computing, 2013.
- [20] "Evolución de las Tecnologías móviles" [Online] <http://tecnologiaparato2.blogspot.com.es/2014/02/evolucion-de-las-tecnologias-moviles.html>

- [21] “Bluetooth” [Online] <http://www.bluetooth.com/Pages/about-bluetooth-sig.aspx>
- [22] Burdett, Arnold. “BCS glossary of computing and ICT”. Capítulo A12 Networking. 2013
- [23] Burdett, Arnold. “BCS glossary of computing and ICT”. Capítulo A12 Networking. 2013
- [24] “Android Distribution Updated for February 2015” [Online] <http://www.droid-life.com/2015/02/02/android-distribution-updated-for-february-2015-lollipop-makes-first-appearance-at-1-6/>
- [25] “Java” [Online] <https://www.java.com/es/about/>
- [26] “JavaFX” [Online] <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>
- [27] “SceneBuilder” [Online] <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>
- [28] “Socrative” [Online] <http://help.socrative.com/>
- [29] “Blackboard Learn” [Online] http://www.wilmu.edu/onlinelearning/blackboard_mobile.aspx
- [30] “Métrica v3” [Online] http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.VYcL4_mvGUk
- [31] Thakur, Dinesh. “Prototyping Model in Software Engineering” [Online] <http://ecomputernotes.com/software-engineering/explain-prototyping-model>
- [32] “Diagrama de Gantt” [Online] <http://www.gantt.com/index.htm>
- [33] “Man-in-the-middle” [Online] <http://www.veracode.com/security/man-middle-attack>

Anexo A - Manual de Usuario TestUC3M

Requisitos para la instalación

Para la instalación de la aplicación TestUC3M para Android, es necesario cumplir los siguientes requisitos:

- Versión mínima de Android 2.3.3
- 512MB de RAM
- Procesador de un núcleo a 1GHz
- 6MB de espacio libre

Los requisitos anteriores aseguran un correcto funcionamiento de la aplicación. Sin embargo es posible que funcione de igual forma con menos RAM o procesador.

Instalación de la aplicación

La aplicación podrá ser descargada desde la página de la Universidad. La instalación consiste en copiar el fichero TestUC3M.apk al dispositivo móvil e instalarlo, aceptando los permisos necesarios para su instalación.

Uso de la aplicación

Al iniciar la aplicación se presenta la vista de conexión de la aplicación. El usuario debe introducir su NIA en el campo indicado y los datos del servidor al que va a conectarse. Los datos del servidor le serán proporcionados por el profesor.



Figura 49 - Ventana de inicio de TestUC3M

Si ya ha establecido una conexión con éxito previamente, los campos aparecerán ya rellenos con esos datos. Para establecer la conexión una vez introducidos todos los datos, el usuario debe pulsar el botón “Conectar”.



Figura 50 - Introducción de datos para la conexión

Todos los campos son obligatorios, si el usuario no introduce alguno de ellos la aplicación se lo recordará mediante notificación. Del mismo modo se le notificará al usuario si el NIA introducido no es válido.

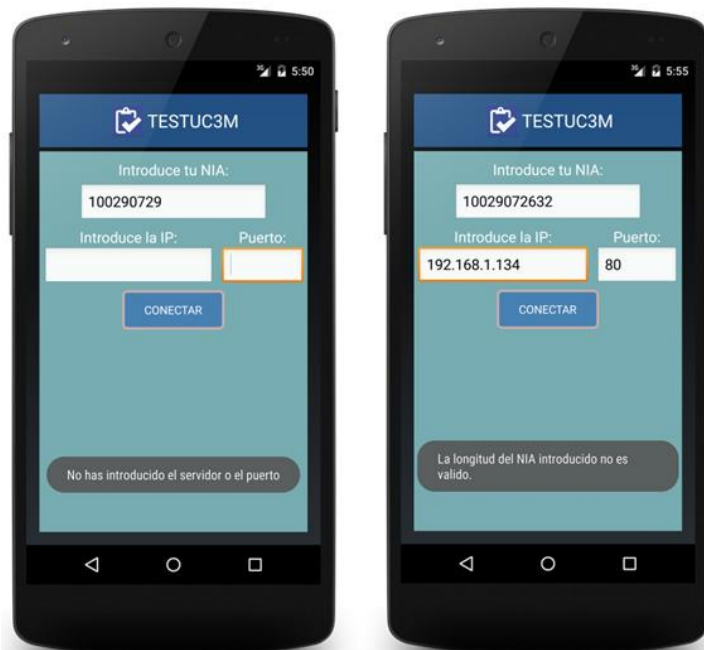


Figura 51 - Notificaciones de errores de conexión

Cuando se realice la conexión se mostrará una notificación por pantalla informando de que se ha establecido con éxito. Acto seguido se muestra la vista de espera, que informa al usuario de que está a la espera de que el profesor dé por iniciado el test. En esta pantalla el usuario solo debe esperar a que la aplicación muestre la siguiente vista.



Figura 52 - Ventana de conexión realizada y espera

Tras esta vista se pueden encontrar dos opciones, dependiendo de si el test está protegido por contraseña o no.

Si el test a realizar cuenta con una contraseña, se mostrará la vista para la introducción de la contraseña. Pulsando el botón Aceptar, la aplicación comprobará si es correcta y nos dejará continuar si lo es.



Figura 53 - Ventana de introducción de contraseña

Si no lo es, se avisa al usuario mediante una notificación. El usuario puede volver a introducirla en caso de ser incorrecta.



Figura 54 - Notificación de contraseña incorrecta

Si el test a realizar no cuenta con una contraseña o si se introduce la contraseña correcta, se avanza a la siguiente vista, que es la vista de test.

Aquí se presentan diferencias relacionadas con la configuración del test. Se tienen diferentes casos en los que la vista y el funcionamiento varían según el caso.

Caso 1. Test con respuesta única

En este caso el test es de tipo respuesta única. Esto quiere decir que solo se podrá seleccionar una respuesta o ninguna en cada pregunta.

En la pantalla, se representa en la parte superior el nombre del test y la pregunta en la que se encuentra del número total de preguntas. Una fila por debajo se encuentran los botones de navegación Atrás y Siguiente para la navegación entre preguntas. El enunciado de cada pregunta se representa en medio de la pantalla.

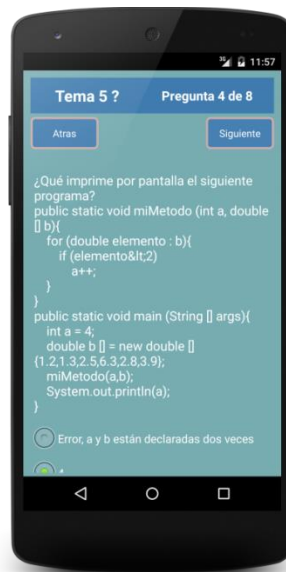


Figura 55 - Vista general de test de respuesta única

Las opciones disponibles para contestar se muestran debajo del enunciado. El usuario en este caso solo podrá marcar una respuesta o ninguna en cada pregunta. Simplemente tiene que seleccionar la respuesta que desee y el indicador de respuesta elegida cambiará.

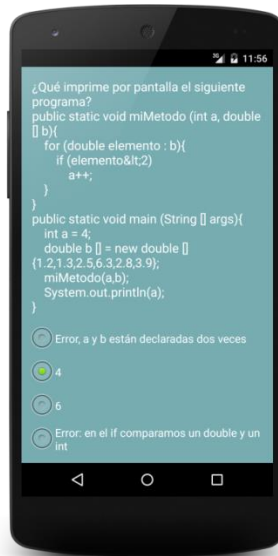


Figura 56 - Selección de respuesta

Cuando esté satisfecho con sus elecciones, el usuario debe ir a la última pregunta y pulsar en el botón Finalizar. Esto dará por terminado el test.

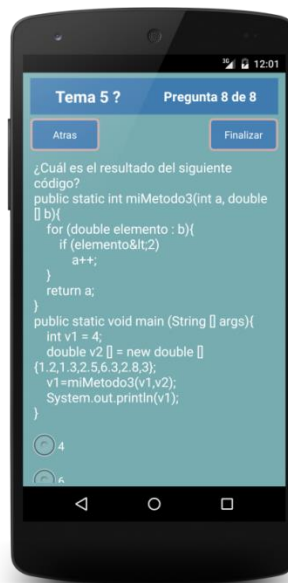


Figura 57 - Vista de botón finalizar test de respuesta única

Este es el caso básico de un test con respuesta única. Pero dentro de este caso pueden darse otras circunstancias que se explican a continuación.

Caso 1.1 Test con límite de tiempo

En el caso de que el test tenga una restricción de tiempo para responder, el escenario es prácticamente el mismo. El usuario solo puede responder una pregunta dentro del tiempo señalado. Si se acaba el tiempo se guardará como respuesta la última que haya marcado. Para finalizar el test manualmente, el alumno podrá ir a la última pregunta y pulsar el botón Finalizar.

En la vista de test, el contador de tiempo se localiza entre los botones de navegación de preguntas.

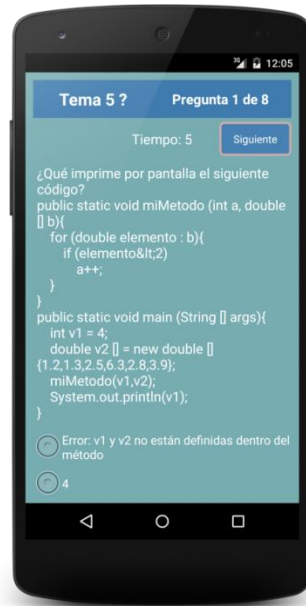


Figura 58 - Vista de tiempo límite

Caso 1.2 Sin posibilidad de modificación de respuesta

La restricción aplicada en este caso impide al usuario modificar la respuesta una vez haya respondido. Las preguntas se realizan en orden y el profesor decide cuando avanzar a la siguiente.

El cambio principal en esta vista está en los botones de navegación. Se cuenta con un botón Enviar para enviar la respuesta elegida de esa pregunta. Una vez enviada, se muestra un mensaje de espera hasta que llegue la siguiente.



Figura 59 - Vista de envío de respuesta única, pregunta a pregunta

Caso 2. Test con múltiples respuestas

En este caso el test puede tener una o varias respuestas válidas. El usuario podrá marcar el número de respuestas que crea conveniente o ninguna.

Se representa en la parte superior el nombre del test y la pregunta en la que se encuentra del número total de preguntas. Una fila por debajo se encuentran los botones de navegación Atrás y Siguiente para navegar por las preguntas del test. El enunciado de cada pregunta se representa en medio de la pantalla.

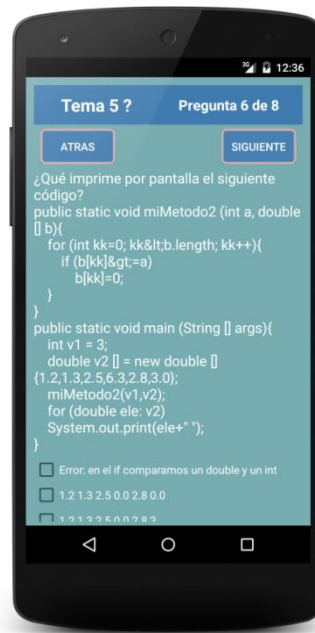


Figura 60 - Vista general de test de respuesta múltiple

Las opciones disponibles para contestar se muestran debajo del enunciado. El usuario en este caso podrá marcar una o varias respuestas o no marcar ninguna, en cada pregunta.



Figura 61 - Marcar una o más opciones

Cuando esté satisfecho con sus elecciones, el usuario debe ir a la última pregunta y pulsar en el botón Finalizar. Esto dará por terminado el test.

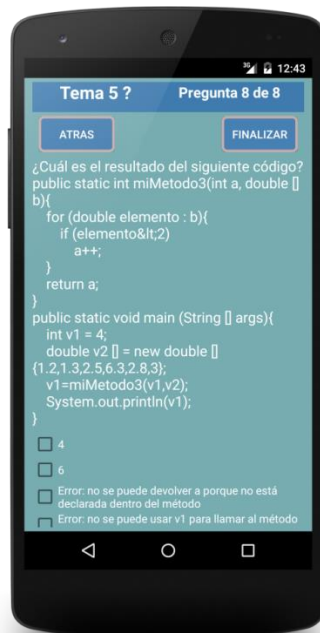


Figura 62 - Vista de botón finalizar test de respuesta múltiple

Este es el caso básico de un test con respuesta múltiple. Pero dentro de este caso pueden darse otras circunstancias que se explican a continuación.

Caso 2.1 Test con límite de tiempo

En el caso de que el test tenga una restricción de tiempo para responder, el escenario es prácticamente el mismo. El usuario solo puede responder una pregunta dentro del tiempo señalado. Si se acaba el tiempo se guardará como respuesta la última que haya marcado. Para finalizar el test manualmente, el alumno podrá ir a la última pregunta y pulsar el botón Finalizar.

En la vista de test, el contador de tiempo se localiza entre los botones de navegación de preguntas.

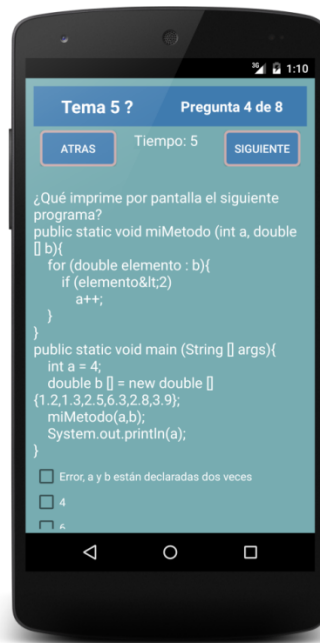


Figura 63 - Vista de tiempo límite en test de respuesta múltiple

Caso 2.2 Sin posibilidad de modificación de respuesta

La restricción aplicada en este caso impide al usuario modificar la respuesta una vez haya respondido. Las preguntas se realizan en orden, y el profesor decide cuando avanzar a la siguiente.

El cambio principal en esta vista está en los botones de navegación. Se cuenta con un botón Enviar para enviar la respuesta elegida de esa pregunta. Una vez enviada, se muestra un mensaje de espera hasta que llegue la siguiente.



Figura 64 - Vista de envío de respuesta múltiple, pregunta a pregunta

Finalizando el test

Tras terminar el test, sea del tipo que sea, se muestra la vista final.



Figura 65 - Vista de finalización de test

Esta vista informa al usuario de que se ha finalizado con éxito el test y muestra un botón para salir de la aplicación. Una vez pulsado se cierra la aplicación.

Anexo B - Manual de Usuario TestUC3M (Escritorio)

Requisitos para la instalación

Para la instalación de la aplicación TestUC3M, es esencial tener instalado en el ordenador lo siguiente:

- Sistema Operativo: Windows, Mac o Linux
- Java 8 instalado
- Pantalla de resolución mínima 1024x768 px

Instalación de la aplicación

No es necesaria su instalación pues se puede ejecutar directamente a través de un fichero portable.

Uso del programa

A continuación se detalla el uso del programa.

Vista principal

Tras ejecutar la aplicación muestra la vista principal del programa.

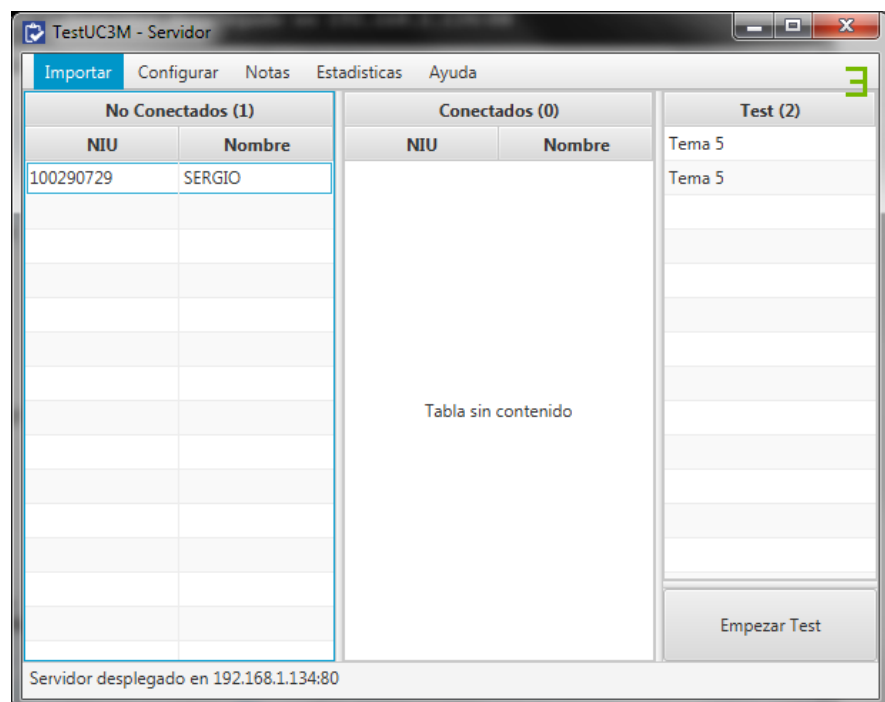


Figura 66 - Vista principal servidor

Aquí están presentes todas las funcionalidades del programa. Se observan también las tablas de información como la tabla de usuarios conectados y no conectados, o la lista de test importados en la aplicación.

Se detallan a continuación las funcionalidades del programa.

Importar ficheros de alumnos o test

Mediante la opción **Importar**, podemos seleccionar si queremos importar una lista de alumnos o de test.

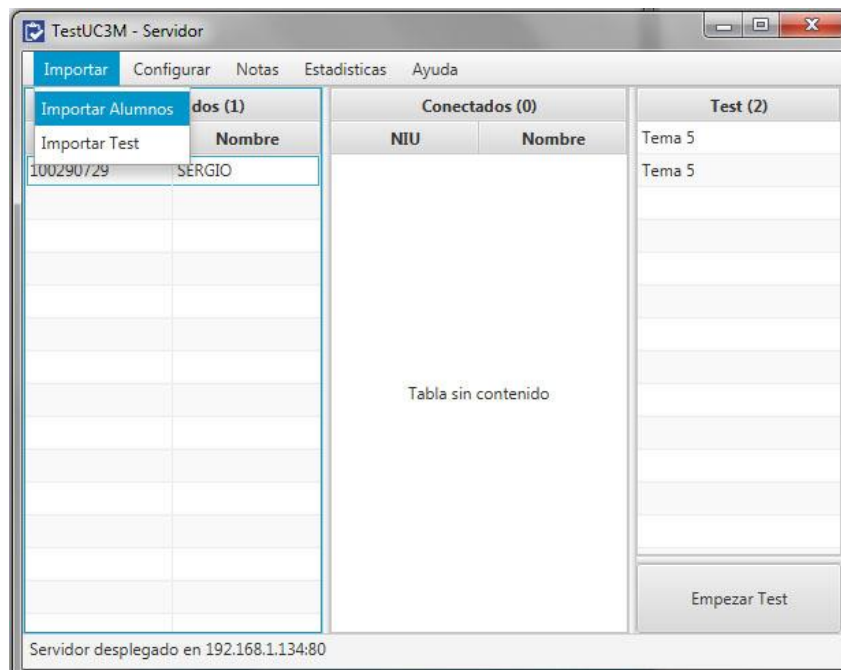


Figura 67 - Opción de importar

Se abre una ventana de navegación que permite seleccionar el fichero del que se quieren importar datos, ya sean alumnos o test. El formato de los ficheros soportados se muestra abajo a la derecha.

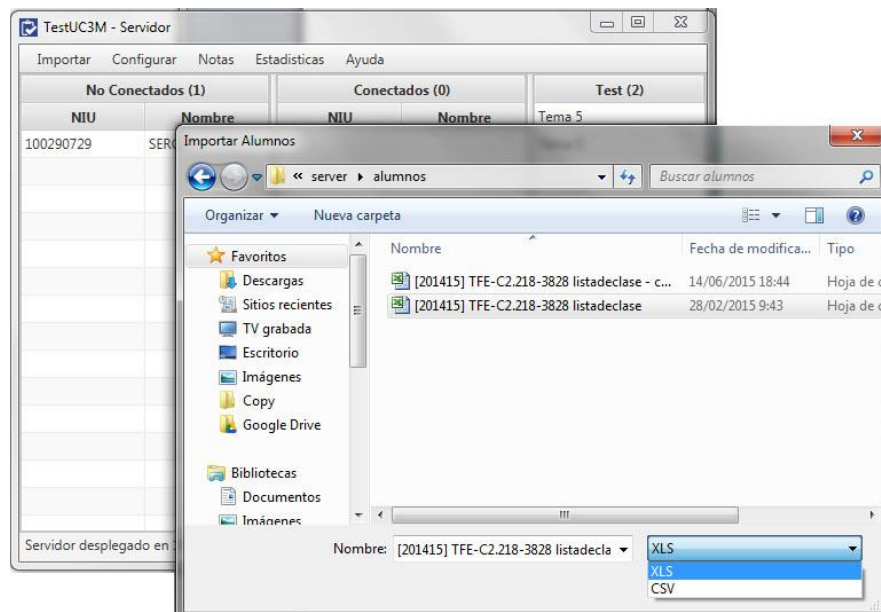


Figura 68 - Ventana de importación

Una vez seleccionado el fichero, el programa importa la selección y la muestra por pantalla mediante las tablas de alumnos y de test.

Configurar test

Para configurar un test en primer lugar se debe seleccionar un test de la lista, de lo contrario se avisara al usuario de que no ha seleccionado ninguno.

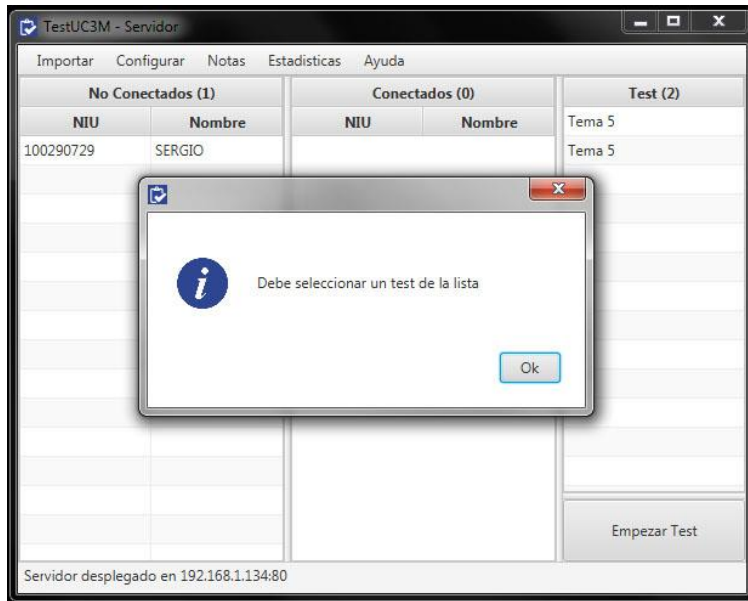


Figura 69 - Aviso de selección de test para configurar

Cuando se ha seleccionado un test, se pulsa en Configurar y se abre la ventana para la configuración del test.

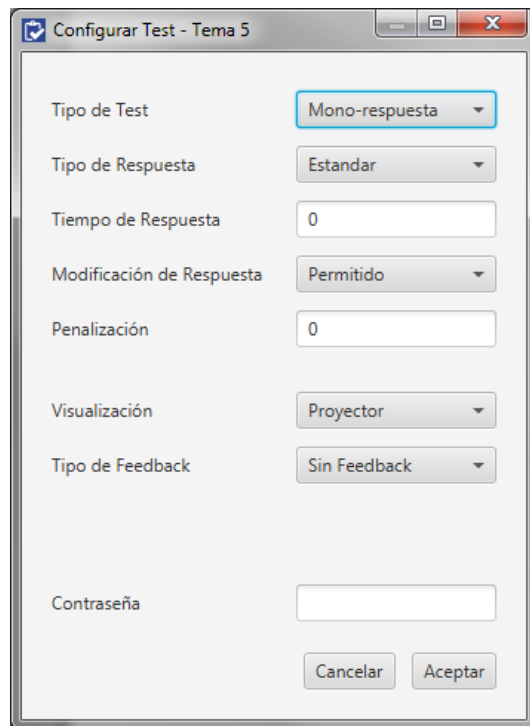


Figura 70 - Ventana de configuración de test

Aquí se pueden ver todas las diferentes opciones que se pueden configurar.

El tipo de test indica el tipo de respuesta posible que soporta el test. Las opciones son Mono-respuesta y Multi-respuesta.

A screenshot of a form field labeled 'Tipo de Test'. It is a dropdown menu with 'Mono-respuesta' selected and highlighted in blue. Below it, the label 'Tipo de Respuesta' is visible, with a dropdown menu showing 'Multi-respuesta' as an option.

Figura 71 - Selección tipo de test

El tipo de respuesta indica si es una respuesta Estándar o de otro tipo. En esta versión solamente se tiene respuesta Estándar.

En el campo tiempo de respuesta el usuario puede introducir un valor numérico que se traduce en segundos. Este tiempo será el que tenga el alumno por respuesta para contestar. No admite valores negativos. Si se deja en 0 no habrá tiempo límite para la pregunta.

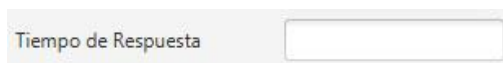
A screenshot of a form field labeled 'Tiempo de Respuesta'. It is an empty text input box with a light gray border.

Figura 72 - Selección de tiempo de respuesta

El campo de modificación de respuesta permite elegir entre Permitido y Prohibido.

A screenshot of a form field labeled 'Modificación de Respuesta'. It is a dropdown menu with 'Permitido' selected and highlighted in blue. Below it, the label 'Penalización' is visible, with a dropdown menu showing 'Prohibido' as an option.

Figura 73 - Selección de modificación de respuesta

Si se marca permitido el alumno podrá navegar entre las respuestas con total libertad y respondiéndolas en el orden que desee.

Si se marca prohibido el test se realizara en orden pregunta a pregunta. El paso lo marcará el profesor pasando a la siguiente pregunta cuando todos los alumnos hayan contestado o cuando le parezca oportuno.

El campo penalización permite al usuario establecer la penalización por pregunta que crea oportuna. Esto afectara al cálculo de notas final.

A screenshot of a form field labeled 'Penalización'. It is a text input box containing the value '0.0'.

Figura 74 - Selección de penalización

El campo visualización permite decidir si el enunciado del test se podrá ver solo en el proyector a través del ordenador donde se ejecute el servidor, o si se podrá ver en el dispositivo móvil del alumno, o en ambos sitios.

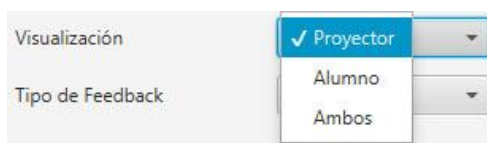


Figura 75 - Selección de tipo de visualización

El campo contraseña se utiliza para proteger el test con una contraseña, definida en este campo. El alumno solo podrá ver el test cuando introduzca la contraseña establecida. Si se deja en blanco no habrá contraseña.

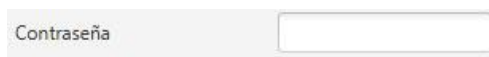


Figura 76 - Campo de introducción de contraseña

Una vez definidas estas combinaciones, el usuario puede guardarlas pulsando el botón Aceptar, o descartarlas pulsando el botón Cancelar. Ambas opciones devuelven a la interfaz principal. Si se han hecho los cambios, se muestra el test configurado con una marca al lado de su nombre.

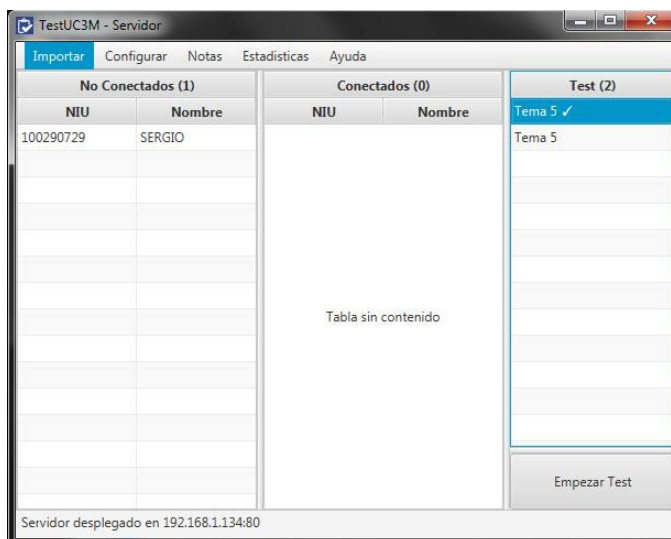


Figura 77 - Vista de test configurado

Si se desea volver a configurar el test, se vuelve a seleccionar Configurar y las opciones previamente introducidas se mostrarán en la ventana.

Iniciar test

Una vez configurado el test, se puede proceder a su inicio.

Antes de comenzar el programa se asegura de que todos los alumnos de la lista estén conectados, si no es así, avisará al usuario y le permitirá decidir si quiere continuar o no.

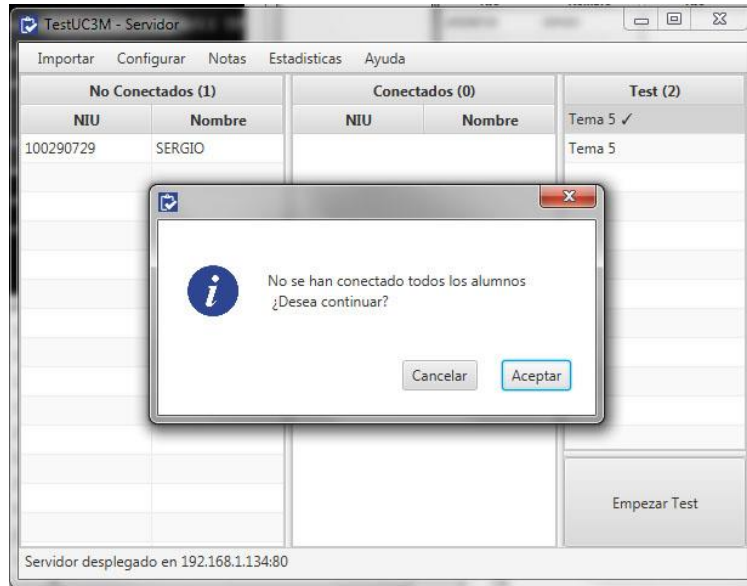


Figura 78 - Aviso de inicio de test

Una vez se decide continuar, se muestra la ventana de navegación por preguntas del test.

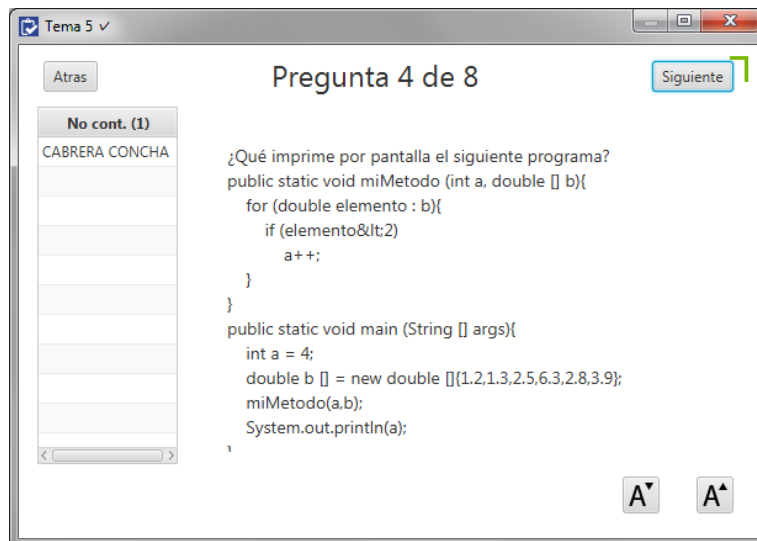


Figura 79 - Vista general de navegación de test

En esta ventana, en la parte superior se encuentran los botones de navegación por preguntas Atras y Siguiente. En medio está la información de la pregunta en la que se está del total de preguntas.

A la izquierda se encuentra la tabla de alumnos que aún no han completado el test.

En medio se encuentra el enunciado de la pregunta. Debajo a la derecha están los botones para aumentar o disminuir el tamaño del texto del enunciado.

Cuando se llega a la última pregunta, se puede dar por finalizado el test mediante el botón Acabar.

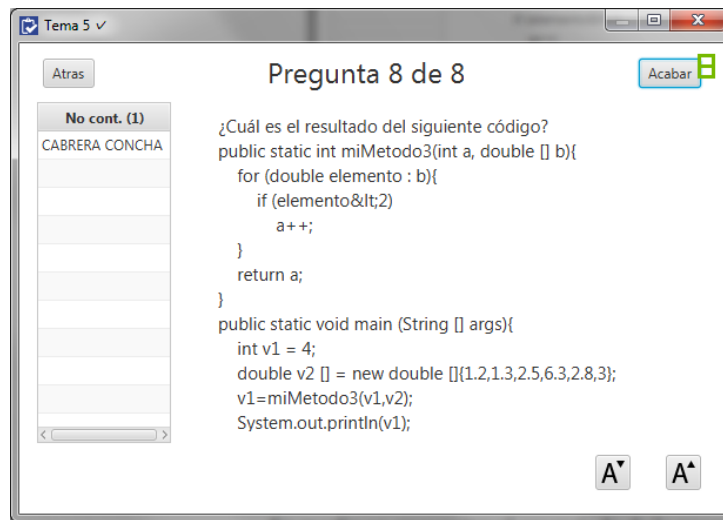


Figura 80 - Vista de botón Acabar

Al intentar acabar el test ya sea a través del botón Acabar o cerrando la ventana del test, una ventana emergente preguntará al usuario si realmente quiere dar por terminado el test.

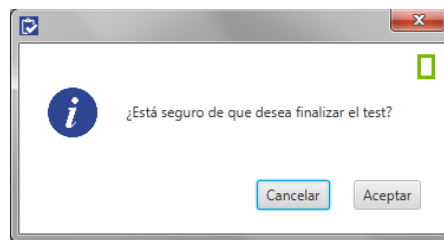


Figura 81 - Ventana de confirmación de finalización de test

Cuando pulse Aceptar, se cerrará la ventana y se dará por finalizado el test.

Notas

Una vez realizado el test, el usuario puede consultar los resultados en la opción del menú Notas.



Figura 82 - Opción de Notas

Hay dos opciones para la obtención de las notas.

Exportar notas

Si selecciona la opción Exportar, las notas se exportarán en un archivo de tipo Excel o de tipo texto plano, a elección del usuario.

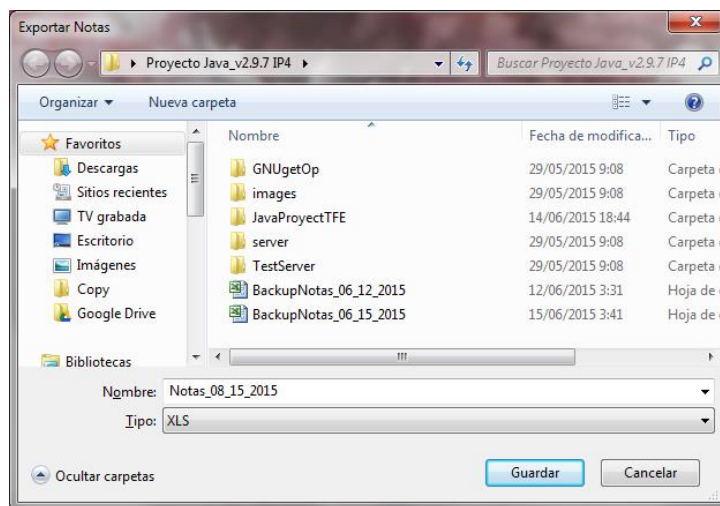


Figura 83 - Ventana de exportación de notas

Una vez se pulsa Guardar, el fichero se crea y se cierra la ventana.

Enviar por Correo

Si selecciona la opción Enviar por Correo, las notas se enviarán por correo a la cuenta de Gmail que especifique. Previamente tendrá que habilitar el acceso a la cuenta de esta aplicación en la configuración de su cuenta Gmail. Se abre una ventana para pedir las credenciales de la cuenta.

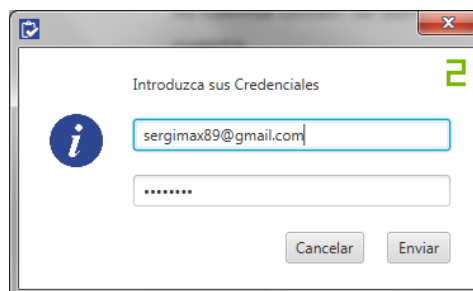


Figura 84 - Ventana de introducción de datos de correo electrónico

Cuando se pulsa en Enviar el fichero de notas se envía al correo introducido y se cierra la ventana. Si se pulsa en Cancelar, se cierra la acción y no se envía nada.

Estadísticas

La opción estadísticas del menú permite acceder a la opción Ver Gráfico para ver el grafico de notas obtenido en el último test.



Figura 85 - Opción de Estadísticas

Si no se hubiese realizado un test con anterioridad del que sacar notas para el grafico, se avisará al usuario de que no está disponible en este momento.

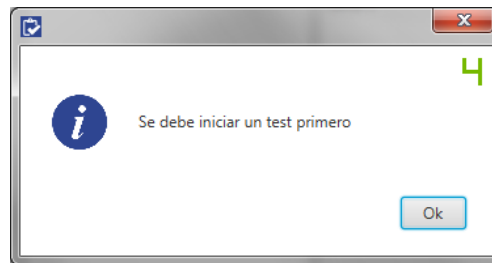


Figura 86 - Aviso de estadística inexistente

Cuando se pulsa en Ver gráfico, se abre una ventana nueva con el grafico de las notas.

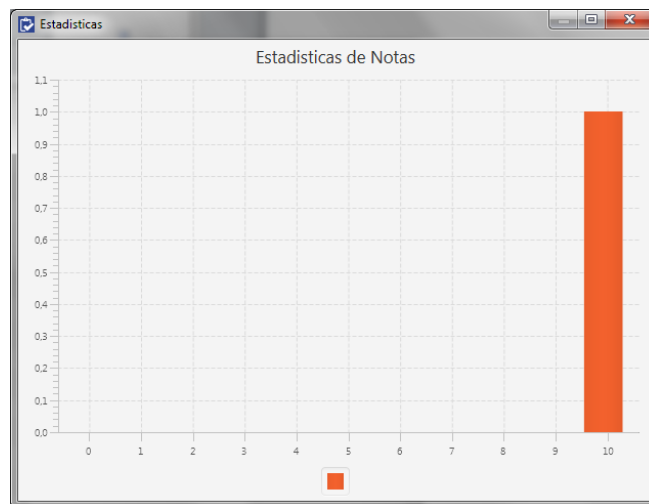


Figura 87 - Ventana de gráfico de barras

En la parte inferior tenemos la calificación en una escala del 0 al 10. Cada barra representa el número de alumnos que han sacado esa nota en el último test realizado. Para salir de esta ventana simplemente basta con cerrarla.