



Universidad
Carlos III de Madrid

Departamento de Informática

TRABAJO FIN DE GRADO

APLICACIÓN MÓVIL PARA FOMENTAR LA SEGURIDAD VIAL EN ADOLESCENTES Y JÓVENES

Autor: Roberto Esteban Santiago

Tutor: Javier García Guzmán

Cotutor: Alberto Heredia García

Leganés, Junio de 2012

Título: Aplicación móvil para fomentar la seguridad vial en adolescentes y jóvenes

Autor: Roberto Esteban Santiago

Director: Javier García Guzmán

Cotutor: Alberto Heredia García

EL TRIBUNAL

Presidente: Alejandro Calderón Mateos

Vocal: Francisco Javier García Blas

Secretario: Jesús Salas Martínez

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día 6 de Julio de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

*“No preguntes que pueden
hacer tus compañeros por ti.
Pregunta qué puedes hacer tú
por tus compañeros”*

Magic Johnson

Agradecimientos

Gracias a este apartado, voy a poder dar las gracias a todas las personas que me han ayudado en una de las etapas más importantes de mi vida, y por eso no me quiero referir solo a esta fase de la universidad, sino que todas estas personas me han ayudado desde el mismo momento que empezó mi formación educativa.

En primer lugar, quiero agradecer a mi padre, mi madre y mi hermano, su apoyo incondicional que me han dado durante toda mi vida, tanto en el aspecto educativo, como en el aspecto personal. Ellos me han enseñado todo lo posible para llegar hasta donde he llegado, porque estas tres personas tan importantes en mi vida luchan y luchan cada día para conseguir sus objetivos, que aunque parezcan dispares, todos llevan a lo mismo, a la unión familiar. Gracias a este esfuerzo que hacen día a día, me han hecho a mí luchar hasta conseguir la meta que deseaba. Sois los mejores.

Seguidamente, y no con menos mérito, me gustaría agradecerle, con todo el apoyo y la ayuda que me ha proporcionado, a mi novia, Soledad. Aunque ella no estuviera desde los inicios mencionados anteriormente, durante esta última fase de la universidad, ha estado ahí como la que más, ayudándome en lo que necesitara, dándome apoyo y sobretodo estando a mi lado. Muchas gracias por convertirte en una pieza totalmente necesaria en mi vida, sin ti, estos últimos cuatro años, no hubiera sido lo mismo.

También tengo que agradecer mucho a mis amigos y a mi familia, ellos han estado siempre que los he necesitado, aunque no tanto en los momentos educativos, pero si en los momentos personales que tanta falta hacen para poder avanzar hasta cumplir el objetivo deseado. Muchos de estos amigos, o “familia indirecta” me han tratado como si de un hijo o un hermano se tratase, a ellos les quiero hacer una mención especial por hacerme tan feliz durante todo este tiempo.

Como no, agradecer a mis compañeros de universidad, que con los cuales, en los últimos cuatro años nos hemos tirado mucho tiempo juntos, realizando prácticas, estudiando, y sobretodo pasándolo bien, porque aunque haya habido épocas de mucho agobio, siempre había alguien que conseguía sacar una sonrisa a todos y nos olvidáramos por momento de dicho agobio. Gracias por toda vuestra ayuda.

Por último dar las gracias a mi tutor, Javier, y a todos sus ayudantes como son Juan Carlos, Alberto, etc. A Javier, le doy las gracias por haberme brindado la posibilidad de realizar este gran proyecto, porque gracias a su insistencia y apoyo, durante todo el proyecto, hemos hecho que este trabajo haya sido exitoso. Como no, agradecer a todas las personas que han estado en el laboratorio ayudándome sin esperar nada a cambio, además de haber podido pasar buenos momentos en él.

Espero no dejarme a nadie, por ello quiero agradecer a todas esas personas, que están o por desgracia ya no, que han deseado lo mejor para mí y siempre han estado a mí lado. Sin todos vosotros nada hubiera sido posible. Me acordaré siempre de todos vosotros.

Resumen

Hoy en día, los teléfonos inteligentes forman parte de la vida de todo tipo de personas, desde los usuarios más jóvenes a los más adultos, y con objetivos diferentes, como pueden ser por cuestiones de trabajo, estudio, ocio o incluso seguridad.

Debido al gran avance de las tecnologías y a los nuevos diseños de terminales ligeros e inalámbricos, cada vez son más las personas que utilizan el teléfono móvil o las tablets en lugar del ordenador, principalmente para el acceso a aplicaciones de ocio y estudio, así como para la navegación en Internet.

De este modo, los mercados virtuales de estos teléfonos móviles inteligentes lanzan una gran variedad de aplicaciones destinadas a diferentes tipos de usuarios, con el propósito de conseguir un beneficio gracias a la descarga de dichas aplicaciones. Dicho beneficio puede ser directo, que consistiría en el cobro de la aplicación por cada descarga de la misma. O por otro lado, el beneficio puede ser indirecto, donde la aplicación puede ser gratuita o de coste muy bajo pero incluyendo publicidad, o simplemente que la aplicación proponga una idea interesante y bien desarrollada y aumente el prestigio de la empresa que la desarrolló.

Siguiendo estos argumentos, y teniendo en cuenta el gran número de personas que utilizan los dispositivos iPhone y/o iPod, el presente trabajo muestra el desarrollo de una aplicación para dichos dispositivos. La aplicación está orientada a todas aquellas personas que se encuentran en un periodo de estudio del permiso de conducción de automóviles. Se ha desarrollado una aplicación con una interfaz sencilla y amigable, que permita a los usuarios aprender las normas de circulación con el fin de examinarse del test que demuestra que una persona es apta para continuar con las pruebas prácticas del permiso de conducción de automóviles.

De este modo, el usuario de la aplicación puede evitar tener que desplazarse a la autoescuela, así como tener que estar en frente de un ordenador. Simplemente accediendo a la aplicación desde su teléfono móvil iPhone o desde su dispositivo iPod, podrá estudiar todos los temas teóricos que enseñan en las autoescuelas con la posibilidad de ver una breve descripción de cada una de ellas, así como un vídeo que orientará al usuario a entender el contenido de cada tema. Una vez aprendida la lección, se podrán realizar test relacionados con aquello que se ha estudiado de modo que el usuario de la aplicación estará preparado para realizar el examen teórico del permiso de conducción de automóviles.

Abstract

Nowadays, smartphones are part of life for every kind of people, from youngest to the adult users, and with different objectives, such as work issues, study, leisure or even security.

Due to great advances in technology and new terminal designs (light and wireless mobile terminals), more and more people using the mobile phone or tablets instead of computers, primarily for access to leisure and study applications and for Internet browsing.

Thus, virtual markets of these smartphones launch a variety of applications for different types of users, in order to get a benefit by downloading these applications. This benefit can be a direct benefit, which would be based in charging the application by either download on the Internet. On the other hand, the benefit may be indirect benefit, where the application can be free or very low cost but including advertising or simply the application proposes an interesting and well developed idea and this causes an increase in the prestige of the company that developed it.

According to these arguments, and taking into account the large number of people using the iPhone and iPod, this paper shows the development of an application for such devices. The application is aimed at people who are in a study period of car driving license. It has been developed an application with a simple and friendly interface, allowing users to learn the traffic rules in order to take the exam that demonstrate a person is able to continue with the practical tests of car driving license.

In this way the user could avoid having to go to driving school as well as staying in front of a computer. Simply accessing the application from his iPhone or from his iPod, user is able to study every theoretical lesson teaching in driving schools. Application provides a brief description of each lesson, and a video that will guide the user understand the contents of each lesson. After learning the lesson, tests can be made regarding what has been studied, so the user of the application will be ready for the theoretical examination of car driving license.

Índice general

Agradecimientos.....	6
Resumen	8
Abstract	9
Índice de tablas.....	15
Índice de ilustraciones.....	18
1. Introducción	23
1.1. Motivación	23
1.2. Objetivos	25
1.3. Etapas del proyecto	26
1.4. Glosario de términos.....	27
1.5. Organización Documento	28
2. Estado del Arte	31
2.1. Análisis de aplicaciones móviles y web.....	31
2.1.1. Aplicaciones Relevantes	31
2.1.2. Aplicaciones descartadas.....	38
2.1.3. Conclusión de aplicaciones analizadas.....	40
2.1.4. Criterios de Búsqueda	40
2.2. Entorno de Desarrollo Móvil.....	41
2.2.1. Sistema Operativo	41
2.2.2. Lenguaje de Programación	42
2.2.3. API a utilizar	43
2.2.4. Entorno de desarrollo.....	43
2.3. Desarrollo Ágil.....	44
2.3.1. SCRUM	45
3. Especificación de Requisitos	54
3.1. Historias de Usuario.....	54
3.2. Casos de Uso	60

3.3.	Requisitos de Restricción	65
4.	Diseño.....	68
4.1.	Alternativas de diseño	68
4.1.1.	Alternativa de diseño 1.....	68
4.1.2.	Alternativa de diseño 2.....	69
4.1.3.	Alternativa de diseño 3.....	70
4.1.4.	Alternativa de Diseño 4	71
4.1.5.	Elección de alternativa de diseño.....	72
4.2.	Paper prototype.....	73
4.2.1.	Diseño 1	73
4.2.2.	Diseño 2	76
4.2.3.	Elección de diseño	79
4.3.	Diagrama de Componentes	80
4.4.	Diagrama de Clases.....	82
4.5.	Diagramas de Secuencia	95
4.6.	Tarjetas CRC.....	101
4.7.	Base de datos.....	105
5.	Pruebas del sistema	109
6.	Planificación	119
6.1.	Planificación Inicial.....	119
6.2.	Planificación Final	121
7.	Presupuesto.....	127
7.1.	Presupuesto Inicial.....	127
7.1.1.	Coste Personal	127
7.1.2.	Coste Elementos software y hardware	128
7.1.3.	Coste Total del Proyecto.....	129
7.2.	Presupuesto Final.....	130
7.2.1.	Coste Personal	130
7.2.2.	Coste Elementos software y hardware	130
7.2.3.	Coste Total del Proyecto.....	131
7.3.	Herramientas Hardware	132

8.	Conclusiones y Líneas Futuras.....	136
8.1.	Conclusiones	136
8.2.	Líneas Futuras	138
8.2.1.	Vídeos en servidores	138
8.2.2.	Parsear datos	139
8.2.3.	Otro objetivo de la aplicación	139
8.2.4.	Aplicación para el iPad.....	140
8.2.5.	Otros sistemas operativos	140
9.	Bibliografía	143
9.1.	Libros de Consulta.....	143
9.2.	Recursos electrónicos	143

Índice de tablas

Tabla 1. Historias de Usuario	54
Tabla 2. Tareas HU-1.....	55
Tabla 3. Tareas HU-2.....	56
Tabla 4. Tareas HU-3.....	57
Tabla 5. Tareas HU-4.....	58
Tabla 6. Tareas HU-5.....	59
Tabla 7. Tabla modelo Casos de Uso	60
Tabla 8. Caso de uso Ver tema	61
Tabla 9. Caso de uso Ver definición tema	62
Tabla 10. Caso de uso Ver vídeo del tema.....	62
Tabla 11. Caso de uso Contestar Test.....	63
Tabla 12. Caso de uso Ver Resultados	64
Tabla 13. Caso de uso Ver Resultados Anteriores	64
Tabla 14. Caso de uso Compartir en redes sociales	64
Tabla 15. Tabla tipo Requisitos de Restricción	65
Tabla 16. Requisito de Restricción 1.....	65
Tabla 17. Requisito de Restricción 2.....	65
Tabla 18. Requisito de Restricción 3.....	65
Tabla 19. Clase ViewController	83
Tabla 20. Clase DetailViewController	85
Tabla 21. Clase TestResultViewController	87
Tabla 22. Clase TestViewController	89
Tabla 23. Clase ResultViewController.....	91
Tabla 24. Clase CorrectViewController.....	93
Tabla 25. Clase HelpViewController	94
Tabla 26. Clase SingletonClass	94
Tabla 27. Clase SqlLiteController	94
Tabla 28. Tarjeta CRC ViewController.....	101
Tabla 29. Tarjeta CRC DetailViewController	101
Tabla 30. Tarjeta CRC TestResultViewController.....	102
Tabla 31. Tarjeta CRC TestViewController.....	102
Tabla 32. Tarjeta CRC ResultViewController	103
Tabla 33. Tarjeta CRC CorrectViewController	103
Tabla 34. Tarjeta CRC HelpViewController	104
Tabla 35. Tarjeta CRC SingletonClass.....	104

Tabla 36. Tarjeta CRC SqlLiteController	104
Tabla 37. Tabla modelo pruebas de sistema	109
Tabla 38. Pruebas de sistema HU-1	110
Tabla 39. Pruebas de sistema HU-2	111
Tabla 40. Pruebas de sistema HU-3	113
Tabla 41. Pruebas de sistema HU-4	114
Tabla 42. Pruebas de sistema HU-5	116
Tabla 43. Presupuesto inicial - Coste personal	128
Tabla 44. Presupuesto inicial - Coste elementos sw y hw	128
Tabla 45. Presupuesto inicial - Coste Total	129
Tabla 46. Presupuesto final - Coste personal	130
Tabla 47. Presupuesto inicial - Coste elementos sw y hw	131
Tabla 48. Presupuesto final - Coste Total	131

Índice de ilustraciones

Ilustración 1. Seguridad Vial para jóvenes.....	32
Ilustración 2. Toca la señal.....	32
Ilustración 3. Toca la señal func-1	33
Ilustración 4. Toca la señal func-2	33
Ilustración 5. Toca la señal func-3	33
Ilustración 6. Toca la señal func-4	33
Ilustración 7. Toca la señal func-5	33
Ilustración 8. Toca la señal func-6	33
Ilustración 9. Toca la señal func-7	33
Ilustración 10. Toca la señal func-8	33
Ilustración 11. Toca la señal func-9	33
Ilustración 12. TargetBox Tráfico.....	34
Ilustración 13. TargetBox Tráfico func-1.....	34
Ilustración 14. TargetBox Tráfico func-2.....	34
Ilustración 15. El velocímetro	34
Ilustración 16. El velocímetro func-1.....	35
Ilustración 17. El velocímetro func-2.....	35
Ilustración 18. El velocímetro func-3.....	35
Ilustración 19. El velocímetro func-4.....	35
Ilustración 20. El velocímetro func-5.....	36
Ilustración 21. El velocímetro func-6.....	36
Ilustración 22. RACE	36
Ilustración 23. RACE func-1.....	37
Ilustración 24. RACE func-2.....	37
Ilustración 25. RACE func-3.....	37
Ilustración 26. RACE func-4.....	37
Ilustración 27. CEA	38
Ilustración 28. DGT	38
Ilustración 29. De dos en dos.....	39
Ilustración 30. ConfortAuto	39
Ilustración 31. CAS	39
Ilustración 33. Arquitectura iOS.....	41
Ilustración 32. iOS 5	41
Ilustración 34. Xcode	43
Ilustración 35. Scrum	45

Ilustración 36. Ciclo de Vida Scrum	46
Ilustración 37. Roles Scrum.....	49
Ilustración 38. Diagrama de Casos de Uso	60
Ilustración 39. Alternativa de Diseño 1.....	68
Ilustración 40. Alternativa de Diseño 2.....	69
Ilustración 41. Alternativa de diseño 3	70
Ilustración 42. Alternativa de diseño 4	71
Ilustración 43. Diseño 1 Vista logo.....	73
Ilustración 44. Diseño 1 Vista Bienvenida.....	73
Ilustración 45. Diseño 1 Vista Temas	74
Ilustración 46. Diseño 1 Vista Detalle	74
Ilustración 47. Diseño 1 Vista Pregunta 1	74
Ilustración 48. Diseño 1 Vista Última pregunta	74
Ilustración 49. Diseño 1 Vista resultados.....	75
Ilustración 50. Diseño 2 Vista Logo.....	76
Ilustración 51. Diseño 2 Vista Bienvenida.....	76
Ilustración 52. Diseño 2 Vista Temas	76
Ilustración 53. Diseño 2 Vista Detalle	76
Ilustración 54. Diseño 2 Vista Resultados anteriores	77
Ilustración 55. Diseño 2 Vista Pregunta 1	77
Ilustración 56. Diseño 2 Vista Ayuda	77
Ilustración 57. Diseño 2 Vista Pregunta 14.....	77
Ilustración 58. Diseño 2 Vista Pregunta final.....	78
Ilustración 59. Diseño 2 Vista Corrección	78
Ilustración 60. Diseño 2 Vista Corrección Pregunta Acertada.....	78
Ilustración 61. Diseño 2 Vista Corrección Pregunta Fallada.....	78
Ilustración 62. Diagrama de componentes.....	80
Ilustración 63. Diagrama de Clases.....	82
Ilustración 64. Diagrama de secuencia HU – 1	95
Ilustración 65. Diagrama de secuencia HU – 2	96
Ilustración 66. Diagrama de Secuencia HU - 3.....	97
Ilustración 67. Diagrama de secuencia HU – 4	98
Ilustración 68. Diagrama de secuencia HU – 5 Funcionalidad Twitter.....	99
Ilustración 69. Diagrama de secuencia HU – 5 Funcionalidad Facebook.....	99
Ilustración 70. Diagrama de secuencia HU – 5 Funcionalidad email.....	100
Ilustración 71. Diagrama entidad-relación	105
Ilustración 72. Modelo relacional	106
Ilustración 73. Planificación Inicial.....	120
Ilustración 74. Planificación Final 1	123
Ilustración 75. Planificación Final 2	124
Ilustración 76. Vídeos en servidores.....	138

Ilustración 77. Otro objetivo de la aplicación.....	139
Ilustración 78. Aplicación para el iPad.....	140
Ilustración 79. Otros sistemas operativos	140

Capítulo 1: Introducción

1. Introducción

En este apartado introductorio se van a mostrar las motivaciones y los objetivos del proyecto, las etapas que se han seguido en él. Además, se muestra el glosario de términos de este documento. Finalmente se mostrará cómo está estructurada toda la documentación.

1.1. Motivación

Durante estos últimos años la venta de los Smartphone ha tenido un gran auge. Hoy en día, la gran mayoría de personas tienen un teléfono inteligente en sus manos, desde los más jóvenes, hasta empresarios que realizan gran parte de su trabajo desde estos dispositivos. El motivo por el que tantas personas tengan estos móviles, son las grandes ventajas que ofrecen estos teléfonos móviles.

Por todo esto el número de aplicaciones que se han creado para estos dispositivos es enorme. Existen aplicaciones de todo tipo, desde las típicas aplicaciones de ofimática, a las aplicaciones sin sentido, que en cuanto a utilidad se refiere, no sirven prácticamente para nada.

Tener conocimiento acerca de cómo realizar este tipo de aplicaciones, es muy importante para un estudiante de grado en ingeniería informática, ya que este tipo de aplicaciones y el lenguaje para programar estas aplicaciones, no se enseñan durante la carrera, por lo que te pueden dar un plus extra para encontrar un trabajo en un futuro.

Estas aplicaciones te dan una visión diferente a la hora de diseñar, ya que se tiene que diseñar en otras dimensiones a las que normalmente se han diseñado, y se tienen que dar una usabilidad mayor que en las aplicaciones para un ordenador, ya que el manejo de estas aplicaciones debe ser más fácil.

El conjunto de estas motivaciones, hacen de este trabajo, un proyecto bastante interesante, ya que hoy en día estas aplicaciones son el presente y por supuesto el futuro. Esto es así, porque según está avanzando la tecnología, estos tipos de dispositivos, reemplazarán a los ordenadores de sobremesa y portátiles. Actualmente, esto ya está ocurriendo con las tablets, que están de moda, ya que tienen prácticamente las mismas prestaciones que un ordenador.

Una motivación extra para realizar este trabajo de fin de grado es que el proyecto se ha realizado en iOS, y los dispositivos que utilizan este sistema operativo, dispositivos de marca Apple, son los más vendidos en cuanto a Smartphone se refiere.

En cuanto a la finalidad de la aplicación realizada, se ha pensado en una aplicación relacionada con el tráfico y los jóvenes. En un principio se pensó en una aplicación para concienciar a los jóvenes de los peligros que existen en la carretera, como por ejemplo ir en moto sin casco, o en coche sin cinturón. Sin embargo, se pensó en una aplicación que los principales usuarios, los jóvenes, utilizarían más, una aplicación en la que estos puedan realizar los test de autoescuela y aprender la lección relacionada con cada test.

Esta aplicación se realizó con este objetivo para resolver el problema que pueden tener todas las personas que están sacándose el carnet de conducir, normalmente jóvenes. Estas personas normalmente no suelen tener demasiado tiempo libre, y con una aplicación de este tipo podrían aprender lo necesario para sacarse el permiso de conducir en cualquier lugar, como por ejemplo cuando viajan en tren, autobús, o en cualquier lugar sin la necesidad de tener que ir a la autoescuela.

Este objetivo de la aplicación se pensó así, porque las personas que están sacándose el carnet de conducir, se lo sacan realizando test y viendo los fallos que tienen. Además se les da la posibilidad de aprender la teoría del tema mediante un vídeo.

La aplicación podría ser de utilidad para una autoescuela, de manera que pudieran complementar las clases teóricas de los profesores con dicha aplicación, de manera que tuvieran más clientes por la comodidad que ofrecen.

1.2. Objetivos

El objetivo principal de este proyecto es poder aprender a desarrollar aplicaciones para dispositivos Apple. Además de aprender a implementar la aplicación, el otro objetivo era aprender cómo funcionan las metodologías ágiles, ya que actualmente este tipo de metodologías están muy de moda y son muy utilizadas en las empresas.

En cuanto a los objetivos específicos del proyecto son:

- Crear una aplicación en la que los usuarios puedan aprender las normas de circulación.
- Utilizar la aplicación para realizar los test que examinan las normas aprendidas en el tema estudiado.
- Dar la posibilidad a las personas que estén sacándose el carnet de conducir, de utilizar esta aplicación como complemento para lo aprendido en la autoescuela.
- Realizar una aplicación para dispositivos móviles que se pueda utilizar en cualquier lugar sin necesidad de conexión a la red.
- Realizar una aplicación lo más genérica posible para poder ser utilizada en otros ámbitos distinto al de las autoescuelas.
- Desarrollar una aplicación amigable y fácil de utilizar para el usuario.

1.3. Etapas del proyecto

En cuanto a las etapas que se han seguido para realizar correctamente el proyecto, en primer lugar, se realizó un análisis de las aplicaciones existentes y se estudió la viabilidad que tenía realizar la aplicación. Con el resultado obtenido se hizo el estado del arte.

Una vez observado que si era viable desarrollar el sistema, porque no había muchas aplicaciones parecidas en el mercado, se empezó a desarrollar la aplicación.

Como la metodología utilizada, era una metodología ágil, para desarrollar la aplicación se iban marcando los objetivos según llegaban las iteraciones, pudiendo añadir nueva funcionalidad a la aplicación al llegar a cada iteración.

Por último, una vez completada la aplicación se realizan pruebas para comprobar que todo lo implementado funciona correctamente. Aunque durante el desarrollo del proyecto, se han ido realizando estas pruebas, al final se han realizado todas ellas por si se había modificado algún apartado.

Antes de todas estas etapas se realizó una etapa de formación para poder realizar la aplicación correctamente, ya que anteriormente no se tenían conocimientos sobre la programación en iOS.

1.4. Glosario de términos

- **AppStore:** tienda virtual de Apple.
- **Bluetooth:** es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos.
- **Framework:** conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
- **Gestos multitáctiles:** operaciones que se realizan sobre los dispositivos táctiles con más de un dedo.
- **iMac:** ordenador de sobremesa de la marca Apple.
- **iOS (iPhone OS):** es el sistema operativo de los dispositivos portátiles de Apple.
- **iPad:** tablet de Apple pensado para disfrutar de música, videojuegos o wifi en cualquier lugar.
- **iPhone:** familia de teléfonos inteligentes de la marca Apple.
- **iPod Touch:** dispositivo de Apple pensado para disfrutar de música, videojuegos o wifi en cualquier lugar.
- **Parsear:** proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal dada.
- **Refactoring:** es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.
- **Stakeholders:** agrupa a trabajadores, organizaciones sociales, accionistas y proveedores, entre muchos otros actores clave que se ven afectados por las decisiones de una empresa. Generar confianza con estos es fundamental para el desarrollo de una organización.

1.5. Organización Documento

En el siguiente apartado se va a mostrar cómo están estructurados los capítulos, mostrando la información de cada uno:

- **Capítulo 1:** El presente capítulo, es el capítulo introductorio de la memoria, el cual inicia al lector de esta. Se incluyen las motivaciones y los objetivos que han llevado a realizar este Trabajo de Fin de Grado, además del glosario de términos y esta organización.
- **Capítulo 2:** En este capítulo, se analizarán las aplicaciones existentes para hacer un estudio de viabilidad del sistema. Además se estudiará el entorno de desarrollo que se utilizará para implementar la aplicación y la metodología que se llevará a cabo.
- **Capítulo 3:** Una vez estudiada la viabilidad del sistema, en el capítulo 3 se va a definir con detalle la funcionalidad completa de la aplicación. Como se ha utilizado Scrum, se ha realizado una extracción de requisitos diferente a lo habitual, que se explicará al comienzo de este capítulo.
- **Capítulo 4:** Ya conocida la funcionalidad de la aplicación, en este capítulo se va a mostrar el diseño que se ha realizado para dicha aplicación, para seguirlo a lo largo de la implementación. En este apartado se mostrarán desde las diferentes alternativas del prototipo, hasta los distintos diagramas necesarios para la aplicación.
- **Capítulo 5:** En este capítulo se van a mostrar las pruebas que han sido realizadas para comprobar que la aplicación funcionaba correctamente. Estas pruebas se han realizado por historias de usuario para comprobar que se probaba toda la funcionalidad.
- **Capítulo 6:** En el capítulo 6 se mostrará la planificación del proyecto, se realizará una inicial y otra final.
- **Capítulo 7:** En este capítulo se calculará el presupuesto, tanto inicial como final, del proyecto. Además se incluye un apartado de herramientas hardware, para especificar los elementos expuestos en el presupuesto e indicar para que han sido utilizados.
- **Capítulo 8:** En este capítulo se exponen las conclusiones extraídas de la realización de este trabajo de fin de grado. Además, se indican las líneas futuras que se podrían seguir llevar a cabo con la aplicación desarrollada.
- **Capítulo 9:** En el último capítulo se ha mostrado la bibliografía, incluyendo las referencias electrónicas y los libros utilizados.

Capítulo 2: Estado del arte

2. Estado del Arte

En este capítulo se va a realizar un estudio de viabilidad del sistema sobre las aplicaciones existentes en el mercado. Con este estudio, se pretende adquirir una mayor comprensión del entorno de estas aplicaciones y valorar otras similares con el fin de evaluar si la realización de la aplicación es viable en el mercado actual. Además se estudiará el entorno de desarrollo que se utilizará para implementar la aplicación y la metodología que se llevará a cabo.

2.1. Análisis de aplicaciones móviles y web

En este apartado se va a realizar un estudio de viabilidad del sistema, donde se va a contextualizar las aplicaciones similares a las que se van a realizar durante este proyecto, es decir, la problemática que quieren resolver, y por lo tanto la problemática que se quiere resolver en este proyecto que se va a desarrollar, también con estas aplicaciones similares se quiere obtener la plataforma tecnológica para el uso de la aplicación, quienes son los usuarios interesados o Stakeholders, y la población a la que va dirigida. Además, en este apartado se va a describir las principales funcionalidades de la aplicación, también se va a realizar un análisis crítico de la aplicación, es decir, indicar puntos fuertes y débiles.

Finalmente se especificarán cuales han sido los criterios de búsqueda que se han seguido, sobre qué herramientas de búsqueda y si la búsqueda devuelve otras aplicaciones que se han descartado, se explicará el porqué.

2.1.1. Aplicaciones Relevantes

En este apartado vamos a mostrar las aplicaciones relevantes que hemos encontrado para nuestro proyecto, tras la búsqueda en internet y el AppStore de un dispositivo móvil de Apple. Aunque se han encontrado muchas aplicaciones relacionadas, las observadas que no tienen demasiado que ver con el objetivo de nuestra aplicación, no se han especificado en este apartado.

FUNDACIÓN MAPFRE. Seguridad Vial para jóvenes

Esta aplicación de la Fundación Mapfre, tiene como objetivo evitar que los jóvenes sean víctimas de accidentes de tráfico, y que estos sean actores y prescriptores de Seguridad Vial. De esta manera los jóvenes colaboran para hacer de la conducción una actividad más segura. Con el programa que han desarrollado, quieren resolver el problema que tienen los adolescentes para aprender consejos prácticos acerca de cómo evitar riesgos, factores que influyen al volante y modos seguros de actuación. Esta aplicación ofrece 6 temas que tienen que “estudiar” los usuarios que estén interesados y con lo aprendido en dichos temas realizar unos tests, para demostrar lo aprendido, además se podrá colaborar con dicha aplicación dando la opinión.



Ilustración 1. Seguridad Vial para jóvenes

La plataforma tecnológica de este proyecto es una aplicación web, que a través de un navegador en un PC, se podrán utilizar los diferentes servicios que ofrece.

Los principales usuarios de esta aplicación son jóvenes que están interesados en la Seguridad Vial. También tiene como usuarios objetivo, aquellos que quieren obtener su carnet de conducir, ya que colaboran con diferentes autoescuelas para dar facilidades a estos usuarios. La zona en la que quiere actuar esta aplicación es toda España, ya que colabora con autoescuelas de diversas ciudades de nuestro país.

FUNDACIÓN MAPFRE. Toca la señal

La aplicación, *Toca la señal*, también de la Fundación Mapfre es un juego que tiene como objetivo enseñar como participar en el tráfico de una forma más segura. Con este juego, se puede aprender a distinguir los distintos tipos de señales que ordenan el tráfico. Antes de empezar a jugar, se hace un repaso de la tipología de las señales, y una vez realizado el breve resumen, la aplicación ofrece 15 niveles de juego.



Ilustración 2. Toca la señal

La plataforma tecnológica de esta aplicación es una aplicación a la que se puede acceder desde dispositivos móviles de la marca Apple, es decir está desarrollada en iOS.

Aunque los principales usuarios de esta aplicación son niños, porque el objetivo de la aplicación es jugar, también puede ser útil para adolescentes que quieran

obtener el carnet de conducir, ya que esta aplicación muestra la tipología de las diferentes señales, enseñando el significado general de cada tipo, además puede entretener a estos usuarios a la vez que aprenden. Además, la población a la que está destinada es a la española, ya que muestra las señales de nuestro país y el juego está totalmente en castellano. La funcionalidad de esta aplicación es la siguiente:



Ilustración 3. Toca la señal func-1

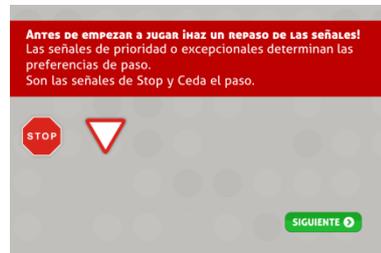


Ilustración 4. Toca la señal func-2



Ilustración 5. Toca la señal func-3



Ilustración 6. Toca la señal func-4



Ilustración 7. Toca la señal func-5



Ilustración 8. Toca la señal func-6



Ilustración 9. Toca la señal func-7



Ilustración 10. Toca la señal func-8



Ilustración 11. Toca la señal func-9

Target Box. Tráfico

Sencilla aplicación que muestra una lista de las señales de tráfico de España con imágenes y códigos de referencia, dichas señales están clasificadas por categoría. Es una aplicación que además de venir bien para adultos que quieren recordar las señales, viene bien para los jóvenes adolescentes que quieran obtener su carnet de conducir. Sin embargo, como punto débil de la aplicación se puede remarcar que la información de las diferentes señales es algo escueta y podía dar más información al pulsar en cada una de las señales.



Ilustración 12. TargetBox Tráfico

La aplicación es bastante simple y fácil de utilizar como se puede observar en las siguientes capturas:

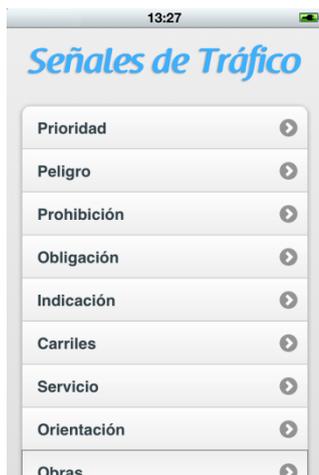


Ilustración 13. TargetBox Tráfico func-1



Ilustración 14. TargetBox Tráfico func-2

XION. El Velocímetro

Es una buena aplicación, no es la típica aplicación de hacer test, es una aplicación Online asistida por una red de autoescuelas de toda España. El velocímetro es la primera autoescuela Online con los mismos servicios que las autoescuelas tradicionales donde un usuario se podrá preparar tanto para el examen teórico como para el examen práctico.

Además, El Velocímetro puede ser utilizado



Ilustración 15. El velocímetro

desde diferentes plataformas, es decir, se puede utilizar esta aplicación desde la web o desde dispositivos móviles de Apple como son el Ipod Touch y el Iphone.

El objetivo de esta aplicación, desde sus diferentes plataformas, es que el usuario pueda estudiarse los diferentes temas, incluso ver video de temas, y una vez estudiados los temas que este pueda realizar test de temas, test de exámenes y test de errores, estos últimos se irán generando automáticamente en función de los fallos del usuario. Además de todo esto, esta aplicación por cada pregunta de test tiene su explicación, pero además se pueden tener “profesores online” a los que se les puede escribir diferentes dudas, a las cuales ellos te contestarán. Por último, también se puede solicitar las clases prácticas y ser atendido por un profesor que atenderá al usuario y que le podrá enseñar a conducir.

En las siguientes imágenes se pueden observar las siguientes aplicaciones:



Ilustración 18. El velocímetro func-1



Ilustración 17. El velocímetro func-2



Ilustración 16. El velocímetro func-3

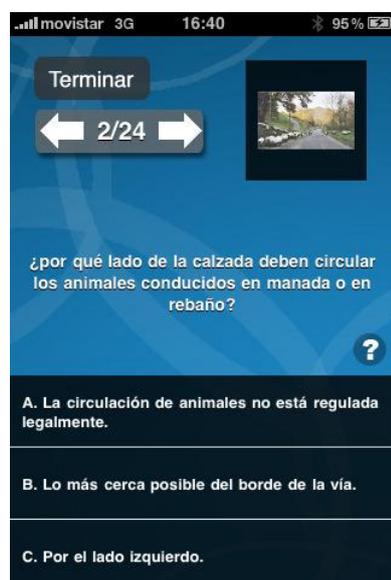


Ilustración 19. El velocímetro func-4



Ilustración 21. El velocímetro func-5



Ilustración 20. El velocímetro func-6

RACE. Seguridad Vial

La aplicación sobre Seguridad Vial que ofrece RACE, es una aplicación bastante completa de la que se puede obtener información interesante para el proyecto que vamos a realizar. Esta aplicación, ayudará a mejorar los conocimientos sobre Seguridad Vial, y a la vez se dispone de diferentes teléfonos de emergencia, informa del tráfico con multitud de opciones y un espacio para que el usuario ayude a mejorar el estado de la vía o que denuncie situaciones que puedan mejorar.



Ilustración 22. RACE

Esta aplicación, por su contenido, está dirigida a personas adultas, aunque se puede sacar cierta información para nuestros Stakeholders, es decir, para adolescentes. El contenido que podemos obtener es la parte de información que ofrece acerca de la formación de Seguridad Vial.

Esta aplicación, con plataforma tecnológica para dispositivos móviles de Apple, muestra la siguiente interfaz:



Ilustración 23. RACE func-1



Ilustración 25. RACE func-2



Ilustración 26. RACE func-3



Ilustración 24. RACE func-4

2.1.2. Aplicaciones descartadas

En esta apartado vamos a mostrar algunas de las aplicaciones descartadas explicando los diferentes motivos. Como aclaración, se quiere decir que no se han incluido en este apartado, todos los documentos o noticias que aparecen acerca de estos temas. Las aplicaciones descartadas son las siguientes:

Mundo docente. Mendoza.edu.ar

Esta aplicación se ha descartado, ya que no cumple la misma funcionalidad que se quiere realizar para este proyecto, porque esta aplicación web, tiene como objetivo, enseñar a docentes y familiares a concienciar a sus respectivos alumnos o hijos. Además, todo está centrado más para niños, que para adolescentes que son los Stakeholders de este proyecto.

Comisariado Europeo del automóvil

Esta aplicación también ha sido descartada, porque el objetivo de esta aplicación no es lo que se está buscando para el proyecto que se está realizando. Esta aplicación, al igual que la anterior, es una página Web, pero el contenido de esta página Web, únicamente son largos textos que te explican ciertas normas de circulación. Además esta aplicación está dirigida a un público más adulto, teniendo únicamente un apartado relacionado con los jóvenes.



Ilustración 27. CEA

Dirección General de Tráfico

La DGT, al igual que la CEA, proporciona un apartado de la página Web oficial de la DGT, la cual está dividida en varios apartados con enlaces a



Ilustración 28. DGT

documentos PDF, sin que el usuario interactúe directamente con la aplicación. Por este motivo no se considera esta aplicación de utilidad para nuestro proyecto. Además no hace apenas referencia a los jóvenes y adolescentes.

Fundación Mapfre. De dos en dos

Se ha descartado la aplicación *De dos en dos* ya que esta aplicación no tiene como objetivo enseñar, sino que es un juego en el que tiene que emparejarse señales iguales, sin mostrar ninguna información acerca de ellas. Es una aplicación que está enfocada más para niños, que para adolescentes, que es lo que se quiere desarrollar en este proyecto. Por todos estos motivos no se considera esta aplicación, como una aplicación relevante para este proyecto.



Ilustración 29. De dos en dos

Confort Auto. Seguridad Vial

Esta aplicación se ha descartado especialmente porque está únicamente dedicada a los niños, para que aprendan la Seguridad Vial jugando y realizando unos tests en compañía de sus padres. El juego que se propone en el juego es colorear distintas imágenes. Aunque es una aplicación interesante, lo es para niños, no para adolescentes.



Ilustración 30. ConfortAuto

The Black Leman. CAS

La aplicación CAS, Control de Alcohol en Sangre, tiene como objetivo educar sobre los peligros que ocasiona el exceso de alcohol entre las personas. Sin embargo en ningún instante, la aplicación te recomienda no beber, solo te dice, aproximadamente, el porcentaje de alcohol en sangre que tendría el usuario. Se ha marcado como no relevante esta aplicación porque no tiene como objetivo enseñar a jóvenes y adolescentes en ningún ámbito de la Seguridad Vial.



Ilustración 31. CAS

2.1.3. Conclusión de aplicaciones analizadas

Como conclusión a las aplicaciones analizadas se ha sacado la conclusión de que no existe ninguna otra aplicación con el mismo objetivo al que pretendemos hacer nosotros. Aunque si existen aplicaciones similares ninguna tiene el mismo fin que la aplicación que se va a desarrollar. En general, todas las aplicaciones tienen un usuario principal distinto al nuestro, aunque se intenta asemejar no lo son, su usuario principal normalmente son o los adultos o los niños, pero no se suelen centrar en los adolescentes. La aplicación más parecida es el velocímetro, pero en nuestro caso, se quiere realizar una aplicación más dinámica y sencilla de utilizar.

2.1.4. Criterios de Búsqueda

Los criterios de búsqueda que se han seguido para encontrar las aplicaciones anteriores han sido mediante un buscador del siguiente modo:

- Vía Web en el buscador Google:
 - Seguridad Vial Jóvenes.
 - Seguridad Vial Adolescentes.
 - Educación Vial.
- Dispositivo Móvil Apple, a través de Apple Store.
 - Tráfico.
 - Seguridad Vial.

Estas búsquedas se realizaron en Febrero del 2012.

2.2. Entorno de Desarrollo Móvil

2.2.1. Sistema Operativo



Ilustración 32. iOS 5

En primer lugar se va a definir el sistema operativo que utiliza el iPhone, dispositivo móvil utilizado para el desarrollo del proyecto, dicho sistema operativo es **iOS**, en concreto la versión iOS 5.

iOS tiene una interfaz de usuario basada en el concepto de la manipulación directa, usando gestos multitáctiles, es decir, en la que se interactúa con el dispositivo utilizando su pantalla táctil de manera que reconoce múltiples puntos de contacto. Esta interacción con el sistema operativo consiste en deslices, toques, etc., que según el entorno en el que se esté trabajando puede tener diferentes resultados. Además de esto, tiene acelerómetros internos, que son utilizados en algunas aplicaciones al agitar el teléfono, o lo más normal como es rotarlo para poder verla aplicación en modo horizontal o en modo vertical.

Como dato iOS deriva de Mac OS X, que este a su vez deriva de Darwin BSD, y por lo tanto iOS es un sistema operativo de Unix.

iOS, al igual que Mac OS X, sostiene su base sobre cuatro capas:

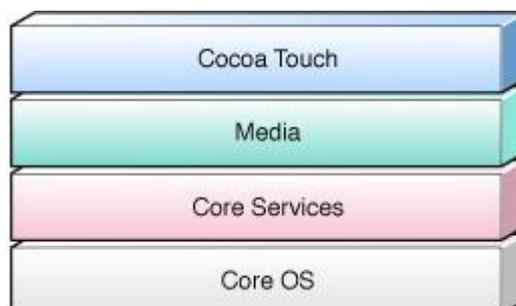


Ilustración 33. Arquitectura iOS

iOS, en comparación con Mac OS X, se diferencia en la primera capa (Cocoa Touch), que ha sido la única modificada en iOS, el resto han quedado igual.

iOS 5 con respecto a las anteriores versiones a la hora de programar, se diferencia en que para diseñar la interfaz anteriormente se utilizaban diversos ficheros .xib que correspondían a cada vista de la interfaz, mientras que con iOS 5 se utiliza un único fichero .storyboard en el que se pueden tener las diferentes vistas, además de que se puede tener la interconexión entre las diferentes vistas y mostrar la transición entre ellas. [\[1\]](#)

2.2.2. Lenguaje de Programación

El lenguaje de programación que se utiliza para la programación en iOS y para el desarrollo de aplicaciones para dispositivos móviles de la marca Apple, es **Objective-C**.

Objective-C fue creado por Brad Cox y la corporación StepStone en 1980. En 1988 fue acogido como lenguaje de programación de NEXTSTEP. En 1992 fue liberado bajo licencia GNU para el compilador GCC.

Objective-C es un lenguaje de programación orientado a objetos, se le puede considerar como un superconjunto de C. Para realizar Objective-C han cogido las ideas de Smalltalk, pero lo han hecho de manera más eficiente. El modelo de programación se basa en enviar mensajes a instancias de objetos, que tiene una gran diferencia con otros lenguajes orientados a objetos. En Objective-C no se llama a un método, sino que se envía un mensaje, y la diferencia entre estos dos conceptos radica en cómo el nombre del código o método es ejecutado.

En este lenguaje es necesario que la interfaz y la implementación de una clase estén en bloques de código separados, siendo la interfaz puesta en un archivo de cabecera (.h) y la implementación en un archivo de implementación (.m). [\[2\]](#)

2.2.3. API a utilizar

La API que se utiliza como complemento de Objective-C es Cocoa Touch. Cocoa Touch es una de las capas más importantes de iOS. Está compuesto por *UIKit* (User Interface Kit) y *Foundation frameworks*, las cuales proporcionan herramientas básicas y la infraestructura necesaria para implementar los aspectos gráficos de la aplicación.

Además de esto, también incluye otros marcos que proporcionan servicios para acceder a las características del dispositivo, como pueden ser los contactos del usuario. La capa de Cocoa Touch es el framework de desarrollo para iPhone y se compone de lo siguiente:

- ✓ *Eventos y controles multitouch.*
- ✓ *Soporte del acelerómetro.*
- ✓ *Vista jerárquica.*
- ✓ *Localización de aplicaciones.*
- ✓ *Alertas.*
- ✓ *Vista web.*
- ✓ *Selector de contactos.*
- ✓ *Selector de imágenes.*
- ✓ *Soporte de la cámara.* [\[3\]](#)

2.2.4. Entorno de desarrollo



Ilustración 34.

El entorno de desarrollo que se utiliza para realizar aplicaciones para dispositivos Apple es **Xcode**. Xcode por defecto tiene instalado el frameworks anteriormente explicado, Cocoa Touch. Xcode es una herramienta de desarrollo fácil de usar, aunque lo suficientemente potente, ya que los desarrolladores de Apple utilizan esta herramienta.

Xcode tiene diferentes características que nos ofrece diferentes facilidades, tales como:

- ✓ Asistente de redacción.
- ✓ Editor de código fuente.
- ✓ Interface Builder integrado (Storyboard).
- ✓ Simulador iOS.
- ✓ Compilador integrado.
- ✓ Depurador gráfico.
- ✓ Ayuda con Refactoring.
- ✓ Subversion integrado.
- ✓ Etc.

Para este proyecto hemos utilizado dos versiones de Xcode, la versión 4.2.1. que tiene el simulador iOS 5.0 y la versión 4.3.2. que tiene el simulador iOS 5.1.

2.3. Desarrollo Ágil

Para la realización de este proyecto, se ha optado por la utilización de una metodología ágil. Esta decisión ha sido tomada por varios motivos que favorecen a este tipo de desarrollo, como son entre otros:

- ✓ Intentan evitar problemas habituales como retrasos, complejidad, costes, etc.
- ✓ Son efectivas con requisitos cambiantes, reducen tiempos y mantienen la calidad.
- ✓ Se basa en iteraciones cortas, normalmente entre 1 y 4 semanas.
- ✓ Al final de cada iteración se debe presentar una demo, sin errores.

El manifiesto ágil se basa en una serie de premisas:

- Centrarse en los individuos y sus interacciones, en lugar de sobre los procesos y herramientas.
- Es mejor tener software funcionando, que documentación extensiva.
- Es mejor poder responder ante el cambio, que seguir un plan.
- Preferible la colaboración con el cliente (en nuestro caso el tutor), que una negociación contractual.

Todas estas premisas que hemos descrito anteriormente, son las que se han seguido a la hora de desarrollar el proyecto.

Además de estas premisas del manifiesto ágil, la metodología ágil tiene unos principios que hemos ido siguiendo a lo largo del desarrollo del proyecto:

- ✓ Se acepta que los requisitos cambien.
- ✓ Se debe entregar frecuentemente software que funcione con el menor intervalo de tiempo posible.
- ✓ Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesiten y confiar en ellos para conseguir finalizar el trabajo.
- ✓ El diálogo cara a cara es el método más eficiente y efectivo.
- ✓ El software que funciona es la medida principal de progreso.
- ✓ La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- ✓ La simplicidad es esencial.

Existen distintos tipos de metodologías ágiles, pero dentro de estos tipos existen más prescriptivas como son RUP o XP, o más adaptativas como es KANBAN, y una intermedia a es SCRUM, que es la que se está llevando a cabo dentro de este proyecto.

2.3.1. SCRUM

Historia

El concepto de Scrum tiene su origen en el estudio de 1986 sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y Estados Unidos, como fueron las cámaras de fotos, fotocopiadores, automóviles, ordenadores, etc. Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos del tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyectos muy similares realizados anteriormente. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinares con la colaboración entre los jugadores de Rugby y su formación de Scrum (melé en español).



Ilustración 35. Scrum

En 1993 se realizó el primer Scrum para desarrollo de software y en 1995 el proceso fue normalizado. En 2001 un grupo de personas muy relevantes en lo que empezaba a ser el desarrollo ágil escribieron los valores fundamentales de los procesos ágiles.

Desde 1995 miles de proyectos en todo el mundo han utilizado Scrum para el desarrollo de productos, tanto en pequeñas empresas, como en multinacionales como son "Google", "Microsoft", "Oracle" e "IBM", entre otras dentro del tipo software, y de otro tipo como "BBC", "Amazon", Electronic Arts, etc. [\[4\]](#)

Ciclo de Vida

El ciclo de vida de Scrum es el siguiente:

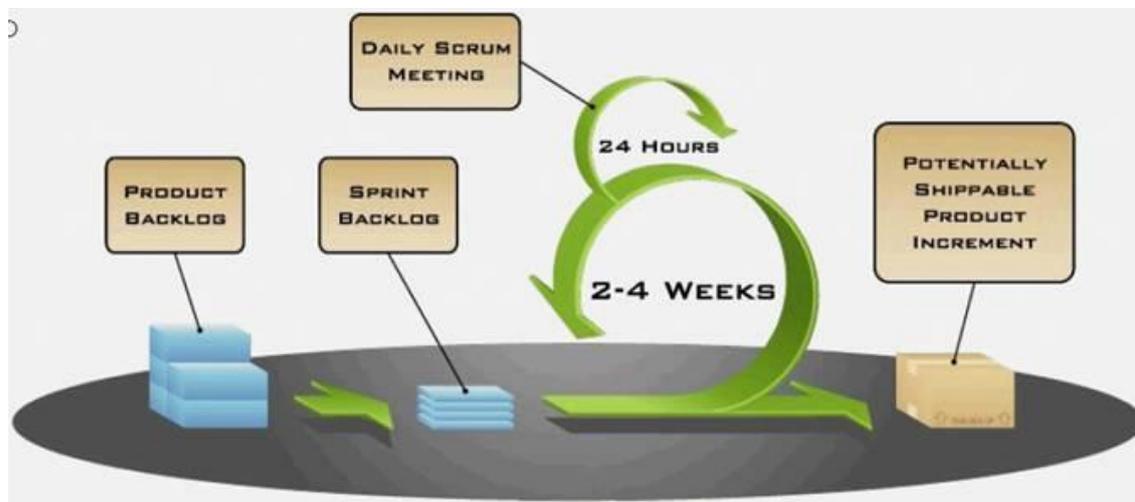


Ilustración 36. Ciclo de Vida Scrum

Donde el **Product Backlog** es:

- Listado con los requisitos del sistema. Contiene: features, requisitos de desarrollo (no funcionales), tareas de investigación y bugs.
- Realizar este listado es responsabilidad del Product Owner.
- Es una combinación de funcionalidades y tareas.
- Es un documento dinámico que incorpora constantemente las necesidades del sistema. Se mantiene durante todo el ciclo de vida (hasta el retiro del sistema).

Y el **Sprint Backlog** es:

- Listado de tareas determinadas por el equipo para realizar en un sprint y lograr al final del mismo un incremento de funcionalidad.
- Las tareas de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo.
- En Scrum se debe ir registrando los tiempos día a día para poder armar el gráfico de avance del proyecto.
- El equipo agrega tareas cuando lo crea necesario, pudiendo eliminar las que considere innecesarias, y ajusta estimaciones a medida que se avanza.

El **Sprint** es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad:

- Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas "carreras".

- La duración máxima de un sprint es de 30 días.
- Durante el sprint no se puede modificar el trabajo que se ha acordado en el Sprint Backlog.
- Sólo es posible cambiar el curso de un sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por algunas de las razones siguientes:
 - La tecnología acordada no funciona.
 - Las circunstancias del negocio han cambiado.
 - El equipo ha tenido interferencias.

Otro aspecto importante en el ciclo de vida de la metodología Scrum, son las reuniones, las cuales las podemos dividir en:

- Reunión de planificación Release.
- Reunión de planificación Sprint.
- Scrum Diario.
- Demostración o revisión del Sprint.
- Retrospectiva del Sprint.

Reunión de planificación de Release

En estas reuniones intervienen el Product Owner, el Scrum Manager y el Team. El objetivo es la creación del Product backlog y se realiza al comienzo del proyecto. La reunión consiste en:

- Construcción del Product Backlog.
- Discusión acerca de los costes, riesgos, fechas, estimaciones, etc.
- Realizar un diseño a muy alto nivel.

Reunión de planificación Sprint

En estas reuniones intervienen el Scrum Manager y el equipo (también debe estar disponible el cliente ya sea vía teléfono o email). En esta reunión el Product Owner presenta el Product Backlog y entre todos se seleccionan las tareas a realizar en el sprint, estas reuniones no pueden durar más de una jornada y se realizan antes de cada sprint. Esta reunión consiste en primer lugar en seleccionar del Backlog y seleccionar las tareas prioritarias y fijar la meta del sprint. Una vez realizado esto, se planifica el trabajo del sprint, definiendo la arquitectura, el diseño de la funcionalidad y la definición-división de las tareas del Backlog del Sprint.

Scrum Diario

En estas reuniones intervienen el Scrum Manager y el equipo, cuyo objetivo es identificar los posibles problemas y obstáculos para poder resolverlos cuanto antes. Esta reunión debe tener una duración aproximada de 15 minutos y normalmente se realiza en el mismo sitio y de pie. Esto se realiza diariamente durante el sprint.

En esta reunión se habla de lo que se ha hecho, de lo que se va hacer y de si ha existido algún obstáculo o problema.

Revisión del Sprint. Demo

En esta reunión intervienen el Product Owner, los Stakeholders el Scrum Manager y el equipo. El objetivo de esta reunión es la inspección del incremento integrado, también se realiza una demostración del proyecto, se obtienen los feedbacks y por último se decide que se hará en el próximo sprint.

Es aconsejable que esta reunión dure 2 horas. Como máximo puede durar 4. La reunión sigue los siguientes pasos:

- Presentación del objetivo del Sprint.
- Presentación de funcionalidad realizada.
- Revisión de progreso: historias no realizadas, historias a medias.
- Anotar mejoras, observaciones y cambios solicitados.

Retrospectiva

En esta reunión intervienen Product Owner, el Scrum Manager y el equipo. El objetivo es revisar como se ha desarrollado el Sprint. Se discute sobre procesos, prácticas, comunicación, entorno, herramientas... en esta reunión, también se proponen mejoras de anteriores retrospectivas y tienen que durar entre 30 y 60 minutos. [5]

Equipos de desarrollo en SCRUM. Roles y Responsabilidades

El equipo de desarrollo de Scrum está formado por dos tipos de roles, los **roles comprometidos** y los **roles involucrados**.

Dentro de los roles comprometidos se encuentran el *Product Owner* – es el que representa al cliente, aunque no tiene porque serlo –, el *Scrum Manager* – es el que gestiona los problemas que puedan surgir – y el *Team Manager*. Las responsabilidades de estos son las siguientes:

- Product Owner:
 - Decide la funcionalidad del producto.
 - Es el responsable del Product Backlog y de su priorización.
 - Representa a los usuarios.
 - Acepta o rechaza el trabajo realizado en el sprint.
- Scrum Manager:
 - Lleva adelante el equipo.
 - Elimina posibles obstáculos.
 - Motiva al equipo.
 - Permite la colaboración entre los diferentes roles.
- Team Scrum:
 - Seleccionan la meta del Sprint.
 - Crean el producto.
 - Interdisciplinados.
 - Comprometidos con la entrega.
 - Estiman las tareas y definen el sprint.

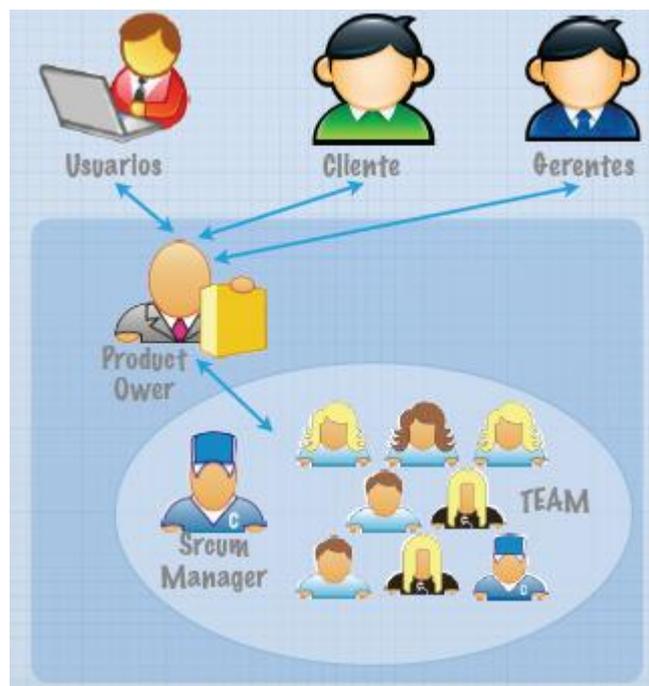


Ilustración 37. Roles Scrum

Dentro de los roles comprometidos se van a señalar diferentes aspectos del Team Scrum. Están formados por analistas, diseñadores, encargados de calidad, desarrolladores, documentadores, etc.

Estos deben conocer y comprender la visión del Product Owner, además de aportar y colaborar. Ellos son los que comparten el objetivo y la responsabilidad del sprint, además se respetan las aportaciones y opiniones de todos y todos participan en las decisiones.

Los otros roles definidos anteriormente, los roles involucrados son los usuarios, los stakeholders y los gerentes.

Prácticas para el desarrollo de software en SCRUM

El desarrollo de software ha tenido una evolución lenta si se le compara con otras áreas de las tecnologías de información, en particular lo que se refiere a las metodologías de trabajo en el desarrollo de software y se le compara a estas con el desarrollo de las componentes de hardware y software básico.

Hoy en día, que aumenta exponencialmente la complejidad del software, el desarrollo metodológico no va avanzando con la misma velocidad o al menos la adopción de buenas prácticas por parte de las organizaciones.

Por ello algunas de las buenas prácticas para el correcto desarrollo software en SCRUM, son las siguientes:

- ✓ Se trabaja en base a iteraciones (entre 15 días y un mes).
- ✓ Equipos de trabajo pequeños (entre 7 y 9 personas).
- ✓ En cada iteración se implementan funcionalidades de usuario, las cuales deben ser entregables.
- ✓ El usuario debe participar en las iteraciones.
- ✓ No es necesario que se especifique todo desde el comienzo.
- ✓ Se pueden cambiar los requisitos a lo largo del proyecto.
- ✓ Cada equipo tiene un líder.
- ✓ Se realizan reuniones diarias, de unos 15 min. de duración.

Todos estos puntos son muy importantes, pero es destacable en la metodología SCRUM, el tercer punto, ya que en otras metodologías no se tiene porqué realizar una entrega funcionable en cada iteración, mientras que en este caso sí, y se podrá medir la parte del proyecto por la que estamos en el momento de dicha iteración, para medirlo se pueden utilizar las pruebas que han sido definidas, viendo cuáles se pasan y cuáles no.

SCRUM es más que una metodología, una colección de buenas ideas, la cual se basa en la transparencia, ya que todos los involucrados en el proyecto saben que está pasando con el desarrollo de una determinada aplicación en periodos cortos de tiempo. [\[6\]](#)

Capítulo 3: Especificación de Requisitos

3. Especificación de Requisitos

En este apartado se va a mostrar una especificación de requisitos detallada, utilizando en este caso la técnica de las historias de usuario. Las historias de usuario son una manera simple para describir una tarea concisa que aporta valor al usuario o al negocio. Esta técnica es muy utilizada en las metodologías ágiles, ya que en estas metodologías se utiliza el formalismo del Product Backlog y del Sprint Backlog, en los cuales se registran las historias de usuario y las tareas, respectivamente.

Finalmente se va a mostrar un diagrama de casos de uso y seguidamente los casos de uso mostrados en detalle.

3.1. Historias de Usuario

A continuación se van a mostrar las historias de usuario que definen la aplicación.

ID	OBJETIVO
HU – 1	Ver información de cada tema
HU – 2	Ver resultados de test anteriores
HU – 3	Realizar test
HU – 4	Ver resultado del test realizado
HU – 5	Compartir en redes sociales

Tabla 1. Historias de Usuario

Una vez definidas las historias de usuario, en los siguientes subapartados se van a mostrar las especificaciones detalladas de cada historia de usuario con sus tareas correspondientes.

HU - 1. Ver información de cada tema

La especificación detallada de la historia de usuario 1 es:

- Mostrar listado de temas con su nombre y una imagen representativa.
- Mostrar una breve descripción del tema.
- Poder pulsar sobre la descripción para ver un vídeo y aprender el tema en detalle.
- Se podrá seleccionar un test para entrar en detalle de dicho test.

Esta historia de usuario tiene las siguientes tareas:

ID	TAREA	OBJETIVO RELACIONADO
T - 1 - 1	Mostrar listado de temas en celdas	HU - 1
T - 1 - 2	Mostrar información de cada tema en una celda	HU - 1
T - 1 - 3	Mostrar video explicativo	HU - 1
T - 1 - 4	Mostrar listado de test a realizar	HU - 1
T - 1 - 5	Obtener datos de la base de datos	HU - 1
T - 1 - 6	Definición de pruebas de sistema	HU - 1
T - 1 - 7	Ejecución de pruebas de sistema	HU - 1

Tabla 2. Tareas HU-1

HU – 2. Ver resultados de test anteriores

La especificación detallada de la historia de usuario 2 es:

- Mostrar, en caso de que existan, las fechas en las que se realizó el test anteriormente.
- Mostrar el número de aciertos que se tuvieron en dicho test en la fecha mencionada anteriormente.
- Poder seleccionar dicha fecha con dicho número de aciertos para ver el resultado específico del test.
- Poder realizar el test.

Esta segunda historia de usuario tiene las siguientes tareas:

ID	TAREA	OBJETIVO RELACIONADO
T - 2 – 1	Mostrar fecha de los test realizados anteriormente	HU – 2
T - 2 – 2	Mostrar listado de los resultados de los test realizados anteriormente	HU – 2
T - 2 – 3	Dar la posibilidad de hacer el test	HU – 2
T - 2 – 4	Obtener datos de la base de datos	HU – 2
T - 2 – 5	Definición de pruebas de sistema	HU – 2
T - 2 – 6	Ejecución de pruebas de sistema	HU – 2

Tabla 3. Tareas HU-2

HU - 3. Realizar test

La especificación detallada de la historia de usuario 3 es:

- Mostrar la diferente información de la pregunta(imagen, número de pregunta, pregunta y respuestas)
- Poder seleccionar una respuesta.
- En caso de que la pregunta sea la 1 dar la posibilidad de ver ayuda para realizar test.
- En caso de que la pregunta sea la última dar la posibilidad de corregir el test.
- En caso de que la pregunta no sea ni la primera ni la ultima dar la posibilidad de volver al inicio de la aplicación.

Esta historia de usuario tiene las siguientes tareas:

ID	TAREA	OBJETIVO RELACIONADO
T - 3 - 1	Mostrar número de la pregunta	HU - 3
T - 3 - 2	Poder visualizar la imagen correspondiente a cada pregunta	HU - 3
T - 3 - 3	Mostrar la pregunta	HU - 3
T - 3 - 4	Mostrar las respuestas	HU - 3
T - 3 - 5	Poder seleccionar una celda con su respuesta correspondiente	HU - 3
T - 3 - 6	Poder cambiar la respuesta seleccionada	HU - 3
T - 3 - 7	En la última pregunta dar la posibilidad de corregir el test	HU - 3
T - 3 - 8	En todas las preguntas, a excepción de primera y la última, te debe dejar volver a la HU - 2	HU - 3
T - 3 - 9	En la primera pregunta dar la posibilidad de ver una ayuda para realizar el test	HU - 3
T - 3 - 10	Obtener datos de la base de datos	HU - 3
T - 3 - 11	Definición de pruebas de sistema	HU - 3
T - 3 - 12	Ejecución de pruebas de sistema	HU - 3

Tabla 4. Tareas HU-3

HU - 4. Ver resultados del test realizado

La especificación detallada de la historia de usuario 4 es:

- Se podrá ver un resumen del total de aciertos y fallos del test realizado.
- Se podrá ver en cada pregunta si se ha fallado o se ha acertado dicha pregunta.
- Se podrá seleccionar una pregunta para ver la respuesta que he seleccionado y cuál es la correcta.

Esta historia de usuario tiene las siguientes tareas:

ID	TAREA	OBJETIVO RELACIONADO
T - 4 - 1	Mostrar el total de fallos y aciertos de la respuesta a los test	HU - 4
T - 4 - 2	Mostrar en detalle, por cada pregunta, si se ha obtenido un acierto o fallo	HU - 4
T - 4 - 3	En la lista de preguntas, poder acceder a cada pregunta en particular	HU - 4
T - 4 - 4	En cada pregunta en particular señalar cuál es la contestada y cuál es la correcta	HU - 4
T - 4 - 5	Insertar en la base de datos los resultados	HU - 4
T - 4 - 6	Obtener datos de la base de datos	HU - 4
T - 4 - 7	Definición de pruebas de sistema	HU - 4
T - 4 - 8	Ejecución de pruebas de sistema	HU - 4

Tabla 5. Tareas HU-4

HU – 5. Compartir en redes sociales

La especificación detallada de la historia de usuario 4 es:

- Si nos encontramos en la vista de contestar el test se podrá seleccionar la opción de compartir la información.
- La información se podrá compartir en twitter.
- La información se podrá compartir en facebook.
- La información se podrá compartir por correo electrónico.

Esta historia de usuario tiene las siguientes tareas:

ID	TAREA	OBJETIVO RELACIONADO
T - 5 - 1	Mostrar una lista con las diferentes posibilidades para compartir la información	HU – 5
T - 5 - 2	Permitir compartir la información deseada en la red social twitter	HU – 5
T - 5 - 3	Permitir compartir la información deseada en la red social facebook	HU – 5
T - 5 - 4	Permitir compartir la información deseada mediante correo electrónico	HU – 5
T - 5 - 5	Definición de pruebas de sistema	HU – 5
T - 5 - 6	Ejecución de pruebas de sistema	HU – 5

Tabla 6. Tareas HU-5

3.2. Casos de Uso

En este apartado se va a mostrar un diagrama de casos de uso y a continuación se especificará cada caso de uso en particular. Un caso de uso ayuda a especificar más en particular lo que el sistema debe hacer desde el punto de vista del usuario, es decir, describen un uso del sistema y cómo este interactúa con el usuario. A continuación se va a mostrar el diagrama de casos de uso:

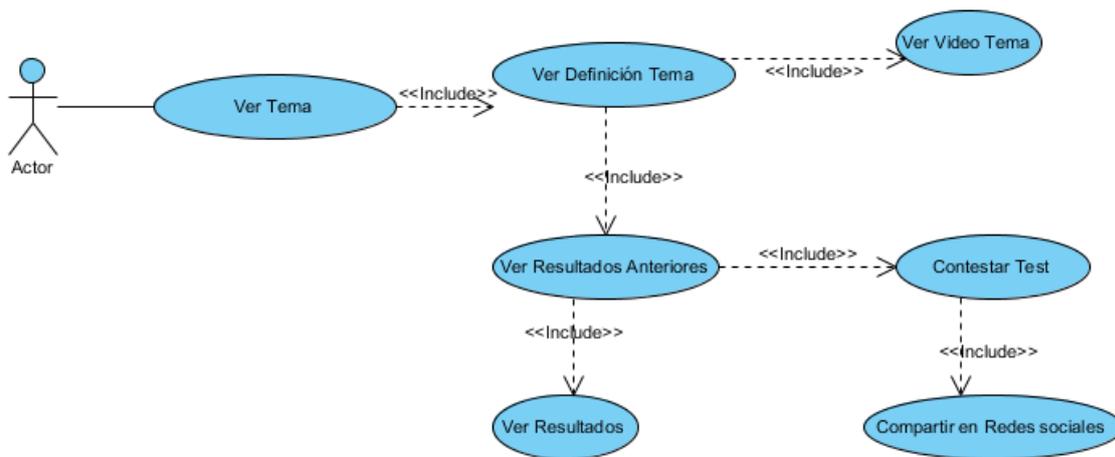


Ilustración 38. Diagrama de Casos de Uso

Una vez mostrado el diagrama se van a especificar los diferentes casos de uso con la siguiente tabla:

Nombre:
Autor:
Descripción:
Actores:
Precondiciones:
Flujo Normal:
Flujo Alternativo:
Postcondiciones:

Tabla 7. Tabla modelo Casos de Uso

Donde:

- Nombre: es el nombre identificativo del caso de uso.
- Autor: la persona que ha realizado el caso de uso.
- Descripción: es una breve descripción de cada caso de uso.
- Actores: aquello que interactúan con el sistema.
- Precondiciones: hechos que se han de cumplir para que el flujo del evento se pueda llevar a cabo.
- Flujo normal: corresponde a la ejecución normal y exitosa del caso de uso.
- Flujo alternativo: nos va a permitir indicar qué es lo que hace el sistema en los casos menos frecuentes o inesperados.
- Postcondiciones: hechos que se han de cumplir si el flujo de eventos normal se ha ejecutado correctamente.

Nombre: <i>Ver tema</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá ver los diferentes temas y seleccionar uno
Actores: Usuario dispositivo
Precondiciones: N/A
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. El sistema le muestra un listado con los diferentes temas 3. El usuario selecciona un tema
Flujo Alternativo: N/A
Postcondiciones: El sistema le cambia de vista

Tabla 8. Caso de uso Ver tema

Nombre: <i>Ver definición tema</i>
Autor: Roberto Esteban Santiago
Descripción: Se ve una breve descripción del tema y/o video
Actores: Usuario dispositivo
Precondiciones: El usuario ha seleccionado un tema determinado
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario selecciona el tema 2. El sistema le muestra una o varias descripciones 3. El usuario pulsa en la descripción y ve una mayor explicación con un video
Flujo Alternativo: <ol style="list-style-type: none"> 3. El usuario selecciona el test que desea realizar
Postcondiciones: Se abre el vídeo

Tabla 9. Caso de uso Ver definición tema

Nombre: <i>Ver video del tema</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá ver un video descriptivo del tema
Actores: Usuario dispositivo
Precondiciones: El usuario ha seleccionado un video para ser visualizado
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario selecciona el vídeo 2. El sistema muestra el vídeo
Flujo Alternativo: N/A
Postcondiciones: N/A

Tabla 10. Caso de uso Ver vídeo del tema

Nombre: <i>Contestar Test</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá contestar a las diferentes preguntas de los test
Actores: Usuario dispositivo
Precondiciones: El usuario ha seleccionado un tema El usuario ha seleccionado un test
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario selecciona el test 2. El sistema le devuelve una pregunta 3. El usuario lee la pregunta y las respuestas y selecciona una de estas respuestas 4. El usuario pasa a la pregunta siguiente con un ligero deslíz 5. El sistema le muestra otra pregunta hasta llegar a la última 6. El usuario, en la última pregunta pulsa el botón corregir
Flujo Alternativo: <ol style="list-style-type: none"> 4. El usuario pulsa el botón de volver al inicio para regresar a la selección del test
Postcondiciones: Se le muestra al usuario los resultados obtenidos

Tabla 11. Caso de uso Contestar Test

Nombre: <i>Ver resultados</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá ver los resultados obtenidos en el test
Actores: Usuario dispositivo
Precondiciones: El usuario elije un tema El usuario elije un test El usuario contesta el test o desde los resultados anteriores accede a esta vista El usuario pulsa el botón de corregir
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario ve el resumen de la corrección y cada pregunta en particular 2. El usuario selecciona una pregunta 3. El sistema le muestra la respuesta que ha contestado y si es la correcta o no
Flujo Alternativo: <ol style="list-style-type: none"> 2. El usuario vuelve al inicio de la aplicación
Postcondiciones: N/A

Tabla 12. Caso de uso Ver Resultados

Nombre: <i>Ver Resultados Anteriores</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá ver los resultados realizados anteriormente del test seleccionado
Actores: Usuario dispositivo
Precondiciones: El usuario elije un tema El usuario elije un test
Flujo Normal: 1. El usuario selecciona un tema 2. El usuario selecciona un test
Flujo Alternativo: N/A
Postcondiciones: Se observan los resultados en detalle del test seleccionado, de la fecha determinada

Tabla 13. Caso de uso Ver Resultados Anteriores

Nombre: <i>Compartir en redes sociales</i>
Autor: Roberto Esteban Santiago
Descripción: El usuario podrá compartir información en distintas redes sociales
Actores: Usuario dispositivo
Precondiciones: El usuario está realizando un test El usuario está visualizando la corrección de un test
Flujo Normal: 1. El usuario pulsa el botón de compartir en el BottomBar 2. El sistema le muestra las distintas posibilidades en las que puede compartir 3. El usuario elije la opción a compartir
Flujo Alternativo: N/A
Postcondiciones: Se publica en la red social determinada el texto deseado

Tabla 14. Caso de uso Compartir en redes sociales

3.3. Requisitos de Restricción

En cuanto a los requisitos de restricción, esta aplicación no tiene demasiados, ya que no se necesitara ni internet para acceder a los datos, ni bluetooth, ni restricciones de seguridad. Los requisitos de restricción que tiene la aplicación se van a describir con la siguiente tabla:

ID	RR-X
Descripción	
Necesidad	Esencial/Opcional
Prioridad	Alta/Media/Baja

Tabla 15. Tabla tipo Requisitos de Restricción

Donde:

- ID: es el identificador de cada requisito.
- Descripción: detalla la restricción del requisito.
- Necesidad: puede tener dos valores y describe la necesidad de cumplirlo o no.
- Prioridad: puede tener tres valores y describe la prioridad del requisito.

ID	RR-1
Descripción	Para publicar mensajes en las redes sociales o enviar por correo se debe tener iniciada la cuenta
Necesidad	Esencial
Prioridad	Alta

Tabla 16. Requisito de Restricción 1

ID	RR-2
Descripción	Se debe tener instalada en el dispositivo una versión superior o igual a iOS 5.0
Necesidad	Esencial
Prioridad	Alta

Tabla 17. Requisito de Restricción 2

ID	RR-3
Descripción	El sonido ambiente no debe ser demasiado alto para escuchar los vídeos de la descripción del tema
Necesidad	Esencial
Prioridad	Alta

Tabla 18. Requisito de Restricción 3

Capítulo 4: Diseño

4. Diseño

En este apartado se van a mostrar todo lo relacionado con el diseño de la aplicación, en primer lugar aparecerá las alternativas de diseño en cuanto a la arquitectura, seguidamente se mostrará un diseño de la interfaz de usuario de la aplicación. Una vez definida la interfaz se va a incluir el diagrama de componentes, el de clases, y finalmente se mostrará el diagrama de secuencia con sus correspondientes tarjetas CRC. Además se incluye el diseño de la base de datos utilizada.

4.1. Alternativas de diseño

En este apartado se van a explicar diferentes alternativas de diseño que se pensaron al comienzo de la aplicación, para ello se van a mostrar unos diagramas conceptuales y se explicarán cada uno de ellos.

4.1.1. Alternativa de diseño 1

Esta alternativa consiste en guardar los datos de la aplicación y los vídeos en el dispositivo móvil:



Ilustración 39. Alternativa de Diseño 1

En este caso se ha pensado en guardar todos los textos en el código fuente y los vídeos en la aplicación, por lo que todo residiría en el dispositivo móvil, además el código fuente sería difícil de modificar, ya que se encontraría mucha cantidad de información dentro de él.

4.1.2. Alternativa de diseño 2

La segunda alternativa consiste en guardar los datos de la aplicación en una base de datos y los vídeos en el dispositivo móvil:

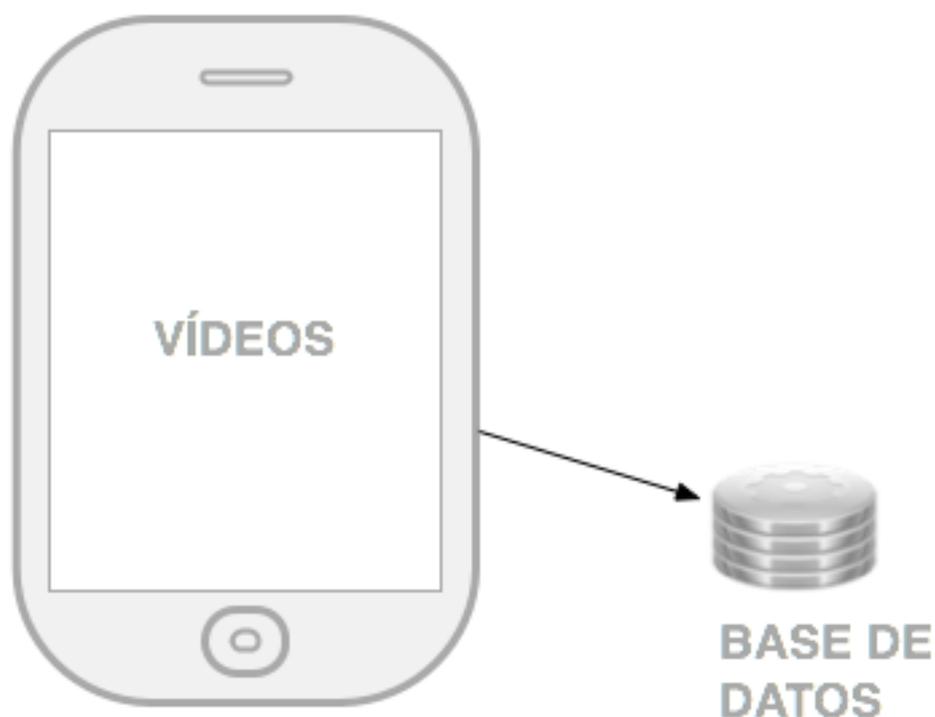


Ilustración 40. Alternativa de Diseño 2

En este caso se ha pensado en guardar todos los textos en una base de datos que residiría en la aplicación, pero toda la información ya no se encontraría en el código, sino que se encontraría en una base de datos, además los datos se encontrarían más organizados. En cuanto a los vídeos de la aplicación, al igual que en la anterior alternativa residiría en el dispositivo móvil. De este modo el código fuente sería más fácil de modificar, ya que se encontraría mucha menos cantidad de información dentro de él.

Otra opción similar a esta, sería guardar los datos en lugar de una base de datos, en un fichero de texto o en un fichero XML, pero es más cómodo utilizar una base de datos.

4.1.3. Alternativa de diseño 3

Esta alternativa consiste en guardar los datos de la aplicación en una base de datos y los vídeos en un servidor externo:

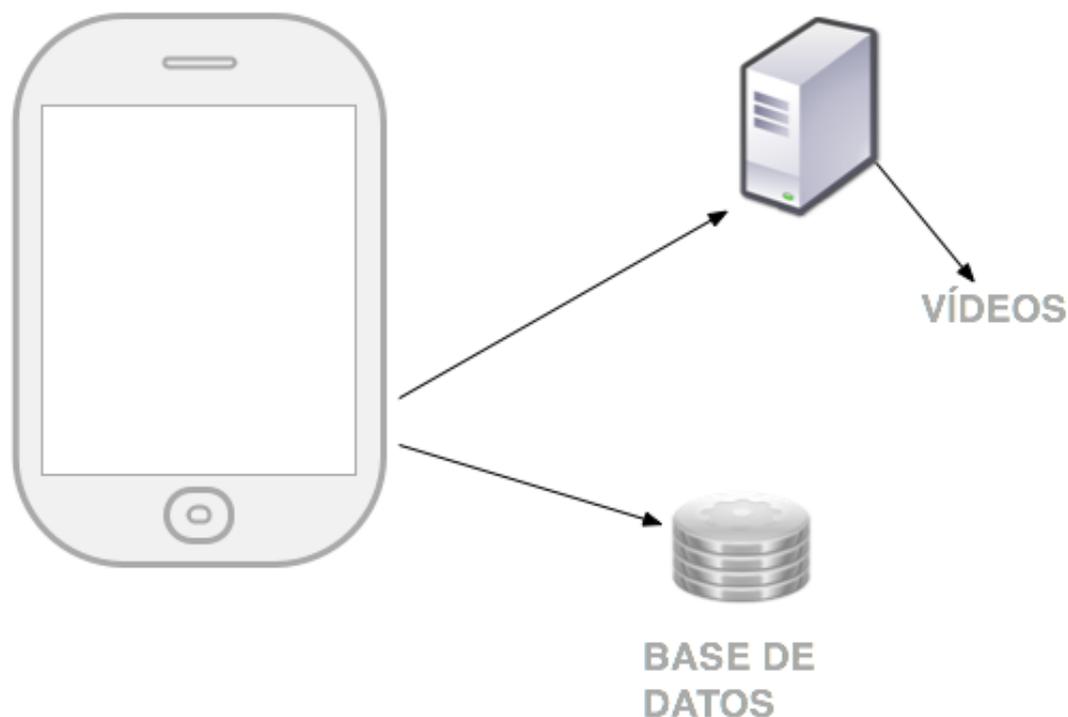


Ilustración 41. Alternativa de diseño 3

En este caso, al igual que en la alternativa anterior se ha pensado en guardar todos los textos en una base de datos que residiría en la aplicación, pero en cuanto a los vídeos de la aplicación, en este caso residirían en un servidor externo a la aplicación, al cual se accedería desde el código fuente. Esta alternativa presenta la ventaja que la aplicación ocuparía bastante menos, ya que los vídeos ahora ocupan la gran parte de la aplicación. Sin embargo, como desventaja para utilizar la aplicación de este modo ya existiría la restricción de que el usuario para ver estos vídeos debería hacer uso de internet.

4.1.4. Alternativa de Diseño 4

Esta última alternativa consiste en guardar los datos de la aplicación en una base de datos y que esta, al igual que los vídeos, resida en un servidor externo:

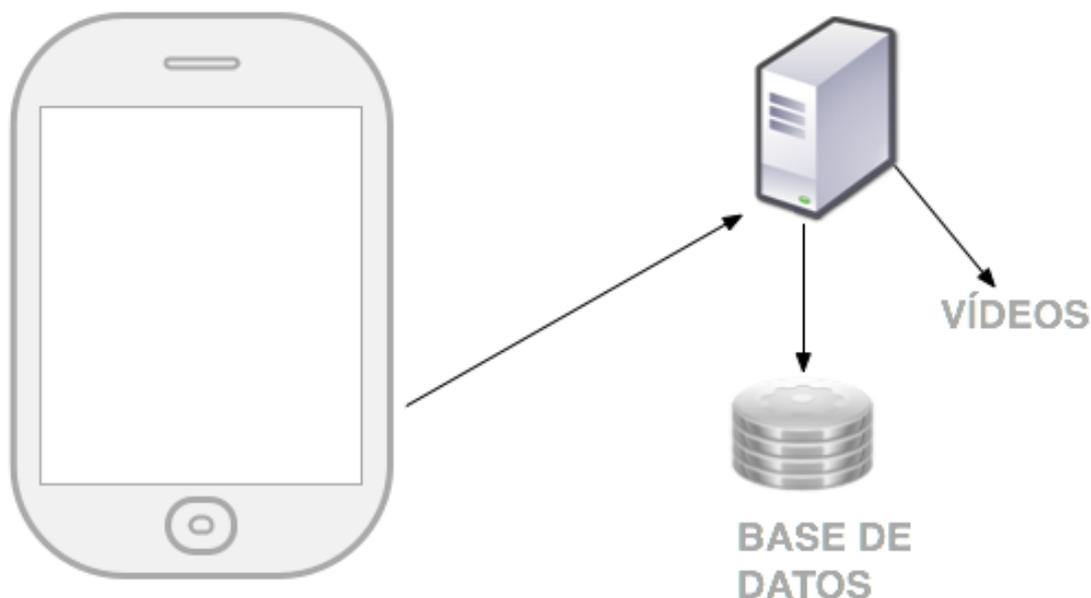


Ilustración 42. Alternativa de diseño 4

En este último caso, al igual que en las dos alternativas anteriores se ha pensado en guardar todos los textos en una base de datos, pero en este caso, al igual que ocurría en la alternativa anterior con los vídeos de la aplicación, residirían en un servidor externo a la aplicación. Por lo tanto, tanto los vídeos como la base de datos, se encontrarían en dicho servidor y la aplicación ocuparía menos, pero al igual que pasaba en la alternativa anterior, presenta la desventaja que para utilizar la aplicación existiría la restricción de que el usuario debería hacer uso de internet.

4.1.5. Elección de alternativa de diseño

Aunque en un principio se iba a realizar la alternativa de diseño 3, finalmente se realizó la alternativa de diseño 2, por falta de infraestructuras, además se consideró la desventaja que tenía la alternativa de diseño 3 y 4 de tener la necesidad de utilizar internet. La alternativa de diseño 1 no se tuvo en cuenta, ya que el código iba a ser muy difícil de modificar en caso de que fuese necesario.

4.2. Paper prototype

En este apartado se van a mostrar los diseños que se han realizado para desarrollar la aplicación siguiendo una guía. Dichos diseños se han realizado con la aplicación OmniGraffle que es utilizada para construir diversos tipos de diagramas, organigramas o ilustraciones. Esta aplicación es sencilla de utilizar, ya que se utiliza la técnica de drag-and-drop. Para poder hacer este prototipo se han utilizado diversas plantillas de iPhone que pueden ser descargadas y añadirlas a las librerías que ya trae incorporadas la aplicación.

Una vez mostrado los dos diseños se explicará en un apartado cual ha sido la elección y la causa de ello.

4.2.1. Diseño 1

En este apartado se va a mostrar el primer diseño que se realizó.



Ilustración 43. Diseño 1 Vista logo



Ilustración 44. Diseño 1 Vista Bienvenida



Ilustración 45. Diseño 1 Vista Temas



Ilustración 46. Diseño 1 Vista Detalle

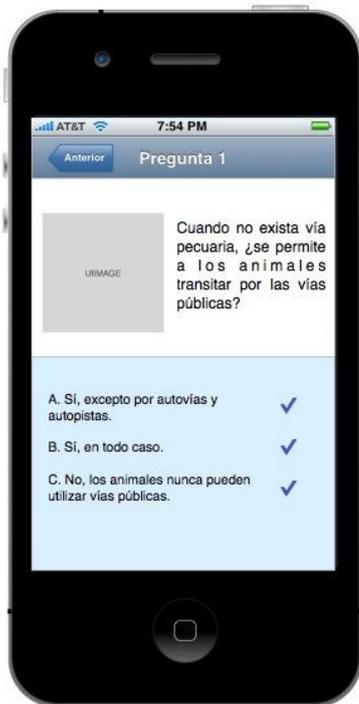


Ilustración 47. Diseño 1 Vista Pregunta 1

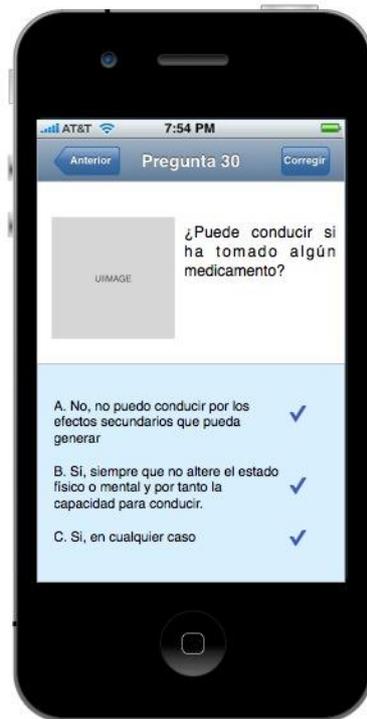


Ilustración 48. Diseño 1 Vista Última pregunta

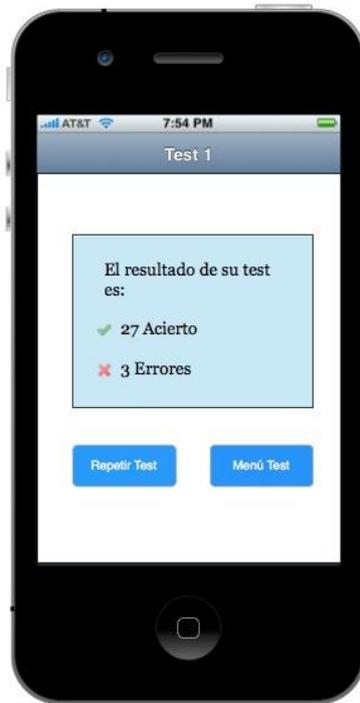


Ilustración 49. Diseño 1 Vista resultados

4.2.2. Diseño 2

A continuación, se puede observar el segundo prototipo:



Ilustración 50. Diseño 2 Vista Logo



Ilustración 51. Diseño 2 Vista Bienvenida



Ilustración 52. Diseño 2 Vista Temas



Ilustración 53. Diseño 2 Vista Detalle

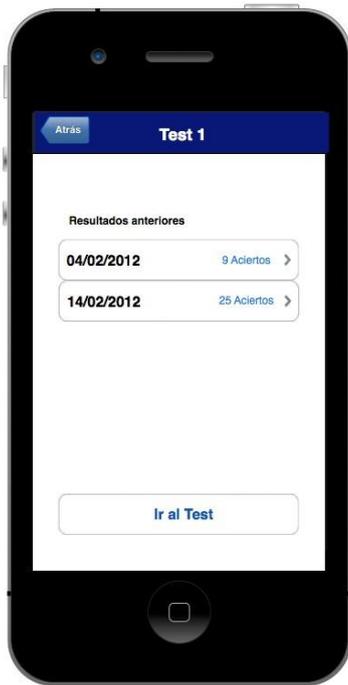


Ilustración 55. Diseño 2 Vista Resultados anteriores

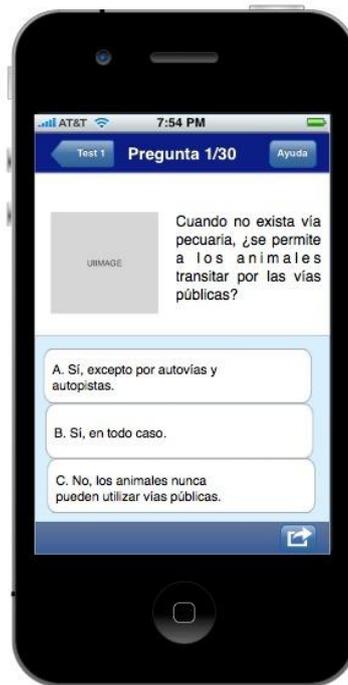


Ilustración 57. Diseño 2 Vista Pregunta 1

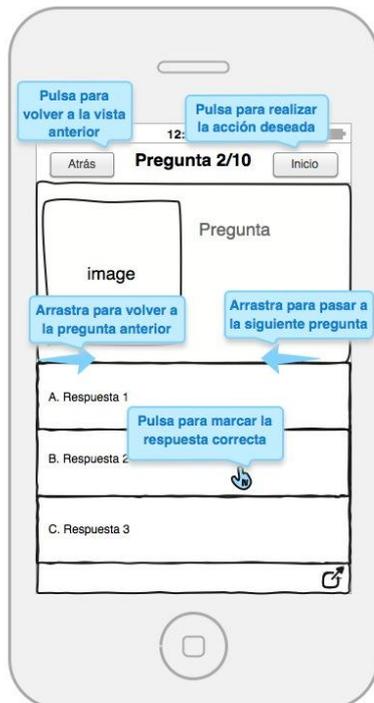


Ilustración 54. Diseño 2 Vista Ayuda

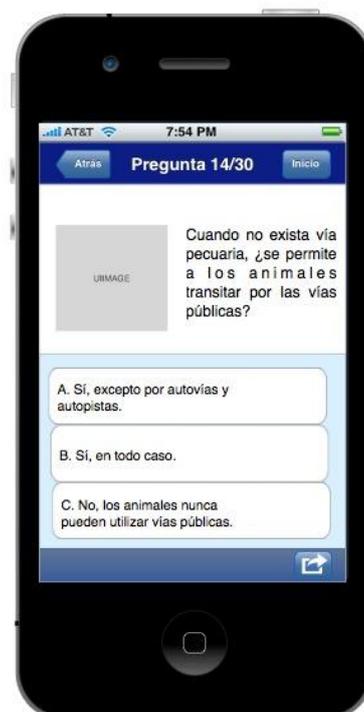


Ilustración 56. Diseño 2 Vista Pregunta 14

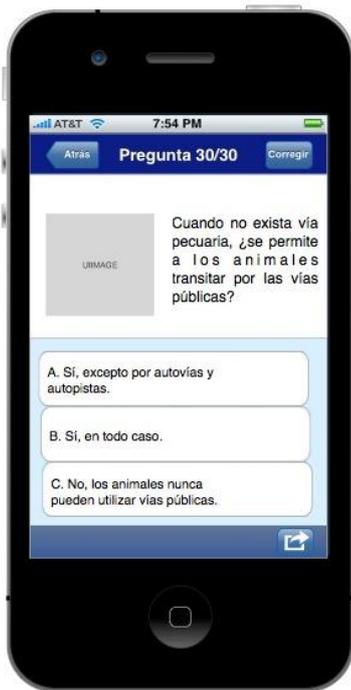


Ilustración 58. Diseño 2 Vista Pregunta final



Ilustración 59. Diseño 2 Vista Corrección

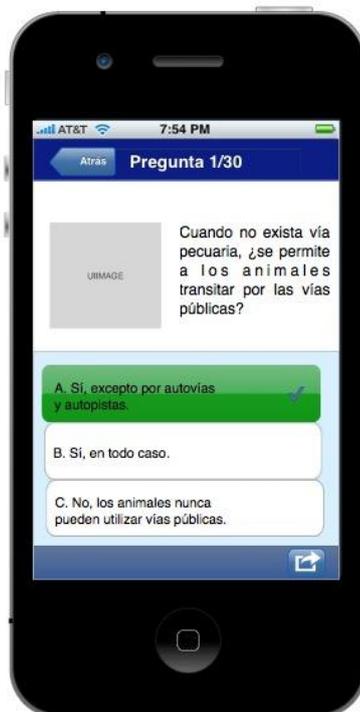


Ilustración 60. Diseño 2 Vista Corrección Pregunta Acertada

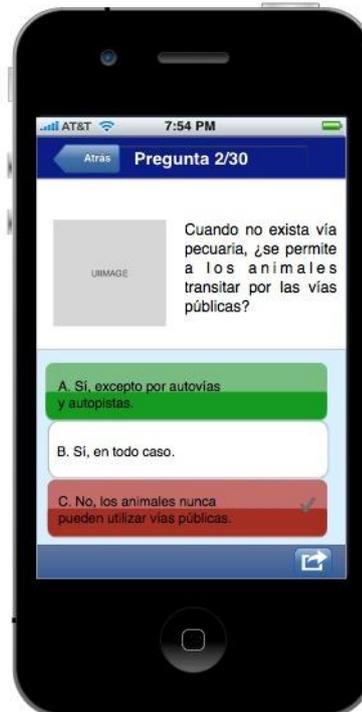


Ilustración 61. Diseño 2 Vista Corrección Pregunta Fallada

4.2.3. Elección de diseño

Para la realización de la aplicación se ha elegido el Diseño 2, por varios motivos que serán explicados a continuación:

- En la vista en la que se muestra una lista de todos los temas, se puede observar que no se ha modificado prácticamente nada, únicamente se ha modificado el color de la barra de navegación para mejorar todo el diseño de la aplicación.
- En la vista de detalle, en el diseño 2 se ha modificado toda la estructura, ya que se consideró que la información se vería mejor en una única celda y que sería más intuitivo para el usuario. Además, en este diseño se da la posibilidad de que en un tema se puedan realizar más de un test, y se va al test señalado para tener más información que antes.
- En la vista de los resultados anteriores del test seleccionado, es exclusiva del diseño 2, ya que se pensó en añadir la funcionalidad de poder ver el resultado que se había obtenido al realizar un test en una fecha determinada, además si se selecciona dicho test, se puede ver el resultado del test en detalle. Además, esta vista es la que da la posibilidad de ir a realizar el test y no la de detalle como ocurría en el diseño 1.
- Con respecto a la vista de las preguntas, en el diseño 2 se muestran 3 imágenes para poder observar el botón que sale a la derecha de la barra de navegación, en el que se pueden observar las diferentes opciones que hay en las diferentes preguntas (si es la pregunta 1 aparece el botón de ayuda, si es la última pregunta aparece el botón para corregir, y si no es ninguna de esta saldrá el botón para volver al inicio), esta es una de las ventajas que tiene el diseño 2 con respecto al diseño 1 en esta vista, ya que se le da mayor funcionalidad, además en el diseño 2 se puede observar en la barra de navegación el número de preguntas totales que hay. Por último también se ha modificado la elección de la respuesta, ya que según los estándares que suelen seguir las aplicaciones en iOS, es mejor hacer la elección con una celda, además que con esta opción da la posibilidad de marcar la respuesta que se había contestado de una manera más visible. Además se ha añadido la funcionalidad de compartir en redes sociales (botón abajo a la derecha), que anteriormente no estaba.
- Por último, en el diseño 1, únicamente se muestra únicamente el número de preguntas acertadas y el número de preguntas falladas, mientras que en el diseño 2, se muestra en detalle que pregunta se ha acertado y cuál se ha fallado, dando además la posibilidad de ver dicha pregunta de nuevo resaltando la respuesta contestada (con un tick y en caso de que sea errónea, marcada la celda en rojo) y la respuesta correcta (marcando la celda en verde).

4.3. Diagrama de Componentes

En este apartado vamos a mostrar el diagrama de componentes de la aplicación. Un diagrama de componentes ilustra las piezas del software, controladores embebidos, etc. que conforman un sistema. Estos diagramas tienen un nivel más alto de abstracción que el mostrado anteriormente, diagrama de clases, ya que normalmente un componente se implementa por una o más clases en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema. Estos diagramas contienen:

- Componentes.
- Interfaces.
- Relaciones de dependencia, generalización, asociación y realización.
- Paquetes o subsistemas. [7]

El diagrama de componentes es el siguiente:

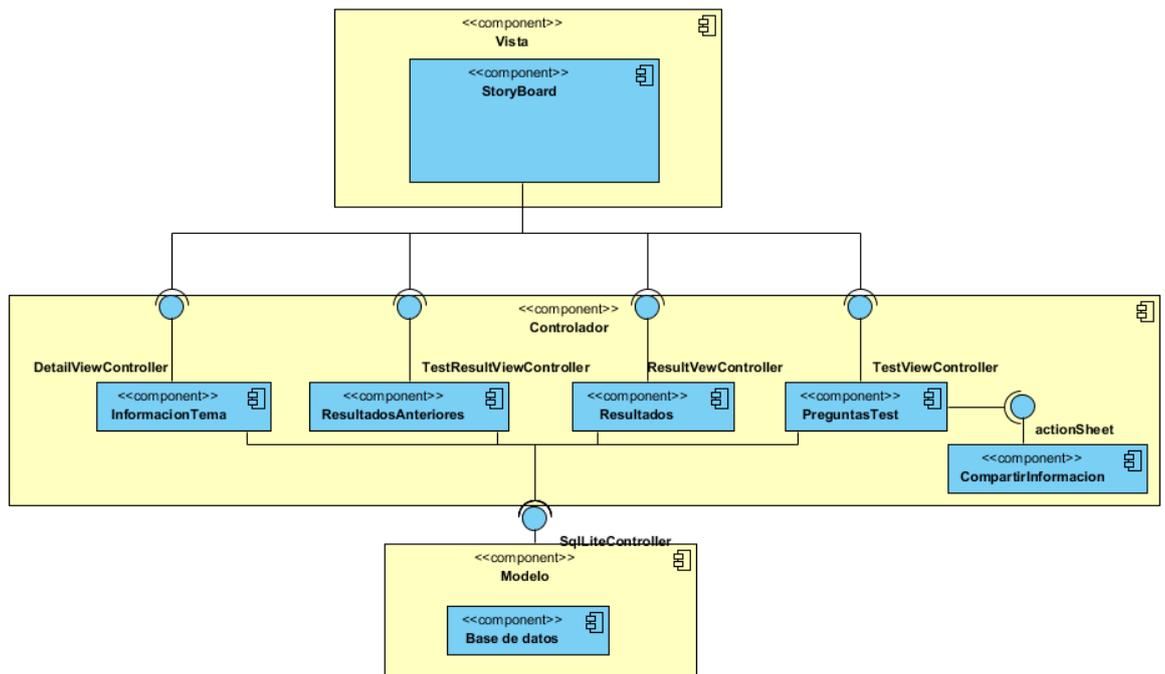


Ilustración 62. Diagrama de componentes

En el diagrama se puede ver cómo está estructurada la aplicación en un modelo-vista-controlador. Dentro de la capa vista, se encuentra únicamente el StoryBoard, esto es una peculiaridad de trabajar con un Xcode superior a la versión 4.2.1, ya que en estas versiones en el StoryBoard se unifican todas las vistas, aunque cada una de estas vistas esté identificada con una clase, mientras que en versiones anteriores para cada vista existía un fichero “.xib” que representaba cada vista.

Además, más internamente en el diagrama, podemos observar que la “Base de Datos” ofrece una única interfaz para “InformacionTema”, “ResultadosAnteriores”, “Resultados” y para “PreguntasTest”, y cada una de estas le ofrece una interfaz diferente al “Storyboard”, por lo explicado anteriormente, que el StoryBoard está compuesto por varias interfaces. Además “CompartirInformación” le ofrece una interfaz a “PreguntasTest”, para que esta pueda compartir la información deseada en Facebook, Twitter o por email.

4.4. Diagrama de Clases

A continuación se va a mostrar el diagrama de clases de la aplicación que se ha desarrollado, mostrando cada clase con sus atributos y operaciones y las relaciones que existen entre ellas. Una vez mostrado el diagrama, se especificará cada clase junto a los atributos y métodos correspondientes:

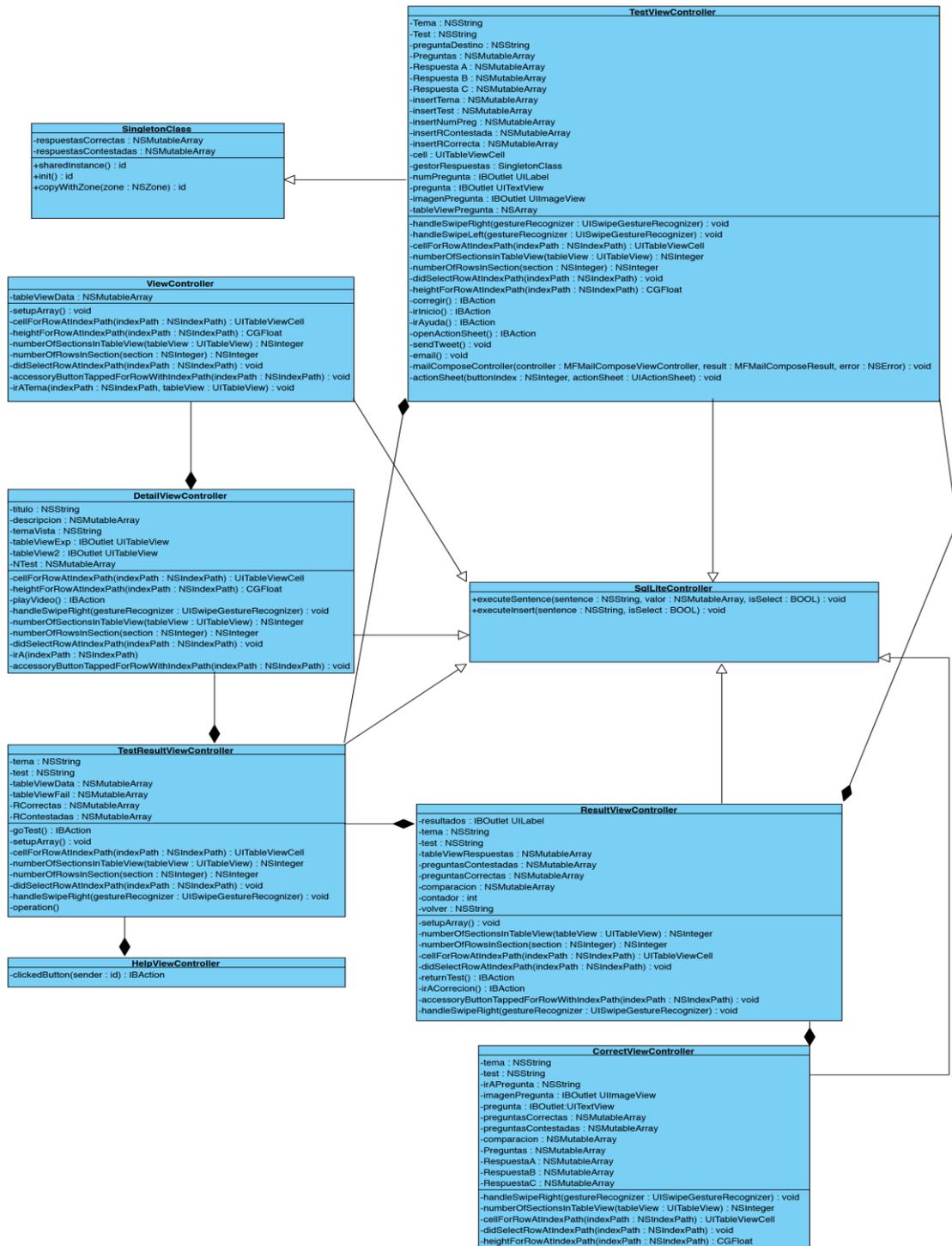


Ilustración 63. Diagrama de Clases

A continuación se va a especificar de cada clase sus métodos y atributos:

Nombre	ViewController
Atributos	<ul style="list-style-type: none"> • tableViewData: es de tipo NSMutableArray y sirve para almacenar los nombres de los diferentes temas para mostrarlos en la pantalla inicial
Métodos	<p style="text-align: center;">setupArray()</p> <p><i>Descripción:</i> Coge los datos de la base de datos y los mete en un array para mostrarlos en la pantalla principal</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">numberOfSectionsInTableView(UITableView tableView)</p> <p><i>Descripción:</i> Nos devuelve el número de secciones que tiene la tabla</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">numberOfRowsInSection(NSInteger section)</p> <p><i>Descripción:</i> Nos devuelve el número de filas que tendrá cada sección que tiene la tabla contabilizando los datos que se mostrarán</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">cellForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza la tabla que se mostrará y como se mostrarán sus datos</p> <p><i>Tipo:</i> UITableViewCell</p> <p style="text-align: center;">heightForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza el tamaño de cada fila</p> <p><i>Tipo:</i> CGFloat</p> <p style="text-align: center;">irATema(NSIndexPath indexPath, UITableView tableView)</p> <p><i>Descripción:</i> Nos redirige al tema seleccionado pasándole los parámetros necesarios a la otra vista</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">didSelectRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Nos redirige al tema seleccionado al pulsar en la celda, llamando al método irATema</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">accessoryButtonTappedForRowWithIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Nos redirige al tema seleccionado al pulsar en el checkmark, llamando al método irATema</p> <p><i>Tipo:</i> void</p>

Tabla 19. Clase ViewController

Nombre	DetailViewController
Atributos	<ul style="list-style-type: none"> • titulo: es de tipo NSString y coge el valor de la vista anterior y lo muestra en la barra superior(Navigation Bar) • descripcion: es de tipo NSMutableArray y sirve para almacenar las breves descripciones de los diferentes temas para mostrarlos en la tabla en la cual se pulsa para ver el vídeo • temaVista: es de tipo NSString y sirve para identificar el tema en el que nos encontramos • tableViewExp: es de tipo IBOutlet UITableView y sirve para mostrar una tabla en la que se podrá pulsar para ver el vídeo • tableView2: es de tipo IBOutlet UITableView y sirve para mostrar en una tabla los diferentes test que tenga un tema en la que se podrá pulsar para ir al test correspondiente • NTest: es de tipo NSMutableArray y sirve para mostrar en tableView2 los diferentes test
Métodos	<p style="text-align: center;">handleSwipeRight(UISwipeGestureRecognizer gestureRecognizer)</p> <p><i>Descripción:</i> Permite volver a la vista anterior deslizando el dedo hacia la derecha</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">numberOfSectionsInTableView(UITableView tableView)</p> <p><i>Descripción:</i> Devuelve el número de secciones que tiene la tabla</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">numberOfRowsInSection(NSInteger section)</p> <p><i>Descripción:</i> Devuelve el número de filas que tendrá cada sección de cada tabla contabilizando los datos que se mostrarán</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">cellForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza las dos tablas que se mostrarán y como se mostrarán sus datos</p> <p><i>Tipo:</i> UITableViewCell</p> <p style="text-align: center;">heightForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza el tamaño de cada fila en cada tabla</p> <p><i>Tipo:</i> CGFloat</p> <p style="text-align: center;">irA (NSIndexPath indexPath, UITableView tableView)</p> <p><i>Descripción:</i> Redirige al test seleccionado o al video dependiendo en que tabla pulses, además si pulsamos en la tabla de los diferentes test se le pasará los parámetros necesarios a la siguiente vista</p> <p><i>Tipo:</i> void</p>

	didSelectRowAtIndexPath(NSIndexPath indexPath)
	<i>Descripción:</i> Redirige al video o test seleccionado al pulsar en la celda, llamando al método irA
	<i>Tipo:</i> void
	accessoryButtonTappedForRowWithIndexPath(NSIndexPath indexPath)
	<i>Descripción:</i> Redirige al video o test seleccionado al pulsar en el checkmark, llamando al método irA
	<i>Tipo:</i> void
	playVideo()
	<i>Descripción:</i> Reproduce el vídeo
	<i>Tipo:</i> void

Tabla 20. Clase DetailViewController

Nombre	TestResultViewController
Atributos	<ul style="list-style-type: none"> • tema: es de tipo NSString y sirve para identificar el tema en el que nos encontramos • test: es de tipo NSString y sirve para identificar el test en el que nos encontramos • tableViewFail: es de tipo NSMutableArray y sirve para almacenar el número de aciertos obtenidos en el test seleccionado para mostrarlos en la tabla junto a la fecha en la que se realizó el test • tableViewData: es de tipo NSMutableArray y sirve para almacenar las fechas en las que se realizó el test seleccionado para mostrarlos en la tabla junto a los aciertos • RCorrectas: es de tipo NSMutableArray y sirve para almacenar las respuestas correctas del test seleccionado • RContestadas: es de tipo NSMutableArray y sirve para almacenar las respuestas contestadas de la fecha seleccionada
Métodos	<p style="text-align: center;">handleSwipeRight(UISwipeGestureRecognizer gestureRecognizer)</p> <p><i>Descripción:</i> Permite volver a la vista anterior deslizando el dedo hacia la derecha</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">numberOfSectionsInTableView(UITableView tableView)</p> <p><i>Descripción:</i> Devuelve el número de secciones que tiene la tabla</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">numberOfRowsInSection(NSInteger section)</p> <p><i>Descripción:</i> Devuelve el número de filas que tendrá cada sección de la tabla contabilizando los datos que se mostrarán</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">cellForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza la tabla que se mostrará y como se mostrarán sus datos</p> <p><i>Tipo:</i> UITableViewCell</p> <p style="text-align: center;">didSelectRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Redirige a los resultados del test seleccionado al pulsar en la celda, pasándole los parámetros necesarios para poder mostrar los aciertos y fallos obtenidos.</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">setupArray()</p> <p><i>Descripción:</i> Coge la fecha y los aciertos de la base de datos y los mete en dos arrays para mostrarlos en la tabla de la vista</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">goTest()</p> <p><i>Descripción:</i> Redirige al pulsar el botón, a las preguntas del test que se ha seleccionado anteriormente.</p> <p><i>Tipo:</i> IBAction</p>

Tabla 21. Clase TestResultViewController

Nombre	TestViewController
Atributos	<ul style="list-style-type: none"> • Tema: es de tipo NSString y sirve para identificar el tema en el que nos encontramos • Test: es de tipo NSString y sirve para identificar el test en el que nos encontramos • preguntaDestino: es de tipo NSString y sirve para acceder a la pregunta siguiente en la que estamos • numPregunta: es de tipo IBOutlet UILabel y sirve para mostrar el número de pregunta en la que nos encontramos • pregunta: es de tipo IBOutlet UITextView y sirve para mostrar el texto de la pregunta que estamos viendo en el test • imagenPregunta: es de tipo IBOutlet UIImageView y sirve para mostrar la imagen de la pregunta correspondiente • gestorRespuestas: es del tipo de la clase SingletonClass y sirve para gestionar los valores de las respuestas, tanto para las respuestas correctas como para las contestadas • cell: es de tipo UITableViewCell y será la que gestionará las celdas de la tabla de las preguntas • Preguntas: es de tipo NSMutableArray y sirve para recuperar de la base de datos la pregunta que se mostrará en la vista • RespuestaA: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta A que se mostrará • RespuestaB: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta B que se mostrará • RespuestaC: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta C que se mostrará • insertTema: es de tipo NSMutableArray y sirve para insertar el número del tema del test que se ha realizado en la base de datos • insertTest: es de tipo NSMutableArray y sirve para insertar el número de test que se ha realizado en la base de datos • insertNumPreg: es de tipo NSMutableArray y sirve para insertar el numero de cada pregunta que se ha contestado en la base de datos • insertRContestada: es de tipo NSMutableArray y sirve para insertar cada respuesta contestada a cada pregunta que se ha contestado en la base de datos • insertRCorrecta: es de tipo NSMutableArray y sirve para

	insertar cada respuesta correcta de todas las pregunta en la base de datos
Métodos	handleSwipeRight(UISwipeGestureRecognizer gestureRecognizer) <i>Descripción:</i> Permite volver a la vista anterior, en el caso de que sea la pregunta 1, o volver a la pregunta anterior si nos encontramos en cualquiera de las otras preguntas. Esto se realiza deslizando el dedo hacia la derecha <i>Tipo:</i> void
	handleSwipeLeft(UISwipeGestureRecognizer gestureRecognizer) <i>Descripción:</i> Permite ir a la pregunta siguiente excepto en el caso de que sea la última pregunta que ya no se podrá pasar a una pregunta posterior <i>Tipo:</i> void
	numberOfSectionsInTableView(UITableView tableView) <i>Descripción:</i> Devuelve el número de secciones que tiene la tabla <i>Tipo:</i> NSInteger
	numberOfRowsInSection(NSInteger section) <i>Descripción:</i> Devuelve el número de filas que tendrá cada sección de la tabla contabilizando los datos que se mostrarán <i>Tipo:</i> NSInteger
	cellForRowAtIndexPath(NSIndexPath indexPath) <i>Descripción:</i> Personaliza la tabla que se mostrará y como se mostrarán sus datos <i>Tipo:</i> UITableViewCell
	didSelectRowAtIndexPath(NSIndexPath indexPath) <i>Descripción:</i> Guarda la pregunta que se ha contestado en un array <i>Tipo:</i> void
	heightForRowAtIndexPath(NSIndexPath indexPath) <i>Descripción:</i> Personaliza el tamaño de cada fila en la tabla <i>Tipo:</i> CGFloat
	irInicio(id sender) <i>Descripción:</i> Redirige a la vista de inicio <i>Tipo:</i> IBAction
	corregir(id sender) <i>Descripción:</i> Redirige a la vista en la que se muestran los resultados obtenidos en el test y se procesan los datos que necesitará dicha vista para mostrar los datos, como por ejemplo calcular los errores <i>Tipo:</i> IBAction
	irAyuda(id sender) <i>Descripción:</i> Redirige a la vista de ayuda <i>Tipo:</i> IBAction
	openActionSheet(id sender)

Métodos	<i>Descripción:</i> Abre las acciones que se podrán usar para compartir en distintas redes
	<i>Tipo:</i> IBAcción
	sendTweet
	<i>Descripción:</i> Sirve para poder mandar un tweet
	<i>Tipo:</i> void
	email()
	<i>Descripción:</i> Sirve para gestionar lo que aparecerá por defecto en el email
	<i>Tipo:</i> void
	mailComposeController(MFMailComposeViewController controller, MFMailComposeResult result, NSError error)
	<i>Descripción:</i> Sirve para gestionar lo que se realizará con el email
<i>Tipo:</i> void	
actionSheet(UIActionSheet actionSheet, NSInteger buttonIndex)	
<i>Descripción:</i> Gestiona el botón que se pulsará con el actionSheet	
<i>Tipo:</i> void	

Tabla 22. Clase TestViewController

Nombre	ResultViewController
Atributos	<ul style="list-style-type: none"> • tema: es de tipo NSString y sirve para identificar el tema en el que nos encontramos • test: es de tipo NSString y sirve para identificar el test en el que nos encontramos • volver: es de tipo NSString y sirve para identificar desde que vista se ha accedido • resultados: es de tipo IBOutlet UILabel y mostrará los resultados obtenidos • preguntasContestadas: es de tipo NSMutableArray y sirve para comparar sus valores con preguntasCorrectas • preguntasCorrectas: es de tipo NSMutableArray y sirve para comparar sus valores con preguntasContestadas • comparacion: es de tipo NSMutableArray y sirve para guardar la comparación entre preguntasCorrectas y preguntasContestadas • tableViewRespuestas: es de tipo NSMutableArray y sirve para guardar el numero de la pregunta de cada respuesta • contador: es de tipo int y contabiliza el número de errores que se ha cometido
Métodos	<p style="text-align: center;">handleSwipeRight(UISwipeGestureRecognizer gestureRecognizer)</p> <p><i>Descripción:</i> Permite volver a la vista anterior, en el caso de que la vista en la que se viene sea en la que se selecciona un test ya realizado</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">setupArray()</p> <p><i>Descripción:</i> inicializa los valores de la tabla de cada pregunta</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">numberOfSectionsInTableView(UITableView tableView)</p> <p><i>Descripción:</i> Devuelve el número de secciones que tiene la tabla</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">numberOfRowsInSection(NSInteger section)</p> <p><i>Descripción:</i> Devuelve el número de filas que tendrá cada sección de la tabla contabilizando los datos que se mostrarán</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">cellForRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Personaliza la tabla que se mostrará y como se mostrarán sus datos</p> <p><i>Tipo:</i> UITableViewCell</p> <p style="text-align: center;">didSelectRowAtIndexPath(NSIndexPath indexPath)</p> <p><i>Descripción:</i> Nos redirige al tema seleccionado al pulsar en la celda, llamando al método irACorreccion</p> <p><i>Tipo:</i> void</p>

	accessoryButtonTappedForRowWithIndexPath(NSIndexPath indexPath)
	<i>Descripción:</i> Nos redirige al tema seleccionado al pulsar en el checkmark, llamando al método irACorreccion
	<i>Tipo:</i> void
	returnTest(id sender)
	<i>Descripción:</i> Redirige a la vista en la que se ha seleccionado un test realizado anteriormente
	<i>Tipo:</i> IBAction
	irACorreccion(id sender)
	<i>Descripción:</i> Redirige a la corrección de la pregunta seleccionada pasándole los datos necesarios
	<i>Tipo:</i> IBAction

Tabla 23. Clase ResultViewController

Nombre	CorrectViewController
Atributos	<ul style="list-style-type: none"> • tema: es de tipo NSString y sirve para identificar el tema en el que nos encontramos • test: es de tipo NSString y sirve para identificar el test en el que nos encontramos • irAPregunta: es de tipo NSString y sirve para acceder a la pregunta siguiente en la que estamos • pregunta: es de tipo IBOutlet UITextView y sirve para mostrar el texto de la pregunta que estamos viendo en el test • imagenPregunta: es de tipo IBOutlet UIImageView y sirve para mostrar la imagen de la pregunta correspondiente • Preguntas: es de tipo NSMutableArray y sirve para recuperar de la base de datos la pregunta que se mostrará • RespuestaA: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta A que se mostrará • RespuestaB: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta B que se mostrará • RespuestaC: es de tipo NSMutableArray y sirve para recuperar de la base de datos la respuesta C que se mostrará • comparacion: es de tipo NSMutableArray y sirve para ver como se mostrará cada celda con los datos obtenidos de la vista anterior • preguntasCorrectas: es de tipo NSMutableArray y sirve para ver como se mostrará cada celda con los datos obtenidos de la vista anterior • preguntasContestadas: es de tipo NSMutableArray y sirve para ver como se mostrará cada celda con los datos obtenidos de la vista anterior
Métodos	<p style="text-align: center;">handleSwipeRight(UISwipeGestureRecognizer gestureRecognizer)</p> <p><i>Descripción:</i> Permite volver a la vista anterior deslizando el dedo hacia la derecha</p> <p><i>Tipo:</i> void</p> <p style="text-align: center;">numberOfSectionsInTableView(UITableView tableView)</p> <p><i>Descripción:</i> Devuelve el número de secciones que tiene la tabla de la vista</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">numberOfRowsInSection(NSInteger section)</p> <p><i>Descripción:</i> Devuelve el número de filas que tendrá cada sección de la tabla contabilizando los datos que se mostrarán</p> <p><i>Tipo:</i> NSInteger</p> <p style="text-align: center;">cellForRowAtIndexPath(NSIndexPath indexPath)</p>

	<i>Descripción:</i> Personaliza la tabla que se mostrará y como se mostrarán sus datos
	<i>Tipo:</i> UITableViewCell
	heightForRowAtIndexPath(NSIndexPath indexPath)
	<i>Descripción:</i> Personaliza el tamaño de cada fila
	<i>Tipo:</i> CGFloat

Tabla 24. Clase CorrectViewController

Nombre	HelpViewController
Atributos	--
Métodos	<p>clickedButton(id sender) <i>Descripcion:</i> Vuelve a la pantalla anterior al pulsar el botón <i>Tipo:</i> IBAction</p>

Tabla 25. Clase HelpViewController

Nombre	SingletonClass
Atributos	<ul style="list-style-type: none"> • respuestasContestadas: es de tipo NSMutableArray y sirve para gestionar los datos de las respuestas que contesta el usuario • respuestasCorrectas: es de tipo NSMutableArray y sirve para gestionar los datos de las respuestas correctas del test
Métodos	<p>sharedInstance() <i>Descripcion:</i> Sirve para compartir la instancia que queramos utilizar <i>Tipo:</i> SingletonClass</p> <p>init() <i>Descripcion:</i> Inicializamos la instancia deseada <i>Tipo:</i> id</p> <p>copyWithZone(NSZone zone) <i>Descripcion:</i> Sirve para no generar múltiples copias de la instancia <i>Tipo:</i> id</p>

Tabla 26. Clase SingletonClass

Nombre	SqlLiteController
Atributos	--
Métodos	<p>executeSentence (NSString sentence, NSMutableArray valor, BOOL isSelect) <i>Descripcion:</i> Se podrá ejecutar una consulta a la base de datos guardando su valor en <i>valor</i> <i>Tipo:</i> void</p> <p>executeInsert (NSString sentence, BOOL isSelect) <i>Descripcion:</i> Se podrá ejecutar una inserción a la base de datos <i>Tipo:</i> void</p>

Tabla 27. Clase SqlLiteController

4.5. Diagramas de Secuencia

En este apartado vamos a mostrar los diagramas de secuencia de la aplicación para cada historia de usuario. Un diagrama de secuencia es un tipo de diagrama de interacción en el que se muestran las clases como líneas de vida y junto a sus interacciones en el tiempo representadas como líneas desde el origen hasta el destino. Estos diagramas son útiles porque indican que objetos se comunican con qué otros objetos y que se pasan entre ellos. [8] A continuación se van a mostrar los diagramas de secuencia:

- ✓ Historia de usuario 1 – Ver información de cada tema:

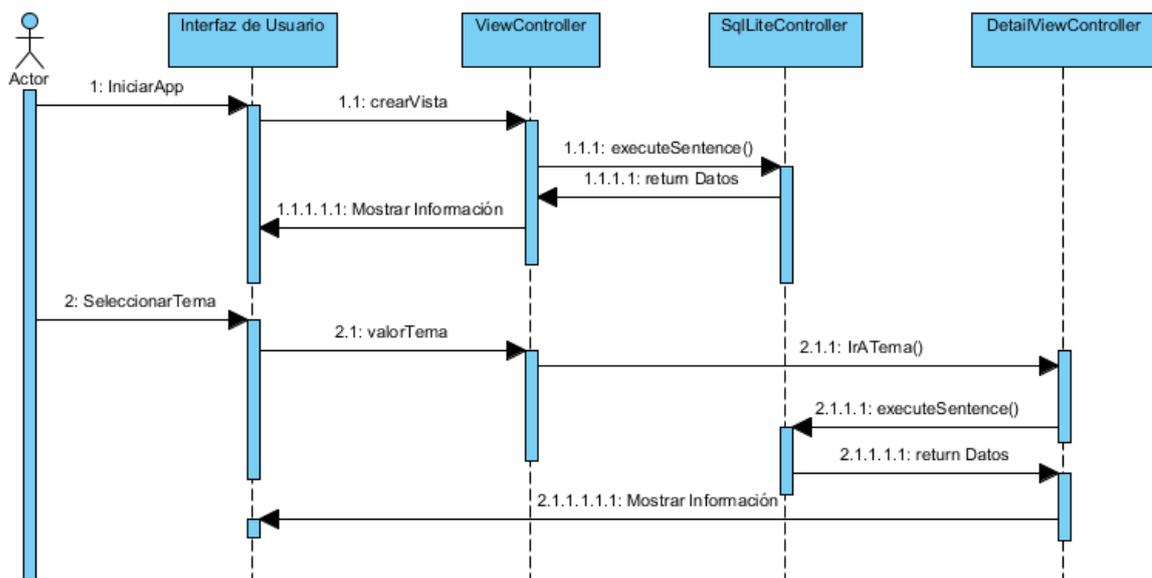


Ilustración 64. Diagrama de secuencia HU – 1

El anterior diagrama de secuencia comienza con el usuario iniciando la aplicación, al iniciar la aplicación se crea una vista automáticamente, ViewController, y dicha vista llama a la base de datos para coger los datos necesarios, se devuelve y se muestra la información. A continuación, el usuario selecciona un tema y la vista ViewController coge el valor seleccionado y se va a la vista de detalle, la cual ejecuta una consulta a la base de datos, esta le devuelve los datos y finalmente se muestran los datos.

✓ Historia de usuario 2 – Ver resultados de test anteriores:

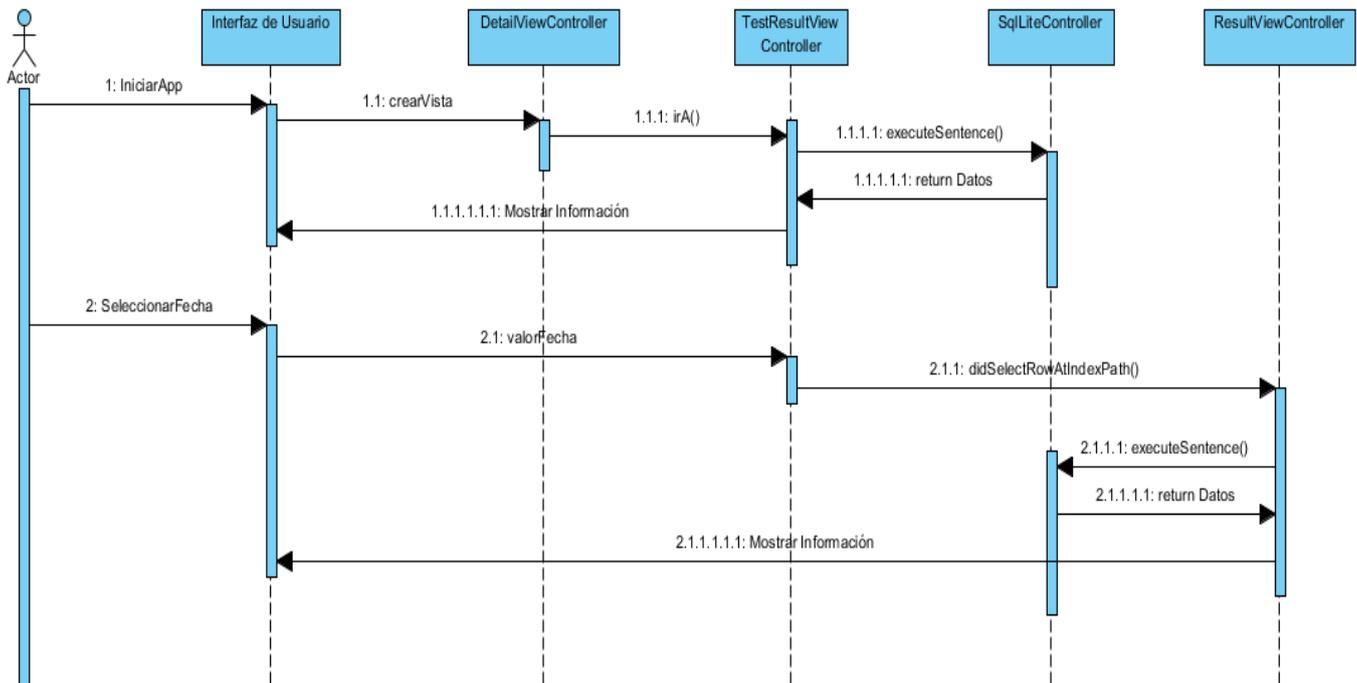


Ilustración 65. Diagrama de secuencia HU – 2

El anterior diagrama de secuencia, al igual que el primero comienza con el usuario iniciando la aplicación, por si se ha cerrado la aplicación, y al iniciar la aplicación se crea una vista automáticamente, DetailViewController, y dicha vista redirige a TestResultView Controller, para ver los resultados de los test ya realizados y esta clase llama a la base de datos para coger los datos necesarios, se devuelven los datos y se muestra la información. A continuación, el usuario selecciona una Fecha de un test y la vista TestResultView Controller coge el valor seleccionado y se va a la vista de los resultados de los test, ResultViewController, la cual ejecuta una consulta a la base de datos, esta le devuelve los datos y finalmente se muestran los datos.

✓ Historia de usuario 3 – Realizar test:

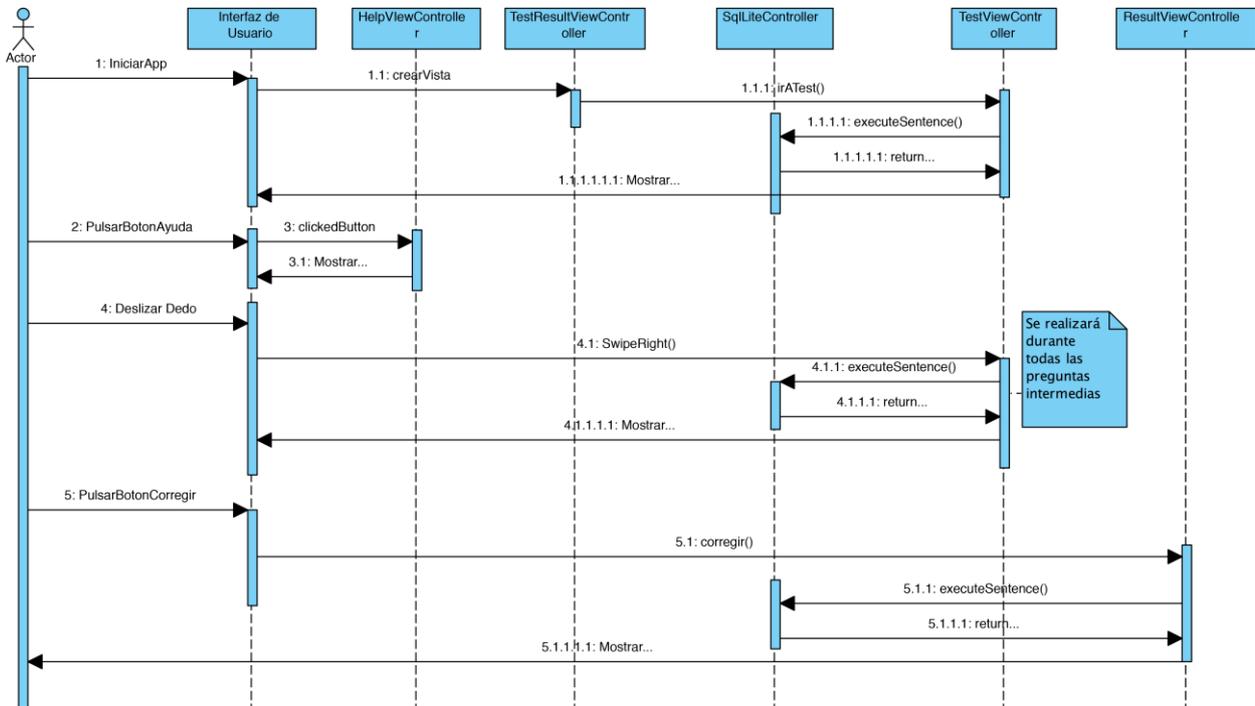


Ilustración 66. Diagrama de Secuencia HU - 3

El anterior diagrama de secuencia, al igual que el de la historia de usuario 2 comienza con el usuario iniciando la aplicación, por si ha cerrado la aplicación, y al iniciar la aplicación se crea una vista automáticamente, TestResultViewController, y dicha vista redirige a TestViewController, para poder realizar el test y esta clase llama a la base de datos para coger los datos necesarios, se devuelven los datos y se muestra la información. Además, existe la posibilidad de que el usuario pulse el botón de ayuda, que le redirige a la vista de ayuda y le muestra la información, en el caso de que no desee ver la ayuda deslizará el dedo sobre la pantalla, y le mostrará la siguiente pregunta, hasta llegar a la ultima, que tendrá que pulsar el botón corregir y le mandará a ResultViewController, el cual ejecuta una consulta a la base de datos, esta le devuelve los datos y finalmente se muestran los datos.

✓ Historia de usuario 4 – Ver resultados del test realizado:

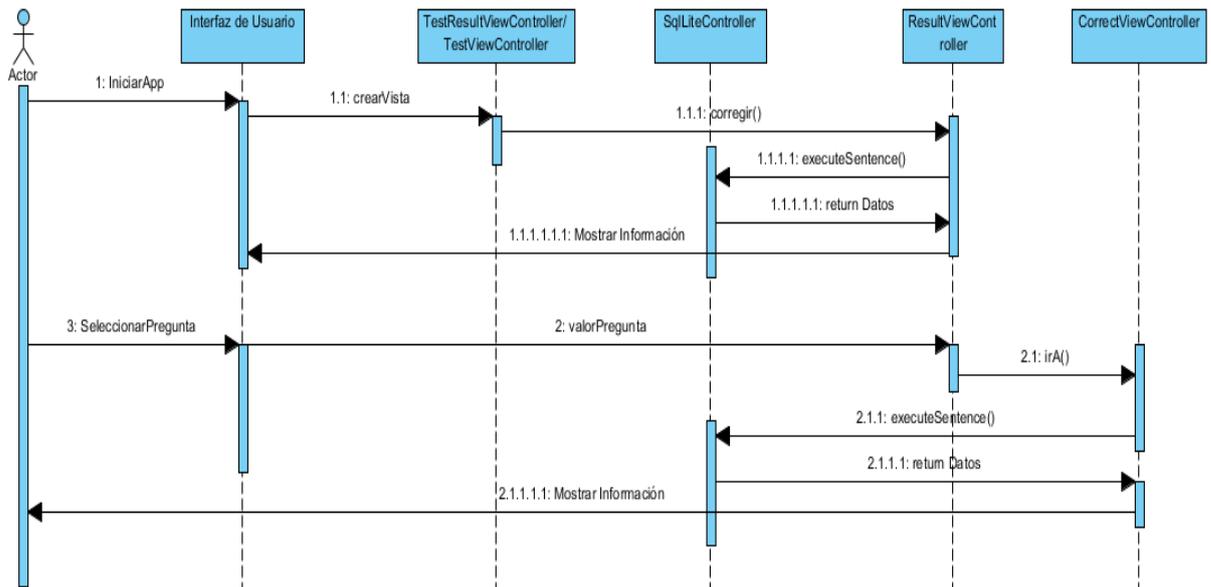


Ilustración 67. Diagrama de secuencia HU – 4

El diagrama de secuencia de la HU - 4, al igual que los dos anteriores comienza con el usuario iniciando la aplicación, por si ha cerrado la aplicación, y al iniciar la aplicación se crea una vista automáticamente, TestResultViewController o TestViewController, en este caso se ponen las dos clases como una, porque tienen la misma funcionalidad para acceder a los resultados, pero se accede desde vistas diferentes, y dichas vistas redirigen a ResultViewController, para ver los resultados del test seleccionado o realizado y esta clase llama a la base de datos para coger los datos necesarios, se devuelven los datos y se muestra la información. A continuación, el usuario selecciona una pregunta de los resultados y la vista ResultViewController coge el valor seleccionado y se va a la vista de la corrección de los resultados de los test, CorrectViewController, la cual ejecuta una consulta a la base de datos, esta le devuelve los datos y finalmente se muestran los datos.

- ✓ Historia de Usuario 5 – Compartir en redes sociales - Funcionalidad Twitter:

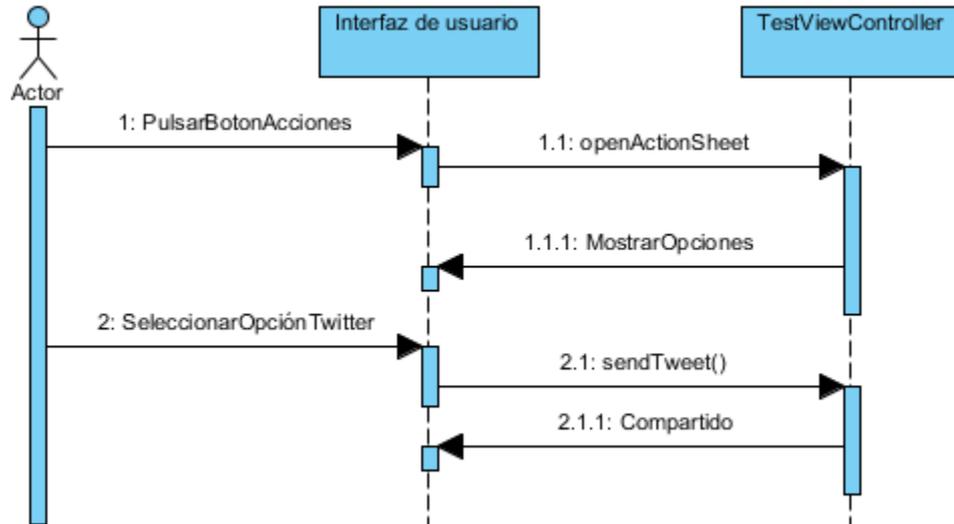


Ilustración 68. Diagrama de secuencia HU – 5 Funcionalidad Twitter

- ✓ Historia de usuario 5 – Compartir en redes sociales - Funcionalidad Facebook:

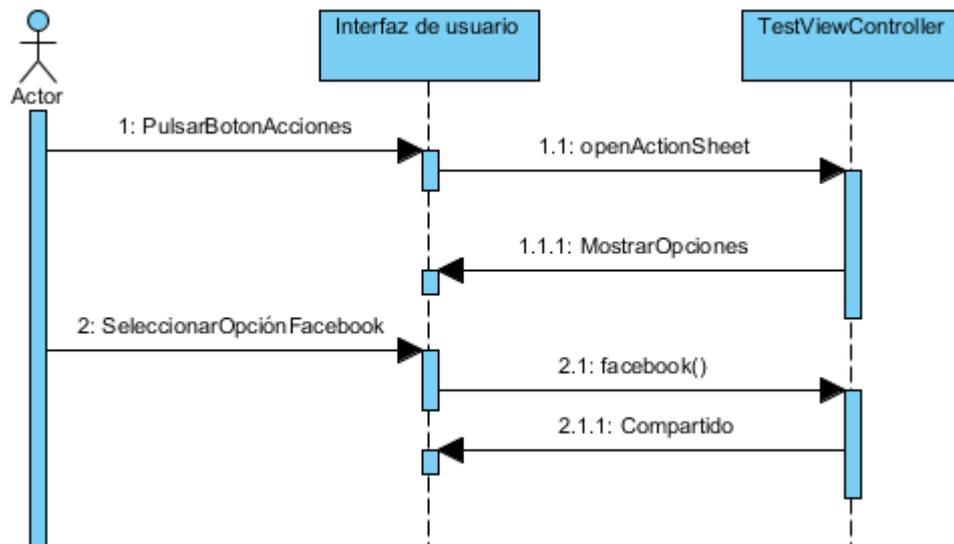


Ilustración 69. Diagrama de secuencia HU – 5 Funcionalidad Facebook

- ✓ Historia de usuario 5 – Compartir en redes sociales - Funcionalidad Email:

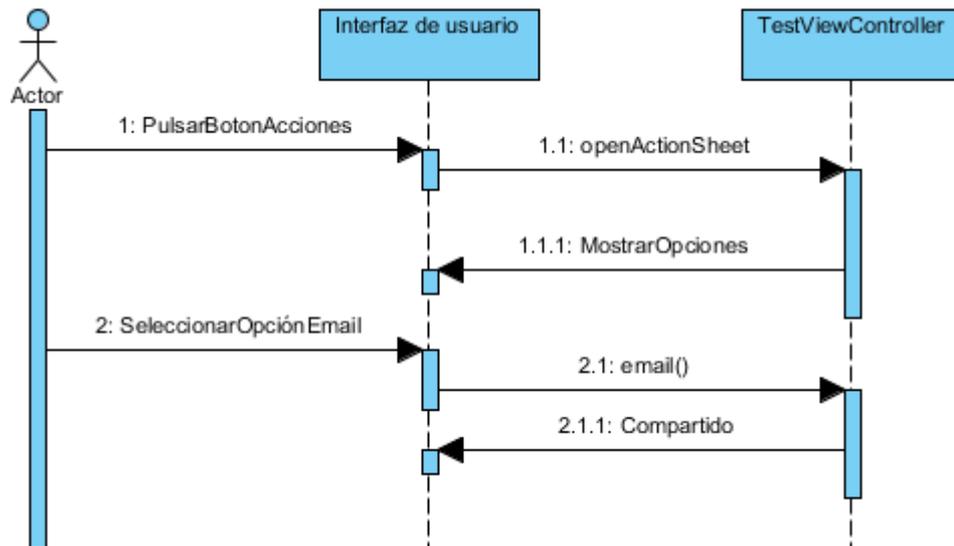


Ilustración 70. Diagrama de secuencia HU – 5 Funcionalidad email

Los tres diagramas anteriores tienen la misma funcionalidad pero con distinto fin, es decir, en ambos se desea publicar cierta información en redes sociales o enviar por correo, pero todos se hacen de la misma manera.

En estos diagramas cuando el usuario se encuentra en la vista de las preguntas, pulsa el botón para que se le muestren las acciones que puede realizar y la clase TestViewController le muestra las opciones, a continuación el usuario selecciona la opción deseada y la clase TestViewController se encarga de compartir dicha información llamando al método correspondiente.

4.6. Tarjetas CRC

Una vez realizados el diagrama de clases y los diagramas de secuencia se van a especificar las diferentes clases de la aplicación con las tarjetas CRC (Class Responsibility Collaborator), algo muy usual en las metodologías ágiles. Estas tarjetas son una ayuda para el diseño software orientado a objetos.

ViewController	
Superclases:	--
Subclases:	DetailViewController
Propiedades:	NSMutableArray tableViewData
Responsabilidades:	Clase responsable de mostrar los diferentes temas de la autoescuela obteniendo la información de la base de datos
Precondiciones:	Debe existir una base de datos con datos
Postcondiciones:	Se selecciona una celda y se pasa a la siguiente vista
Colaboraciones:	DetailViewController, TestViewController, ResultViewController

Tabla 28. Tarjeta CRC ViewController

DetailViewController	
Superclases:	ViewController
Subclases:	TestResultViewController
Propiedades:	NSString titulo NSMutableArray descripción NSString temaVista IBOutlet UITableView tableViewExp IBOutlet UITableView tableView2 NSMutableArray NTest
Responsabilidades:	Clase responsable de mostrar una breve explicación del tema y un video con el mismo fin, además de mostrar los diferentes test que tiene dicho tema
Precondiciones:	Haber seleccionado un tema en la vista anterior
Postcondiciones:	Mostrar el vídeo o ir a los resultados anteriores del test seleccionado
Colaboraciones:	DetailViewController, TestResultViewController

Tabla 29. Tarjeta CRC DetailViewController

TestResultViewController	
Superclases:	DetailViewController
Subclases:	TestViewController
Propiedades:	NSString tema NSString test NSMutableArray tableViewData NSMutableArray tableViewFail NSMutableArray RCorrectas NSMutableArray RContestadas
Responsabilidades:	Clase responsable de mostrar, en caso de que existan, los resultados obtenidos del test realizado en una fecha determinado, además de dar la posibilidad de realizar dicho test
Precondiciones:	Haber seleccionado un test en la vista anterior
Postcondiciones:	Mostrar los resultados del test seleccionado o ir a las preguntas
Colaboraciones:	DetailViewController, TestViewController

Tabla 30. Tarjeta CRC TestResultViewController

TestViewController	
Superclases:	TestResultViewController
Subclases:	ResultViewController
Propiedades:	IBOutlet UILabel numPregunta IBOutlet UITextView pregunta IBOutlet UIImageView imagenPregunta NSString Tema NSString Test NSString preguntaDestino NSArray tableViewPregunta SingletonClass gestorRespuestas UITableViewCell cell NSMutableArray Preguntas NSMutableArray RespuestaA NSMutableArray RespuestaB NSMutableArray RespuestaC NSMutableArray insertTema NSMutableArray insertTest NSMutableArray insertNumPreg NSMutableArray insertRContestada NSMutableArray insertRCorrecta
Responsabilidades:	Clase responsable de mostrar las diferentes preguntas del test y tema seleccionado, esta clase se reutiliza para ver cada pregunta
Precondiciones:	Haber seleccionado un test en la vista anterior.
Postcondiciones:	Mostrar los resultados del test seleccionado o volver al inicio
Colaboraciones:	ViewController, HelpViewController, TestResultViewController, ResultViewController

Tabla 31. Tarjeta CRC TestViewController

ResultViewController	
Superclases:	TestViewController, TestResultViewController
Subclases:	CorrectViewController
Propiedades:	IBOutlet UILabel resultados NSString tema NSString test NSString volver NSMutableArray tableViewRespuestas NSMutableArray preguntasContestadas NSMutableArray preguntasCorrectas NSMutableArray comparacion int contador
Responsabilidades:	Clase responsable de mostrar los resultados de cada pregunta y los aciertos y fallos totales
Precondiciones:	Haber realizado un test completo o haber seleccionado en la vista de la clase TestResultViewController un test realizado anteriormente
Postcondiciones:	Mostrar el detalle de cada pregunta
Colaboraciones:	TestViewController, TestResultViewController, CorrectViewController

Tabla 32. Tarjeta CRC ResultViewController

CorrectViewController	
Superclases:	ResultViewController
Subclases:	--
Propiedades:	IBOutlet UIImageView imagenPregunta IBOutlet UITextView pregunta NSArray tableViewPregunta NSString tema NSString test NSString irAPregunta NSMutableArray preguntasCorrectas NSMutableArray preguntasContestadas NSMutableArray comparacion NSMutableArray Preguntas NSMutableArray RespuestaA NSMutableArray RespuestaB NSMutableArray RespuestaC
Responsabilidades:	Clase responsable de mostrar la especificación de cada pregunta, señalando cuál ha seleccionado el usuario y cuál es la correcta
Precondiciones:	Haber seleccionado una pregunta en la vista anterior
Postcondiciones:	--
Colaboraciones:	ResultViewController

Tabla 33. Tarjeta CRC CorrectViewController

HelpViewController	
Superclases:	TestViewController
Subclases:	--
Propiedades:	--
Responsabilidades:	Clase responsable de mostrar la ayuda de cómo interactuar con la vista de las preguntas
Precondiciones:	--
Postcondiciones:	--
Colaboraciones:	TestViewController

Tabla 34. Tarjeta CRC HelpViewController

SingletonClass	
Superclases:	--
Subclases:	TestViewController
Propiedades:	NSMutableArray respuestasCorrectas NSMutableArray respuestasContestadas
Responsabilidades:	Clase responsable de gestionar los arrays de las preguntas
Precondiciones:	--
Postcondiciones:	Inicializar una única vez el array requerido
Colaboraciones:	TestViewController

Tabla 35. Tarjeta CRC SingletonClass

SqlLiteController	
Superclases:	--
Subclases:	ViewController, Detail ViewController, TestResult ViewController, Test ViewController, Result ViewController, Correct ViewController
Propiedades:	--
Responsabilidades:	Clase responsable de gestionar como se interactúa con la base de datos
Precondiciones:	--
Postcondiciones:	Poder insertar o coger datos en la base de datos
Colaboraciones:	ViewController, Detail ViewController, TestResult ViewController, Test ViewController, Result ViewController, Correct ViewController

Tabla 36. Tarjeta CRC SqlLiteController

4.7. Base de datos

En este apartado se va a mostrar el diagrama entidad-relación de la base de datos y el modelo relacional. Para realizar este proyecto se ha utilizado SQLite que es una pequeña librería programada en C que implementa un motor de base de datos multiplataforma. En nuestro caso se ha utilizado como gestor de base de datos, pero puede ser usada como una extensión de PHP. [9]

A continuación, se va a mostrar el diagrama entidad-relación, que sirve principalmente para representar las entidades relevantes de la base de datos, así como sus interrelaciones y propiedades:

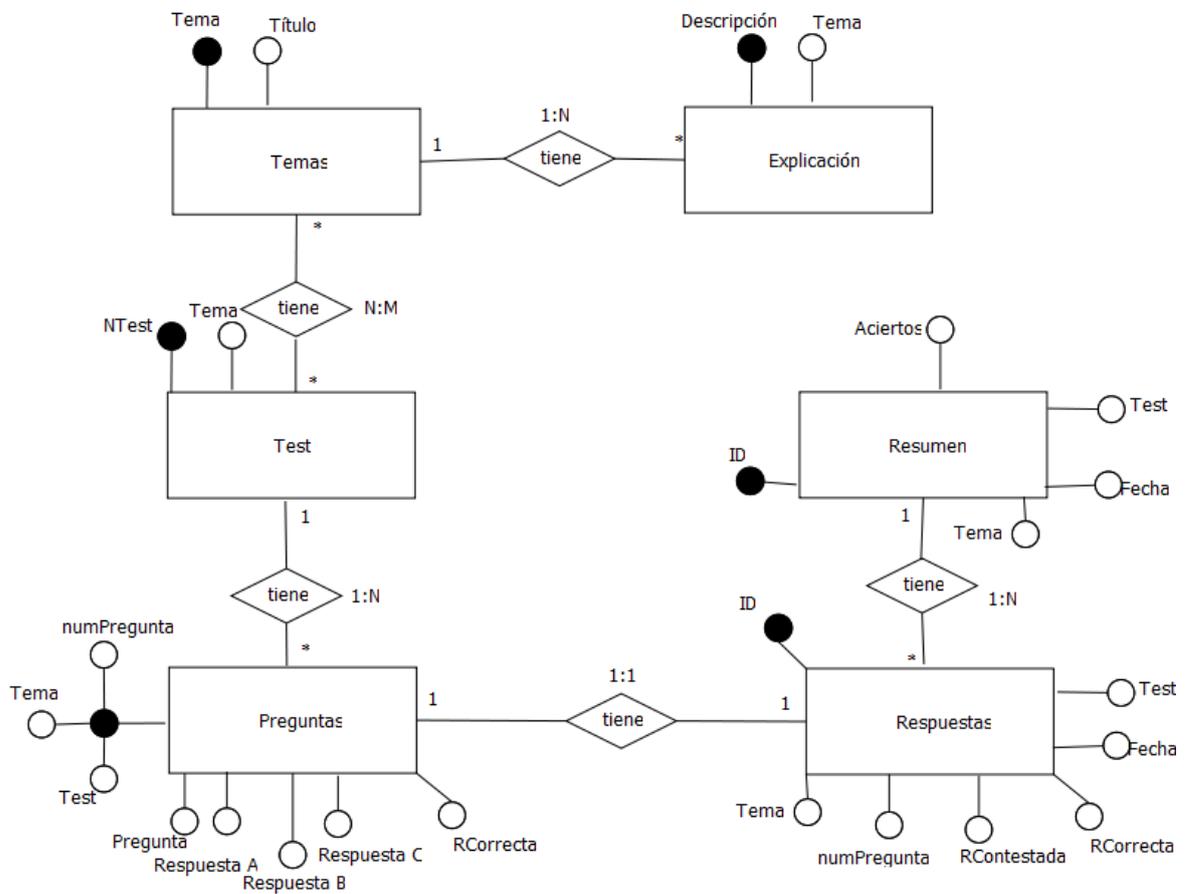


Ilustración 71. Diagrama entidad-relación

Seguidamente se mostrará el modelo relacional, que sirve para modelar y administrar datos dinámicamente, mediante relaciones:

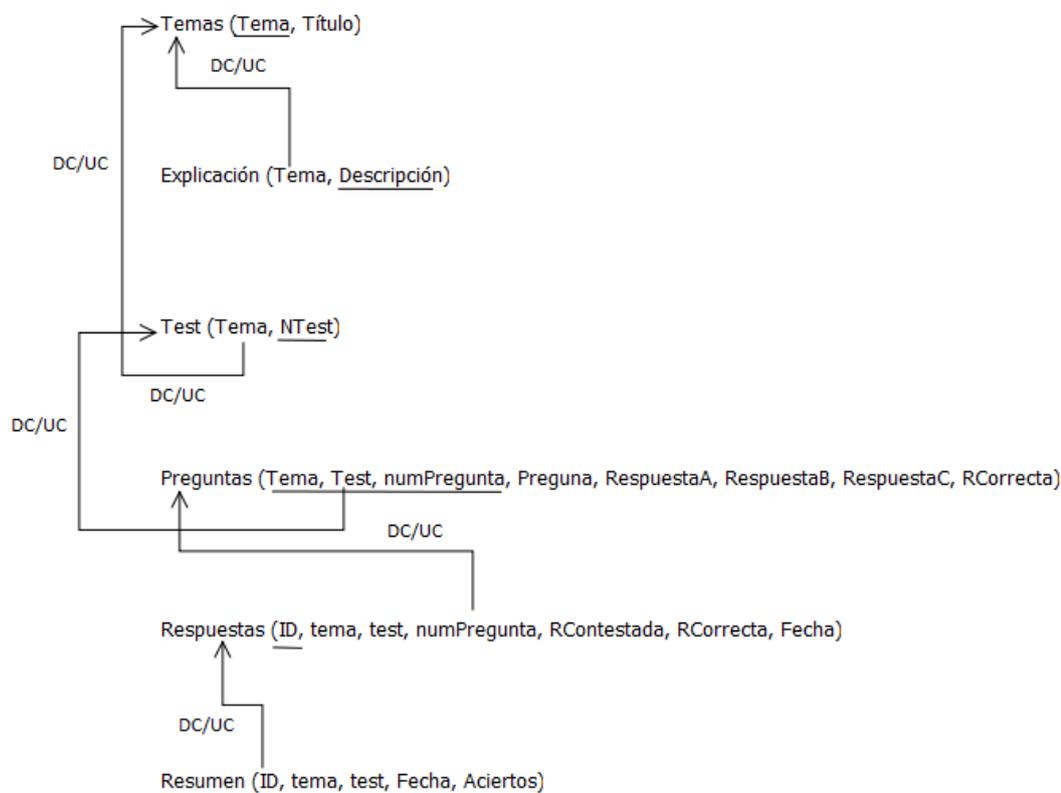


Ilustración 72. Modelo relacional

En el diagrama entidad-relación y modelo relacional mostrados anteriormente se pueden observar varias peculiaridades, como que todas las tablas tienen el atributo tema, se ha realizado de ese modo, para que sea más fácil de gestionar a la hora de acceder a estos datos.

Para realizar la aplicación se podría haber utilizado un sistema de ficheros, pero por facilidad y comodidad de uso, se ha utilizado una base de datos.

Capítulo 5: Pruebas del sistema

5. Pruebas del sistema

En este apartado de la memoria se van a especificar las pruebas que se van a realizar para comprobar que la aplicación funciona correctamente y cumple todas las funcionalidades descritas anteriormente. Todas estas pruebas se han ido realizando iteración a iteración, pero una vez terminada la aplicación se va a comprobar que se siguen pasando todas las pruebas.

A continuación, se va a mostrar la tabla modelo que se va a seguir para exponer las pruebas y que se vean de una forma más clara:

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – Y-X			dd-mm-aa	

Tabla 37. Tabla modelo pruebas de sistema

Donde:

- ID: es un identificador único de la prueba donde Y identifica el número de la historia de usuario y X identifica el número de prueba, el cuál comenzará en 1 e irá aumentado en una unidad por cada prueba.
- Propósito: identifica una descripción de la prueba y el fin que queremos obtener al realizar dicha prueba.
- Pasos: sirve para ayudar a un usuario que no sepa nada acerca de la aplicación a realizar la prueba.
- HU: identifica la historia de usuario con la que está relacionada la prueba, donde X puede ser un número comprendido entre 1 y 5.
- Fecha: indica la fecha en la que ha sido realizada la prueba.
- Resultado: indica si la prueba ha sido pasada satisfactoriamente o no, puede tener dos valores: éxito o fracaso.

Historia de usuario 1: Ver información de cada tema

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – 1 - 1	Al abrir la aplicación visualizar la vista de bienvenida de la aplicación y que desaparezca	1.Abrir la aplicación	04-05-12	Éxito
PA – 1 - 2	Al abrir la aplicación visualizar los 16 test con sus imágenes correspondientes	1. Abrir la aplicación	04-05-12	Éxito
PA – 1 - 3	Seleccionar un test y que te lleve a la vista de detalle	1.Abrir la aplicación 2.Seleccionar un tema	04-05-12	Éxito
PA – 1 - 4	Volver desde la vista de detalle a la vista de los temas y no exista ninguna celda seleccionada	1.Abrir la aplicación 2.Seleccionar un tema 3.Pulsar botón Temas	04-05-12	Éxito
PA – 1 - 5	Visualizar al pulsar en un tema una breve descripción del tema y los test que tiene dicho tema y poder ver un vídeo del tema al seleccionar la descripción	1.Abrir la aplicación 2.Seleccionar un tema 3.Pulsar en la celda donde aparece la descripción del tema	04-05-12	Éxito
PA – 1 - 6	En la vista de detalle poder seleccionar un test y que te redirija a la vista siguiente	1.Abrir la aplicación 2.Seleccionar un tema 3.Pulsar en una celda donde aparecen los test	04-05-12	Éxito

Tabla 38. Pruebas de sistema HU-1

Historia de usuario 2: Ver resultados de test anteriores

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – 2 -1	Una vez seleccionado un tema y un test abrir la vista con los resultados de test realizados anteriormente y en el caso de que no se haya realizado nunca que no se muestre ningún resultado. Además debe dar la posibilidad de realizar el test con el botón Ir al test	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test	04-05-12	Éxito
PA – 2 - 2	Una vez seleccionado un tema y test abrir la vista con los resultados de test realizados anteriormente y en el caso de que se haya realizado alguna vez dicho test se muestre la fecha en la que se hizo y el número de aciertos. Además debe dar la posibilidad de realizarlo con el botón Ir al test	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test	04-05-12	Éxito
PA – 2 – 3	Seleccionar un resultado de un test anterior y que te redirija a la vista en detalle de los resultados	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4.Seleccionar un resultado	04-05-12	Éxito
PA – 2 – 4	Desde la vista en detalle de los resultados poder volver a la vista de resultados anteriores y que no aparezca ninguno seleccionado	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4.Seleccionar un resultado 5. Pulsar botón Test X(donde x es el número de test seleccionado)	04-05-12	Éxito

Tabla 39. Pruebas de sistema HU-2

Historia de usuario 3: Realizar test

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – 3 - 1	Una vez seleccionado un tema y un test se debe y se haya pulsado el botón para ir al test, se debe abrir la vista donde aparecerá la información de cada pregunta	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test	05-05-12	Éxito
PA – 3 - 2	En la vista de la pregunta 1, aparecerá un botón de ayuda que te informará de cómo realizar el test. Aparecerá una vista con dicha información	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar botón Ayuda	05-05-12	Éxito
PA – 3 - 3	En la vista de ayuda dar la posibilidad de volver a la pregunta 1. Aparecerá de nuevo la vista con la pregunta	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar botón Ayuda 6. Pulsar botón volver	05-05-12	Éxito
PA – 3 - 4	En la vista de las preguntas, desde la pregunta 2 hasta la penúltima, deberá aparecer un botón de inicio que te permitirá volver al inicio de la aplicación	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Deslizar el dedo hacia la izquierda 6. Pulsar botón Inicio	05-05-12	Éxito
PA – 3 - 5	En la vista de las preguntas, en la última pregunta, deberá aparecer un botón para corregir el test que se ha realizado y al pulsarlo deberá aparecer la vista de los resultados	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Deslizar el dedo hacia la izquierda hasta la última pregunta e ir seleccionando o no una respuesta 6. Pulsar botón Corregir	05-05-12	Éxito
PA – 3 - 6	En la vista de las preguntas, poder ir de una pregunta a otra,	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test	05-05-12	Éxito

hacia la pregunta siguiente y anterior, y que en el caso de que se haya seleccionado una respuesta se te muestre la que has seleccionado al realizarlo	4. Pulsar el botón Ir al Test 5. Deslizar el dedo hacia la izquierda e ir seleccionando o no una respuesta 6. Deslizar el dedo hacia la derecha y comprobar que la respuesta sigue marcada
--	--

Tabla 40. Pruebas de sistema HU-3

Historia de usuario 4: Ver resultado del test realizado

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – 4 - 1	Una vez pulsado el botón corregir en la última pregunta del test se debe mostrar una lista de todas las preguntas indicando si se ha fallado o acertado cada pregunta, además de mostrar un resumen de todas ellas	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar un tema 3. Seleccionar un test 4. Pulsar el botón Ir al Test 5. Deslizar el dedo hacia la izquierda hasta la última pregunta e ir seleccionando o no una respuesta 6. Pulsar botón Corregir 	06-05-12	Éxito
PA – 4 - 2	Una vez seleccionado un test realizado anteriormente se debe mostrar una lista de todas las preguntas indicando si se ha fallado o acertado cada pregunta, además de mostrar un resumen de todas ellas	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar un tema 3. Seleccionar un test 4. Seleccionar un resultado 	06-05-12	Éxito
PA – 4 - 3	En la vista de los resultados del test se selecciona una pregunta marcada con error y se debe mostrar una vista con dicha pregunta y la respuesta que ha seleccionado en rojo y la correcta en verde	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar un tema 3. Seleccionar un test 4. Seleccionar un resultado 5. Seleccionar una pregunta con error 	06-05-12	Éxito
PA – 4 - 4	En la vista de los resultados del test se selecciona una pregunta marcada con acierto y se debe mostrar una vista con dicha pregunta y la respuesta que ha seleccionado en verde	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar un tema 3. Seleccionar un test 4. Seleccionar un resultado 5. Seleccionar una pregunta con acierto 	06-05-12	Éxito

Tabla 41. Pruebas de sistema HU-4

Historia de usuario 5: Compartir en redes sociales

ID	PROPÓSITO	PASOS	FECHA	RESULTADO
PA – 5 - 1	En la vista de los test cuando se desee compartir información por una red social debe aparecer en la barra inferior un botón que te da la posibilidad de hacerlo por las redes sociales de twitter, facebook o enviar un correo	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar el botón de mostrar acciones	06-05-12	Éxito
PA – 5 - 2	Una vez pulsado el botón de mostrar las acciones en la vista de test, se debe probar a compartir en twitter y en dicha red social se debe ver el comentario	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar el botón de mostrar acciones 6. Pulsar el botón Publicar en twitter	06-05-12	Éxito
PA – 5 - 3	Una vez pulsado el botón de mostrar las acciones en la vista de test, se debe probar a compartir en facebook y en dicha red social se debe ver el comentario	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar el botón de mostrar acciones 6. Pulsar el botón Publicar en facebook	06-05-12	Éxito
PA – 5 - 4	Una vez pulsado el botón de mostrar las acciones en la vista de test, se debe probar a compartir por correo electrónico y que el destinatario del correo lo reciba	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al Test 5. Pulsar el botón de mostrar acciones 6. Pulsar el botón Enviar por correo	06-05-12	Éxito
PA – 5 - 5	Una vez pulsado el botón de mostrar las acciones en la vista de test, se debe probar a cancelar la	1.Abrir la aplicación 2.Seleccionar un tema 3.Seleccionar un test 4. Pulsar el botón Ir al	06-05-12	Éxito

acción	Test 5. Pulsar el botón de mostrar acciones 6. Pulsar el botón Cancelar
--------	---

Tabla 42. Pruebas de sistema HU-5

Capítulo 6: Planificación

6. Planificación

En este apartado se van a realizar dos planificaciones, la que se hizo al comienzo del proyecto y la que se ha hecho una vez terminada el trabajo. Para realizar la planificación inicial se ha realizado una planificación orientativa de las fases que se iban a seguir a lo largo del proyecto, pero como he dicho es orientativo, porque al ser una metodología ágil, durante el desarrollo del proyecto, la planificación no se sigue al pie de la letra, es decir, la planificación que se va realizando es una planificación adaptativa, en la que se va realizando iteración a iteración y al finalizar cada una de las iteraciones se piensa en lo que mejor puede venir para la siguiente iteración.

Con respecto a la planificación final, se van a mostrar los datos reales utilizados para la realización del proyecto.

6.1. Planificación Inicial

Como he dicho anteriormente se va a especificar una planificación orientativa que se realizó al comienzo del proyecto para las distintas fases del mismo, indicando también quienes han participado en cada fase:

- **Descripción del problema y reuniones:** la primera fase de cada iteración consistía en centrar una idea de lo que iba a consistir la iteración y que tareas íbamos a realizar en dicha iteración. En la fase inicial del proyecto la reunión para describir el problema fue más larga, ya que se tenía que ver por donde se iba a encaminar el proyecto.
 - Tiempo estimado: 8 horas
 - Participan: Desarrollador del TFG y tutor del TFG
- **Documentación:** en esta fase se encuentra toda la documentación del proyecto, es decir, se incluye desde la obtención de requisitos, en este caso historias de usuario con sus correspondientes tareas y casos de usos, hasta el diseño.
 - Tiempo estimado: 70 horas
 - Participan: Desarrollador del TFG
- **Estudio del lenguaje:** en esta etapa se pretendía que el desarrollador se familiarizara con el entorno y aprendiera el lenguaje Objective-C y todo lo relacionado con la programación para iOS.
 - Tiempo estimado: 50 horas
 - Participan: Desarrollador del TFG y experto en la materia
- **Implementación:** aquí se incluye toda la planificación de la aplicación.
 - Tiempo estimado: 160 horas
 - Participan: Desarrollador del TFG
- **Pruebas del sistema:** en esta fase se probaría el correcto funcionamiento del sistema.
 - Tiempo calculado: 10 horas

- Participan: Desarrollador del TFG
- **Presentación:** una vez terminada la implementación y entregada la documentación se realizará esta fase para la exposición del trabajo realizado.
 - Tiempo calculado: 2 horas
 - Participan: Desarrollador del TFG

Si se suman todas las horas del proyecto dan un total de 300 horas, que es lo que se estima que tiene que durar el trabajo de fin de grado, pero como bien he dicho, esto no se ha seguido al pie de la letra y en el siguiente apartado se van a describir las diferentes iteraciones que se han seguido a lo largo del proyecto. A continuación se muestra el diagrama de Gantt de la planificación orientativa que se había pensado:

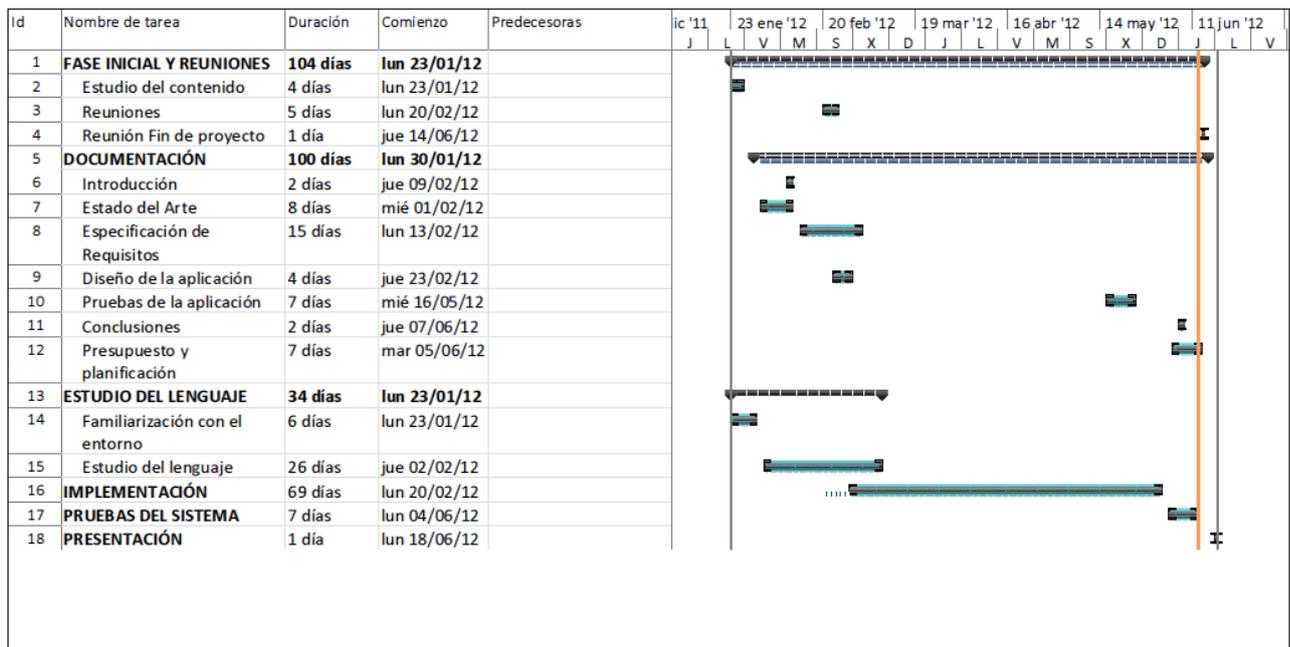


Ilustración 73. Planificación Inicial

6.2. Planificación Final

Como se ha mencionado anteriormente en este apartado vamos a describir los objetivos de las diferentes iteraciones que han tenido lugar a la hora de la implementación del proyecto, ya que la documentación se ha ido llevando a cabo mientras se realizaba dicha implementación, con un esfuerzo mayor al final de la implementación.

- ✓ **Iteración 1:** esta fase consistía en primer lugar, crear el proyecto y realizar la primera historia de usuario, en la que el usuario puede ver los diferentes temas y puede seleccionar uno de ellos para ver su breve descripción y su vídeo.
- ✓ **Iteración 2:** en esta iteración se desarrolló la parte con más trabajo de la aplicación, en la que se debía poder contestar a los test, es decir, la historia de usuario 3.
- ✓ **Iteración 3:** en esta fase ya se podían ver los resultados obtenidos al realizar el test y la corrección con la especificación de dichos resultados. Además se añadió la funcionalidad de compartir en diferentes redes sociales.
- ✓ **Iteración 4:** en esta última iteración además de realizar algunas mejoras sobre lo realizado anteriormente se añadió la posibilidad de ver los resultados de los test realizados anteriormente.

Como se ha realizado en la planificación inicial se va a especificar finalmente lo que se ha tardado para cada fase de la descripción del proyecto:

- **Descripción del problema y reuniones:** en esta fase se han utilizado más horas que en la planificación inicial, ya que se han realizado más reuniones de las esperadas
 - Tiempo estimado: 8 horas
 - Tiempo gastado: 10 horas
 - Participan: Desarrollador del TFG y tutor del TFG
- **Documentación:** el aumento de tiempo en este apartado ha sido mayor porque al haber aumentado la funcionalidad de la aplicación se ha tenido que gastar más tiempo en explicar dichas funcionalidades, en hacer diagramas, etc.
 - Tiempo estimado: 70 horas
 - Tiempo gastado: 95 horas
 - Participan: Desarrollador del TFG
- **Estudio del lenguaje:** en esta etapa se ha tardado aproximadamente lo que se calculó inicialmente.
 - Tiempo estimado: 50 horas
 - Tiempo gastado: 55 horas
 - Participan: Desarrollador del TFG y experto en la materia

- **Implementación:** como se aumentó la funcionalidad el tiempo ha sido bastante mayor, ya que además estas nuevas funcionalidades tenían más complejidad.
 - Tiempo estimado: 160 horas
 - Tiempo gastado: 210 horas
 - Participan: Desarrollador del TFG
- **Pruebas del sistema:** en esta fase se ha tardado menos tiempo de lo esperado.
 - Tiempo estimado: 10 horas
 - Tiempo gastado: 8 horas
 - Participan: Desarrollador del TFG
- **Presentación:** esta fase como no se ha realizado estimamos que se tardará lo estimado.
 - Tiempo estimado: 2 horas
 - Participan: Desarrollador del TFG

En este caso si se suman todas las horas del proyecto dan un total de 380 horas, mientras que se habían estimado 300 horas, lo estipulado para un trabajo de fin de grado, por lo que nos hemos desviado en 80 horas. Esto es normal, ya que al ser el primer año en el que se llevan a cabo trabajos de fin de grado, los que dirigen estos trabajos, es normal y comprensible que quieran implementar más funcionalidad que la que debería tener, porque están acostumbrados a los antiguos proyectos fin de carrera.

A continuación se va a mostrar en un Diagrama de Gantt lo especificado en los apartados anteriores:

Id	Nombre de tarea	Duración	01 febrero					01 marzo	
			09/01	23/01	06/02	20/02	05/03		
1	FASE INICIAL Y REUNIONES	104 días							
2	Estudio del contenido	4 días							
3	Reunión Iteración 1	1 día							
4	Reunión Iteración 2	1 día							
5	Reunión Iteración 3	1 día							
6	Reunión Iteración 4	1 día							
7	Reunión Fin de proyecto	1 día							
8	DOCUMENTACIÓN	95 días							
9	Introducción	2 días							
10	Estado del Arte	4 días							
11	Especificación de Requisitos	9 días							
12	Diseño de la aplicación	6 días							
13	Pruebas de la aplicación	10 días							
14	Conclusiones	2 días							
15	Presupuesto	3 días							
16	ESTUDIO DEL LENGUAJE	28 días							
17	Familiarización con el entorno	10 días							
18	Estudio del lenguaje	20 días							
19	IMPLEMENTACIÓN	95 días							
20	Iteración 1 - HU-1	18 días							
21	Iteración 2 - HU - 3	22 días							
22	Iteración 3 - HU - 4 y HU - 5	21 días							
23	Iteración 4 - HU - 2	15 días							
24	PRUEBAS DEL SISTEMA	3 días							
25	PRESENTACIÓN	1 día							

Ilustración 74. Planificación Final 1

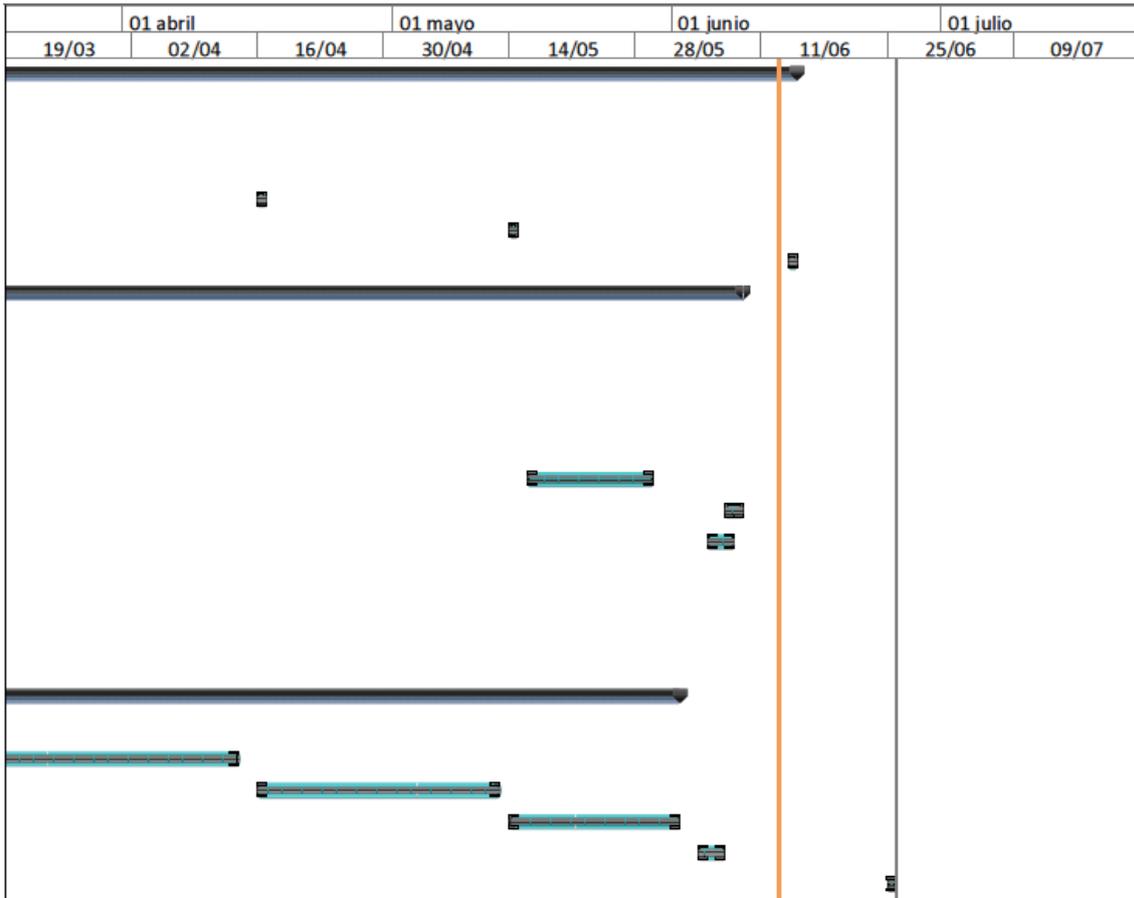


Ilustración 75. Planificación Final 2

Capítulo 7: Presupuesto

7. Presupuesto

En este apartado se va a presentar una estimación detallada de los costes de realización de este trabajo de fin de grado. Estos costes se presentan agrupados en diversas categorías.

Los grupos en los que se han dividido los costes son:

- Costes del personal involucrado en el proyecto.
- Costes de elementos software y hardware.

En este proyecto no es necesario incluir costes en dietas y viajes por lo que este apartado no será incluido en el proyecto. Por el mismo motivo no se añadirán costes de materiales fungibles, como puede ser material para impresora, etc.

Para el cálculo del presupuesto se deben tener en cuenta los siguientes aspectos:

- Los costes que se mostrarán estarán expresados en Euros a no ser que se especifique lo contrario.
- Se tomarán dos decimales para las cantidades económicas, redondeando en el caso de que sea necesario.

Además, al final del cálculo del presupuesto se pondrán las herramientas hardware utilizadas para la realización del proyecto, ya que en el coste de los equipos aparecen dichos elementos.

7.1. Presupuesto Inicial

En este primer apartado del presupuesto vamos a indicar el presupuesto que se planteó al inicio del proyecto, para ello vamos proponer el coste del personal que se tenía pensado realizar junto con el material que se tenía pensado utilizar.

7.1.1. Coste Personal

Para calcular el coste del personal se van a detallar las personas que han participado en el proyecto y el peso que han llevado en él. El coste por hora de cada persona del proyecto ha sido definido por un valor aproximado de lo que se está cobrando en la actualidad.

En la siguiente tabla se va a mostrar el nombre de las personas que han participado en el proyecto, su categoría, su coste a la hora, el número de horas totales

que han dedicado al proyecto y finalmente el coste total. Además se mostrará una fila con el coste total del personal, sumando el coste personal de cada uno.

En la tabla, aparecen Adrián Casado y Javier Sánchez, que han sido los especialistas a los que se les ha realizado alguna pregunta del proyecto en el caso de que hubiera habido dudas.

Además aparece el tutor y director del proyecto como Ingeniero Sénior, el cotutor, Alberto Heredia García, no ha sido incluido porque ha trabajado ayudando al tutor.

Nombre	Categoría	Euros/hora	Horas Totales	Coste Total
Roberto Esteban Santiago	Ingeniero Junior	20	300	6000 €
Javier García Guzmán	Ingeniero Sénior	36	15	540 €
Adrián Casado Rivas/Javier Sánchez Hernández	Ingeniero Junior	20	15	300 €
TOTAL:				6840 €

Tabla 43. Presupuesto inicial - Coste personal

7.1.2. Coste Elementos software y hardware

El coste del material, en este apartado se refiere a los diversos equipos informáticos que se preveían necesarios para el desarrollo del trabajo de fin de grado. En la siguiente tabla se mostrará el producto, su precio, el periodo de amortización en meses y el tiempo que ha sido utilizado, también en meses. Además en este apartado también se mostrará la licencia de desarrollador de iOS que se ha tenido que adquirir.

Al igual que en el apartado anterior al final de la tabla se mostrará el coste total.

Producto	Precio	Periodo de amortización	Uso del producto	Coste para el proyecto
iMac 21.5'	1.145 €	48	5	119,27 €
Licencia de Desarrollador iOS	80€	12	5	33,33 €
TOTAL				152,60 €

Tabla 44. Presupuesto inicial - Coste elementos sw y hw

7.1.3. Coste Total del Proyecto

Por último en este apartado se va a mostrar el coste total que se había planificado para el proyecto sumando el coste del personal y el coste de los equipos:

Descripción	Coste Total
Personal	6840 €
Equipos	152,60 €
COSTE TOTAL PROYECTO	6992,60 €

Tabla 45. Presupuesto inicial - Coste Total

Por lo tanto el proyecto, según el presupuesto inicial, el proyecto tendría un coste de **6992,60 €**.

7.2. Presupuesto Final

Por último, dentro del apartado de presupuesto, vamos a realizar el cálculo real que ha tenido finalmente el proyecto, especificando las horas reales que se han empleado para realizarlo y los materiales que finalmente se han utilizado. Este presupuesto final, sería lo que hubiera costado realmente el proyecto si se hubiera realizado en una empresa.

7.2.1. Coste Personal

En este caso al igual que en el coste del personal del presupuesto inicial se van a detallar las personas que han participado, la categoría, lo que cuesta a la hora cada persona, las horas que han trabajado en el proyecto y el coste total.

También, como se ha dicho en el presupuesto inicial, en la tabla, además del tutor, aparecen Adrián Casado y Javier Sánchez, que han sido los especialistas a los que se les ha realizado alguna pregunta del proyecto en el caso de que hubiera habido dudas.

Nombre	Categoría	Euros/hora	Horas Totales	Coste Total
Roberto Esteban Santiago	Ingeniero Junior	20	380	7600 €
Javier García Guzmán	Ingeniero Sénior	36	19	684 €
Adrián Casado Rivas/Javier Sánchez Hernández	Ingeniero Junior	20	19	380 €
TOTAL:				8664 €

Tabla 46. Presupuesto final - Coste personal

7.2.2. Coste Elementos software y hardware

Con el coste de los elementos software y hardware, en este apartado se han incrementado los costes, ya que no se tenía pensado la utilización del ordenador portátil ASUS A53S, ni el disco duro Western Digital My Passport, ni el dispositivo móvil iPhone 4S, ya que para hacer las pruebas se pensaba en utilizar el simulador del Xcode. Al final de este apartado aparecerá el subapartado de [Herramientas Hardware](#), en las cuales se explicará en detalle cada dispositivo y para que han sido usados.

Al igual que en el presupuesto inicial, se va a mostrar el producto, su precio el periodo de amortización, el uso del producto y el coste para el proyecto. Además, al final de la tabla se mostrará el coste total de todos los materiales. Además el uso del producto se ha modificado con respecto al presupuesto inicial, ya que se tenía pensado que se usaran 5 meses y se han usado 6.

Producto	Precio	Periodo de amortización	Uso del producto	Coste para el proyecto
iMac 21.5'	1.145 €	48	6	143,12 €
Licencia de Desarrollador iOS	80 €	12	6	40 €
ASUS A53S	690 €	48	6	86,25 €
Western Digital My Passport	90 €	48	6	11,25 €
iPhone 4S	699 €	18	6	233 €
TOTAL				513,62 €

Tabla 47. Presupuesto inicial - Coste elementos sw y hw

7.2.3. Coste Total del Proyecto

Por último en este apartado se va a mostrar el coste total que supondría el proyecto sumando el coste del personal y el coste de los equipos:

Descripción	Coste Total
Personal	8664 €
Equipos	513,62 €
COSTE TOTAL PROYECTO	9177,62 €

Tabla 48. Presupuesto final - Coste Total

Por lo tanto el proyecto si se hubiera realizado en una empresa tendría un coste de **9177,62 €**.

7.3. Herramientas Hardware

En este apartado se van a especificar las herramientas hardware que se han utilizado para el desarrollo del trabajo de fin de grado.

iMac

El ordenador que he utilizado para desarrollar la implementación del proyecto es un iMac de 21'5 pulgadas. El iMac tiene las siguientes especificaciones:

- Procesador: 2.5 GHz Intel Core i5
- Memoria RAM: 8 GB DDR3
- Tarjeta Gráfica: AMD Radeon HD 6750M 512 MB
- Software: Mac OS X Lion 10.7.3
- Disco duro: 500 GB

iPhone 4S

Para poder probar la aplicación desarrollada, se ha utilizado un iPhone 4S. Este Smartphone es una evolución del iPhone 4. Estos teléfonos inteligentes con pantalla táctil, fueron de los primeros Smartphone con calidad aparecidos en España. Hoy en día, estos móviles son los más vendidos en España y las aplicaciones que existen para este tipo de teléfonos superan el medio millar.

Este tipo de teléfonos pueden ser considerados como pequeños ordenadores, ya que las características que van teniendo en cuanto al software y al hardware son muy similares a los que tienen los ordenadores actualmente. Sin irnos más lejos el modelo usado en este proyecto tiene 64 GB de memoria de almacenamiento y 512 MB DRAM. Además tienen una CPU ARM Cortex-A9 de doble núcleo. En cuanto a la resolución de la pantalla, la del iPhone 4S es de 960x640 px a 326 píxeles por pulgada.

Para probar esta aplicación se tiene instalada la versión de iOS 5.1. En primer lugar cuando se tenía la versión de Xcode 4.2.1. se tenía instalada la versión 5.0.

ASUS A53S

Este ordenador portátil ha sido usado para diferentes tareas como ha sido realizar la documentación del proyecto, con la herramienta Microsoft Word, avanzar en la codificación con una máquina virtual instalada en dicho PC. Además en dicha máquina virtual se instaló también el programa SQLite Manager para introducir los diferentes datos de la aplicación a la base de datos.

Este ordenador portátil tiene las siguientes características:

- Procesador Intel Core i7-2670QM a 2,20GHz.
- Memoria RAM 4 GB.
- Disco duro 500 GB.
- Tarjeta gráfica nVIDIA GEFORCE 1 GB.

Western Digital My Passport

Se ha utilizado un disco duro para almacenar las imágenes, vídeos, documentación y código fuente del proyecto. Este disco tiene 500GB de capacidad y su tamaño es de 2.5 pulgadas.

Capítulo 8: Conclusiones y Líneas Futuras

8. Conclusiones y Líneas Futuras

En este capítulo de la memoria se van a exponer las conclusiones personales sobre el desarrollo del Trabajo de Fin de Grado, además de unas líneas futuras que se podrían realizar si se continuara con el proyecto.

8.1. Conclusiones

En esta sección se van a exponer las conclusiones que se han extraído tras la finalización del proyecto desarrollado. En este apartado además de hacer las conclusiones personales, se van a mostrar las dificultades que se han tenido durante todo el desarrollo del proyecto.

Con respecto a la aplicación que se ha desarrollado, estoy bastante satisfecho porque considero que se ha realizado una aplicación bastante interesante, ya que la considero de mucha utilidad. Además, los objetivos que se habían establecido al principio, no es que se hayan cumplido, sino que se han superado, ya que se han realizado más funcionalidades que las que se habían establecido como primeros objetivos. También me complace, que al ver la aplicación realizada, no tiene mucho que envidiar a algunas aplicaciones que se encuentran en el AppStore, por lo que considero que ha sido un trabajo bien hecho.

En cuanto a la metodología utilizada, Scrum de tipo ágil, pienso que es una metodología bastante interesante, ya que no te tienes que ceñir a una planificación cerrada y puedes trabajar más “libremente” hacia un objetivo, por lo que considero que ha funcionado bastante bien, ya que aunque en alguna iteración no se ha podido terminar la funcionalidad asignada a dicha iteración, en la siguiente iteración se ha podido solucionar, además de realizar lo planificado para esa iteración. Por todo esto, considero que no ha habido ningún problema al utilizar este tipo de metodología, y los que ha habido de no llegar a tiempo, no se consideran problemas, ya que se han solucionado rápidamente.

En cuanto a la dificultad del proyecto, finalmente ha sido menor de la esperada a la elección del proyecto, pero esto ha sido debido a la fase de formación que tuvimos al inicio de este, ya que pudimos aprender bastante y practicar en muchos aspectos que se utilizaron finalmente en la aplicación.

Los problemas que se han tenido a la hora de implementar la aplicación han sido relacionados con la base de datos, ya que no existe demasiada información acerca de SQLite. Un problema fue, que no se imprimían correctamente los acentos, y creíamos que era problema de la base de datos, pero el problema era a la hora de imprimir dichos datos. El otro problema que ha habido ha sido a la hora de utilizar el patrón de diseño Singleton, ya que no los había utilizado nunca durante la carrera y tuve que buscar mucha información.

Como conclusión personal, este proyecto me ha aportado mucho, ya que aprendido un lenguaje que considero que es el futuro y que es bastante interesante por las posibilidades que suministra, además he aprendido a trabajar con una metodología ágil. También al haber estado durante todo el cuatrimestre asistiendo al laboratorio a realizar el trabajo de fin de grado, me he adaptado a como se trabajaría en una empresa.

Finalmente, creo que a nivel profesional este trabajo me ha abierto muchas puertas, ya que he aprendido un lenguaje y una metodología, que actualmente están de moda y que considero que serán también el futuro.

8.2. Líneas Futuras

En este apartado se va a explicar las líneas futuras que podría tener la aplicación realizada. Aunque la aplicación tiene toda la funcionalidad deseada implementada, se podrían añadir mejoras u otras líneas a las que aplicar dicha aplicación.

8.2.1. Vídeos en servidores

La primera y una de las más sencillas de las líneas futuras sería meter los vídeos, que se muestran en la vista del detalle del tema, en un servidor, pero que por falta de infraestructuras en este momento no ha sido posible realizarlo. De esta manera la aplicación ocuparía bastante menos de lo que ocupa, y se podrían tener vídeos más largos.

Con la nueva funcionalidad descrita anteriormente se podría añadir en la vista de detalle, más vídeos y más definiciones, de manera que hubiera más explicaciones de un mismo tema, y el alumno pudiera entender mejor el tema.

La desventaja sería, que como se ha explicado en las alternativas de diseño, para ver los vídeos de la aplicación se necesitaría conexión a internet.

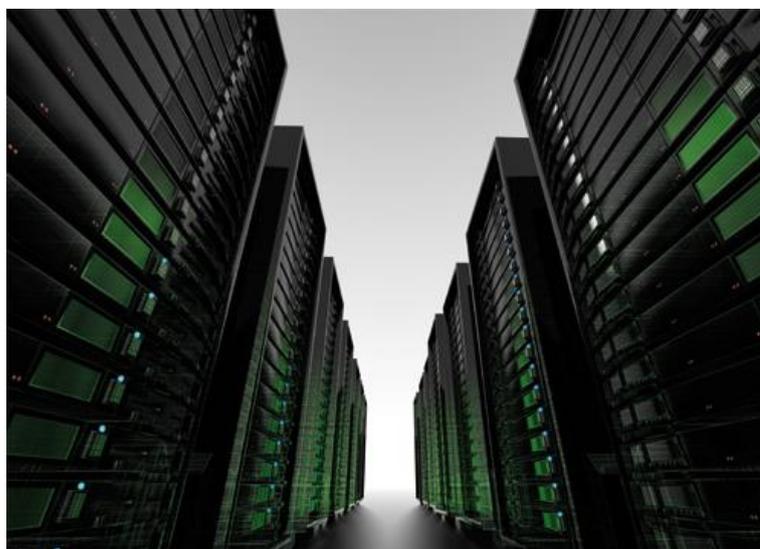


Ilustración 76. Vídeos en servidores

8.2.2. Parsear datos

Otra posible línea futura sería poder introducir los datos de las preguntas de una manera automática parseando una página web y no introducir los datos uno a uno como se ha hecho para esta aplicación.

Con esta nueva funcionalidad se podría aumentar bastante el número de test y el número de preguntas y no tener únicamente un test por cada tema.

8.2.3. Otro objetivo de la aplicación

Una nueva funcionalidad, que sería cambiar el objetivo de esta aplicación, pero si está relacionado con ella, sería poder ampliar esta aplicación a todo tipo de exámenes de test. Por ejemplo un buen fin de esta aplicación sería para la propia universidad, en una asignatura, en lugar de hacer test en papel, o por el ordenador, lo podrían hacer con su propio dispositivo y la corrección sería automática. Además, se podría ofrecer un test a cada alumno, para evitar la copia. Además, se podrían gestionar los usuarios que acceden a la aplicación, para controlar que test ha realizado cada uno y que nota han obtenido. En otra era tecnológica, esta funcionalidad no sería viable, pero hoy en día, la mayoría de los jóvenes tienen Smartphone, por lo que sería una funcionalidad viable.

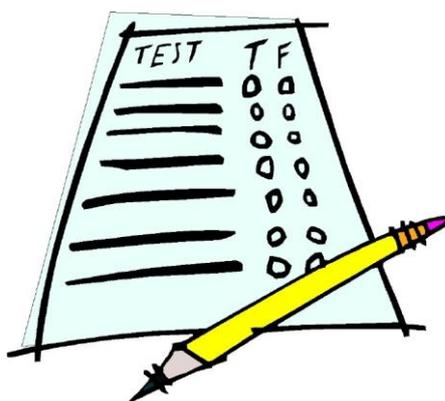


Ilustración 77. Otro objetivo de la aplicación

La aplicación ya se ha realizado pensando en dicho fin, es decir, está programada para que cambiando únicamente la información, pudiera tener dicho objetivo, además para la mejor anterior de meter más datos, también está implementada la función de meter más test por cada tema.

8.2.4. Aplicación para el iPad

Otra posible mejora para el proyecto, sería realizar la aplicación para el dispositivo iPad. Para realizar este cambio, en cuanto a la funcionalidad no habría que modificar nada, únicamente se tendría que volver a rediseñar la aplicación y acoplarla al tamaño del iPad, ya que el tamaño del iPhone es 960x640 píxeles, mientras que el del iPad es 1024x768 píxeles. Además, habría que modificar el tamaño de las fotos para que no se vieran difuminadas, ya que dichas fotos han sido reducidas para que ocuparan menor tamaño.



Ilustración 78. Aplicación para el iPad

8.2.5. Otros sistemas operativos

Otro giro que podría dar la aplicación, sería realizar la aplicación para otros sistemas operativos de dispositivos móviles, como pueden ser Windows Phone o Android, ya que cada vez los móviles con estos sistemas operativos tienen mayor salida en el mercado y podría suponer un gran auge para esta aplicación.



Ilustración 79. Otros sistemas operativos

Sin embargo, la implementación de esta nueva funcionalidad si sería más complicada que las anteriores ya que habría que implementar de nuevo toda la funcionalidad y no se parece la forma de programar en iOS que en los otros dos sistemas operativos.

Capítulo 9: Bibliografía

9. Bibliografía

A continuación se mostrarán los recursos electrónicos y los libros usados para la realización del trabajo de fin de grado.

9.1. Libros de Consulta

[1] Erica Sadun, The iOS 5 Developer's Cookbook: Core Concepts and Essential Recipes for iOS Programmers, Addison-Wesley, Noviembre 2011.

[2] John Ray, iPhone Application Development in 24 Hours, Sams, Octubre 2010.

9.2. Recursos electrónicos

[3] Antonio Rodriguez, Cocoa Touch

<<http://www.iphonesoftware.es/2008/04/15/el-sistema-operativo-de-iphone/>>

Ultimo acceso: Junio 2012

[4] Proyectos ágiles, Historia de Scrum

<<http://www.proyectosagiles.org/historia-de-scrum>>

Ultimo acceso: Junio 2012

[5] Luis Miguel Gracia Luis, Ciclo de vida Scrum

<<http://unpocodejava.wordpress.com/2010/02/24/scrum-un-resumen-con-muchos-monigotes/>>

Ultimo acceso: Junio 2012

[6] Alejandro Barros, Prácticas para el desarrollo de Scrum

<<http://www.alejandrobarrros.com/content/view/560804/Nuevas-practicas-de-desarrollo-de-software.html>>

Ultimo acceso: Junio 2012

[7] Ingenieriasoftwaredos, Diagrama de componentes

<<http://ingenieriasoftwaredos.wikispaces.com/Diagrama+de+componentes+y+objetos>>

Ultimo acceso: Junio 2012

[8] Sparx Systems, Diagrama de secuencia

<http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html>

Ultimo acceso: Junio 2012

[9] Ministerio de educación y ciencia, SQLite

<<http://usuarios.pntic.mec.es/sqlite.php>>

Ultimo acceso: Junio 2012