MejoraTuIngles

Memoria de Proyecto

Daniel Cortés Fernández

Trabajo de Fin de Carrera - Ingeniería Técnica en Telecomunicación (2012-2013)

Consultor de Proyecto: Manel Llopart Vidal

Abstract

- Versión en español:

En el presente trabajo de fin de carrera se analizará y se implementará un proyecto enmarcado en el ámbito del aprendizaje electrónico orientado a dispositivos móviles (m-learning).

En una primera fase del proyecto se introduce el problema: un curso de gramática inglesa virtualizado. Esta idea surge por la observación del alto porcentaje de personas que intentan aprender inglés en España pero que no pueden adaptarse a un horario de clases por diferentes motivos como incompatibilidad de horarios o imposibilidad de desplazamiento.

Se realiza un "Estudio del Arte", presentando el estado actual del e-learning a nivel mundial, definiendo los diferentes tipos de e-learning y, finalmente, explicando el concepto de m-learning.

En la fase de diseño e implementación se realiza una rápida introducción a las metodologías ágiles y la toma de requisitos mediante historias de usuario. Se aíslan los casos de uso detectados realizando diagramas de solución técnica en UML que sirven de ayuda a la implementación.
El resultado final es la creación de un prototipo que puede servir como experiencia piloto para un futuro desarrollo de un proyecto más complejo.

- English version:

In the current finishing carrier document I have analyzed and implemented a project related with the electronic learning over mobile devices (m-learning).

In the first phase it is described the problem: one virtualized English grammar course. This idea arise thanks to observe the amount of people who try to learn English in Spain but they can't adapt to a timetable or go to class.

I have performed a study (Art state) where the e-learning state in the world is described and the different kinds of e-learning are introduced. Finally you can find the m-learning concept with more details.

In the design and implementation phase I have performed a quick introduction to agile methodologies and the requirements capture using user stories. Each user case is identified by technical diagrams using UML to help to generate the code.

Like final result, the prototype is finished and it can serve like pilot experience in order to perform, in future, a more complex product.

Índice de Capítulos

1. Introducción 9

2. Objetivos y alcance del proyecto 11

3. Definición del problema 12

3.1 Introducción 12

3.2 E-learning 12

3.2.1 Tipos de Elearning 14

3.2.2 E-learning sobre plataformas móviles: m-learning 15

3.2.2.1 Ventajas de las plataformas móviles 16

3.2.2.2 Inconvenientes de las plataformas móviles 17

3.3 Motivación del proyecto 18

3.4 Casos de estudio 18

4. Planificación 20

4.1 Introducción 20

4.2 Planificación 20

4.2.1 Entregables 20

4.3 Material 23

4.3.1 Equipos 23

4.4 Plan de contingencia 24

4.4.1 Análisis de Riesgos y Acciones Mitigadoras 24

4.4.1.1 Problemas de integración entre Software 24

4.4.1.2 Entregas coincidentes con otras asignaturas 25

4.4.1.3 Avería de la estación de trabajo 25

4.5 Diagramas de Gantt completos 26

5. Análisis de la aplicación 27

5.1 Recopilación de requisitos mediante Historias de Usuario 27

5.1.1 Historia de "Estudiar unidad didáctica" 27

5.1.2 Historia de "Realizar examen" 28

5.1.3 Historia de “Seguimiento de progresos” 28

5.2 Identificación de Casos de Uso 28

6. Diseño del Sistema 30

6.1 Diagrama de clases del Sistema 30

6.1.1 Diagrama de Beans 31

6.1.2 Diagrama de clases 32
6.2 Diagramas de solución técnica 33
6.2.1 Estudio de lección 33
6.2.2 Realización de test 34
6.2.3 Registrar progreso 36
6.2.4 Consultar progreso 37
6.3 Diseño del modelo de datos 38
6.3.1 Lección 39
6.3.2 Ejercicio 39
6.3.3 Pregunta 39

7. Interfaz de Usuario 40
7.1 Diseño de la interfaz gráfica de usuario 40
7.1.1 Estética 41
7.1.2 Experiencia de usuario 41
7.1.3 Independencia del dispositivo 41
7.2 Presentación 42
7.3 LeccionView 44
7.4 Listado de lecciones 46
7.5 Test 47
7.6 Control de progreso 52
7.7 Ayuda 54

8. Implementación 56
8.1 Soluciones técnicas 56
8.1.1 Operaciones con Base de Datos 56
8.1.2 Carga de datos 58
8.2 Especificación de características y permisos 60
8.1.1 Ubicación del Software 61
8.1.2 Permisos 61
8.1.3 Versión de Android necesaria 61

9. Plan de pruebas funcionales 62
9.1 Emulador 62
9.2 Estudio de lección 63
9.3 Realización de test 64
9.3.1 Desarrollo del test 64

9.3.2 Finalización de test 66
9.4 Registrar progreso 67
9.5 Consultar progreso 69
9.6 Ayuda 70

10. Conclusiones y líneas futuras 71
10.1 Evolución del proyecto 71
10.2 Dificultades encontradas 71
10.3 Líneas futuras 71
10.3.1 Descarga dinámica de nuevo contenido 72
10.3.2 Contenido multimedia 72

11. Anexo I - Terminología aplicable 73

12. Anexo II – Tecnologías y Metodologías de Desarrollo 75
12.1 Tecnologías de Desarrollo 75
12.1.1 ANDROID 75
12.1.2 Eclipse 76
12.1.3 SQLite 76
12.2 Metodología de Desarrollo 77
12.2.1 Formación de un grupo de trabajo SCRUM 77
12.3 Ciclo de vida en aplicaciones Android 79
12.4 Introducción a las interfaces visuales en Android 80

13. Anexo III – Configuración del Entorno de Desarrollo 82
13.1 Eclipse 82
13.1.1 Los proyectos en Eclipse 82
13.1.2 Perspectivas en Eclipse 83
13.1.3 Instalación y configuración 83
13.2 Android SDK 83
13.2.1 Instalación y configuración 83
13.3 Integración del Android SDK en Eclipse 84
13.3.1 Android VIRTUAL DEVICE 86
13.3.2 Perspectiva DDMS 88
13.4 Ejecutando la aplicación en AVD o terminal real 90

14. Anexo IV – Bibliografía 93
14.1 Bibliografía 93

14.2 Fuentes de imágenes utilizadas.....93

Índice de Ilustraciones

Ilustración 1 - Desglose de tareas del Proyecto de TFC (PEC1) 22
Ilustración 2 - Desglose de tareas de Estado del Arte 22
Ilustración 3 - Desglose de tareas de Diseño de Implementación 22
Ilustración 4 - Desglose de tareas de la Memoria 23
Ilustración 5 - Desglose de tareas de la Presentación 23
Ilustración 6 - Desglose de tareas de la Presentación 23
Ilustración 7 - Diagrama Gantt PEC1 26
Ilustración 8 - Diagrama de Gantt PEC2 26
Ilustración 9 - Diagrama de Gantt PEC3 26
Ilustración 10 - Diagrama de Gantt PEC4.....26
Ilustración 11 - Diagrama de Gantt PEC5a 26
Ilustración 12 - Diagrama de Gantt PEC5b 26
Ilustración 9 - Diagrama de casos de uso.....29
Ilustración 10 - Beans.....31
Ilustración 11 - Diagrama de clases 32
Ilustración 12 – Estudio de lección 33
Ilustración 13 – Funcionalidad del test 35
Ilustración 14 – Registrar progreso 36
Ilustración 15 – Consultar progreso.....37
Ilustración 16 - Modelo de datos.....38
Ilustración 17 – Pantalla de presentación 42
Ilustración 18 - Código de Layout principal 43
Ilustración 19 - Código de layout de menú de la página principal.....43
Ilustración 20 – Lección cargada 44
Ilustración 21 - Utilización del WebView 45
Ilustración 22 - Menú de LeccionView 45
Ilustración 23 - Listado de lecciones 46
Ilustración 24 - Listado de lecciones 46
Ilustración 25 - Configuración de elemento de una lista.....47
Ilustración 26 - Pantalla de test 48

Ilustración 27 - Áreas definidas en la vista 48
Ilustración 28 - Configuración del área 1 49
Ilustración 29 - Resto de áreas 50
Ilustración 30 - Navegación de la prueba de conocimiento 51
Ilustración 31 - Menú contextual del test 51
Ilustración 32 - Control de progreso 52
Ilustración 33 - Layout de control de progreso 53
Ilustración 34 - Configuración de la cabecera 54
Ilustración 35 - Configuración del cuerpo 54
Ilustración 36 - Vista de ayuda 55
Ilustración 37 - Layout de ayuda 55
Ilustración 38 - Detalle del método onCreate 57
Ilustración 39 - Método onUpgrade de MejorarTuInglesDatabaseHelper 58
Ilustración 40 - XML de lecciones 59
Ilustración 41 - Detalle del fichero AndroidManifest 60
Ilustración 42 - Administrador de aplicaciones de Android 61
Ilustración 43 - Proceso SCRUM 78
Ilustración 44 - Ciclo de vida de una Actividad Android 80
Ilustración 45 - Esquema jerárquico de View 81
Ilustración 46 - Instalar nuevo software 84
Ilustración 47 - Instalación de ADT en Eclipse 85
Ilustración 48 - Configuración del SDK en Eclipse 86
Ilustración 49 - Acceso AVD 86
Ilustración 50 - Pantalla principal del AVD Manager 87
Ilustración 51 - Configuración AVD 88
Ilustración 52 - Perspectiva DDMS 89
Ilustración 53 - Ejecución de la aplicación 90
Ilustración 54 - Selección de dispositivo 91
Ilustración 55 - Inicialización de AVD 92

1. Introducción

El presente documento se corresponde Memoria Final del Trabajo de Fin de Carrera de los Estudios de Ingeniería Técnica de Telecomunicación, especialidad Telemática (en adelante TFC) realizados en la Universidad Oberta de Catalunya.

El objetivo de este documento es la presentación formal de la totalidad del trabajo realizado en relación con este TFC.

En primer lugar se apuntarán los objetivos que se pretenden alcanzar con la realización de este proyecto, así como el alcance que tendrá el mismo.

Una vez fijados los objetivos, realizaremos una introducción sobre el concepto de e-learning, así como sus diferentes tendencias y definiremos un caso de estudio. Mediante la resolución de este caso alcanzaremos los objetivos planteados inicialmente. El caso de estudio elegido podría darse en el mundo laboral: un curso de inglés para dispositivos móviles.

Expondremos el caso de estudio, explicando la funcionalidad que el hipotético cliente querría obtener como resultado final del proyecto.

Una vez introducido el caso de estudio, se establecerá una planificación para abordar el proyecto. En la planificación se realizará un desglose que mostrará las tareas que debemos realizar y un calendario de ejecución estimado para las mismas.

En el apartado de análisis se explicará el enfoque metodológico utilizado para la extracción de requisitos y ejecución del proyecto. Se realizará también un análisis de negocio (a alto nivel) que nos permitirá aislar y estudiar por separado las diferentes funcionalidades solicitadas por el cliente. También estableceremos en esta fase la interfaz gráfica de usuario que, una vez aprobada por el cliente, permitirá crear un prototipo en una fase temprana del proyecto.

El apartado de implementación explica, en primer lugar, cómo se ha configurado el entorno de desarrollo necesario para crear el software. Una vez operativo el entorno,

se explicarán las soluciones técnicas adoptadas para implementar las funcionalidades aisladas y estudiadas en el apartado de análisis.

El documento finalizará con la presentación de un plan de pruebas funcional que ayudará a garantizar el correcto funcionamiento de la aplicación.

Por último, se han añadido anexos destinados a explicaciones cuyo conocimiento se antoja imprescindible para la comprensión del proceso de desarrollo de este proyecto. En el anexo bibliográfico se hace referencia a toda la documentación utilizada para el desarrollo de este TFC.

2. Objetivos y alcance del proyecto

En este capítulo se realiza una síntesis de los objetivos que se pretenden alcanzar con la realización de este TFC, así como de los entregables que se generarán a su finalización.

Los objetivos del proyecto son:

- Conocer y manejar el concepto de e-learning
- Ser capaces de aplicar la tecnología para el desarrollo de un prototipo de aplicación m-learning.
- Gestionar correctamente un desarrollo software relacionado con el ámbito de los sistemas e-learning
- Ser capaces de presentar los resultados obtenidos de forma clara y efectiva.

Con el fin de alcanzar estos objetivos, se desarrollará una aplicación de m-learning consistente en un curso de inglés que funcionará sobre el sistema operativo Android. Este tipo de aplicación engloba los conocimientos y habilidades que pretendemos adquirir con la realización de este TFC.

A la finalización de este proyecto y, como resultado, se obtendrán los siguientes entregables:

- Una memoria final
- Un prototipo instalable, funcionando sobre la versión 2.3 de Android

3. Definición del problema

3.1 Introducción

En la actualidad, los avances tecnológicos, en materia de Telecomunicaciones e Informática han posibilitado la producción de teléfonos inteligentes y tabletas digitales de altas prestaciones a precios asequibles. La combinación de tecnología y precio competitivo ha tenido como resultado que un alto porcentaje de población utilice actualmente teléfonos móviles inteligentes y demande cada vez más aplicaciones pensadas específicamente para estos terminales.

Por otro lado, nos encontramos inmersos en un proceso de evolución en el ámbito educativo. La necesidad de formación continuada durante toda la vida profesional del personal cualificado perteneciente a distintos sectores de la actividad económica hace que sea imprescindible la búsqueda de nuevos sistemas capaces de conciliar los horarios de trabajo, vida familiar y aprendizaje. Se busca el aprovechamiento del tiempo disponible, con la máxima flexibilidad posible y el mínimo consumo de recursos. Nace un nuevo concepto educativo: el e-learning.

3.2 E-learning

Tal y como define Wikipedia, "*Se denomina **aprendizaje electrónico** (conocido también por el anglicismo e-learning) a la educación a distancia completamente virtualizada a través de los nuevos canales electrónicos (las nuevas redes de comunicación, en especial Internet, utilizando para ello herramientas o aplicaciones de hipertexto (correo electrónico, páginas web, foros de discusión, mensajería instantánea, plataformas de formación que aúnan varios de los anteriores ejemplos de aplicaciones, etc.) como soporte de los procesos de enseñanza-aprendizaje.*".

Este nuevo modelo de educación a distancia virtualizada está siendo adoptado cada vez con más relevancia por empresas e instituciones educativas. Las razones principales de esta adopción pueden ser resumidas en los siguientes puntos:

- Eliminación de restricciones geográficas
- Eliminación de restricciones temporales

- Reducción de costes de material
- Incremento en el rendimiento del estudiante
- Acceso en tiempo real
- Mejoras en la retención

La eliminación de las restricciones temporales y geográficas, así como la posibilidad de reaprovechamiento del material educativo, hacen que este tipo de modelo educativo pueda ser empleado con éxito en situaciones dónde un modelo tradicional fracasaría. A continuación se exponen dos ejemplos de escenario de uso:

Ejemplo 1: Actualización para profesionales cualificados

Son muchas las profesiones que están sujetas a continuos cambios en las técnicas y procedimientos empleados en ellas, es por este motivo que los profesionales que las ejercen están sometidos a un proceso de actualización continua que suele tener lugar en tiempos muertos de su actividad laboral o en sus momentos de ocio. Al eliminar las restricciones geográficas y de tiempo, estos profesionales pueden decidir en qué momento y situación pueden revisar una unidad didáctica, visualizar una clase o realizar test de entrenamiento, sin tener que desplazarse o respetar un horario.

Ejemplo 2: Educación en el tercer mundo

El hecho de no necesitar edificaciones y la necesidad de poco personal docente, hacen que el coste de poner en funcionamiento un plan de estudios e-learning sea significativamente menor que un colegio tradicional.

3.2.1 Tipos de Elearning

En la práctica, un proyecto de e-learning puede llevarse a cabo de diferentes formas, adaptando las condiciones de la docencia al caso concreto.

B-learning

También conocido como aprendizaje mixto, supone un híbrido entre la enseñanza presencial y el e-learning, aprovechando los puntos fuertes de cada modelo.

En este tipo de enseñanza el profesor asume el rol tradicional, pero se apoya en material multimedia ya sea de elaboración propia o de terceros. Las tutorías pueden ser llevadas a cabo presencial o virtualmente.

Web Based Training

La capacitación basada en web consiste en la provisión de contenido que pueda ser visualizado con la única ayuda de un navegador web. Esto incluye contenidos en formato HTML, correos electrónicos, foros de debate, etc.

Computer Based Training

La capacitación basada en computador está constituida por programas y tutoriales diseñados para una plataforma concreta. No están sujetos a las restricciones impuestas por los navegadores web, ni a las impuestas por la velocidad de conexión a Internet, es por este motivo que suelen ser más ricos desde el punto de vista gráfico.

Paralelamente a esta clasificación y, atendiendo al dominio temporal, podemos clasificar los tipos de E-learning como síncronos o asíncronos.

E-learning síncrono

Este tipo de e-learning implica la necesidad de conexión temporal sincronizada entre los asistentes al curso y el docente. En cierto modo, simula a un aula tradicional, exceptuando el hecho de que los participantes no necesitan encontrarse en el mismo lugar. Existen múltiples herramientas que el docente puede utilizar para apoyar las explicaciones tales como pizarras digitales, gráficos, etc.

Actualmente existe en el mercado una gran variedad de software que permite realizar esta clase de encuentros virtuales.

Cabe destacar que las sesiones que en algún momento fueron síncronas e interactivas pueden ser grabadas para ser visualizadas a posteriori.

E-learning asíncrono

En el E-Learning asíncrono, el contenido (o courseware) y las actividades están disponibles en un servidor y son accedidas bajo demanda de los propios usuarios. De esta forma, el alumno puede llevar la formación a su propio ritmo, ya que la documentación y los elementos de aprendizaje están disponibles 24 horas al día, 7 días a la semana.

Para la implementación de este tipo de sistemas suele contarse con plataformas LMS (Learning Management System) o CMS (Course Management System). Estos sistemas permiten al alumno el acceso a un menú personalizado con sus cursos y materiales docentes y son capaces de registrar el progreso del alumno para cada materia.

3.2.2 E-learning sobre plataformas móviles: m-learning

Una nueva tendencia dentro del e-learning es el llamado m-learning. El m-learning se basa en la utilización de dispositivos móviles tales como teléfonos, tabletas, PDAs o Pocket PC que son utilizados como vía de acceso a los contenidos docentes.

Este concepto supone un paso más hacia el aprovechamiento completo del tiempo disponible del alumno. La generación de contenidos pensados específicamente para terminales móviles tiene como principales ventajas:

- Sin necesidad de planificación previa
- Posibilidad de aprendizaje “outdoor” utilizando las capacidades de los terminales.

- Posibilidad de acceso inmediato casi en cualquier situación.

No obstante, esta nueva tendencia muestra también limitaciones. En los siguientes apartados se comentan las posibilidades que el m-learning ofrece así como sus posibles inconvenientes.

3.2.2.1 Ventajas de las plataformas móviles

Sin necesidad de planificación previa

Los contenidos diseñados para smartphones, una vez descargados, pueden ser utilizados directamente. Este hecho permite aprovechar cada momento que el usuario considere oportuno, sin necesidad de llevar consigo ningún equipo voluminoso y sin ningún tipo de planificación. Es especialmente útil durante los tiempos de transporte a los diferentes lugares de la actividad diaria.

Aprendizaje "outdoor"

Muchos de los dispositivos móviles comercializados actualmente cuentan con algunas capacidades no disponibles de forma estándar en los ordenadores personales, por ejemplo, localización GPS o cámara fotográfica y de video. Estos elementos añaden nuevas posibilidades en cuanto a:

- Aplicaciones de Realidad Aumentada (RA)

- Lectura de códigos QR

Existen diferentes tipos de aplicaciones de RA, como Google Goggles o Layar, que utilizan diferentes capacidades del dispositivo para mostrar información superpuesta a lo que el usuario está viendo en el mundo real. Google Googles hace uso de la cámara fotográfica para realizar un reconocimiento de imagen y mostrar información adicional de lo que el usuario ve. Este sistema es muy útil para museos o monumentos ya que se pueden obtener datos relevantes de la obras con tan solo enfocar el terminal hacia ellas.

Layar se basa en la posición geográfica del usuario, así como la brújula y los sensores de posicionamiento del dispositivo. Con estos datos la aplicación es capaz de consultar capas de información y superponerlas a las imágenes del mundo real. Un ejemplo de utilización podría ser un paseo virtual por una reconstrucción del foro romano.

Los códigos QR son unos identificadores únicos, a modo de códigos de barras que, ubicados de forma adecuada, permiten obtener información en tiempo real sobre algún elemento. Un ejemplo de utilización sería la creación de un audio-guía en un museo, ubicando un código QR cercano a cada obra se podría descargar un fichero de audio con una explicación de la misma.

Acceso inmediato

Un dispositivo móvil proporciona un acceso prácticamente instantáneo al contenido deseado y, en el caso de los docentes, les permite resolver dudas con escasos márgenes de tiempo.

3.2.2.2 Inconvenientes de las plataformas móviles

Duración de la batería

La corta duración de la batería de los dispositivos móviles, especialmente de los Smartphones, resulta un hándicap importante a la hora de seguir una lección a través de este tipo de dispositivos.

Tamaño de la pantalla

La generación de contenido debe ser adaptada específicamente para dispositivos con tamaños de pantalla reducidos. Este hecho puede resultar incómodo desde el punto de vista de la experiencia de usuario.

Recursos del dispositivo

En general, los terminales móviles presentan un conjunto de recursos reducido respecto a los ordenadores personales en cuanto a la capacidad del microprocesador y memoria se refiere. Este hecho puede originar que los contenidos se ejecuten y visualicen de forma lenta, empeorando considerablemente la experiencia de usuario.

3.3 Motivación del proyecto

Es en este contexto tecnológico y educativo dónde surge la idea de aprovechar las ventajas que ofrecen los Smartphones en cuestiones multimedia para construir un curso de inglés interactivo.

Para llevar a cabo este proyecto se ha elegido el sistema operativo Android por las siguientes razones:

- Alta implantación en el mercado
- Documentación abundante y de libre acceso
- Amplia gama de librerías para la resolución de problemas comunes
- Integración con otros estándares tecnológicos
- Compatibilidad con múltiples dispositivos de diversas marcas.

La funcionalidad detallada de la aplicación se introducirá en el siguiente apartado mediante un caso de estudio real.

3.4 Casos de estudio

Con el fin de alcanzar los objetivos propuestos en este TFC, se creará **un prototipo de curso de inglés para dispositivos móviles**.

El curso estará compuesto por

- Unidades didácticas
- Test de control de conocimiento

- Control de lecciones

Unidades didácticas

El usuario tendrá acceso a una serie de unidades didácticas que tratarán sobre un tema concreto del idioma inglés. Existirá un menú que permitirá navegar entre una lección y otra.

Exámenes

Cada unidad didáctica contará con un examen compuesto de varias preguntas destinadas a comprobar la compresión de la unidad didáctica por parte del alumno. En cada ejercicio se expondrá un enunciado, acompañado por una serie de indicaciones para la resolución. Una vez terminado el examen, el sistema corregirá las preguntas y dará una nota final de la unidad didáctica.

Control de lecciones

Con el fin de que el alumno pueda llevar un seguimiento de las lecciones realizadas y de las notas obtenidas, el sistema contará con una funcionalidad que permitirá al alumno marcar las lecciones que ha finalizado y consultar la nota de los exámenes que ha realizado.

Ayuda

Existirá una sección de ayuda que oriente al usuario sobre la utilización del curso.

4. Planificación

4.1 Introducción

El presente apartado describe el plan de trabajo que será llevado a cabo en relación con el TFC MejoraTuIngles.

En él se realiza un análisis y desglose de las tareas que se llevarán acabo, situándolas en un marco temporal.

Se realiza una enumeración del material utilizado para la realización del proyecto y, para concluir, se realiza una presentación de los riesgos del proyecto y de las acciones mitigadoras que ayudarán a corregir posibles desviaciones en la planificación.

4.2 Planificación

En este apartado se enumeran las entregas que debemos realizar, explicando el objeto de cada una de ellas.

4.2.1 Entregables

Proyecto de TFC - PEC1

En esta entrega se exponen las líneas generales del proyecto que se va a llevar a cabo. Se definen el alcance y los objetivos y se realiza una primera aproximación al trabajo que se realizará.

Estado del arte - PEC2

Esta entrega está dedicada a la recopilación y estudio de documentación con el objetivo de obtener una imagen actual del área de conocimiento sobre la que versa el proyecto. En este caso se ha realizado un estudio sobre la situación del e-learning en general, para centrarse posteriormente en el e-learning orientado a dispositivos móviles

Diseño e Implementación - PEC3

Esta entrega engloba las fases de diseño e implementación del proyecto. En esta fase se identifican casos de uso y se diseñan soluciones técnicas para la implementación. Toda la fase de diseño ha sido recopilada en forma de diagramas UML.

Inmediatamente después se ha dado comienzo a la implementación de la solución, utilizando para ello el entorno Eclipse con los complementos de Android. Para resolver algunas cuestiones técnicas se han utilizado patrones extraídos de recomendaciones de buenas prácticas.

La entrega concluye con un breve plan de pruebas funcionales

Memoria – PEC4

En esta fase se confecciona la memoria que vamos a entregar. La entrega consta de un documento principal, dónde se explica el desarrollo del proyecto, y de varios anexos que abarcan temas de vocabulario y explicaciones técnicas, planificación, montaje del entorno de trabajo, bibliografía y código generado.

Presentación – PEC5a

En esta entrega se realizará una presentación abreviada del trabajo realizado en formato PowerPoint. Esta presentación servirá de apoyo al resto de la documentación y proporcionará una idea general de las tareas abordadas y sus resultados.

Defensa – PEC5b

El proyecto concluirá con una defensa en las que se contestarán las dudas o preguntas que puedan ser realizadas por el equipo docente y el tribunal examinador.

Tareas y actividades

A continuación se muestra un desglose rápido de las tareas programadas para cada una de las entregas.

Tareas del Proyecto de TFC - PEC1

⊟ **Proyecto de TFC - PEC1**	**17 días**
Presentación del problema	4 días
Definición de objetivos	2 días
Planificación	5 días
Recopilación de documentación	1 día
Preparación de la entrega	4 días
Entrega PEC1	1 día

Ilustración 1 - Desglose de tareas del Proyecto de TFC (PEC1)

Tareas de Estado del Arte - PEC2

⊟ **Estado del Arte - PEC2**	**15 días**
Recopilación y estudio de documentación sobre e-learning	7 días
Identificación de Casos de Uso	2 días
Preparación de la entrega	8 días
Estudio de prototipo rápido	6 días
Entrega PEC2	0 días

Ilustración 2 - Desglose de tareas de Estado del Arte

Tareas de Diseño de Implementación - PEC3

⊟ **Diseño e implementación - PEC3**	**33 días**
Preparación entorno desarrollo	1 día
Diagrama de clases	2 días
Diagramas de solución técnica en UML	3 días
Diseño de interface de usuario	5 días
Diseño del plan de pruebas funcionales	1 día
Preparación de la entrega	21 días
Codificación	20 días
Pruebas unitarias y funcionales	1 día
Entrega PEC3	0 días

Ilustración 3 - Desglose de tareas de Diseño de Implementación

Tareas de la Memoria – PEC4

⊟ **Memoria - PEC4**	**18 días**
Confección de memoria	9 días
Preparación de anexos	3 días
Repaso aspectos formales	1 día
Entrega de borrador	1 día
Correcciones finales	4 días
Entrega PEC4	0 días

Ilustración 4 - Desglose de tareas de la Memoria

Tareas de la Presentación – PEC5a

⊟ **Presentación - PEC5 - a**	**6 días**
Realización de esquemas y resúmenes	3 días
Redacción de presentación	2 días
Entrega borrador	0 días
Correcciones finales	1 día
Entrega PEC5 - a	0 días

Ilustración 5 - Desglose de tareas de la Presentación

Tareas de la Defensa – PEC5b

⊟ **Defensa - PEC5 - b**	**7 días**
Contestación de preguntas	7 días

Ilustración 6 - Desglose de tareas de la Presentación

4.3 Material

En este apartado se enumeran todos los materiales utilizados para la realización del proyecto, exceptuando la documentación, que vendrá indicada en el apartado de Bibliografía.

4.3.1 Equipos

- Estación de trabajo principal: iMac 2,8 GHz, i7, 16 GB RAM
- Estación de trabajo secundaria: Intel Quadcore 9966 2.5 GHz, 8 GB RAM
- Samsung Galaxy S (I9000)

Software

- Microsoft Word 2007
- Microsoft Power Point 2007
- Microsoft Project 2003
- Microsoft Windows 7
- MacOS 10.8.2
- Eclipse
- ANDROID 2.3
- Google Maps API
- SQLite
- Acrobat Reader X

4.4 Plan de contingencia

En este proyecto se identifican las posibles situaciones de riesgo que podrían darse en este proyecto así como las medidas correctoras que podrían aplicarse para evitar desviaciones en la planificación.

4.4.1 Análisis de Riesgos y Acciones Mitigadoras

Se han identificado los siguientes riesgos:

4.4.1.1 Problemas de integración entre Software

Descripción del riesgo

Pueden producirse problemas de integración entre los diferentes componentes software que van a utilizarse durante el desarrollo del aplicativo.

Acción mitigadora

En este caso se dedicarán más horas al montaje del entorno hasta conseguir dejarlo plenamente operativo.

4.4.1.2 Entregas coincidentes con otras asignaturas

Descripción del riesgo

Pueden existir entregas que coincidan con otras asignaturas.

Acción Mitigadora

Se tomarán días no-hábiles (sábados y domingos) para incrementar el tiempo total disponible. También se cuenta con la posibilidad de solicitar días de vacaciones en mi empleo habitual.

4.4.1.3 Avería de la estación de trabajo

Descripción

Se avería la estación de trabajo, ya sea por avería física o corrupción del entorno.

Acción Mitigadora

Se cuenta con un repositorio CVS para controlar los ficheros generados durante la ejecución del proyecto. También se cuenta con 2 máquinas operativas y, en caso de avería de una, se continuará el desarrollo en la otra.

13.4.1.4 Tiempo disponible para la ejecución del proyecto

Descripción

El tiempo disponible para la ejecución del proyecto es muy escaso.

Acción mitigadora

Se parte con algunos conocimientos básicos de desarrollo en Android, así como de amplia experiencia en desarrollo en entornos Java/J2EE.

4.5 Diagramas de Gantt completos

	Nombre de tarea	Duración	Comienzo	Fin	Pre	Nombres de los recursos
1	**Proyecto de TFC - PEC1**	**17 días**	**lun 17/09/12**	**mar 09/10/12**		
2	Presentación del problema	4 días	lun 17/09/12	jue 20/09/12		
3	Definición de objetivos	2 días	vie 21/09/12	lun 24/09/12	2	
4	Planificación	5 días	mar 25/09/12	lun 01/10/12	3	
5	Recopilación de documentación	1 día	mar 02/10/12	mar 02/10/12	4	
6	Preparación de la entrega	4 días	mié 03/10/12	lun 08/10/12	5	
7	**Entrega PEC1**	1 día	mar 09/10/12	mar 09/10/12	6	

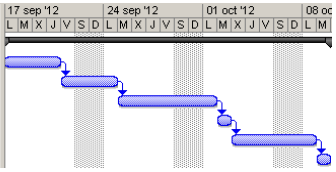


Ilustración 7 - Diagrama Gantt PEC1

8	**Estado del Arte - PEC2**	**15 días**	**mié 10/10/12**	**mar 30/10/12**		
9	Recopilación y estudio de documentación sobre e-learning	7 días	mié 10/10/12	jue 18/10/12		
10	Identificación de Casos de Uso	2 días	vie 19/10/12	lun 22/10/12	9	
11	Preparación de la entrega	8 días	vie 19/10/12	mar 30/10/12	9	
12	Estudio de prototipo rápido	6 días	mar 23/10/12	mar 30/10/12	10	
13	**Entrega PEC2**	0 días	mar 30/10/12	mar 30/10/12	12	

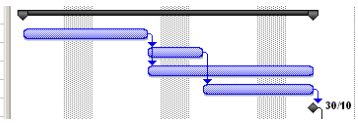


Ilustración 8 - Diagrama de Gantt PEC2

14	**Diseño e implementación - PEC3**	**33 días**	**mié 31/10/12**	**dom 16/12/12**		
15	Preparación entorno desarrollo	1 día	mié 31/10/12	mié 31/10/12	13	
16	Diagrama de clases	2 días	jue 01/11/12	vie 02/11/12	15	
17	Diagramas de solución técnica en UML	3 días	lun 05/11/12	mié 07/11/12	16	
18	Diseño de interface de usuario	5 días	jue 08/11/12	mié 14/11/12	17	
19	Diseño del plan de pruebas funcionales	1 día	jue 15/11/12	jue 15/11/12	18	
20	Preparación de la entrega	21 días	vie 16/11/12	dom 16/12/12	19	
21	Codificación	20 días	vie 16/11/12	jue 13/12/12	19	
22	Pruebas unitarias y funcionales	1 día	vie 14/12/12	dom 16/12/12	21	
23	**Entrega PEC3**	0 días	dom 16/12/12	dom 16/12/12	20	

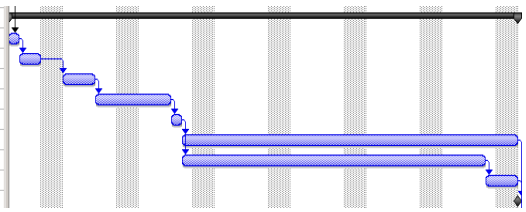


Ilustración 9 - Diagrama de Gantt PEC3

24	**Memoria - PEC4**	**18 días**	**lun 17/12/12**	**mié 09/01/13**		
25	Confección de memoria	9 días	lun 17/12/12	jue 27/12/12	22	
26	Preparación de anexos	3 días	vie 28/12/12	mar 01/01/13	25	
27	Repaso aspectos formales	1 día	mié 02/01/13	mié 02/01/13	26	
28	Entrega de borrador	1 día	jue 03/01/13	jue 03/01/13	27	
29	Correcciones finales	4 días	vie 04/01/13	mié 09/01/13	28	
30	**Entrega PEC4**	0 días	mié 09/01/13	mié 09/01/13	29	

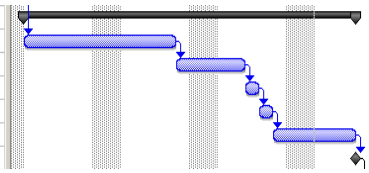


Ilustración 10 - Diagrama de Gantt PEC4

31	**Presentación - PEC5 - a**	**6 días**	**jue 10/01/13**	**jue 17/01/13**		
32	Realización de esquemas y resúmenes	3 días	jue 10/01/13	lun 14/01/13	30	
33	Redacción de presentación	2 días	mar 15/01/13	mié 16/01/13	32	
34	**Entrega borrador**	0 días	mié 16/01/13	mié 16/01/13	33	
35	Correcciones finales	1 día	jue 17/01/13	jue 17/01/13	34	
36	**Entrega PEC5 - a**	0 días	jue 17/01/13	jue 17/01/13	35	

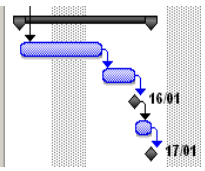


Ilustración 11 - Diagrama de Gantt PEC5a

37	**Defensa - PEC5 - b**	**7 días**	**mié 18/01/12**	**mié 25/01/12**		
38	Contestación de preguntas	7 días	mié 18/01/12	mié 25/01/12		
39	**Fin de Proyecto**	0 días	mié 25/01/12	mié 25/01/12	38	

Ilustración 12 - Diagrama de Gantt PEC5b

5. Análisis de la aplicación

En este capítulo se describe un análisis preliminar del problema para extraer conclusiones y organizar las tareas que debemos llevar a cabo. Este análisis comenzará con una toma inicial de requisitos y proseguirá con un desglose de los mismos utilizando un diagrama de Casos de Uso.

Con el fin de tener una metodología que nos ayude a ejecutar el proyecto se realizó un breve estudio de metodologías de gestión clásicas, sin embargo estas resultaban demasiado pesadas y no ayudaban a salvar dos de los grandes problemas de la ejecución de este proyecto:

- Conocimiento escaso de las tecnologías implicadas
- Escasez de tiempo

Estos factores ponen de manifiesto que se necesitaba una metodología no-exhaustiva, que sirviese de guía para la ejecución del proyecto pero sin grandes exigencias metodológicas ya que sería inviable cumplir con muchas recomendaciones.

Finalmente se ha optado por elegir una combinación entre metodología SCRUM y eXtremeProgramming. Para más información sobre estos paradigmas, consúltense los anexos Anexo I – Terminología Aplicable y Anexo II – Tecnologías y Metodología de Desarrollo.

5.1 Recopilación de requisitos mediante Historias de Usuario

En este apartado se configuran las historias de usuario, fruto del análisis del problema que debemos resolver, expresado en el apartado 4.2, “Casos de estudio”.

5.1.1 Historia de "Estudiar unidad didáctica"

Un usuario que estudia inglés quiere poder hacerlo aprovechando el tiempo que le dedica diariamente al transporte. Para ello quiere tener la posibilidad de estudiar temas de forma guiada en un dispositivo móvil.

5.1.2 Historia de "Realizar examen"

Un usuario que ha concluido una unidad lectiva quiere comprobar el grado de asimilación de la misma, para ello querría tener la posibilidad de realizar un test que le ayude a evaluar los conocimientos adquiridos, indicándole los fallos que cometa.

5.1.3 Historia de “Seguimiento de progresos”

El usuario hace uso del sistema cuando su tiempo se lo permite, este tiempo puede no seguir una pauta regular, teniendo como consecuencia que pueden pasar días entre una sesión de estudio y otra. Al usuario le gustaría poder marcar su nivel de avance, señalando de algún modo las unidades didácticas que ha completado.

5.2 Identificación de Casos de Uso

Los requisitos contenidos en las historias de usuario permiten identificar los casos de uso que manejaremos en este proyecto.
El siguiente diagrama pretende mostrar los casos de uso identificados, así como las relaciones existentes entre ellos:

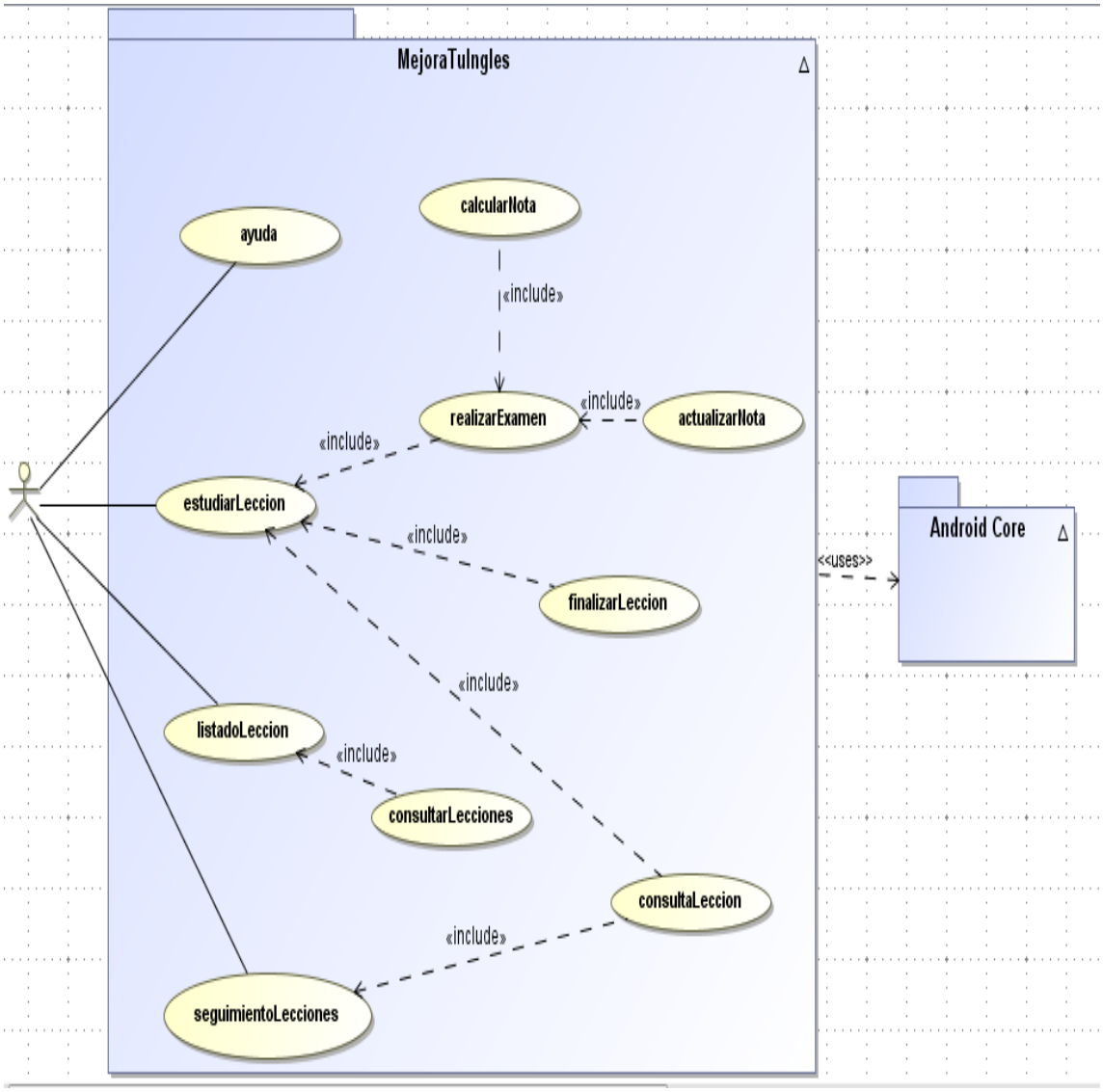


Ilustración 13 - Diagrama de casos de uso

6. Diseño del Sistema

En este capítulo se desarrollará una propuesta de diseño de las funcionalidades identificadas en el capítulo 6. Análisis de la aplicación. Para realizar los diseños se utilizará la técnica de Diagramas de Secuencia, por considerarse suficientemente ilustrativa para este caso. La notación utilizada será UML 2.0.

Como paso previo a la lectura de este capítulo y, con el fin de clarificar conceptos, se recomienda la lectura del Anexo II – Tecnologías y Metodología de Desarrollo dónde se introducirán algunos conceptos sobre la arquitectura de Android que servirán para una mejor comprensión de los diagramas y de las soluciones técnicas adoptadas.

6.1 Diagrama de clases del Sistema

En este apartado se realizarán los diagramas de las clases necesarios para explicar los elementos existentes en la aplicación y las relaciones que se producen entre ellos.

Es necesario establecer una diferenciación, a nivel conceptual, entre dos tipos de elementos:

Beans, son componentes que representan a una entidad del "mundo real", formados por datos simples y métodos de acceso (get y set) a ellos. No tienen ninguna otra funcionalidad asociada.

Resto de clases (***Actividades***, ***Adaptadores***, ***Helpers***, etc...) son las clases que implementan la funcionalidad de la aplicación (vistas, consultas a BBDD, lectura de ficheros, etc).

6.1.1 Diagrama de Beans

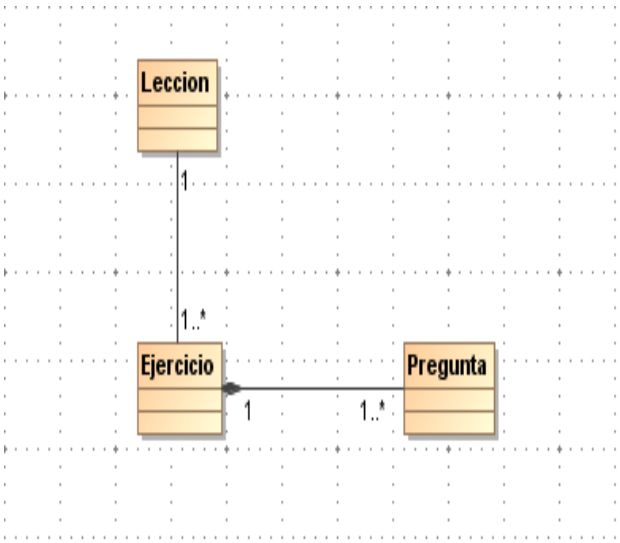


Ilustración 14 - Beans

Como se ha comentado, estas clases son simples almacenes de datos con métodos de establecimiento y acceso.

La clase “Leccion” representa a una unidad didáctica. Una lección puede tener 1 o más ejercicios, mientras que un ejercicio pertenece a una única lección. Se ha modelado como una composición la relación entre Ejercicio y Pregunta con el fin de subrayar la dependencia entre uno y otro de tal forma que el Ejercicio carece de sentido sin preguntas y las preguntas no pueden existir si no están relacionadas con un ejercicio.

6.1.2 Diagrama de clases

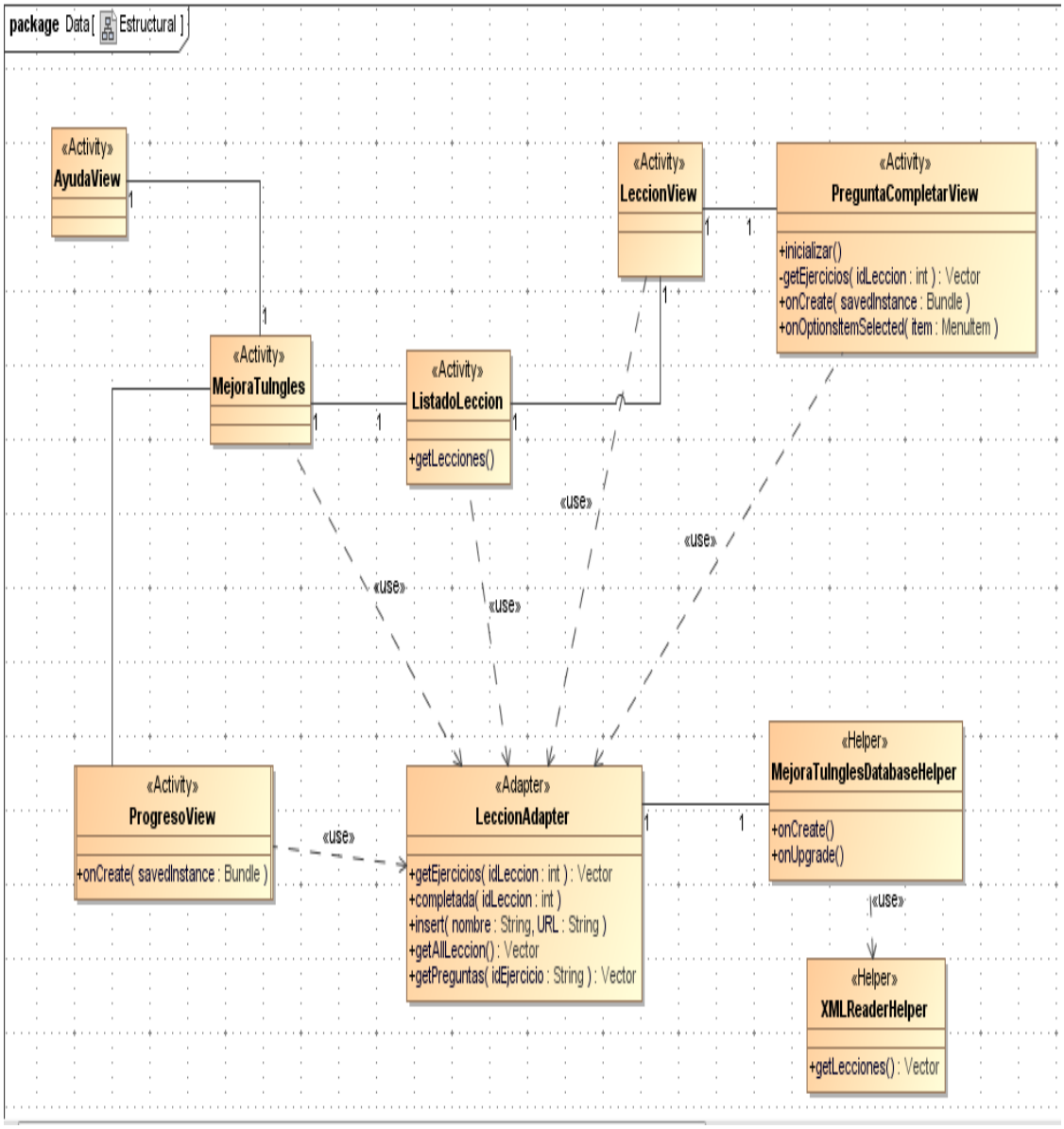


Ilustración 15 - Diagrama de clases

El sistema está constituido por un conjunto de actividades, manejadores y ayudantes, estereotipados como <<Activity>>, <<Handler>> y <<Helper>>, que manejan los objetos de tipo Bean, que son los que contienen los datos. Los conceptos de Handler y Helper son comúnmente aceptados, provienen de diversos frameworks de ayuda a la programación y hacen referencia a clases que operan con algún tipo de dato significativo desde el punto de vista de negocio (Handler) o a clases que ocultan alguna funcionalidad técnicamente completa y muy recurrente, como los accesos a BBDD (***Helper***).

6.2 Diagramas de solución técnica

En este apartado se detallan, en forma de diagramas de secuencia, las soluciones técnicas adoptadas para la resolución de las tareas más importantes de la aplicación. Es necesario apuntar que estos diagramas no son exhaustivos y contienen cierto nivel de abstracción, dejando de lado detalles técnicos en favor de una mejor compresión de la explicación.

6.2.1 Estudio de lección

Esta funcionalidad es la encargada de mostrar por la pantalla la lección que va a estudiar el usuario. Se accede a ella desde varios puntos de la aplicación y permite seleccionar una lección entre un listado de lecciones y mostrarla en pantalla. El proceso puede observarse en la siguiente ilustración.

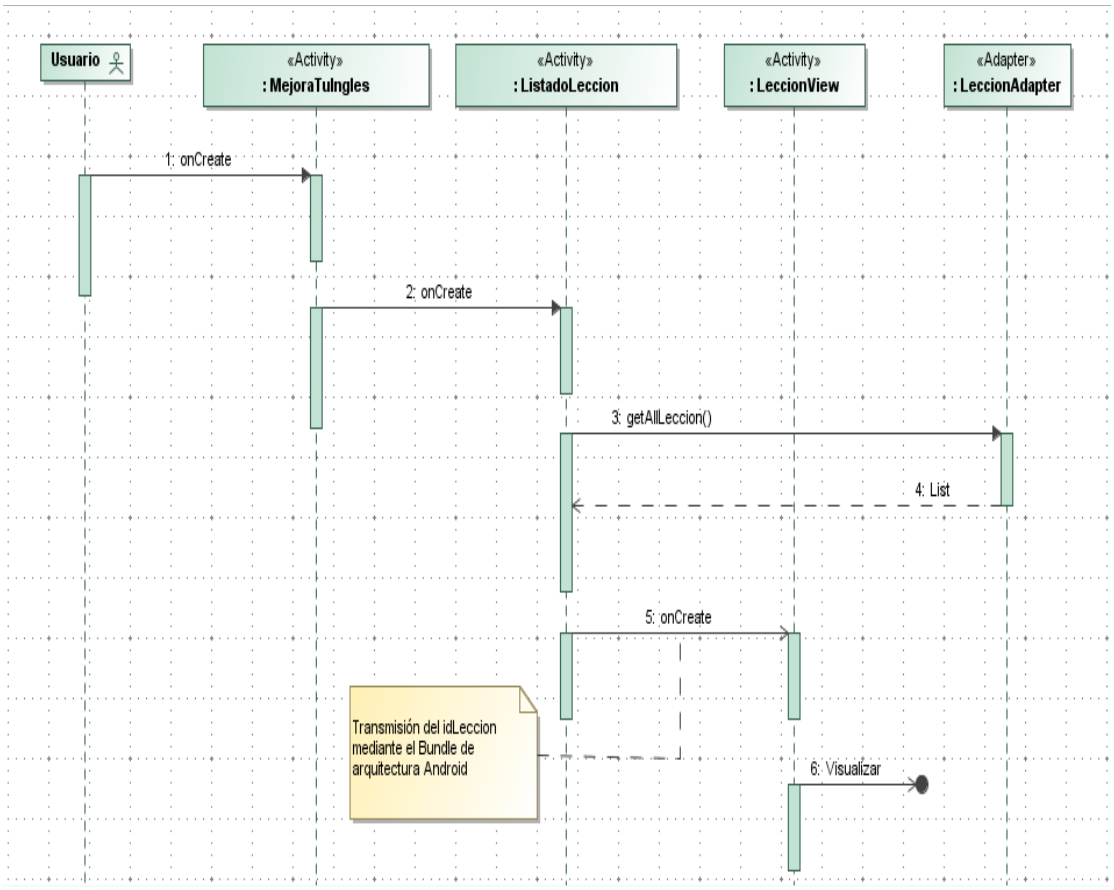


Ilustración 16 – Estudio de lección

6.2.2 Realización de test

Se trata de la funcionalidad técnicamente más compleja de la aplicación. Una vez finalizada una unidad didáctica, el alumno puede realizar un test de comprensión, los test están divididos en ejercicios que, a su vez, se dividen en preguntas. El sistema permitirá al alumno navegar entre las preguntas que componen el test, irá anotando sus respuestas y le proporcionará una nota final. En el siguiente diagrama puede observarse el proceso seguido.

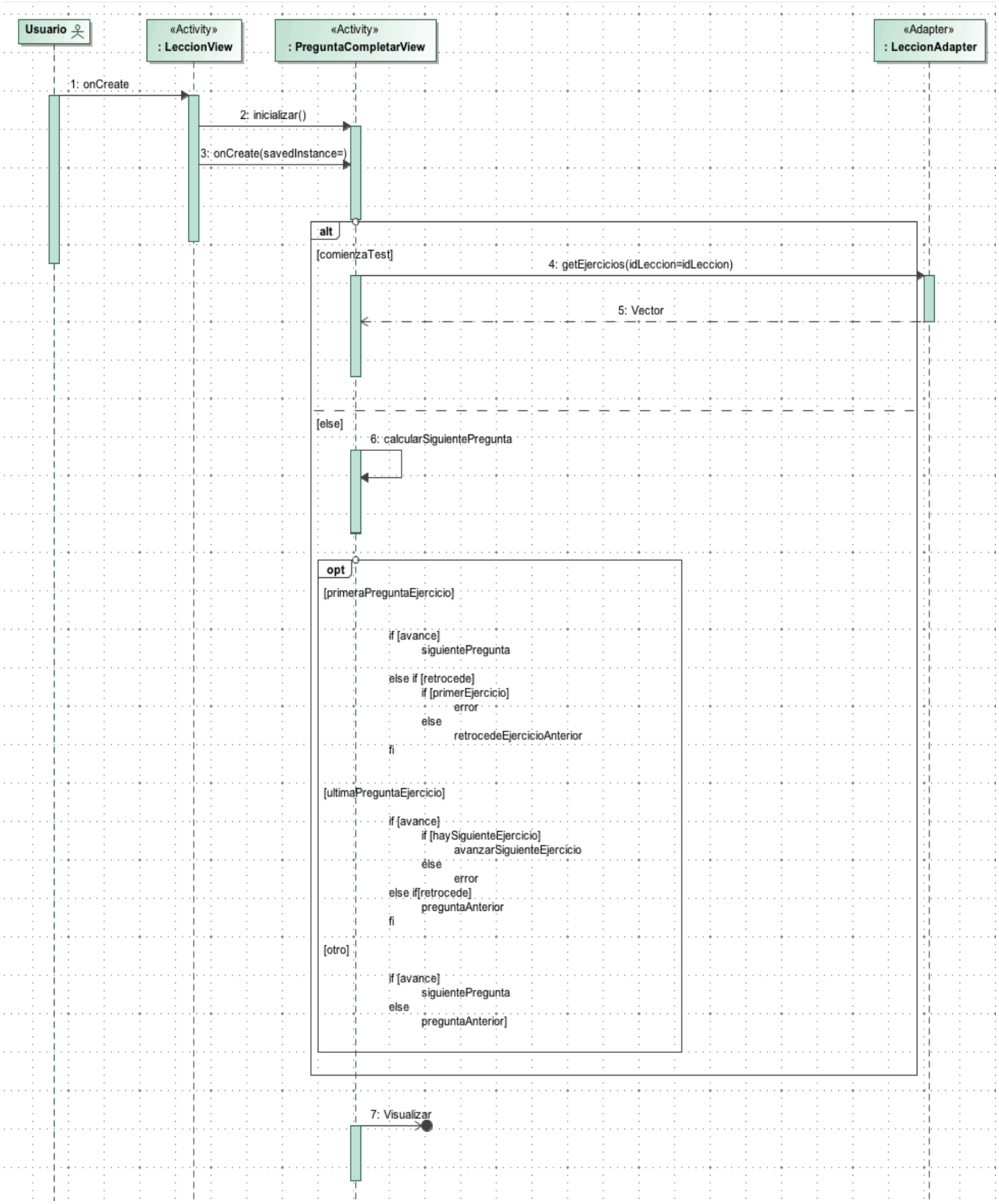


Ilustración 17 – Funcionalidad del test

6.2.3 Registrar progreso

Esta funcionalidad permite al alumno llevar un seguimiento de las unidades que ha cursado. Para ello, una vez finalizada la lección, el alumno puede darla por concluida, quedando este hecho registrado en el sistema.

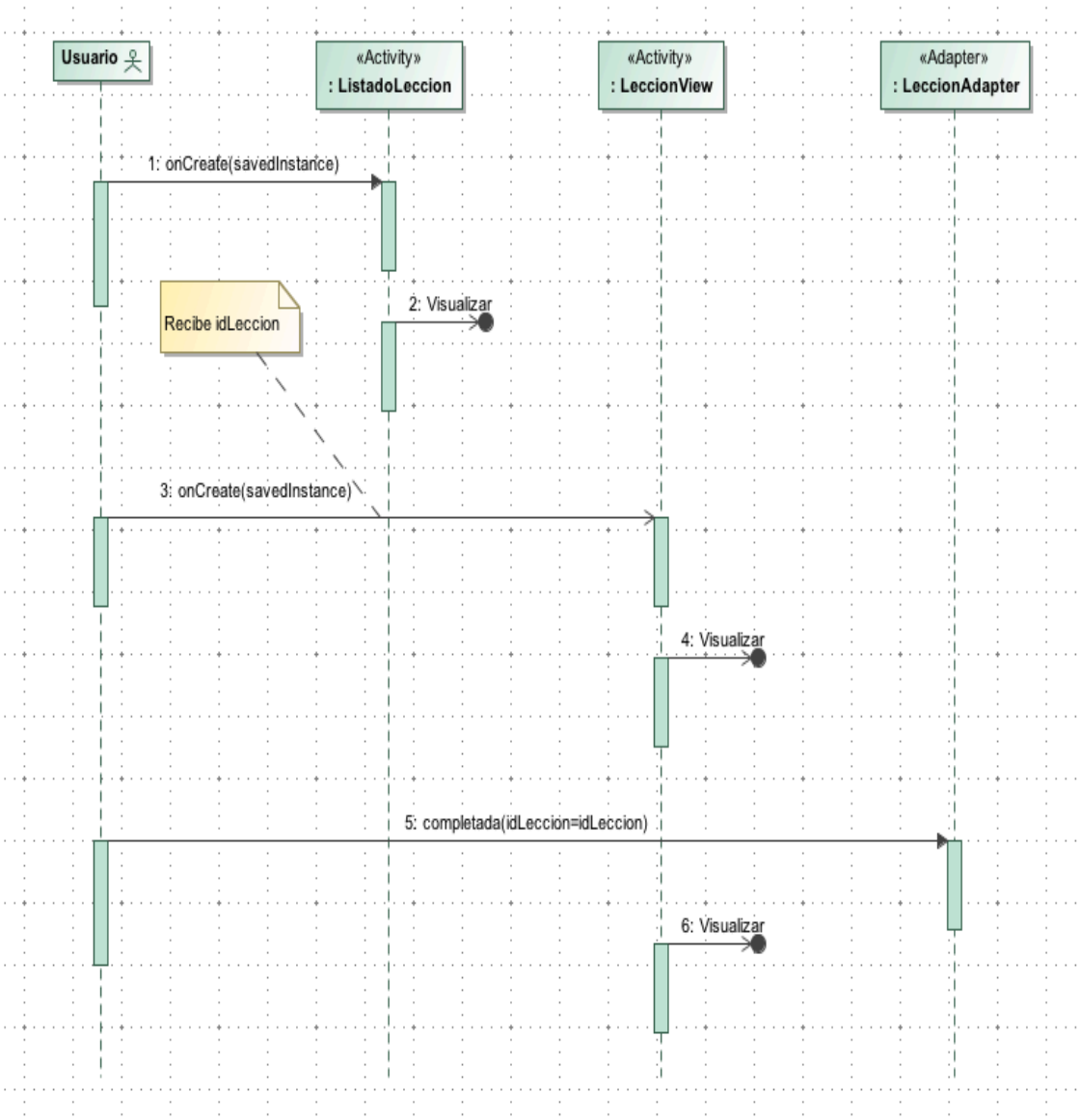


Ilustración 18 – Registrar progreso

6.2.4 Consultar progreso

Esta funcionalidad permite al alumno consultar las unidades que ha cursado y los exámenes que ha realizado, con su respectiva calificación.

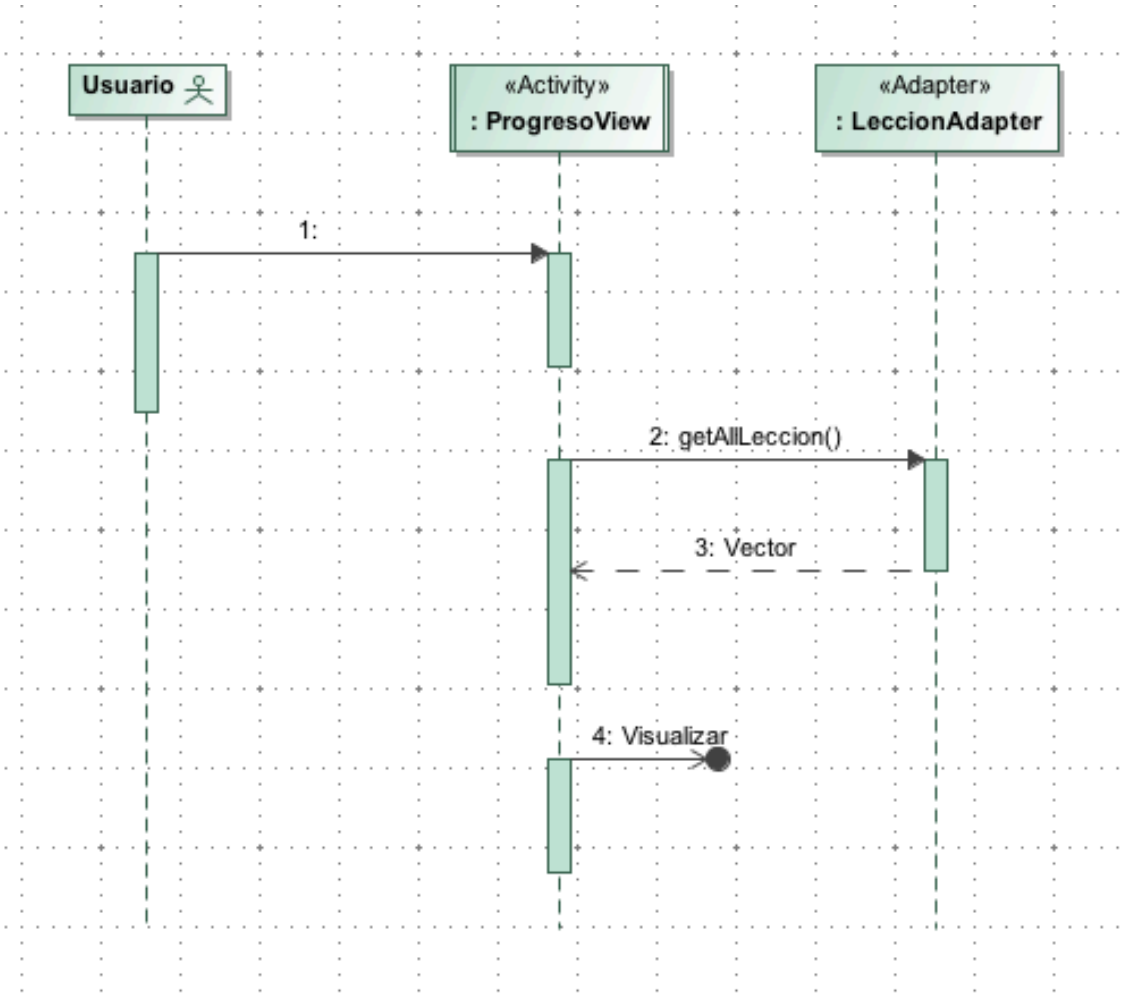


Ilustración 19 – Consultar progreso

6.3 Diseño del modelo de datos

Con el fin de almacenar los datos que manejaremos en el aplicativo, se hace necesario un elemento que nos permita guardar estos datos y recuperarlos de forma ágil cuando sea necesario. Este elemento es el Sistema Gestor de Bases Datos.

A continuación, abordaremos la creación de un modelo de datos que nos permita dar respuesta a las necesidades de gestión de datos de este TFC.

Según Wikipedia, "*un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, un modelo de datos permite describir las estructuras de datos de la base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base).*"

En el caso que nos ocupa se ha considerado suficiente el siguiente modelo.

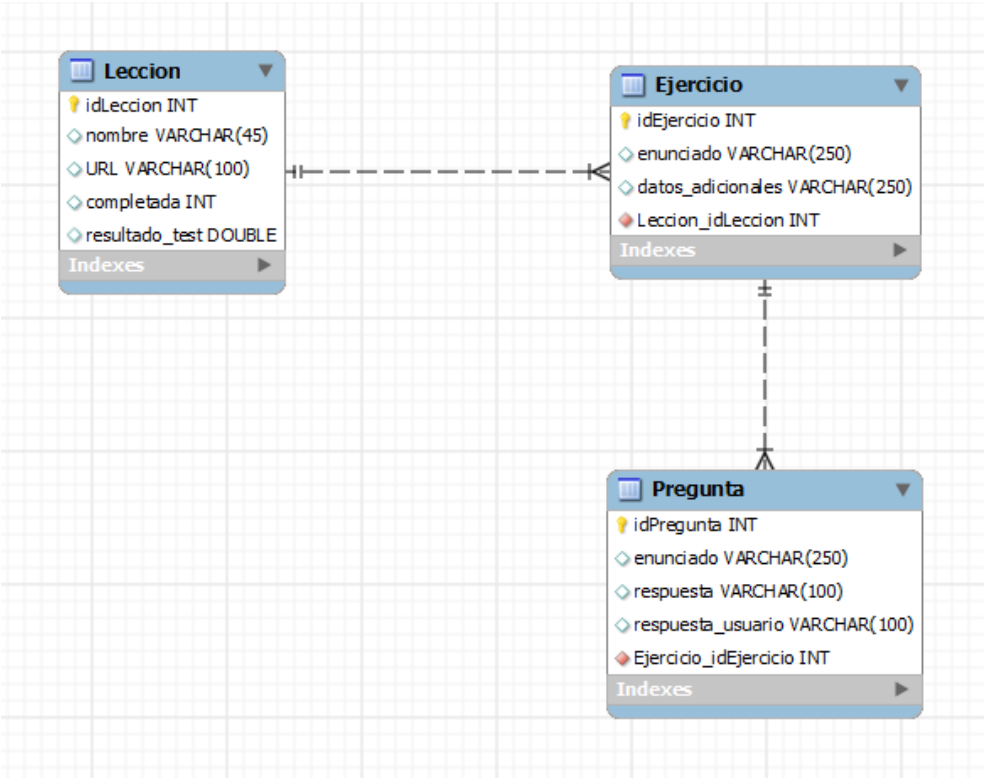


Ilustración 20 - Modelo de datos

Como puede observarse, es un modelo de datos muy simple pero que, al mismo tiempo, permite dar respuesta a nuestras necesidades.

A continuación procedemos a realizar una breve explicación de cada una de las tablas que componen el modelo.

6.3.1 Lección

En esta tabla se almacenarán las lecciones disponibles, así como el estado en el que se encuentran.

6.3.2 Ejercicio

Cada una de las lecciones puede tener una serie de ejercicios de compresión. Estos ejercicios serán almacenados en la tabla "Ejercicio"

6.3.3 Pregunta

Cada uno de los ejercicios está compuesto por una serie de preguntas, la tabla "Pregunta" almacena las preguntas que componen los ejercicios, como puede observarse, existe una relación 1:n entre las tablas "Ejercicio" y "Pregunta".

7. Interfaz de Usuario

La interfaz gráfica de usuario es una parte fundamental de la aplicación. Una elaboración temprana, en forma de prototipo, puede servir para mostrar al cliente los resultados que se obtendrán a la finalización del proyecto y poder realizar correcciones antes de la implementación.

De acuerdo con el manifiesto ágil de desarrollo de software, es deseable "*poder ver anticipadamente cómo se comportan las funcionalidades esperadas sobre prototipos o sobre las partes ya elaboradas del sistema final ofrece una retroalimentación (feedback) muy estimulante y enriquecedor que genera ideas imposibles de concebir en un primer momento; difícilmente se podrá conseguir un documento que contenga requisitos detallados antes de comenzar el proyecto.*"

En este capítulo se introducirán las líneas generales seguidas para el diseño de la interfaz y las motivaciones por las cuales se han adoptado. Para una mejor compresión de la explicación se recomienda consultar el Anexo II – Tecnologías y Metodología de Desarrollo, ya que en él se ofrece una introducción breve a los conceptos que intervienen en el diseño de interfaces en Android.

En este TFC, con fines didácticos, se han implementado las vistas de varias formas distintas, de tal forma que pueda ofrecerse una panorámica de las diferentes técnicas que pueden utilizarse.

7.1 Diseño de la interfaz gráfica de usuario

En la orientación del diseño, se han tenido en cuenta 3 aspectos fundamentales:

- Estética
- Experiencia de usuario
- Independencia del dispositivo

7.1.1 Estética

El diseño de la aplicación a nivel visual resulta fundamental en el éxito o fracaso de cualquier aplicación. En una aplicación para dispositivos móviles, además, resulta fundamental adaptarse al tamaño de la pantalla y organizar los elementos de modo que puedan verse de un solo vistazo.

7.1.2 Experiencia de usuario

En entornos móviles una aplicación debe estar diseñada para llegar fácilmente a las funcionalidades que ofrece, con el menor número de interacciones y cargas posibles y de la forma más cómoda e intuitiva posible.

7.1.3 Independencia del dispositivo

En el mundo de los dispositivos móviles no existe un estándar de pantalla de visualización, y tampoco de recursos disponibles (memoria, cámaras o sensores). De este modo deben utilizarse técnicas que adapten de forma dinámica la aplicación al dispositivo físico que la soporta.

7.2 Presentación

La presentación principal de la aplicación está compuesta de una única imagen ubicada en un LinearLayout.

El LinearLayout es un gestor de posicionamiento que ubica los elementos que contiene uno tras otro, en forma horizontal o vertical. Es el tipo de gestor de posicionamiento básico y a partir de él se pueden llevar a cabo maquetaciones complejas.

Ilustración 21 - Pantalla de presentación

```
activity_mejora_tu_ingles.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <ImageView
        android:id="@+id/logotipo"
        android:src="@drawable/logotipo"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:contentDescription=""
    />
</LinearLayout>
```

Contenedor

Imagen

Ilustración 22 - Código de Layout principal

Para posibilitar la interacción del usuario con la aplicación, a todas las pantallas se les ha añadido un menú contextual que aparece cuando el usuario pulsa el botón del menú del teléfono. Este menú se crea también utilizando un layout particular para menús. El mecanismo que se observa en la figura es el seguido, tanto en esta vista, como en el resto de vistas de la aplicación.

```
activity_mejora_tu_ingles.xml   activity_mejora_tu_ingles.xml
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/MnuOpc1" android:title="Units" android:icon="@drawable/icon_lecciones"></item>
    <item android:id="@+id/MnuOpc2" android:title="Progress" android:icon="@drawable/icon_calendario"></item>
    <item android:id="@+id/MnuOpc3" android:title="Help" android:icon="@drawable/icon_ayuda"></item>
</menu>
```

Ilustración 23 - Código de layout de menú de la página principal

Tal y como puede observarse en la figura, se añade un layout menú y, por cada opción, se añade un “ítem”, que representa al botón que aparecerá en el menú contextual.

7.3 LeccionView

En el diseño de la página de presentación de lección se ha tenido en cuenta la complejidad y el posible dinamismo del contenido que albergará. Las lecciones no serán exactamente iguales entre sí, pueden contener pequeñas diferencias en cuadros explicativos o número de ejemplos. Estás diferencias podrían complicar en exceso la implementación de la presentación, por este motivo se ha juzgado interesante utilizar WebView Layout, obteniendo de esta forma la facilidad de implementación de HTML, así como su flexibilidad para determinados tipos de presentación.

El efecto conseguido es el mostrado en la siguiente ilustración:

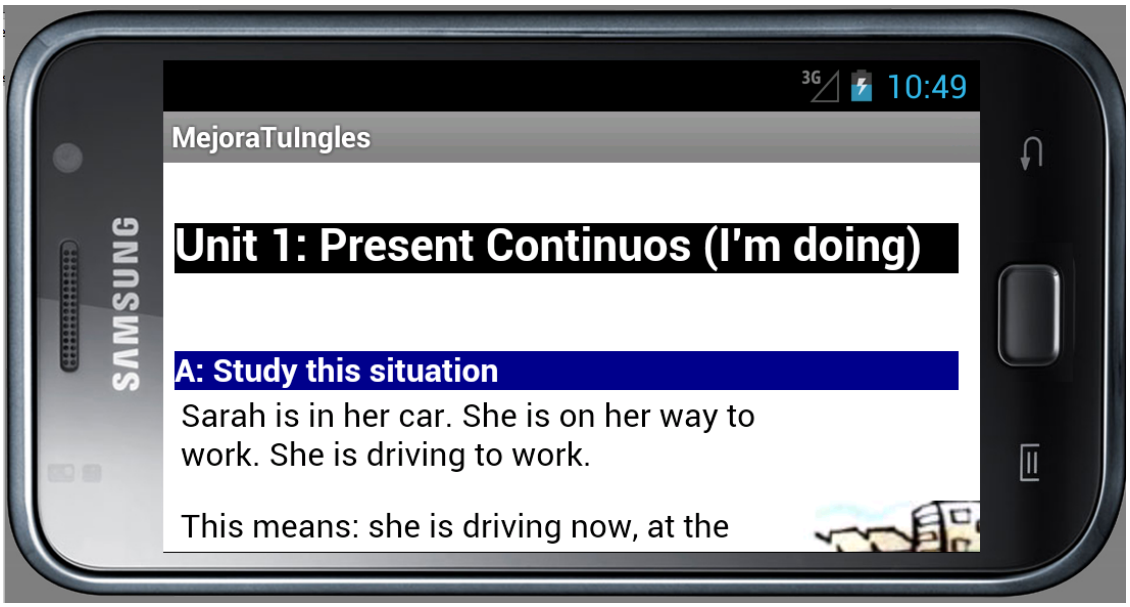


Ilustración 24 – Lección cargada

El código utilizado para la maquetación de la actividad puede observarse en la siguiente ilustración:

```
activity_mejora_tu_ingles.xml   activity_mejora_tu_ingles.xml   leccion_layout.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:padding="4dip"
            android:textSize="16sp"
            />

    <WebView
        android:id="@+id/webview"
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        />

</LinearLayout>
```

Layout WebView

Ilustración 25 - Utilización del WebView

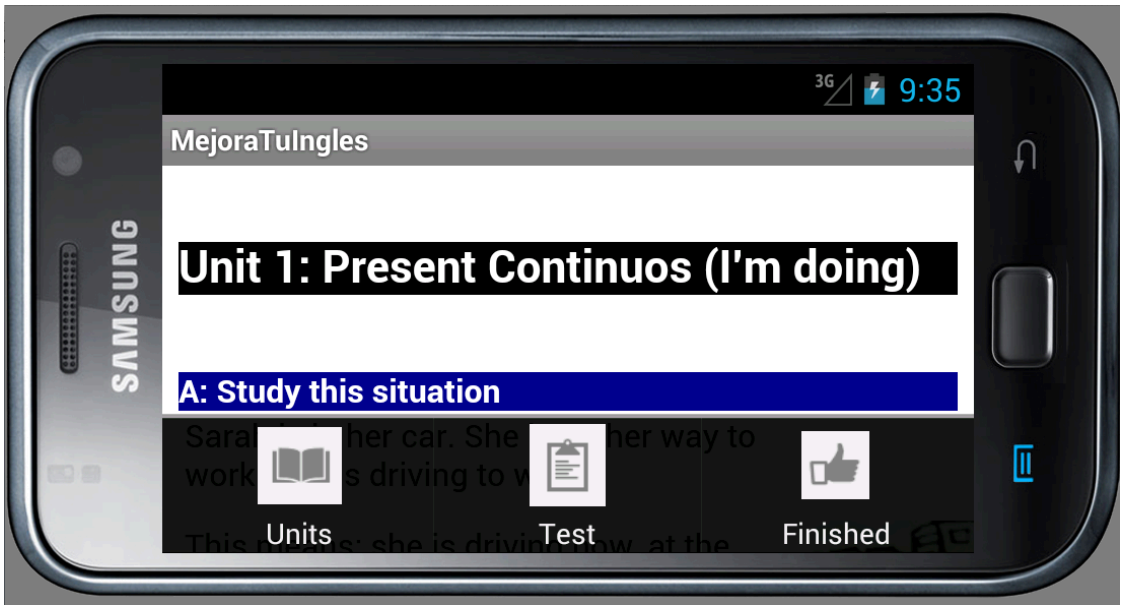


Ilustración 26 - Menú de LeccionView

La página cuenta con un menú contextual de tres opciones. La primera de ellas nos permite regresar al listado de unidades didácticas, mientras que la segunda y tercera nos dan acceso a la prueba de conocimientos de la lección y a marcar la lección como finalizada, de cara al control del progreso.

7.4 Listado de lecciones

Las lecciones catalogadas se presentan en forma de lista. Esta forma de presentación permite navegar de forma cómoda por cada uno de ellos.

Ilustración 27 - Listado de lecciones

Para la implementación de lista se ha utilizado un elemento ListView, configurado tal y como se muestra en el siguiente fragmento de código.

```
activity_mejora_tu_ingles...   activity_mejora_tu_ingles...   leccion_layout.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ListView
        android:id="@+id/android:list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```

listado

Ilustración 28 - Listado de lecciones

Cada elemento de la lista es igualmente configurable mediante un layout.

```
leccion_item.xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:padding="8dip">

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_marginRight="8dip"
       />

    <TextView
        android:id="@+id/row_toptext"

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"

        android:layout_toRightOf="@id/icon"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_centerVertical="true"

        android:textSize="17dp"
        android:textStyle="italic"

        android:singleLine="true"
        android:ellipsize="marquee"/>

</RelativeLayout>
```

Tipo de posicionamiento

icono de avance

nombre de la lección

Ilustración 29 - Configuración de elemento de una lista

Obsérvese que se ha utilizado un posicionamiento relativo en el layout de ítems. El motivo es que es necesario que los ítems convivan en una lista cuyo número de elementos es desconocido en tiempo de diseño.

Con la técnica empleada se está delegando el posicionamiento del bloque ítem en el ListView de la actividad, asegurando de esta forma su correcto pintado en pantalla.

7.5 Test

La pantalla de test nos da acceso a la prueba de conocimiento de cada unidad didáctica. Cada una de las unidades tiene su propio test. En la ilustración 24 puede apreciarse el aspecto de esta vista.

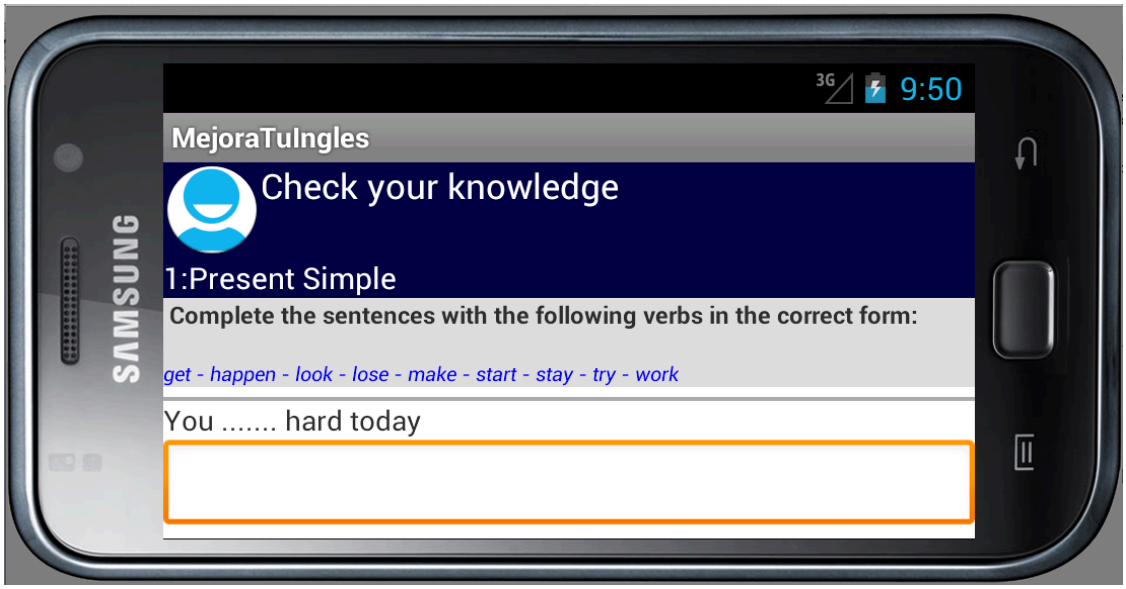


Ilustración 30 - Pantalla de test

La vista está diferenciada en 5 partes, cada una de las cuales se recuadra independientemente en la siguiente figura.

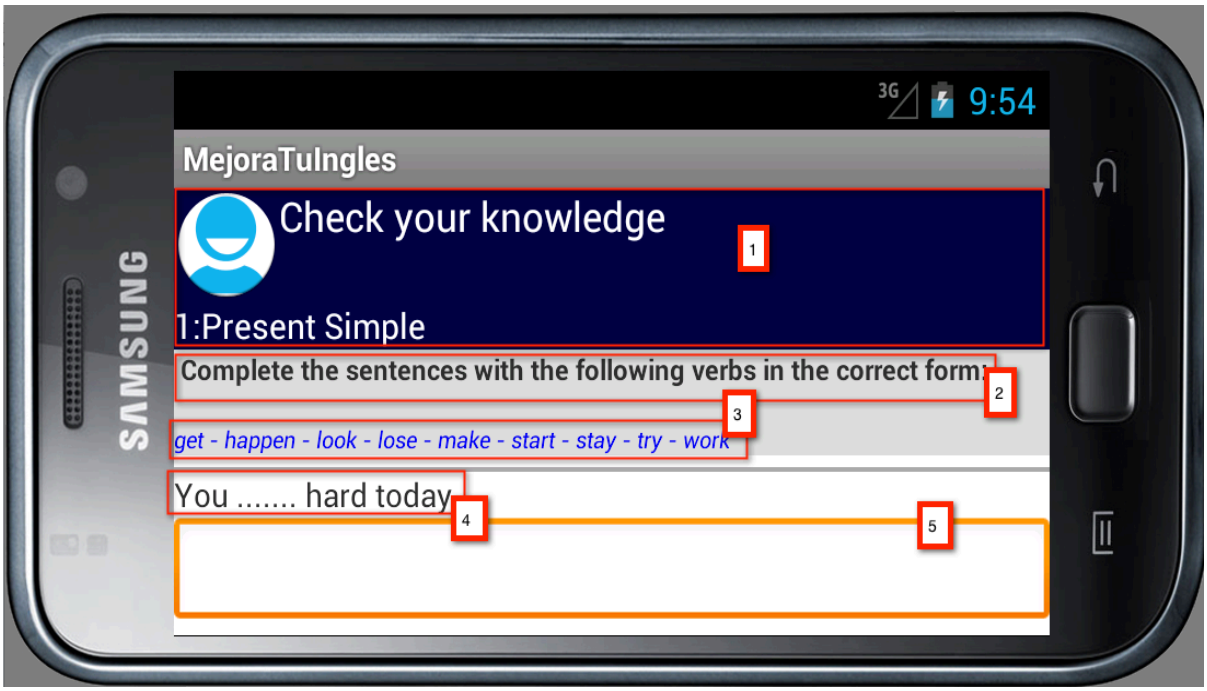


Ilustración 31 - Áreas definidas en la vista

En la primera área se consigna el título de la lección, a modo de introducción. En las áreas 2 y 3 se encuentran el enunciado general del ejercicio y los datos adicionales del este, respectivamente.

Las áreas 4 y 5 corresponden a la pregunta concreta dentro del ejercicio y a su contestación por parte del usuario.

A continuación se muestra la construcción del layout de la vista, señalando los bloques que la forman.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#000044" >

           <ImageView
         android:id="@+id/imageView1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:src="@drawable/ic_launcher"
         android:background="#000044" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Check your knowledge"
            android:textColor="#ffffff"
            android:background="#000044"
            android:textSize="35px"/>
   </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:background="#000044" >

        <TextView
            android:id="@+id/tema"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:textColor="#ffffff"
            android:textSize="30px"

            />

    </LinearLayout>
```

Area 1

Configuración de la cabecera (Imagen y slogan)

Nombre del tema

Ilustración 32 - Configuración del área 1

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#dddddd">
    <TextView
        android:id="@+id/enunciadoGeneral"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textColorHighlight="#000000"
        android:textSize="24px"
        android:textStyle="bold"/>
    <TextView
        android:id="@+id/datos_adicionales"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="20px"
        android:textStyle="italic"
        android:textColor="#0a04f0"/>
</LinearLayout>

<View android:layout_width="fill_parent"
android:layout_height="5dp"
android:background="@android:color/white"/>

<View android:layout_width="fill_parent"
android:layout_height="2dp"
android:background="@android:color/darker_gray"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/pregunta"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView" />

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/respuesta"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>
    <LinearLayout
        android:id="@+id/capaRespuesta"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:background="#000044">
        <ImageView
            android:id="@+id/iconoCheck"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleType="fitEnd"
            android:maxWidth="50px"
            android:maxHeight="50px"/>
        <TextView
            android:id="@+id/respuestaCorrecta"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="25px"
            android:textColor="#ffffff"/>
    </LinearLayout>
```

Área 2

Enunciado del ejercicio y datos adicionales

Área 3

Área 4

Enunciado de la pregunta

Área 5

Zona de respuesta

Ilustración 33 - Resto de áreas

La vista se completa con un menú contextual que permite al usuario navegar entre los ejercicios de una lección y entre las preguntas que los componen.

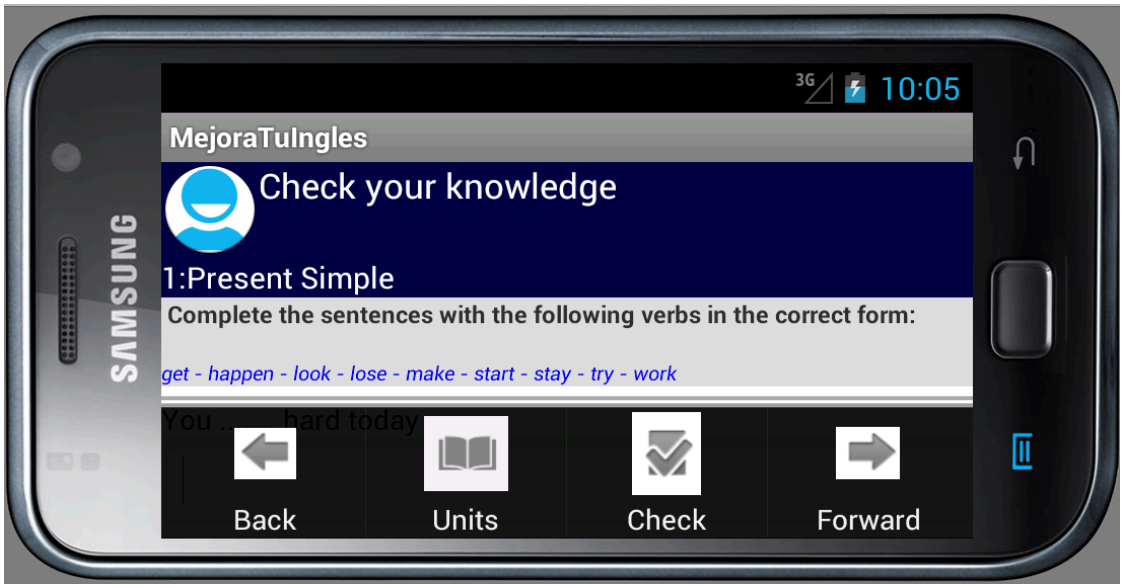


Ilustración 34 - Navegación de la prueba de conocimiento

En la siguiente ilustración puede observarse la configuración de la botonera del menú contextual.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
 <item android:id="@+id/MnuOpc7" android:title="Back" android:icon="@drawable/icon_flecha_retroceso"></item>
 <item android:id="@+id/units" android:title="Units" android:icon="@drawable/icon_lecciones"></item>
 <item android:id="@+id/check" android:title="Check" android:icon="@drawable/icon_corregir"></item>
 <item android:id="@+id/MnuOpc9" android:title="Forward" android:icon="@drawable/icon_flecha_avance"></item>

</menu>
```

Ilustración 35 - Menú contextual del test

7.6 Control de progreso

La pantalla de control de progreso permite al usuario consultar las unidades didácticas realizadas así como las calificaciones de sus pruebas de conocimiento (en caso de haberlas realizado).

Ilustración 36 - Control de progreso

Para la realización de esta tabla se ha utilizado un layout de tipo tabla, dando estilo a las celdas de la cabecera y del cuerpo de la tabla por separado. A continuación mostramos el código del layout, así como el código utilizado para definir la cabecera y el cuerpo de la tabla.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent" android:orientation="vertical"
android:padding="5dp" android:gravity="center_horizontal">

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

      <ImageView
     android:id="@+id/imageView1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:src="@drawable/logo" />

      <TextView android:text="Unit Ranking"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="28dp"
     />

</LinearLayout>

<TableLayout android:id="@+id/cabecera"
android:layout_width="wrap_content" android:layout_height="wrap_content">
</TableLayout>

<ScrollView android:id="@+id/scroll"
android:layout_width="wrap_content" android:layout_height="250dp"
android:scrollbars="vertical" >
   <LinearLayout android:orientation="vertical"
   android:layout_width="wrap_content" android:layout_height="wrap_content">
      <TableLayout android:id="@+id/tabla"
      android:layout_width="wrap_content" android:layout_height="wrap_content">
      </TableLayout>
   </LinearLayout>
</ScrollView>

</LinearLayout>
```

Cabecera de la actividad

Layout de tabla, con scroll incorporado

Ilustración 37 - Layout de control de progreso

Como puede observarse en la imagen, se ha creado un TableLayout para la cabecera de la tabla y otro diferente para el cuerpo de la tabla. Este último está recubierto por un scroll, de tal forma que el aumento del número de lecciones no supone un problema para la visualización ya que esta se adapta automáticamente al crecimiento del número de elementos.

Los códigos para perfilar el aspecto de cabecera y cuerpo de tabla son los que se muestran a continuación.

```xml
<?xml version="1.0" encoding="utf-8"?>
  <shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke android:width="1dp" android:color="#FF2E69AD" />
  <solid android:color="#FFAEE4E8"/>
  <padding android:left="5dp" android:top="10dp"
  android:right="5dp" android:bottom="10dp" />
</shape>
```

Ilustración 38 - Configuración de la cabecera

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke android:width="1dp" android:color="#FF2E69AD" />
  <solid android:color="#FFFFFFFF"/>
  <padding android:left="5dp" android:top="5dp"
  android:right="5dp" android:bottom="5dp" />
</shape>
```

Ilustración 39 - Configuración del cuerpo

7.7 Ayuda

La pantalla de ayuda está pensada para ofrecer al usuario una idea general de las funcionalidades de la aplicación, pero en su propio idioma, en este caso Español.

Presenta una configuración muy sencilla.

Ilustración 40 - Vista de ayuda

Se ha vuelto a optar por WebView dada su facilidad de implementación. La definición del layout es la que se muestra en la ilustración 35.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:padding="4dip"
            android:textSize="16sp"
            />

    <WebView
        android:id="@+id/ayuda"
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        />

</LinearLayout>
```

Contenido de la ayuda

Ilustración 41 - Layout de ayuda

8. Implementación

En este capítulo se analizarán los detalles técnicos concernientes a la implementación del software.
El primer paso para llevar a cabo la implementación será la puesta a punto del entorno de desarrollo, cuyo proceso de instalación es tratado en el Anexo III - Configuración del entorno de desarrollo.

Las explicaciones que a continuación se detallan se caracterizan por tener una fuerte componente técnica, así que, con el fin de resolver dudas terminológicas o referentes a la tecnología empleada, es recomendable la consulta de los Anexos I - Terminología Aplicable y Anexo II - Tecnologías y Metodología de Desarrollo.

Todo el código fuente de la aplicación se entrega por separado en un documento de código.

8.1 Soluciones técnicas

En este apartado se explican las soluciones técnicas más importantes adoptadas para la implementación de las funcionalidades de este TFC.

8.1.1 Operaciones con Base de Datos

El sistema operativo Android puede integrarse con motores de BBDD que respeten la especificación WebSQL. Android incluye, entre las librerías que componen su núcleo, un módulo de soporte a la conexión con SQLite.
SQLite es una BBDD pensada para funcionar en sistemas con recursos escasos (véase Anexo II - Tecnologías y Metodologías de Desarrollo).

Para realizar las operaciones con BBDD se ha escogido una solución basada en ***Helpers*** y ***Adapters***, tal y como se comentó en el capítulo 7 Diseño del Sistema.

La clase principal de la librería de conexión es ***SQLiteOpenHelper***. Esta clase contiene métodos que pueden ser re-escritos para adaptarse al negocio de la aplicación que la utiliza. En el caso que nos ocupa se ha creado la clase ***MejoraTuInglesDatabaseHelper***, que se encargará de las siguientes tareas:

- Creación de un nuevo fichero de BBDD (en caso de no existencia)

- Actualización del fichero de BBDD (en caso de cambio de estructura o versión)

En el inicio de la aplicación se crea una instancia de ***MejoraTuInglesDatabaseHelper***. Esta clase realiza una comprobación de la existencia del fichero de BBDD de la aplicación. Si el fichero no está creado, se procede a la creación mediante el código situado en el método ***onCreate***. En la ilustración 31 puede observarse como se crea la estructura de la BBDD.

```
@Override
public void onCreate(SQLiteDatabase db) {

    int idEjercicio=0;


    Log.d("[MejoraTuIngles]","Ejecutando el script de creacion con modificaciones");

    //Script completo de creaci⊕n de BBDD.
    db.execSQL("CREATE  TABLE \""+LECCION+"\" (\"ID\" INTEGER PRIMARY KEY  AUTOINCREMENT NOT NULL , \"NOMBRE\" VARCHAR NOT NULL  UNIQUE," +
            " \"URL\" VARCHAR NOT NULL  UNIQUE, \"COMPLETADA\" INTEGER, \"RESULTADO_TEST\" VARCHAR );");
    db.execSQL("CREATE  TABLE \""+EJERCICIO+"\" (\"id\" INTEGER PRIMARY KEY  AUTOINCREMENT NOT NULL ,\"id_leccion\" INTEGER, \"enunciado\"" +
            " VARCHAR NOT NULL , \"datos_adicionales\" VARCHAR, \"calificacion\" VARCHAR,  FOREIGN KEY(id_leccion) " +
            "REFERENCES LECCION (id));");
    db.execSQL("CREATE  TABLE \""+PREGUNTA+"\" (\"id\" INTEGER PRIMARY KEY  AUTOINCREMENT NOT NULL ,\"id_ejercicio\" INTEGER, \"enunciado\"" +
            " VARCHAR NOT NULL , \"respuesta\" VARCHAR NOT NULL,  FOREIGN KEY(id_ejercicio) REFERENCES EJERCICIO (id));");
```

Ilustración 42 - Detalle del método onCreate

Si el fichero de BBDD relacionado con la aplicación ya existía, ***MejoraTuInglesDatabaseHelper*** procede a la realización de una concordancia de versiones con el fin de actualizar la estructura de la BBDD en caso necesario. El código para la actualización se encuentra ubicado en el método ***onUpgrade***.

```
@Override
public void onUpgrade(SQLiteDatabase db, int version, int newVersion) {
    Log.d("MejoraTuIngles","Existe un cambio de versión en BBDD. Preparado para actualizar.");

    // TODO Auto-generated method stub
     db.execSQL("DROP TABLE IF EXISTS "+LECCION);
     db.execSQL("DROP TABLE IF EXISTS "+EJERCICIO);
     db.execSQL("DROP TABLE IF EXISTS "+PREGUNTA);

     Log.d("DEBUG","Base de datos borrada");

     onCreate(db);
}
```

Ilustración 43 - Método onUpgrade de MejorarTuInglesDatabaseHelper

Las operaciones CRUD están encapsuladas en una clase de tipo Adapter, de esta forma pueden ser llamadas dónde sean necesarias esas funcionalidades, permitiendo la reutilización de código, la facilidad de mantenimiento (en caso de modificación ha de actuarse en un único lugar) y el orden dentro del código.

8.1.2 Carga de datos

La carga de datos inicial es uno de los problemas relevantes a los que nos enfrentamos. Se ha diseñado un método flexible que permita la introducción de material (lecciones, ejercicios y preguntas) al menor coste y con el menor impacto en el código posible.

Siguiendo este objetivo se ha diseñado una estructura XML conteniendo el listado de lecciones disponibles con sus elementos asociados.

Al detectar la necesidad de creación o actualización de la BBDD, el sistema localiza el fichero XML con la lista de unidades didácticas, lo procesa y lo inserta.

De esta forma se pueden catalogar fácilmente nuevas lecciones, con sus ejercicios y preguntas o eliminar alguna de las existentes sin realizar grandes modificaciones en el código de la aplicación.

En la siguiente ilustración se puede observarse la estructura que presenta una lección.

```xml
<leccion nombre="Present Simple" url="PresentSimple.html">
    <ejercicio>
        <enunciadoGeneral>
            <![CDATA[
            Complete the sentences with the following verbs in the correct form:
             ]]>
             <datosAdicionales>get - happen - look - lose - make - start - stay - try - work</datosAdicionales>
        </enunciadoGeneral>
        <pregunta>
            <enunciado>You ..... hard today</enunciado>
            <respuesta>are working</respuesta>
        </pregunta>
        <pregunta>
            <enunciado>I ..... for Christine. Do you know where she is?</enunciado>
            <respuesta>looking</respuesta>
        </pregunta>

    </ejercicio>
    <ejercicio>
        <enunciadoGeneral>
            <![CDATA[
            Put de verbs into the correct forms. Sometimes you need the negative. (I'm not doing etc.).
             ]]>
             <datosAdicionales></datosAdicionales>
        </enunciadoGeneral>
        <pregunta>
            <enunciado>Please don't make so much noise ..... (try) to work</enunciado>
            <respuesta>I'm trying</respuesta>
        </pregunta>
         <pregunta>
            <enunciado>Let's go out now. It ..... (rain) any more</enunciado>
            <respuesta>isn't raining</respuesta>
        </pregunta>

    </ejercicio>
</leccion>
```

Ilustración 44 - XML de lecciones

La lectura del XML es realizada por la clase ***XMLReaderHelper***, aislando al resto del sistema de los detalles del tratamiento de la información.

8.2 Especificación de características y permisos

La aplicación, para su correcto funcionamiento, deberá contar con una serie de permisos para acceder a diferentes recursos del terminal. Además, el terminal deberá contar con un determinado nivel de sistema operativo Android ya que, en caso contrario, podrían no encontrarse disponibles funcionalidades del núcleo del sistema operativo utilizadas por la aplicación.

En el fichero AndroidManifest.xml se recogen los permisos y características necesarias para operar con la aplicación.

```
MejoraTuInglesDatabaseHelper.java   lecciones.xml   es.uoc.mejoratuingles.activities.MejoraTuIngles Manifest

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="es.uoc.mejoratuingles.activities"
    android:versionCode="1"
    android:versionName="0.14"
    android:installLocation="preferExternal">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:allowBackup="true">

        <activity
            android:name=".MejoraTuIngles"
            android:label="@string/title_activity_mejora_tu_ingles"
            android:screenOrientation="portrait">

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListadoLeccion"></activity>
        <activity android:name=".LeccionView" android:screenOrientation="landscape"></activity>
        <activity android:name=".PreguntaCompletarView" android:screenOrientation="landscape"></activity>
        <activity android:name=".AyudaView"></activity>
        <activity android:name=".ProgresoView"></activity>
        <activity android:name=".CorreccionView"></activity>
    </application>

</manifest>
```

Bloque de permisos

Versión mínima necesaria de Android

Actividades

Ilustración 45 - Detalle del fichero AndroidManifest

8.1.1 Ubicación del Software

Mediante el elemento ***android:installLocation*** se habilita la posibilidad de instalación en la memoria externa del dispositivo. El usuario podrá decidir en qué lugar ubicar el software, flexibilizando así el uso de la memoria interna del dispositivo.

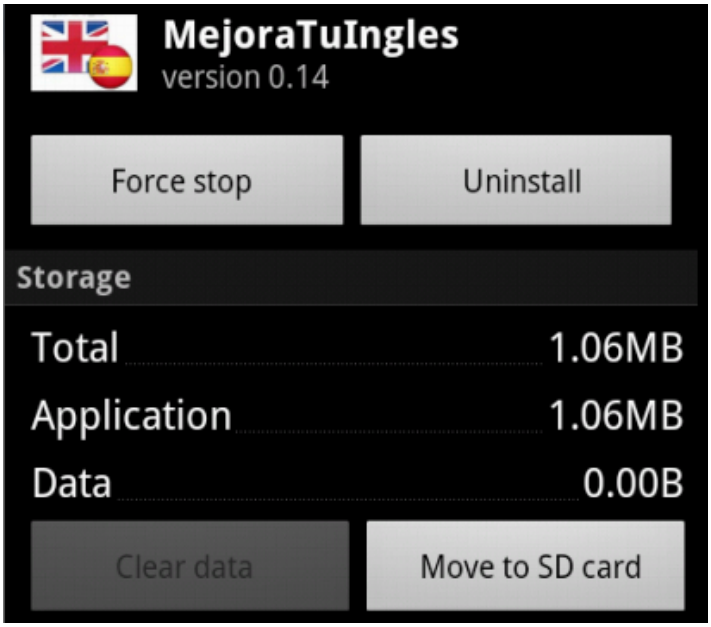


Ilustración 46 - Administrador de aplicaciones de Android

8.1.2 Permisos

Será necesario solicitar permisos para utilizar los siguientes elementos del terminal:

- Conexión a Internet (android.permission.INTERNET)
- Permiso de escritura en la tarjeta de memoria externa (***android.permission.WRITE_EXTERNAL_STORAGE***)

Estos permisos vienen señalados como bloque de permisos en la ilustración 41.

8.1.3 Versión de Android necesaria

El nivel mínimo de Android necesario para el funcionamiento de la aplicación será la versión 8 de SDK. Puede observarse la definición de este atributo en la ilustración 34, bajo el título de versión mínima necesaria.

9. Plan de pruebas funcionales

El plan de pruebas es un documento clave para asegurarnos que se cumplen los requerimientos tomados al cliente.

El plan de pruebas está compuesto por una serie de test funcionales en los que se puede verificar que, ante una acción determinada, el sistema responde de la forma esperada. En el caso que nos ocupa, el plan de pruebas estará compuesto por los casos de uso más críticos desde el punto de vista de usuario.

El plan de pruebas de este TFC se compone de una prueba de verificación de las funcionalidades descritas por los diagramas de solución técnica del capítulo 7, dado que son las funcionalidades claves del proyecto, así como del caso de uso de "Ayuda" (no esquematizado en el mencionado capítulo por considerarse de resolución trivial).

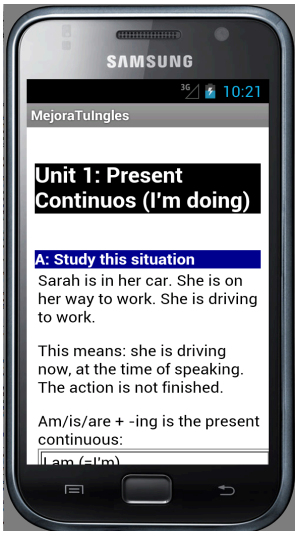
Cada prueba funcional está estructurada de forma que se muestra el punto de vista del usuario, la acción que debe realizar y la respuesta que espera obtener del sistema.

9.1 Emulador

Dada la dificultad de disponer de un terminal real para la ejecución de los test que componen el plan de pruebas, se recomienda la utilización de terminales virtuales. Los Android Virtual Devices (AVDs) son emuladores de terminal Android destinados a la ejecución de aplicaciones sobre una computadora, sin necesidad de disponer de terminal físico. Los AVDs pueden configurarse para emular terminales reales existentes en el mercado ya que cuentan con los llamados Skins, con frecuencia distribuidos por los propios fabricantes, y que permiten adoptar el aspecto de diferentes modelos de dispositivos.

En el Anexo III – Configuración del entorno de desarrollo puede obtenerse información detallada acerca de la configuración de AVDs.

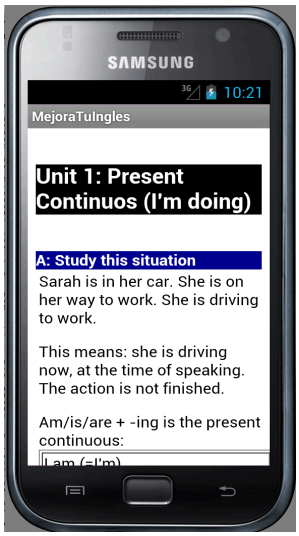
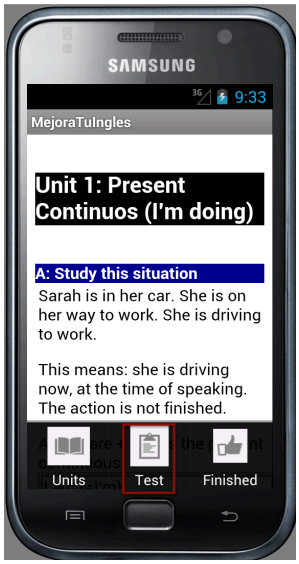
9.2 Estudio de lección

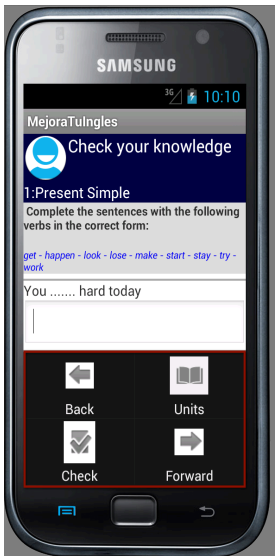
Usuario	Respuesta del sistema
1. El usuario accede a la aplicación desde la pantalla inicial. Pulsa el botón de menú del teléfono y selecciona la opción. 	1. El sistema responde mostrando el listado de lecciones registradas en la aplicación.
2. El usuario selecciona una de las lecciones. 	2. El sistema responde mostrando el contenido de la lección. 

9.3 Realización de test

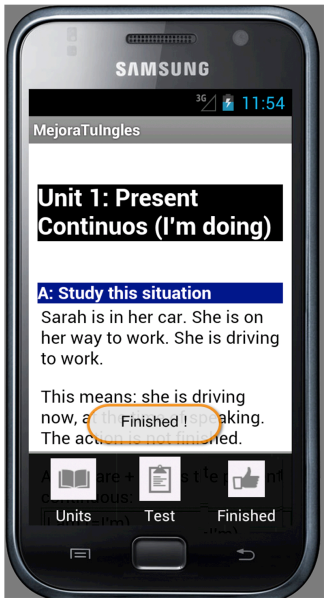
9.3.1 Desarrollo del test

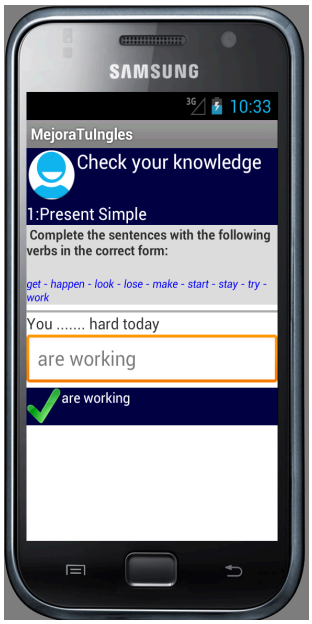
Usuario	Respuesta del sistema
1.El usuario accede a la aplicación desde la pantalla inicial. Pulsa el botón de menú del teléfono y selecciona la opción.	1. El sistema responde mostrando el listado de lecciones registradas en la aplicación.

2. El usuario selecciona una de las lecciones. 	2. El sistema responde mostrando el contenido de la lección. 
3. El usuario pulsa el botón de menú del terminal y selecciona la opción "Test" de las que aparecen en el menú contextual.  	3. El sistema responde mostrando el primer ejercicio de la unidad didáctica y la primera pregunta.

4. El usuario contesta la pregunta utilizando el teclado virtual. Pulsa la tecla “back” del terminal y puede utilizar la navegación para completar todas las preguntas.  	

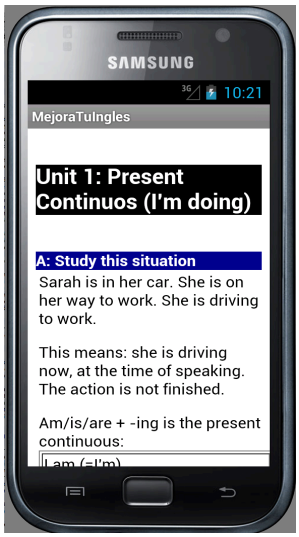
9.3.2 Finalización de test

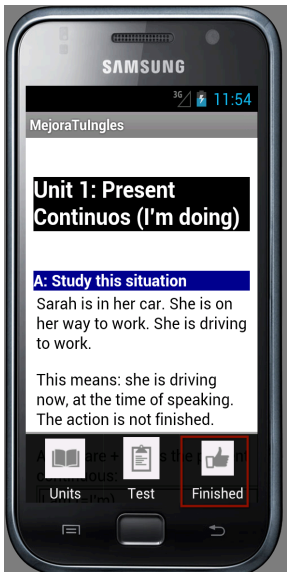
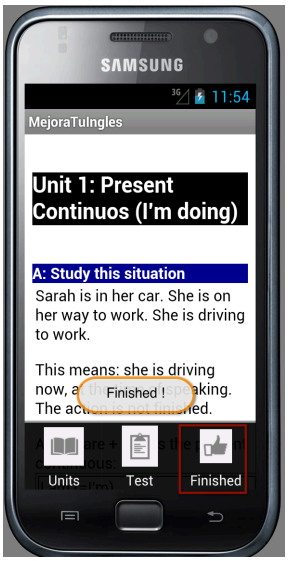
1. El sistema muestra el mensaje de finalización de test.  	1. El usuario pulsa la tecla de corrección del test.

2. El sistema muestra el primer ejercicio, con la primera pregunta. En la vista puede observarse la respuesta correcta y la contestación del usuario.  	2. El usuario puede utilizar los botones de navegación para desplazarse entre las respuestas, viendo fallos y aciertos.

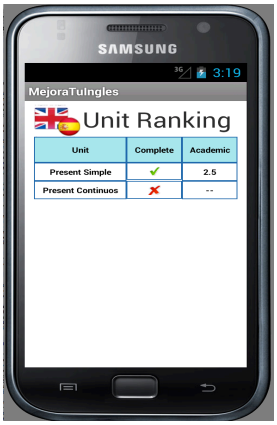
9.4 Registrar progreso

Usuario	Respuesta del sistema
1. El usuario accede a la aplicación desde la pantalla inicial. Pulsa el botón de menú del teléfono y selecciona la opción.	1. El sistema responde mostrando el listado de lecciones registradas en la aplicación.

2. El usuario selecciona una de las lecciones. 	2. El sistema responde mostrando el contenido de la lección. 

3. El usuario selecciona la opción “Finished” del menú contextual. 	3. El sistema anota la finalización de la lección y muestra un mensaje de finalización. 

9.5 Consultar progreso

Usuario	Respuesta del sistema
1. El usuario accede a la aplicación desde la pantalla inicial. Pulsa el botón de menú del teléfono y selecciona la opción “Progress”. 	1. El sistema muestra un cuadro referente al progreso en las diferentes unidades didácticas. 

9.6 Ayuda

Usuario	Respuesta del sistema
1. El usuario accede a la aplicación desde la pantalla inicial. Pulsa el botón de menú del teléfono y selecciona la opción “Help”.	1. El sistema muestra una página de ayuda, en la que se explican las funcionalidades de la aplicación.

10. Conclusiones y líneas futuras

10.1 Evolución del proyecto

La finalidad de este TFC era tomar contacto con aplicaciones e-learning y conseguir implementar una aplicación de aprendizaje para dispositivos móviles enmarcada en este tipo de sistemas.

El resultado del proyecto sería un prototipo de curso de inglés con funcionalidades básicas de estudio y realización de exámenes.

Estos objetivos básicos han sido alcanzados durante la realización de este TFC.

10.2 Dificultades encontradas

Las dos principales dificultades encontradas han sido:

- el tiempo disponible para la implementación
- la compatibilización de este TFC con otras asignaturas

A pesar de partir con algunos conocimientos previos sobre Android y experiencia en programación Java, el escaso tiempo del que se dispone para la implementación hace que el proyecto no pueda ser demasiado ambicioso en lo que a contenidos y funcionalides se refiere, aunque resulta una excelente experiencia de aprendizaje para abordar desarrollos más complejos.

10.3 Líneas futuras

Durante el diseño del sistema se ha prestado especial atención a la posibilidad de reutilización. La idea es utilizar los conocimientos y experiencia adquirida para abordar un producto comercial completo. A continuación se lanzan algunas ideas de desarrollo futuro.

10.3.1 Descarga dinámica de nuevo contenido

Se ha desarrollado un mecanismo de introducción de contenido aislado y modular, que podría ser fácilmente sustituido por llamadas a servicios Web, de esta forma se podría ampliar el contenido dinámicamente desde un servidor externo, sin necesidad de modificaciones adicionales en el código.

10.3.2 Contenido multimedia

En una siguiente fase podrían explotarse las capacidades multimedia de los smarthphones para poder añadir lecciones de dictado y ejercicios de pronunciación.

11. Anexo I - Terminología aplicable

ADT

Complemento que permite la interconexión entre ANDROID SDK y Eclipse.

ANDROID

Es un sistema operativo para dispositivos móviles promovido por Google. Está orientado a su utilización en teléfonos de última generación así como en tabletas.

ANDROID SDK

Es el kit de desarrollo para ANDROID. Provee el conjunto de librerías necesarias para desarrollar, así como un emulador que permite probar los desarrollos.

ANDROID VIRTUAL DEVICE (AVD)

Los AVDs son emuladores que se utilizan para probar las aplicaciones ANDROID durante su fase de desarrollo.

CRUD

Son operaciones básicas de base de datos: alta, baja, modificación y consulta.

Eclipse

Es un entorno de desarrollo integrado (IDE) de código abierto y multilenguaje. Contiene un núcleo básico al que pueden agregarse *plug-ins* con el fin de extender sus capacidades. Su organización interna se basa en la tenencia de un espacio de trabajo (*Workspace*) al que pueden añadirse proyectos de diferentes tipologías (*Web*, J2EE, Java, ANDROID...), estos proyectos pueden interactuar unos con otros mediante las configuraciones adecuadas.

eXtreme Programming (XP)

XP es un enfoque de ingeniería de software, englobado en el marco de los procesos ágiles, que pone el énfasis en la adaptabilidad, en lugar de en la previsibilidad.

En este TFC es imposible adoptar completamente todas las recomendaciones, sin embargo nos hemos guiado por estas:

- Desarrollo iterativo e incremental

- Refactorización de código
- Simplicidad en el código

Plug-in

Aplicado al ámbito del software, se define como un módulo o complemento que puede ser insertado en otro sistema mayor con el fin de ampliar su funcionalidad.

SCRUM

SCRUM es una metodología de desarrollo de proyectos que, en ocasiones, puede verse combinada con otro enfoque de desarrollo más ágil.

Características de esta metodología:

- el software se consigue por iteraciones.
- las iteraciones deben ser cortas y de duración fija.
- debe generarse un producto software al final de cada iteración.

El procedimiento general consiste en la creación de una "pila de historias", cada historia es una funcionalidad de la aplicación. Al principio de cada iteración decidiremos qué historias se implementarán y qué historias anteriores se deberán corregir o completar.

De esta forma, tras sucesivas pasadas, se acaba finalizando el producto software.

SPRINT

Cada una de las iteraciones realizadas por la metodología SCRUM.

SQLite

Es un Sistema Gestor de Bases de Datos (SGBD) relacionales especialmente diseñado para funcionar en sistemas con un número de recursos muy limitado. Esto la hace particularmente interesante para ejecutarse en terminales móviles.

12. Anexo II – Tecnologías y Metodologías de Desarrollo

12.1 Tecnologías de Desarrollo

En este apartado introduciremos las principales tecnologías sobre las que se crearán los módulos definidos en la arquitectura.

12.1.1 ANDROID

ANDROID en un Sistema Operativo, además de una plataforma de Software, basada en el núcleo de Linux. Diseñada en un principio para dispositivos móviles, ANDROID permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java.

ANDROID es una plataforma de código abierto por lo que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de ANDROID).

Inicialmente, ANDROID fue desarrollada por Google Inc. aunque poco después se unió Open Handset Alliance, un consorcio de 48 compañías de Hardware, Software y telecomunicaciones, las cuales llegaron a un acuerdo para promocionar los estándares de códigos abiertos para dispositivos móviles.

Google sin embargo, ha sido quien ha publicado la mayoría del código fuente deANDROID bajo la licencia de Software Apache, una licencia de software libre y de código abierto a cualquier desarrollador.

ANDROID SDK, es el kit de desarrollo estándar de ANDROID, necesario para crear aplicaciones que funcionen sobre esta plataforma. En este kit podemos encontrar todas las herramientas que los programadores necesitan: entornos de desarrollo y depuración, librerías, un emulador de móviles, documentación o tutoriales.

El ANDROID SDK está soportado para sistemas Windows XP o Vista, Mac OS X 10.4.8 o superior y en Ubuntu 6.06 o superior; también es necesario Java 1.5. Está preparado para la integración con el entorno de desarrollo Eclipse mediante el *plug-in* ADT.

La versión r10 de ANDROID SDK, utilizada en este PFC, puede ser descargada desde la web oficial de desarrolladores de ANDROID.

Todas las instrucciones necesarias para la instalación se encuentran en la misma web, que constituye la referencia de cabecera para el desarrollador.

12.1.2 Eclipse

Eclipse es un entorno de desarrollo integrado, de código abierto y multiplataforma.

La plataforma Eclipse se utiliza frecuentemente para crear entornos de desarrollo integrados (IDE). Eclipse emplea *plug-ins* para proporcionar toda su funcionalidad. Cualquier desarrollador puede crear nuevos *plug-ins* y ponerlos a la disposición de la comunidad, incrementando las capacidades de Eclipse.

La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".
El entorno Eclipse organiza sus desarrollos en **proyectos**.
Podríamos definir un proyecto Eclipse como un conjunto de software relacionado, y que puede enmarcarse en una tipología concreta (proyectos Web, proyectos J2EE, proyecto Java). Adicionalmente, Eclipse introduce el concepto de ***workspace*** o espacio de trabajo, que permite ubicar en un marco común proyectos de distintas tipologías.

Un proyecto puede contener diferentes tipos de recursos. Las perspectivas se utilizan a modo de filtro para visualizar únicamente la información que nos interesa obtener en cada momento y ocultar otra que sea irrelevante. Existen diferentes perspectivas predefinidas que vienen de serie con el producto, pero también pueden crearse perspectivas nuevas. Un ejemplo de perspectiva puede ser la perspectiva CVS, que muestra al usuario los depósitos CVS configurados en su entorno de desarrollo local.

12.1.3 SQLite

SQLite es un SGBD de dominio público. Está constituido por una pequeña librería de aproximadamente 500kb, programado en el lenguaje C. Uno de las primeras diferencias entre los motores de Bases de datos convencionales es su arquitecturacliente/servidor, pues SQLite es independiente, simplemente se realizan llamadas a sub rutinas o funciones de las propias librerías de SQLite, lo cual reduce ampliamente la latencia en cuanto al acceso a las bases de datos. Con lo cual

podemos decir que las bases de datos compuestas por la definición de las tablas, índices y los propios datos son guardadas por un solo fichero estándar y en un solo ordenador.

SQLite está especialmente indicado para sistemas con pocos recursos: prototipos, sistemas embebidos o empotrados, elementos domóticos, etc.

12.2 Metodología de Desarrollo

SCRUM propone un desarrollo incremental, basado en sucesivas iteraciones, mejorando el producto en cada una de ellas. Estas iteraciones son conocidas como SPRINTS en la metodología SCRUM. Los SPRINTS tienen que ser de duración corta y fija (a ser posible). Tras cada SPRINT se obtiene un producto final que se evalúa y sobre el que sacan conclusiones para ver en qué estado se encuentra el proyecto (si los avances son los esperados, si la funcionalidad es la requerida por el cliente, etc.).

12.2.1 Formación de un grupo de trabajo SCRUM

Típicamente, un equipo que quiera realizar SCRUM debería estar conformado por:

- Jefe de producto
- Desarrollador/es
- Jefe de Proyecto

Es necesario comprender estos roles para comprender el proceso que va a explicarse a continuación.

El jefe de producto es el encargado de reunirse con el cliente y captar las ideas y funcionalidades que quiere para la aplicación, su responsabilidad es que el proyecto salga adelante en coste y fechas previstos.

Los desarrolladores son las personas que van a traducir todo lo solicitado por el cliente a código informático, con un análisis mínimo previo. Igualmente, van a estimar tiempos de realización de dichas funcionalidades.

El jefe de proyecto, actúa como moderador entre los otros dos colectivos. Vigila que el jefe de producto no exija cosas inalcanzables y que los desarrolladores no den estimaciones de tiempo infinitas y no se pierdan en divagaciones técnicas interminables. Es un árbitro que posee conocimiento técnico y que, en caso de no lograr consenso entre el jefe de producto y desarrolladores, tiene que marcar la línea a seguir (y el resto del equipo se compromete a respetarla).

En la ilustración 41 se observa un proceso típico en SCRUM.

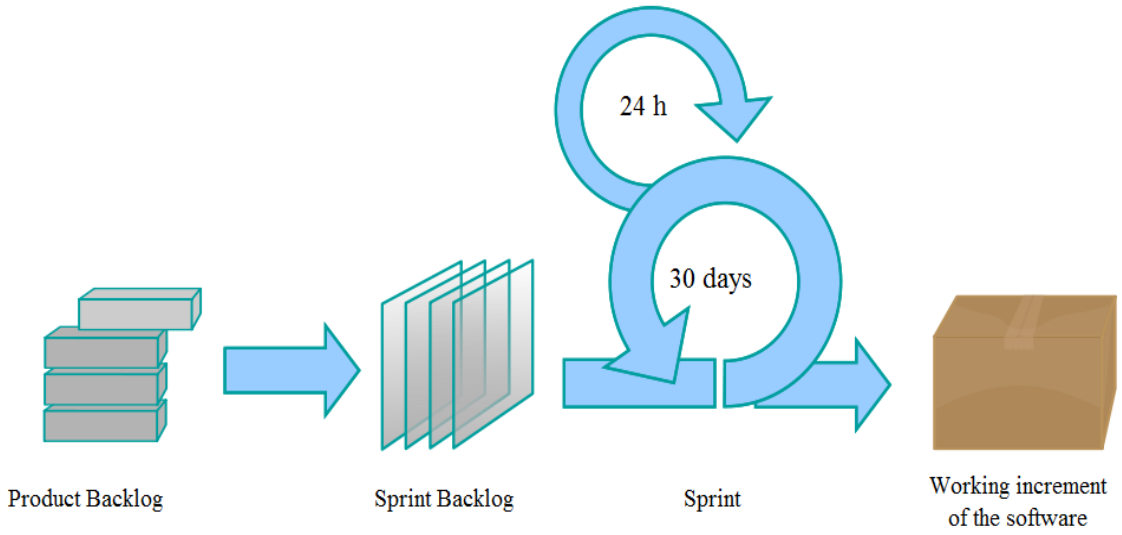


Ilustración 47 - Proceso SCRUM

SCRUM funciona recogiendo los requisitos del producto en una pila (Backlog en la ilustración). Los requisitos son capturados en forma de funcionalidad completa, y cada una de estas funcionalidades recibe el nombre de "Historia de Usuario" o, simplemente, “Historia”.

Una historia puede ser cualquier funcionalidad que el usuario ha pensado que puede hacer con la aplicación.

Previamente a la realización de un SPRINT, se consensúa el número de historias que se incluirán en él. Debe existir un acuerdo entre todos los colectivos del equipo SCRUM, y en caso de no haberlo, será el jefe de proyecto quien decida. El número de historias seleccionadas serán las historias que se implementarán o corregirán en ese SPRINT, este número de historias está representado como SPRINT BACKLOG en la ilustración.

A partir de cada historia, podemos obtener una lista de tareas que habrá que realizar para implementarla. Las tareas son unidades más concretas y se centran en aspectos muy pequeños y determinados de cada historia.

En el caso de este proyecto de fin de carrera, se ha utilizado la filosofía SCRUM para:

- Capturar los requisitos de usuario
- Definir SPRINTS
- Trabajar en el producto de forma incremental, añadiendo pequeñas mejoras tras cada SPRINT

12.3 Ciclo de vida en aplicaciones Android

Las aplicaciones para plataforma Android giran en torno al concepto de Actividad. Una actividad es, según la documentación oficial de Google, "un componente de aplicación que provee una pantalla desde la cual el usuario puede interactuar con el aplicativo con el fin de realizar alguna opción determinada".

Ateniéndonos a esta definición, una aplicación no es más que un conjunto de actividades relacionadas entre sí que cooperan para ofrecer al usuario una serie de funcionalidades, también relacionadas.

El ciclo de vida de una actividad es el siguiente:

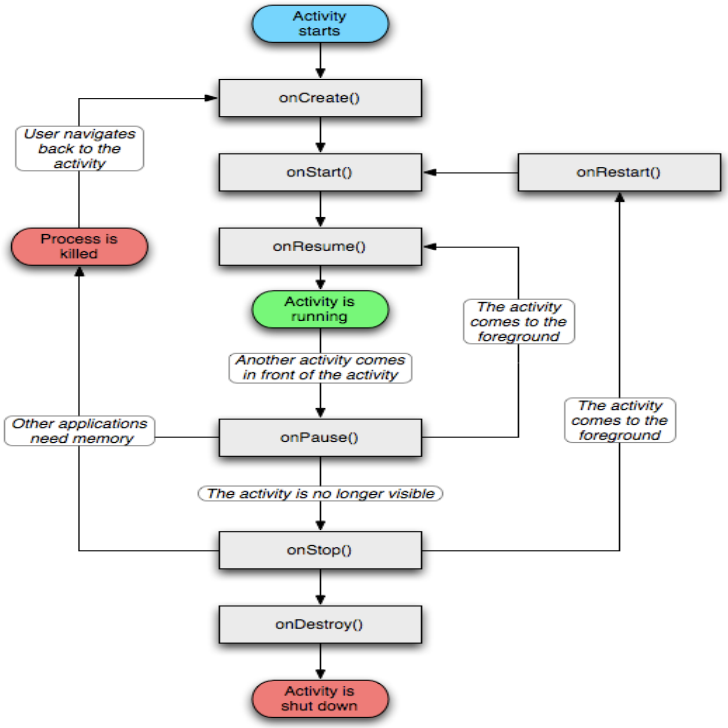


Ilustración 48 - Ciclo de vida de una Actividad Android

El núcleo del Sistema Android es el encargado de llamar a algunos de estos métodos cuando se producen eventos determinados. El desarrollador también puede llamarlos para manipular el estado de una acción en un momento dado.

12.4 Introducción a las interfaces visuales en Android

En Android, las interfaces de usuario se construyen utilizando vistas, grupos de vistas y *Layouts*. Existen muchos tipos de vistas, todas ellas tendrán como clase padre a *View.class*.

Los objetos *View* son las unidades básicas del interfaz de usuario de Android , ofrecen un marco en el que pueden "pegarse" los diferentes elementos de ordenación (*Layouts*) o visuales (*Widgets*). Adicionalmente, son los encargados de realizar una primera detección sobre las acciones de usuario (pulsaciones, desplazamientos…).

Cada *Activity* (pantalla funcional) Android debe tener asociada al menos una *View*. Cuando una actividad Android recibe el foco, el sistema analiza la *View*, como si de un árbol se tratara, y dibuja todos los elementos pegados en la *View*, así como los subelementos que estos últimos puedan contener.

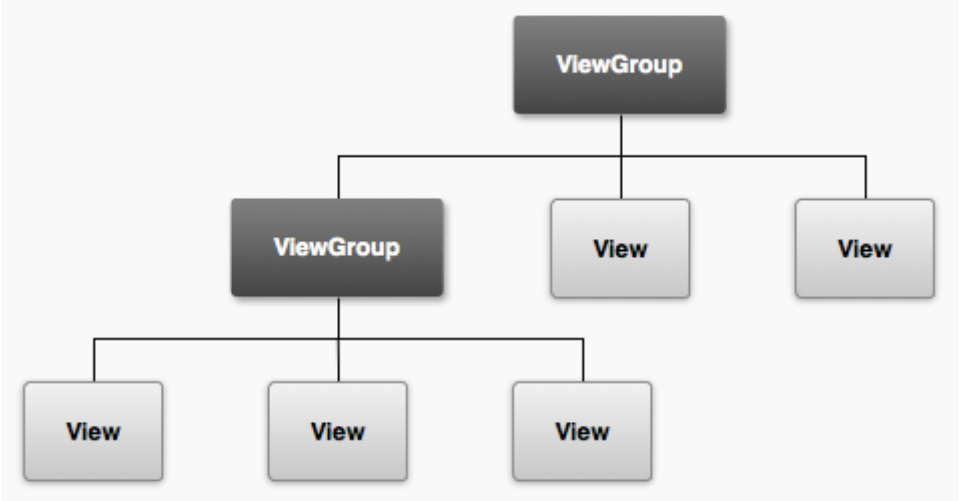


Ilustración 49 - Esquema jerárquico de View

Los *Layouts* son los elementos de posicionamiento que permiten ubicar y ordenar los diferentes elementos en pantalla. Existe diferentes tipos de *Layouts*, según la funcionalidad y el tipo de ordenación que deseemos utilizar. Los *layouts* pueden combinarse entre sí tantas veces como sea necesario, de forma absoluta o relativa, a otros *layouts*.

13. Anexo III – Configuración del Entorno de Desarrollo

13.1 Eclipse

Eclipse es un entorno de desarrollo integrado, de código abierto y multiplataforma.

La plataforma Eclipse se utiliza frecuentemente para crear entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (también llamados plug-ins) para proporcionar toda su funcionalidad. Cualquier desarrollador puede crear nuevos plug-ins para la realización de funcionalidades específicas.

La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

De esta forma, Eclipse proporciona al programador frameworks muy ricos para el desarrollo de aplicaciones: gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, GEF (Graphic Editing Framework - Framework para la edición gráfica) es un módulo de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto hasta editores de diagramas UML, interfaces gráficas para el usuario, etc.

13.1.1 Los proyectos en Eclipse

El entorno Eclipse organiza sus desarrollos en **proyectos**. Podríamos definir un proyecto Eclipse como un conjunto de software relacionado, y que puede enmarcarse en una tipología concreta (proyectos Web, proyectos J2EE, proyecto Java). Adicionalmente, Eclipse introduce el concepto de **workspace** o espacio de trabajo, que permite ubicar en un marco común proyectos de distintas tipologías.

13.1.2 Perspectivas en Eclipse

Un proyecto puede contener diferentes tipos de recursos. Las perspectivas se utilizan a modo de filtro para visualizar únicamente la información que nos interesa obtener en cada momento y ocultar otra que sea irrelevante. Existen diferentes perspectivas predefinidas que vienen de serie con el producto, pero también pueden crearse perspectivas nuevas. Un ejemplo de perspectiva puede ser la perspectiva CVS, que muestra al usuario los depósitos CVS configurados en su entorno de desarrollo local.

13.1.3 Instalación y configuración

La configuración consiste simplemente en descomprimir el fichero en el directorio dónde se desea instalar.

13.2 Android SDK

Android SDK, es el kit desarrollo estándar de Android, necesario para crear aplicaciones que funcionen sobre esta plataforma.

En el kit de desarrollo Android podemos encontrar todas las herramientas que los programadores necesitan: entornos de desarrollo y depuración, librerías, un emulador de móviles, documentación o tutoriales.

El Android SDK funciona en sistemas Windows XP o Vista, Mac OS X 10.4.8 o superior y en Ubuntu 6.06 o superior, también es necesario Java 1.5. Está preparado para la integración con el entorno de desarrollo Eclipse mediante plugins.

Todas las instrucciones necesarias para la instalación se encuentran en la misma web (http://developer.android.com/index.html), que se constituye en la referencia de cabecera para el desarrollador.

13.2.1 Instalación y configuración

Una vez obtenido el software, el siguiente paso será instalarlo. Para ello, ejecutaremos el instalador descargado que cuenta con un ayudante que nos guiará durante el proceso.

13.3 Integración del Android SDK en Eclipse

El plugin ADT para eclipse permite utilizar las funcionalidades del SDK integradas en el entorno Eclipse. Existen varias formas de instalación de ADT, hemos elegido la siguiente:

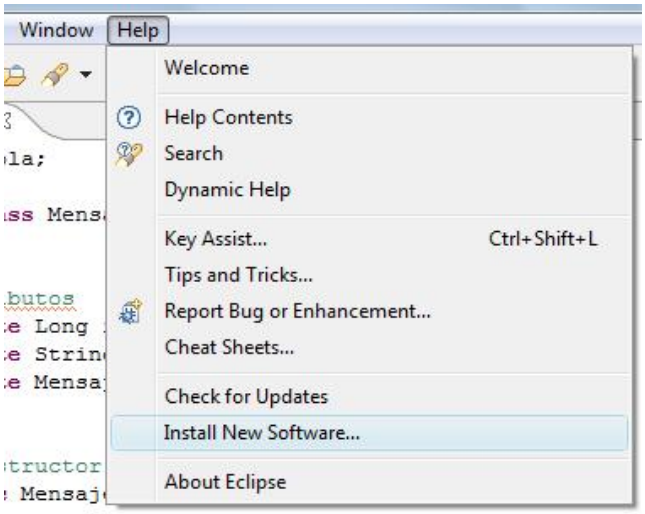


Ilustración 50 - Instalar nuevo software

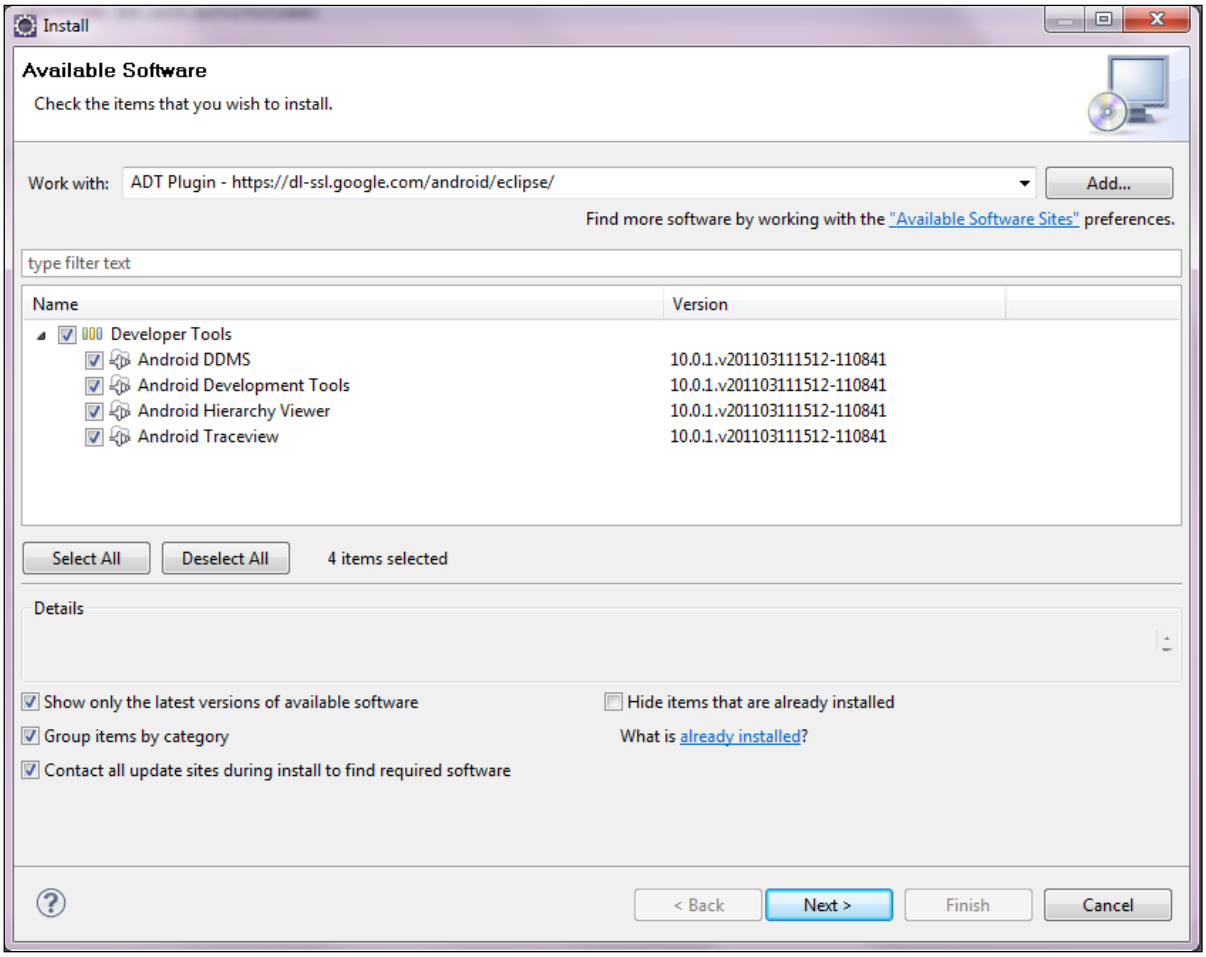


Ilustración 51 - Instalación de ADT en Eclipse

Al finalizar la instalación podremos observar la aparición de una nueva tipología de proyecto (***proyecto Android***), así como nuevas perspectivas (***DDMS***) y opciones (***AVD***). En ***Preferences*** de Eclipse contaremos con un nuevo apartado dedicado a Android.

Para finalizar el proceso, debemos informar a Eclipse de la ubicación en la que se encuentra Android SDK. Para ello debemos ir a la opción ***Window→Preferences→Android***, y cumplimentar el campo ***SDK Location***, tal y como se muestra en la Ilustración 46.

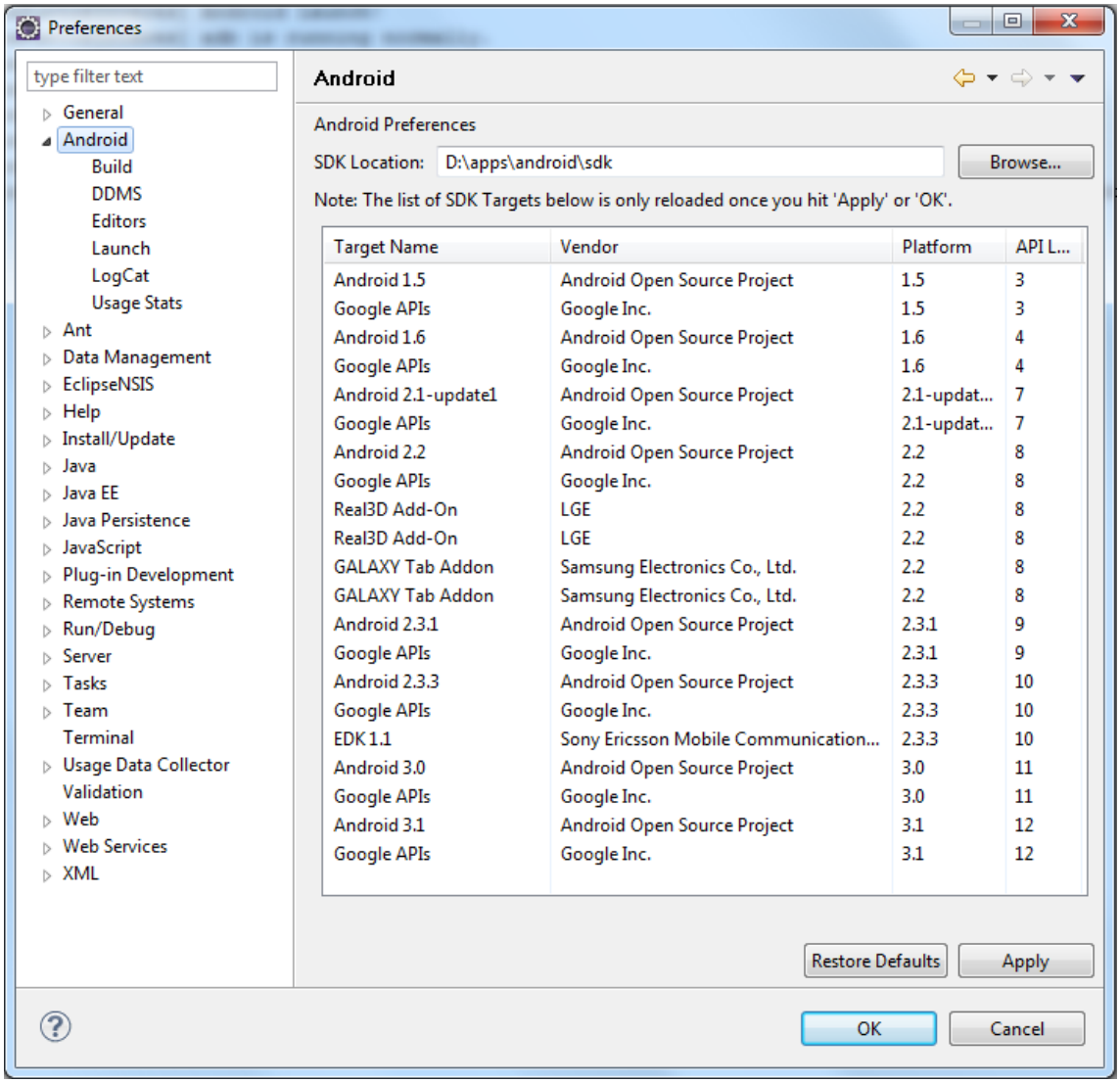


Ilustración 52 - Configuración del SDK en Eclipse

13.3.1 Android VIRTUAL DEVICE

El ***Android Virtual Device*** (en adelante AVD) es un emulador de terminales Android. Se utiliza para pruebas y es distribuido como parte del Android SDK. Podremos acceder a él mediante el nuevo icono aparecido en la barra de herramientas de Eclipse tras la instalación del ADT.

Ilustración 53 - Acceso AVD

Será necesario al menos un AVD para ejecutar las aplicaciones.

La forma de crearlo es la siguiente:

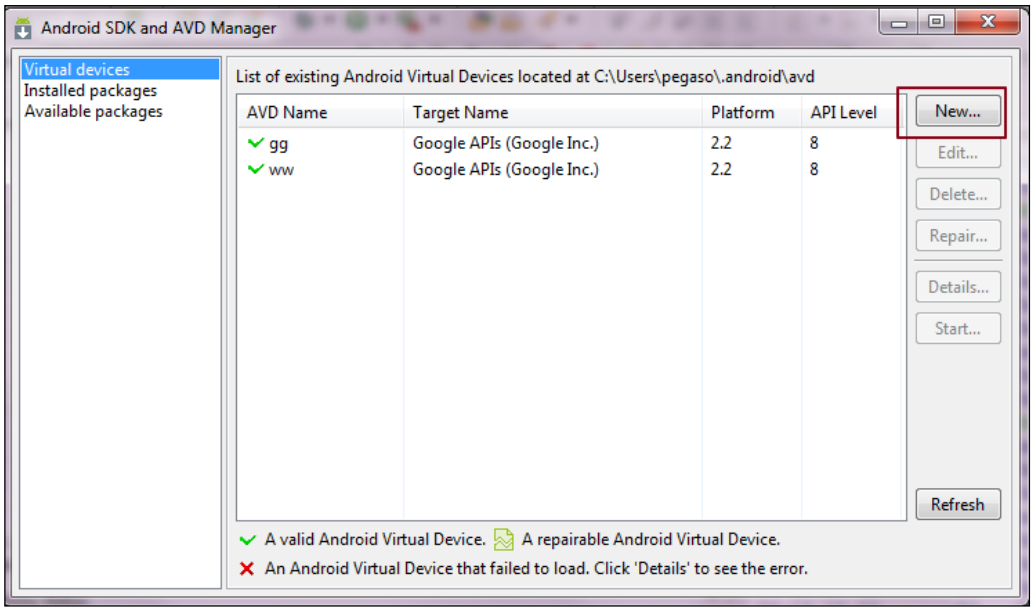


Ilustración 54 - Pantalla principal del AVD Manager

Pulsando "***New***" accedemos a la pantalla de creación del AVD, mostrada en la ilustración 48.

En esta pantalla podremos configurar todas las características del AVD. Estas características van desde el tamaño y definición de la pantalla, al soporte de tarjetas SD, el acelerómetro o el soporte para GPS.

Una característica muy importante del emulador es el **target**. El **target** está constituido por:

- Versión de Android para la que se va a desarrollar
- Nivel de los APIs de Google

En el apartado ***Skin*** podemos configurar aspecto del teléfono y resolución de la pantalla del ***AVD***. Existen diversos ***Skins***, descargables a través de la Web, de teléfonos disponibles en el mercado, de esta forma obtendremos una percepción más

exacta de cómo se visualizará el software en un terminal determinado. Pueden cargarse ***Skins*** descargados desde la opción ***Built-in*** del apartado ***Skin***.

En la opción ***Resolution*** podemos seleccionar diferentes resoluciones. Existen diversos tipos de resoluciones para terminales móviles.

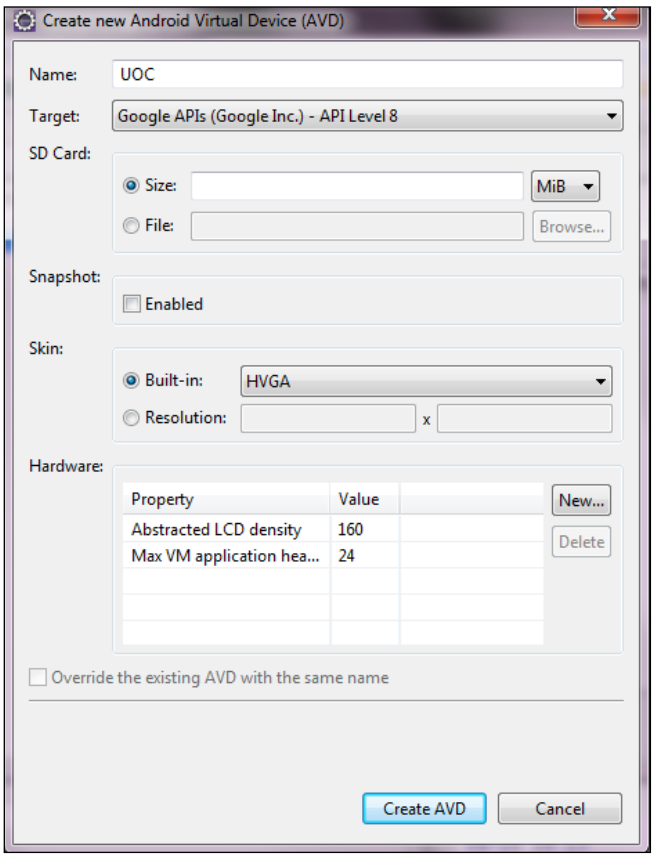


Ilustración 55 - Configuración AVD

13.3.2 Perspectiva DDMS

Desde esta perspectiva de Eclipse podremos manejar los AVDs que hayamos creado. La perspectiva se distribuye en varias pestañas, tal y como se muestra en la ilustración 50.

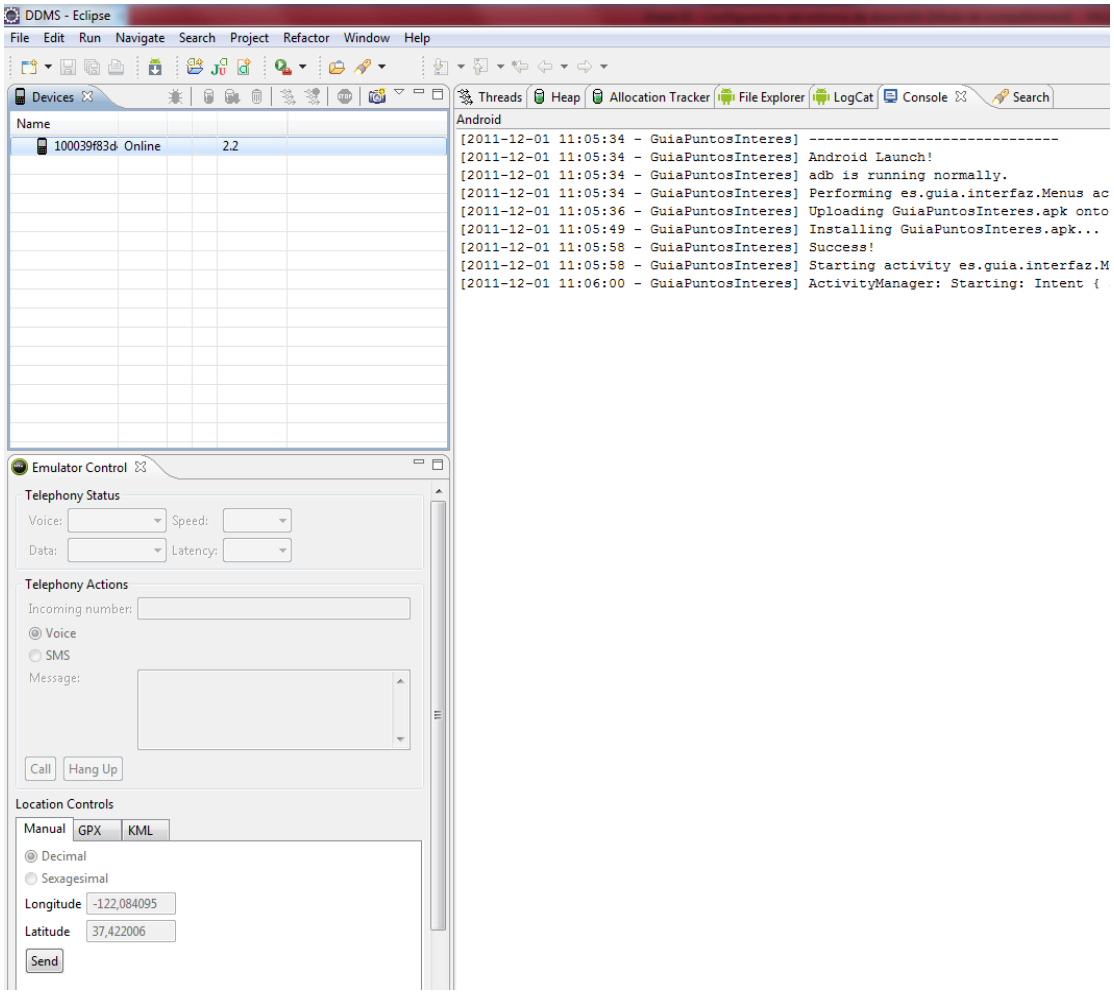


Ilustración 56 - Perspectiva DDMS

En la pestaña ***Devices*** aparecerán todos los AVDs (o terminales reales) arrancados en el sistema. Marcando uno de ellos podremos modificar sus características en tiempo de ejecución.

Los elementos a manejar más importantes son los siguientes:

- Las coordenadas del dispositivo: Podemos manejarlas desde la pestaña ***Emulator*** Control.

- Los ficheros existentes en el dispositivo: Podemos gestionarlos desde la pestaña ***File Explorer***. Esta pestaña nos permite introducir ficheros o descargarlos desde el terminal.

- Consola del sistema: ***LogCat*** es la consola del sistema. Podremos escribir en ella mensajes desde la aplicación. Estos mensajes están clasificados en niveles (debug, info, error…). Esta opción es muy útil para el proceso de depuración.

13.4 Ejecutando la aplicación en AVD o terminal real

La ejecución de la aplicación puede realizarse desde la perspectiva Java de Eclipse, tal y como se muestra a continuación.

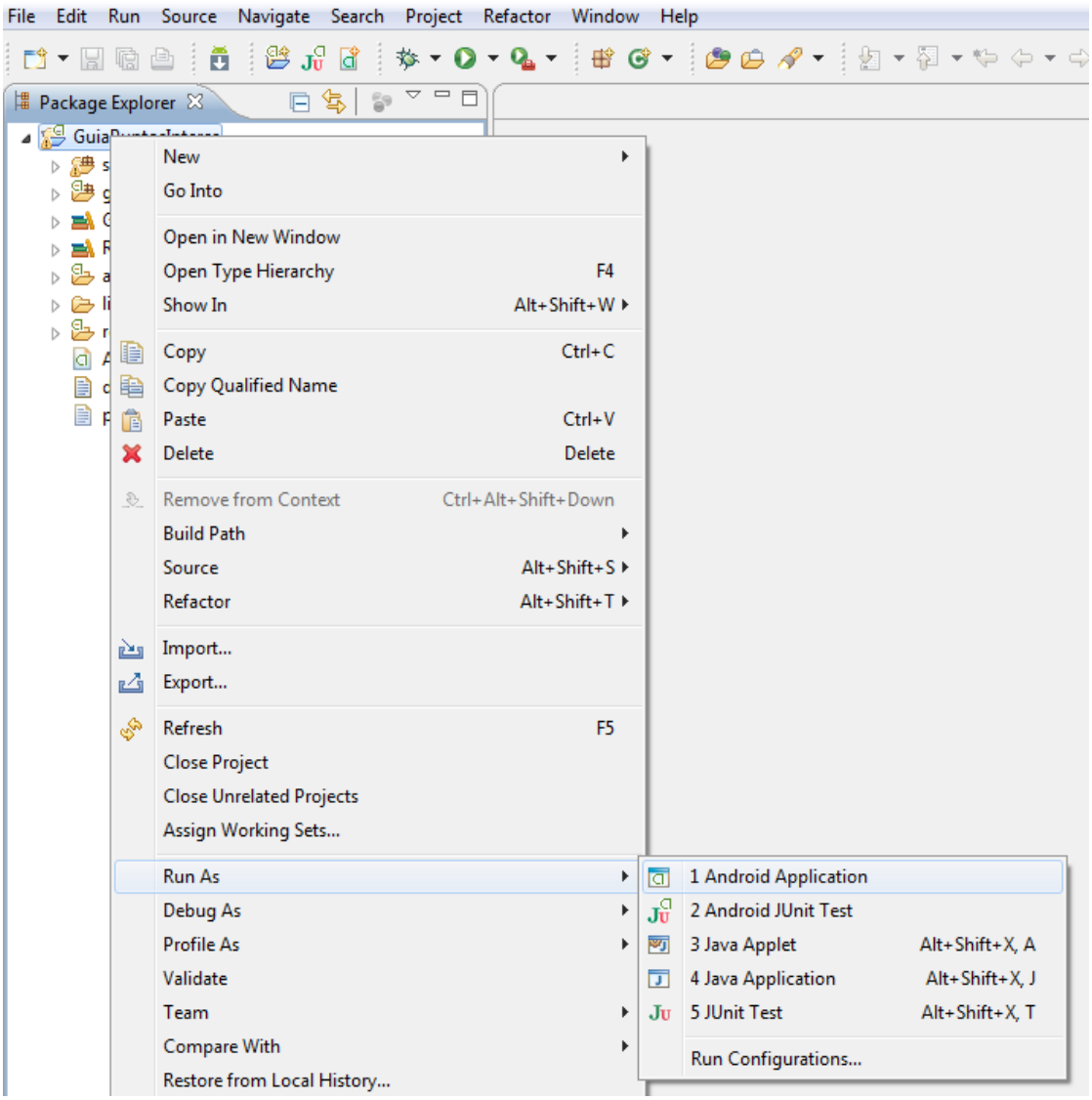


Ilustración 57 - Ejecución de la aplicación

Haciendo click sobre la opción ***Android Application*** aparecerá la pantalla de selección de dispositivo, mostrada en la ilustración 52.

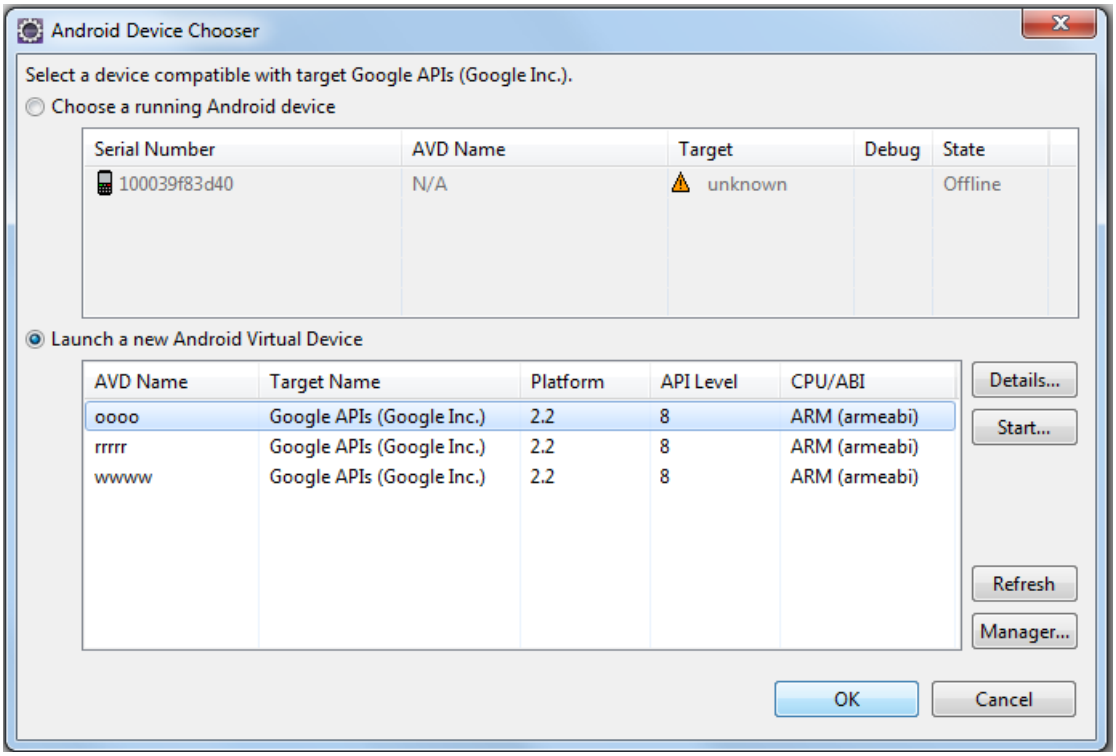


Ilustración 58 - Selección de dispositivo

La pantalla está dividida en dos partes. En la parte superior podremos seleccionar los dispositivos Android en ejecución, detectados por el sistema, y en la parte inferior aparecerán todos los AVDs registrados pero que no se encuentran en ejecución. Seleccionamos el terminal deseado y pulsamos OK.

El proceso de instalación y arranque de la aplicación en el terminal elegido se realizará automáticamente.

Ilustración 59 - Inicialización de AVD

14. Anexo IV – Bibliografía

14.1 Bibliografía

ANDERSON, T. – GARRISON, D.R (2010). El e-learning en el siglo XXI. Barcelona: Octaedro

DIAZ, Alfons. Gestión y Desarrollo de Proyectos. UOC

DIMARZIO, J.F (2008). Android A Programmer's Guide. New York: Mc Graw Hill

KNIBERG, H. (2007). SCRUM y XP desde las trincheras. EEUU: InfoQ

MEIER, R. (2010). Personal Android 2 Application Development. Indianapolis: Wrox

RODRIGUEZ, José Ramón. Metodología y Gestión de Proyectos Informáticos. UOC BATALLER

SÁENZ HIGUERA, Nita. VIDAL OLTRA, Rut. Redacción de Textos Científico-técnicos. UOC

STEELE, J. (2010). The Android Developers CookBook. Boston: Addison-Wesley

http://en.wikipedia.org/wiki/MLearning[22/10/2012]

http://www.andykemp.org.uk/2009/04/comparison-of-vle-solutions/ [24/10/2012]

14.2 Fuentes de imágenes utilizadas

Documento	Ilustración	Fuente
Memoria	Ilustración 41	http://en.wikipedia.org/wiki/Scrum_%28development%29
Memoria	Ilustración 43	http://developer.android.com/images/viewgroup.png