



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SeniorFitness: Aplicación e-health para la valoración de la condición física funcional de personas mayores.

Álvaro González Álvarez

3 de septiembre de 2017



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

SeniorFitness: Aplicación e-health para la valoración de la condición física funcional de personas mayores.

- Departamento: Lenguajes y sistemas informáticos
- Director del proyecto: Raquel Ureña Pérez
- Autor del proyecto: Álvaro González Álvarez

Cádiz, 3 de septiembre de 2017

Fdo: Álvaro González Álvarez

Agradecimientos

Me gustaria agradecer y/o dedicar este texto a ...

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2017 Álvaro González Álvarez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Resumen

Es un hecho que el ejercicio físico puede paliar las limitaciones que va imponiendo el proceso de envejecimiento en las personas, pero este debe ser individualizado a las características de la persona mayor, y es por ello que es necesaria la valoración de la condición física de ésta. La Senior Fitness Test (SFT) es una batería de pruebas para tal valoración, y es una de las pocas que está adaptada a los mayores.

Esta batería evalúa la condición física funcional, entendiendo por este término: *la capacidad física para desarrollar actividades normales de la vida diaria de forma segura, con independencia y sin una excesiva fatiga* (Rikli y Jones, 2001) [6]. Los parámetros de condición física que incluye dicha batería son: fuerza muscular (miembros superiores e inferiores), resistencia aeróbica, flexibilidad (miembros superiores e inferiores) y agilidad.

La aplicación Android que se presenta en este documento tiene como empresa darle al usuario la posibilidad de realizar los ejercicios que componen la SFT para cuantas personas mayores el usuario desee registrar en la aplicación, así como almacenar y llevar un control sobre los resultados obtenidos para cada una de ellas. Las numerosas características y sensores de los que disponen actualmente los dispositivos móviles facilitan en gran medida la monitorización de cada uno de los ejercicios y además reducen los materiales necesarios para obtener y registrar los resultados.

Palabras claves: Android, Aplicación, Senior Fitness Test, Persona Mayor, Ejercicio Físico

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Alcance	2
1.3. Glosario de términos	2
1.4. Organización del Documento	4
2. Descripción general del proyecto	5
2.1. Descripción	5
2.2. Características de la aplicación	5
2.2.1. Componentes de la aplicación	5
2.2.2. Pruebas de valoración de la condición física	6
3. Planificación	11
3.1. Fase inicial	11
3.2. Fase de análisis	11
3.3. Fase de aprendizaje	11
3.4. Fase de desarrollo	12
3.4.1. Metodología de desarrollo	12
3.4.2. Planificación del desarrollo	13
3.5. Pruebas y correcciones	15
3.6. Redacción de la memoria	15
3.7. Diagrama de Gantt	15
4. Análisis de requisitos	19
4.1. Especificación de requisitos del sistema	19
4.1.1. Requisitos de interfaces externas	19
4.1.2. Requisitos funcionales	21
4.1.3. Requisitos no funcionales	35
4.1.4. Reglas de negocio	37
4.1.5. Requisitos de información	37
4.2. Modelo conceptual de datos	38
5. Diseño del sistema	41
5.1. Arquitectura física del sistema	41
5.2. Arquitectura lógica del sistema	42
5.2.1. Android	42
5.2.2. Git	48
5.2.3. SQLite	49
5.3. Patrones de diseño	49

5.3.1. Patrones de diseño creacionales	50
5.3.2. Patrones de diseño estructurales	51
5.4. Diseño físico de datos	51
5.4.1. Base de datos	51
5.5. Diseño de la interfaz de usuario	53
5.5.1. Pantalla principal	55
5.5.2. Alta de persona	56
5.5.3. Detalle de persona	57
5.5.4. Estadísticas de sesión	58
5.5.5. Selección de persona	59
5.5.6. Selección de test	60
5.5.7. Información de test	61
5.5.8. Fuerza de piernas	62
5.5.9. Fuerza de brazos	63
5.5.10. Resistencia Aeróbica	64
5.5.11. Flexibilidad de piernas	65
5.5.12. Flexibilidad de brazos	66
5.5.13. Agilidad	67
5.5.14. Resultados	68
5.6. Uso del dispositivo móvil en la realización de ejercicios	69
5.6.1. Fuerza de piernas (F_Pna)	69
5.6.2. Fuerza de brazos (F_Br)	69
5.6.3. Resistencia aeróbica (Resist)	70
5.7. Diseño de clases UML	70
5.8. Diseño de componentes	72
5.8.1. Diagramas de secuencia	72
6. Implementación del sistema	77
6.1. Solución propuesta	77
6.2. Entorno de construcción	77
6.2.1. Android Studio	78
6.2.2. Android Software Development Kit	79
6.3. Código fuente	79
6.3.1. Activities	79
6.3.2. Listeners	80
6.3.3. Base de Datos	81
6.3.4. Modelo	81
6.3.5. Adapters	81
6.4. Detección de movimiento	82
6.4.1. Sensor de gravedad	82
6.4.2. Lectura para el test Fuerza de brazos	83
6.4.3. Lectura para el test Fuerza de piernas	84
6.4.4. Lectura para el test Resistencia aeróbica	86
7. Pruebas y validaciones	89
7.1. Estrategia	89
7.2. Entorno de pruebas	89
7.3. Pruebas unitarias	90
7.4. Pruebas de integración	90
7.5. Pruebas funcionales	90

7.6. Pruebas no funcionales	91
8. Herramientas utilizadas	93
8.1. Cliente del sistema de control de versiones	93
8.2. Redacción de la memoria	93
8.3. Gestión del proyecto	94
8.4. Realización de diagramas	94
8.5. Ilustración y retoque	95
9. Manual de instalación	97
10. Manual de usuario	99
11. Conclusiones	101
11.1. Objetivos alcanzados	101
Bibliografia y referencias	103
GNU Free Documentation License	105
1. APPLICABILITY AND DEFINITIONS	105
2. VERBATIM COPYING	106
3. COPYING IN QUANTITY	106
4. MODIFICATIONS	107
5. COMBINING DOCUMENTS	108
6. COLLECTIONS OF DOCUMENTS	109
7. AGGREGATION WITH INDEPENDENT WORKS	109
8. TRANSLATION	109
9. TERMINATION	109
10. FUTURE REVISIONS OF THIS LICENSE	110
11. RELICENSING	110
ADDENDUM: How to use this License for your documents	110

Indice de figuras

2.1. Fuerza de piernas	7
2.2. Fuerza de brazos	7
2.3. Resistencia aeróbica	8
2.4. Flexibilidad de piernas	8
2.5. Flexibilidad de brazos	9
2.6. Agilidad	9
3.1. Diagrama de Gantt inicial del proyecto	17
4.1. Diagrama UML: Casos de uso de la aplicación	22
4.2. Diagrama UML: Casos de uso test fuerza de piernas	23
4.3. Diagrama UML: Casos de uso test fuerza de brazos	23
4.4. Diagrama UML: Casos de uso test resistencia aeróbica	24
4.5. Diagrama UML: Casos de uso test agilidad	24
4.6. Diagrama UML: Casos de uso test flexibilidad de piernas	24
4.7. Diagrama UML: Casos de uso test flexibilidad de brazos	25
4.8. Análisis: Modelo conceptual Entidad-Relación	39
5.1. Arquitectura Android	43
5.2. Arquitectura Android: Ciclo de vida de una Activity	45
5.3. Arquitectura Android: Ciclo de vida de un Servicio	46
5.4. Arquitectura Android: Content Provider	47
5.5. Arquitectura Android: Broadcast Receiver	47
5.6. Arquitectura Android: Externalización de recursos	48
5.7. Diagrama Entidad-Relación de la base de datos	52
5.8. Navegación entre pantallas	54
5.9. Pantalla principal	55
5.10. Pantalla de alta de persona	56
5.11. Pantalla de detalle de persona	57
5.12. Pantalla de estadísticas de sesión	58
5.13. Pantalla de selección de persona	59
5.14. Pantalla de selección de test	60
5.15. Pantalla de información de test	61
5.16. Pantalla de test de fuerza de piernas	62
5.17. Pantalla de test de fuerza de brazos	63
5.18. Pantalla de test de resistencia aeróbica	64
5.19. Pantalla de test de flexibilidad de piernas	65
5.20. Pantalla de test de flexibilidad de brazos	66
5.21. Pantalla de test de agilidad	67
5.22. Pantalla de resultados	68

5.23. Colocación del dispositivo móvil para el test Fuerza de piernas	69
5.24. Colocación del dispositivo móvil para el test Fuerza de brazos	70
5.25. Diagrama UML con el diseño de clases de la aplicación	71
5.26. Diagrama de secuencia de registro de persona	72
5.27. Diagrama de secuencia de registro de comienzo de sesión	73
5.28. Diagrama de secuencia de selección de la persona que comenzará la sesión	74
5.29. Diagrama de secuencia de detección de movimiento	75
5.30. Diagrama de secuencia de guardado de resultado	75
 6.1. Interfaz de Android Studio	78
6.2. Efecto de la gravedad sobre el eje Z con el dispositivo colocado de forma paralela sobre el suelo y con la pantalla hacia arriba	82
6.3. Efecto de la gravedad sobre el eje Y con el dispositivo colocado de forma perpendicular (de pie) sobre el suelo	83
6.4. Fuerza de brazos: Efecto de la gravedad sobre el eje Y en la posición inicial	84
6.5. Fuerza de brazos: Efecto de la gravedad sobre el eje Y en la posición final	84
6.6. Fuerza de piernas: Efecto de la gravedad sobre el eje Y en la posición inicial	85
6.7. Fuerza de piernas: Efecto de la gravedad sobre el eje Y en la posición final	85
6.8. Resistencia aeróbica: Efecto de la gravedad sobre el eje Y en la posición inicial	86
6.9. Resistencia aeróbica: Efecto de la gravedad sobre el eje Y en la posición final	86
 8.1. Logo de GitHub	93
8.2. Logo de LATEX	94
8.3. Logo de Gantt Project	94
8.4. Logo de Dia	94
8.5. Logo de Gimp	95

Indice de tablas

3.1. Tiempo estimado y tiempo real empleado en el proyecto	15
4.1. CU01	25
4.2. CU02	26
4.3. CU03	26
4.4. CU04	27
4.5. CU05	27
4.6. CU06	28
4.7. CU07	28
4.8. CU08	29
4.9. CU09	29
4.10. CU10	30
4.11. CU11	30
4.12. CU12	31
4.13. CU13	31
4.14. CU14	32
4.15. CU15	33
4.16. CU16	33
4.17. CU17	34
4.18. CU18	34
4.19. CU19	35
4.20. CU20	35
4.21. RNF01	35
4.22. RNF02	36
4.23. RNF03	36
4.24. RNF04	36
4.25. RNF05	36
4.26. RNF06	37
4.27. RIN01	37
4.28. RIN02	38
4.29. RIN03	38
4.30. RIN04	38

Capítulo 1

Introducción

A continuación, se describe la motivación del presente proyecto y su alcance, y justificaremos la elección de un proyecto como el presente. También se incluye un glosario de términos y la organización del resto de la presente documentación.

1.1. Motivación

Actualmente, si tomamos como referencia la población española, las personas mayores de 65 años representan el 18,4 % del número total de habitantes (INE, 2016) [4], y se espera que este porcentaje continúe aumentando en las próximas décadas. Por tanto, es esperable encontrarnos cada vez con mayor número de personas mayores y a su vez con más limitaciones físicas, que vendrán dadas principalmente por el irreversible proceso de envejecimiento, y muchas de las cuales pueden paliarse con el ejercicio físico. Sin embargo, para conseguir los máximos beneficios del ejercicio para cada persona es de vital relevancia conocer la condición física del mayor para la correcta prescripción de ejercicio, y es en este punto donde entra en juego la Senior Fitness Test (SFT), una batería de pruebas específicamente diseñada para evaluar la condición física funcional de las personas mayores.

Partiendo de esta base y teniendo en cuenta que actualmente es cada vez más frecuente el uso de dispositivos móviles por una amplia gama de usuarios y que el rango de edad de los mismos crece sin detenerse, se determinó que sería interesante desarrollar una aplicación móvil que, con la ayuda de los numerosos sensores de los que disponen los dispositivos móviles de hoy en día, sirviese como herramienta para realizar las diferentes pruebas físicas que componen la SFT, y además diese la posibilidad de registrar y consultar los resultados obtenidos para cada sujeto en las diferentes sesiones realizadas.

De esta forma cualquier profesional dentro del sector de Ciencias de la Salud, de la Actividad Física o del Deporte que esté interesado en llevar un seguimiento de la condición física funcional de personas mayores, puede hacer uso de la aplicación para que ésta contabilice de forma autónoma (sin necesidad de tener que contar a ojo ni con la ayuda de un cronómetro) las repeticiones realizadas y los tiempos obtenidos por los sujetos para cada uno de los diferentes ejercicios de la batería de pruebas, además de quedar registrados en el perfil de cada persona para su estudio.

Debido a todo lo anteriormente mencionado y a que el aprendizaje de las tecnologías empleadas resulta muy útil de cara al mercado laboral por tratarse de tecnologías punteras en la actualidad, puede justificarse un proyecto como el presente como Proyecto de Fin de Carrera.

1.2. Alcance

El alcance del proyecto consiste en la realización de una aplicación Android que, mediante los sensores que contienen los dispositivos móviles actuales tales como el acelerómetro, el giroscopio o el sensor de gravedad, sea capaz de detectar la correcta realización de los ejercicios en los que consiste la SFT.

Para que los sensores del dispositivo móvil sean capaces de realizar las lecturas necesarias para el correcto cálculo del movimiento realizado durante la ejecución del ejercicio, es necesario que el usuario siga las instrucciones dadas por la aplicación para cada uno de los diferentes ejercicios físicos. Por ejemplo, puede ser necesario que el dispositivo móvil se coloque en la extremidad (brazo o pierna) con la que se realizará la actividad.

Durante la realización de las pruebas que hagan uso de los sensores comentados, el dispositivo móvil dará feedback sobre la correcta realización de los ejercicios, emitiendo un sonido con cada repetición realizada de forma satisfactoria y en el mismo instante en el que se contabiliza, sin que sea necesario observar la pantalla del dispositivo para conocer si se ha realizado adecuadamente el ejercicio.

Una vez finalizado cualquiera de los tests, la aplicación da la posibilidad de almacenar el resultado para la persona mayor que esté realizando la batería de pruebas, dándose por concluida la sesión cuando se haya completado el último test que quede pendiente de realizar de la SFT.

El usuario puede consultar en todo momento los resultados obtenidos por cada uno de los sujetos que han ido realizando las pruebas, tanto para la sesión que esté en progreso (con algún test pendiente de finalizar) como para aquellas sesiones que ya se completaron con anterioridad, para las cuales se mostrarán las correspondientes estadísticas.

La aplicación es de código libre bajo la licencia GNU GPL así como las técnicas y herramientas empleadas para su desarrollo. Su uso será en dispositivos móviles Android.

1.3. Glosario de términos

Esta sección contiene una lista ordenada alfabéticamente de los principales términos, acrónimos y abreviaturas específicos del dominio del problema:

- **Android:** Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets y también para relojes inteligentes, televisores y automóviles.
- **API:** La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- **APK:** Un archivo con extensión .apk (Android Application Package, significado en español: Aplicación empaquetada de Android) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android.
- **Aplicación:** Es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

- **Condición física funcional:** Capacidad física para desarrollar actividades normales de la vida diaria de forma segura, con independencia y sin una excesiva fatiga (Rikli y Jones, 2001).
- **e-health / eHealth / eSalud:** Es el término con el que se define al conjunto de Tecnologías de la Información y la Comunicación (TICs) que, a modo de herramientas, se emplean en el entorno sanitario en materia de prevención, diagnóstico, tratamiento, seguimiento, así como en la gestión de la salud, ahorrando costes al sistema sanitario y mejorando la eficacia de este.
- **Framework:** Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Un framework proporciona bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.
- **Git:** Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.
- **GPL:** La Licencia Pública General de GNU es la licencia de derecho de autor más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.
- **IDE:** Un entorno de desarrollo integrado es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.
- **INE:** El Instituto Nacional de Estadística (INE) es un organismo autónomo de España encargado de la coordinación general de los servicios estadísticos de la Administración General del Estado y la vigilancia, control y supervisión de los procedimientos técnicos de los mismos. Entre los trabajos que realiza, destacan las estadísticas sobre la demografía, economía, y sociedad españolas.
- **JAR:** Es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java.
- **Java:** Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.
- **Prueba / Test de valoración de la condición física:** Son las pruebas que se realizan con la finalidad de medir y valorar las diferentes cualidades físicas básicas en sus diferentes facetas. La medición y valoración de estas cualidades nos informa del estado actual del sujeto. Este dato es fundamental para la programación del entrenamiento, ya que nos indica si hay que trabajar de forma específica alguna de ellas, en función de los objetivos a alcanzar, o por el contrario hay que trabajar de forma general.
- **Repositorio:** Es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.
- **Senior Fitness Test (SFT):** Es una batería de pruebas de valoración de la condición física diseñada por Rikli y Jones. Surgió por la necesidad de crear una herramienta que nos permitiese valorar la condición física de los mayores con seguridad así como de forma práctica.
- **SQL:** Es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos.

- **SQLite:** SQLite es un motor de bases de datos muy popular en la actualidad por ofrecer características tan interesantes como su pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y ser de código libre. Android incorpora de serie todas las herramientas necesarias para la creación y gestión de bases de datos SQLite, y entre ellas una completa API para llevar a cabo de manera sencilla todas las tareas necesarias.
- **UML:** El Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un Sistema.
- **XML:** El Lenguaje de Marcas Extensible es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

1.4. Organización del Documento

La estructura del presente documento es la siguiente:

- **Introducción:** Apartado introductorio sobre la motivación para desarrollar este proyecto, su alcance, así como la estructuración de este documento.
- **Descripción general:** Descripción más amplia sobre el proyecto, así como todas las características relevantes que tendrá.
- **Planificación:** En este apartado se detalla la planificación realizada para llevar a cabo el proyecto, así como las distintas etapas en las que está compuesto el mismo.
- **Análisis de requisitos:** En este apartado se detallan los objetivos del sistema, requisitos funcionales, requisitos no funcionales, reglas de negocio, requisitos de información y se representará mediante un diagrama inicial el modelo conceptual de datos del proyecto.
- **Diseño del sistema:** El diseño del sistema viene dado por una arquitectura general del sistema (tanto física como lógica), patrones de diseño, diseño físico y lógico de datos, el diseño de la interfaz de usuario, diseño del modo de empleo del dispositivo móvil para la detección de movimientos y los diagramas de clases y de secuencia.
- **Implementación del sistema:** En este apartado se detallan los aspectos más importantes sobre la implementación del sistema: entorno de construcción y el código fuente.
- **Pruebas y validaciones:** Pruebas realizadas a la aplicación, con el fin de comprobar su correcto funcionamiento y cumplimiento de las expectativas.
- **Herramientas utilizadas:** Herramientas software utilizadas a lo largo del desarrollo del proyecto.
- **Manual de instalación:** Apartado donde se indica cómo realizar la instalación de la aplicación en el sistema.
- **Manual de usuario:** Manual de usuario para el correcto uso de la aplicación.
- **Conclusiones:** Conclusiones obtenidas sobre el proyecto desarrollado, visión futura del mismo y experiencia adquirida.
- **Bibliografía:** Referencias consultadas durante la realización del proyecto.
- **Licencia GNU GFDL:** Texto completo sobre la licencia GNU GFDL en inglés.

Capítulo 2

Descripción general del proyecto

2.1. Descripción

Con la aplicación descrita en el presente documento, el usuario puede registrar en la misma cuantas personas mayores desee. Una vez exista una persona registrada en el sistema, podrá realizar la batería de pruebas de valoración de la condición física en la que consiste la Senior Fitness Test, sirviéndole de ayuda el propio dispositivo móvil donde se encuentre instalada la aplicación.

Para ello, dependiendo de la prueba que se haya seleccionado para realizar entre las disponibles, la aplicación hará uso por ejemplo del sensor de gravedad del dispositivo móvil para determinar si el ejercicio se ha realizado de forma satisfactoria (y por tanto contabilizar la repetición), o bien mostrar en pantalla un cronómetro que se deberá usar si la prueba así lo requiere, entre otros casos.

Una vez realizada cada prueba para la persona mayor en cuestión, el sistema da la posibilidad de almacenar el resultado obtenido por la aplicación, pudiéndose consultar posteriormente en el historial de sesiones realizadas por dicha persona.

2.2. Características de la aplicación

La aplicación ofrece una alternativa libre y gratuita para poder llevar un registro y control de la condición física funcional de personas mayores, pudiéndose consultar los resultados obtenidos y sus estadísticas asociadas en cualquier lugar y momento. Además, es fácilmente adaptable a nuevos tests o pruebas que se deseen ir incluyendo a la aplicación en un futuro.

2.2.1. Componentes de la aplicación

En esta sección se hará una pequeña descripción de los distintos componentes que conforman la aplicación.

Usuario

Usuario de la aplicación, que será quien realice el registro de las personas mayores en el sistema y tenga acceso a las estadísticas y resultados obtenidos para cada una de ellas.

Personas mayores

Son los sujetos que se registran en el sistema y quienes realizan las sesiones de ejercicios correspondientes a los tests que componen la batería Senior Fitness Test.

Tests

Los tests son las pruebas de valoración de la condición física en las que consiste la batería de pruebas denominada Senior Fitness Test. Los resultados obtenidos en cada uno de los tests se podrán almacenar para la persona mayor que haya realizado el ejercicio.

Sesiones

Una sesión es el conjunto de los diferentes tests o pruebas que componen la batería Senior Fitness Test, y se identifican por el día y la hora en el que se comienza con ella. Se considera que una persona ha completado una sesión cuando ya ha realizado cada uno de los tests de la batería de pruebas y por tanto ha obtenido un resultado para cada una de ellas. En caso de que quede algún test por realizar, la sesión se considera en progreso, y por tanto no se marcará como completada hasta obtener el resultado de todas y cada una de las pruebas restantes.

Resultados

Se entiende por resultado el valor obtenido por una persona mayor en un determinado test de una determinada sesión. Dependiendo del test, un resultado puede ser un número de repeticiones, un tiempo en segundos o bien una distancia medida en centímetros.

Estadísticas

Una vez se ha completado una sesión, desde el historial de sesiones realizadas por la persona mayor en cuestión, se pueden consultar las estadísticas calculadas en función de los resultados obtenidos en cada uno de los tests de dicha sesión.

2.2.2. Pruebas de valoración de la condición física

A continuación se listarán las diferentes pruebas o tests de valoración de la condición física que actualmente se han implementado en la aplicación, con una breve descripción para cada una de ellas.

Fuerza de piernas (F_Pna)

Número de veces que es capaz de sentarse y levantarse de una silla durante un tiempo definido (30 segundos por defecto) con los brazos en cruz y colocados sobre el pecho.

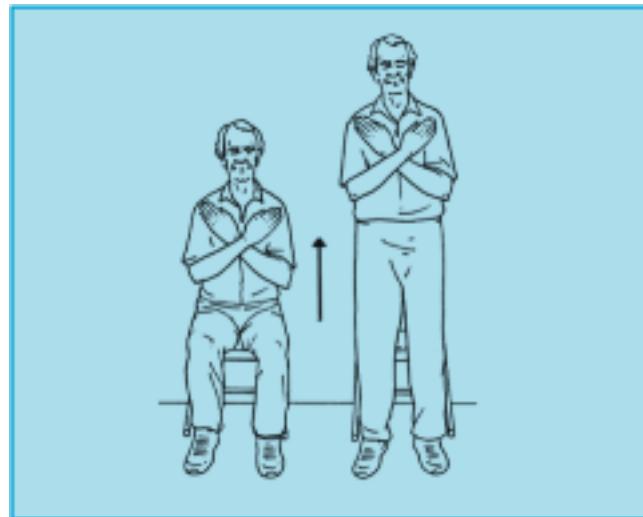


Figura 2.1: Fuerza de piernas

Fuerza de brazos (F_Br)

Número de flexiones de brazo completas, sentado en una silla, que realiza durante un tiempo definido (30 segundos por defecto) sujetando una pesa de 3 libras (2,27 kg) para mujeres y 5 libras (3,63 kg) para hombres.

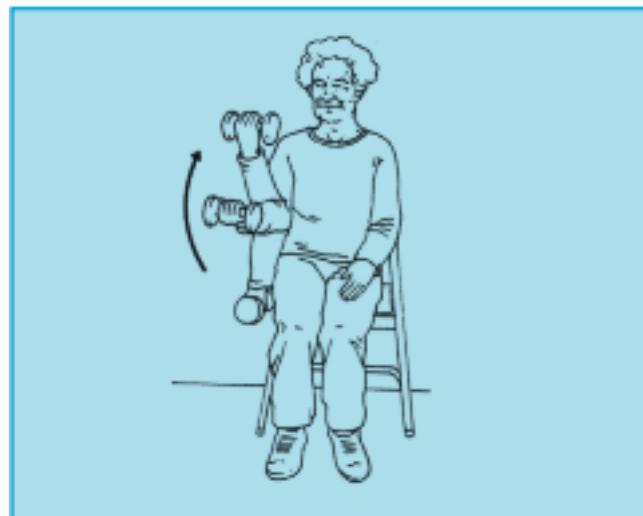


Figura 2.2: Fuerza de brazos

Resistencia aeróbica (Resist)

Número de veces que levanta la rodilla hasta una altura equivalente al punto medio entre la rótula y la cresta ilíaca durante un tiempo definido (2 minutos por defecto). Se contabiliza una vez por cada ciclo (derecha-izquierda).

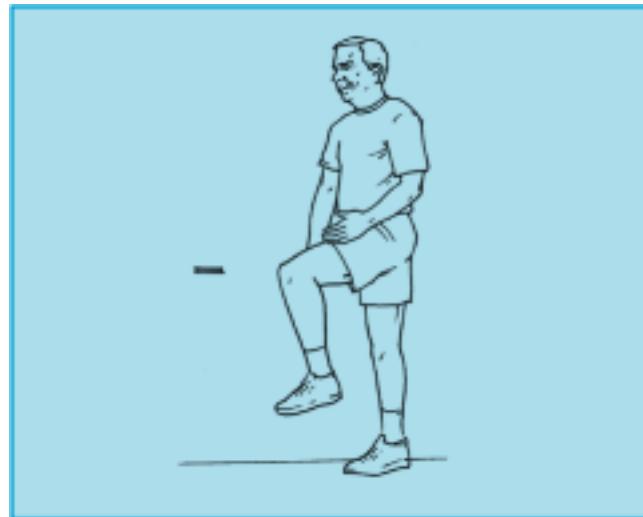


Figura 2.3: Resistencia aeróbica

Flexibilidad de piernas (Flex_Pna)

Sentado en el borde de una silla, estirar la pierna y las manos intentan alcanzar los dedos del pie que está con una flexión de tobillo de 90 grados. Se mide la distancia entre la punta de los dedos de la mano y la punta del pie (positiva si los dedos de la mano sobrepasan los dedos del pie o negativa si los dedos de la mano no alcanzan a tocar los dedos del pie).



Figura 2.4: Flexibilidad de piernas

Flexibilidad de brazos (Flex_Br)

Una mano se pasa por encima del mismo hombro y la otra pasa a tocar la parte media de la espalda intentando que ambas manos se toquen. Se mide la distancia entre la punta de los dedos de cada mano (positiva si los dedos de la mano se superponen o negativa si no llegan a tocarse los dedos de la mano).

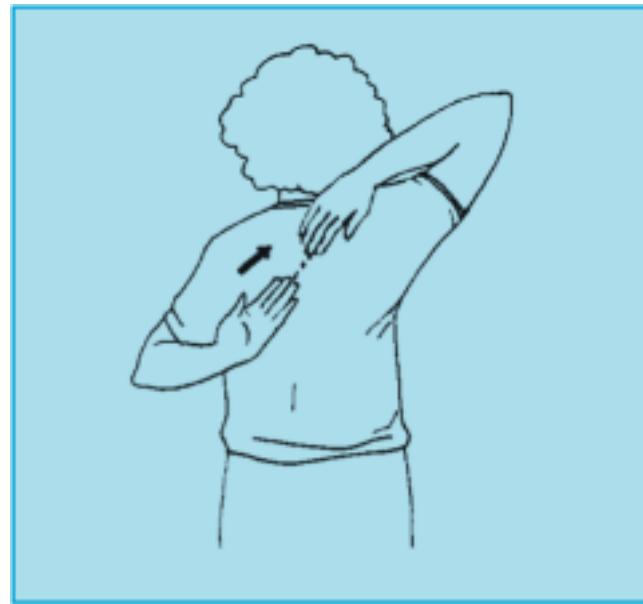


Figura 2.5: Flexibilidad de brazos

Agilidad (Agil)

Partiendo de sentado, tiempo que tarda en levantarse caminar hasta un cono situado a 2,44 metros, girar y volver a sentarse.

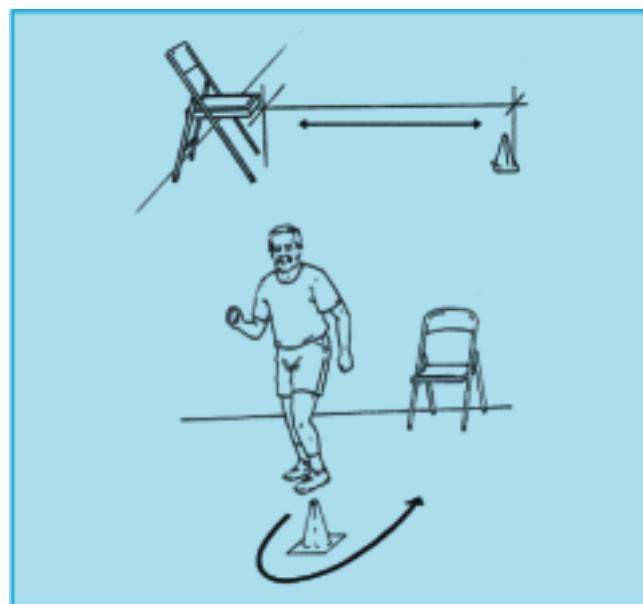


Figura 2.6: Agilidad

Capítulo 3

Planificación

La planificación realizada para el desarrollo del proyecto está dividida en varias partes, tal y como pasamos a describir a continuación.

3.1. Fase inicial

La primera fase consistió en plantear la idea del proyecto con la ayuda de Raquel Ureña. Tras plantear varios enfoques, se decidió realizar este proyecto debido a las motivaciones escritas anteriormente.

También se pensó en qué lenguaje de programación se desarrollaría el proyecto, así como las principales bibliotecas que se usarían durante la realización del mismo, priorizando siempre opciones libres. Debido a que ya tenía suficientes conocimientos de Java, finalmente se optó por utilizar el SDK oficial de Android (Java) por su documentación, comunidad y robustez.

3.2. Fase de análisis

Esta etapa está dividida principalmente en las dos partes siguientes:

- **Especificación de los requisitos:** Estudio de los diferentes requisitos que deberá cumplir la aplicación.
- **Recursos necesarios:** Recursos necesarios que deberemos usar para llevar a cabo el desarrollo del proyecto.

3.3. Fase de aprendizaje

Aunque ya conocía Java en profundidad debido a mi profesión actual, hasta ahora no me había embarcado en el desarrollo de una aplicación Android, por lo que tuve que dedicarle tiempo a aprender algunas reglas de diseño, determinadas sintaxis del lenguaje y API de librerías, así como familiarizarme con el entorno de desarrollo Android Studio, que aunque tiene algunas similitudes respecto a Eclipse (IDE que uso a diario en mi puesto de trabajo), también tiene muchas otras diferencias.

Esta fase se caracterizó por intentar entender código ya escrito así como leer tutoriales y documentación oficial. Se podría dividir en varias etapas principales:

- **Aprendizaje del SDK de Android:** Las aplicaciones Android siguen un diseño específico basado en Activities, así como un sistema de layout propio para la capa de presentación. Dado que no

tenía experiencia previa, me basé en la documentación oficial, la cual me ayudó a que el código evolucionase favorablemente y fuese fácil de mantener.

- **Estudio de los sensores existentes en los dispositivos móviles actuales:** En los dispositivos móviles de hoy en día existen numerosos sensores que se pueden usar para la detección de movimientos. Dado que en ciertos tests la aplicación debe ser capaz de detectar la correcta realización de los ejercicios, fue necesario hacer un estudio sobre los diferentes sensores disponibles (como pueden ser el giroscopio, acelerómetro o el sensor de gravedad) para determinar cual de ellos usar.
- **Aprendizaje del entorno de desarrollo:** Estudio del entorno de desarrollo y de la operativa a seguir para la programación de los diferentes módulos de la aplicación, así como para el testeo, detección y depuración de errores.
- **Aprendizaje de otras bibliotecas:** Para algunas funcionalidades fue necesario el aprendizaje de algunas bibliotecas adicionales, como puede ser por ejemplo SQLite para la persistencia de datos en Android.
- **Aprendizaje del arte:** No estaba muy familiarizado con el retoque de imágenes, pero a partir de tutoriales y pruebas ensayo-error aprendí el manejo básico de los programas que usé para la edición de los iconos e imágenes usadas en la aplicación.

3.4. Fase de desarrollo

Tras la consecución de las etapas anteriores se comenzó el desarrollo del proyecto. Esta etapa del desarrollo es la más extensa de todas y me fue posible llevarla a cabo gracias a los prototipos y pruebas de concepto que iba implementando durante el aprendizaje de los aspectos detallados en el punto anterior.

3.4.1. Metodología de desarrollo

Para el desarrollo del presente proyecto se ha optado por el empleo del modelo de desarrollo iterativo e incremental. Se consideró que esta estrategia de desarrollo sería la idónea dado que, gracias al empleo de las iteraciones o tareas, nos sería de gran utilidad a la hora de hacer frente a la inclusión de nuevos requisitos a medida que fuera avanzando el desarrollo del proyecto.

Por tanto, el empleo de esta metodología implicará que el proceso de desarrollo se divida en iteraciones que abarquen a un segmento del proyecto que ya es funcional. Al ser éste también un modelo incremental, las futuras iteraciones evolucionarán el desarrollo anterior, incluyendo mejoras y respondiendo a los requisitos del sistema que se van cumplimentando y añadiendo a posteriori. Es importante mencionar que cada iteración tratará a un requisito del sistema, y que se tendrán en cuenta primeramente los requisitos básicos que permitan tener una versión funcional del sistema en las primeras iteraciones.

El cliente de este modo va obteniendo en cada iteración una versión más avanzada del proyecto pero siempre funcional, por lo que no tendrá que esperar al final del desarrollo de la aplicación para poder hacer uso del sistema.

El desarrollo iterativo e incremental permite que, al dividir el desarrollo del proyecto en tareas más pequeñas, sea más flexible la respuesta ante cambios y nuevos requisitos que imponga el cliente. Del mismo modo, será más sencillo realizar las pruebas y consideraciones que se hagan tras cada iteración completada, pues si éstas se realizaran una vez se desarrolle todo el proyecto (como ocurre con otras

metodologías como el desarrollo en cascada), la aparición de un error en el diseño implicaría un coste temporal importante comparado con el de detectar un error en una parte o módulo específico del proyecto.

3.4.2. Planificación del desarrollo

Una vez decidida la metodología de desarrollo a emplear en el proyecto, se procede a especificar cada una de las iteraciones que compondrán el desarrollo de éste.

Antes de desarrollar cada iteración, cabe mencionar que existe una etapa previa a la del cumplimiento de las iteraciones en la que se han de especificar los objetivos del sistema, así como las diferentes tecnologías que se usarán y el tiempo que se estima para cada una de las iteraciones asignadas.

Así pues, las iteraciones que se contemplarán en el desarrollo del proyecto son las siguientes:

- **Iteración 1: Implementación de la prueba Fuerza de brazos (F_Br):** En esta iteración se implementa el test *Fuerza de brazos* de la batería de pruebas Senior Fitness Test. Una vez finalizada esta iteración, la aplicación es capaz de contar, con ayuda del sensor de gravedad, las repeticiones que el usuario hace durante la realización del ejercicio y durante un tiempo determinado (en segundos) introducido por el usuario. Además, al ser el primer test que se implementa, el desarrollo se hace de forma que sea posible reutilizar mediante herencia los métodos y funciones encargados de detectar las repeticiones y dar feedback al usuario para el resto de tests que se quieran implementar.
- **Iteración 2: Implementación de la prueba Fuerza de piernas (F_Pna):** En esta iteración se implementa el test *Fuerza de piernas* de la batería de pruebas Senior Fitness Test. La amplitud del desarrollo en esta iteración se reduce, dado que se heredan funcionalidades de la clase padre implementada en la iteración anterior. Una vez finalizada esta iteración, la aplicación ya es capaz de contar las repeticiones que el usuario hace durante la realización de dos tipos de ejercicios diferentes de la SFT.
- **Iteración 3: Implementación de la prueba Resistencia aeróbica (Resist):** En esta iteración se implementa el test *Resistencia aeróbica* de la batería de pruebas Senior Fitness Test. Al igual que en la iteración anterior, es posible heredar de la clase padre implementada en la primera iteración. Una vez finalizada esta iteración, la aplicación ya es capaz de contar las repeticiones que el usuario hace durante la realización de tres tipos de ejercicios diferentes de la SFT.
- **Iteración 4: Implementación de la prueba Agilidad (Agil):** En esta iteración se implementa el test *Agilidad* de la batería de pruebas Senior Fitness Test. En este caso, al ser un tipo de ejercicio diferente en el que es necesario implementar un cronómetro y por tanto no es necesario contar repeticiones, no es posible heredar directamente de la actividad padre, por lo que es necesario hacer modificaciones sobre la misma para que pueda ser reutilizable por ejercicios que requieran un cronómetro, además de hacer la implementación propia del test.
- **Iteración 5: Implementación de la prueba Flexibilidad de piernas (Flex_Pna):** En esta iteración se implementa el test *Flexibilidad de piernas* de la batería de pruebas Senior Fitness Test. En este caso, al ser un tipo de ejercicio diferente en el que es necesario obtener una medida en centímetros y por tanto no es necesario contar repeticiones ni hacer uso de un cronómetro, no es posible heredar directamente de la actividad padre, por lo que es necesario hacer modificaciones sobre la misma para que pueda ser reutilizable por ejercicios de este tipo, además de hacer la implementación propia del test.

- **Iteración 6: Implementación de la prueba Flexibilidad de brazos (Flex_Br):** En esta iteración se implementa el test *Flexibilidad de brazos* de la batería de pruebas Senior Fitness Test. La amplitud del desarrollo se reduce en este caso con respecto a la iteración anterior, dado que en la misma se había adaptado la clase padre heredable para este tipo de ejercicios en los que el resultado a obtener es una medida en centímetros.
- **Iteración 7: Desarrollo del subsistema de almacenamiento de la información:** Llegados a este punto, una vez implementados todos las pruebas de la SFT, se considera necesario empezar con la implementación del sistema encargado de almacenar en una base de datos la información y los resultados obtenidos para cada uno de los tests que se realicen, así como el diseño y el despliegue de la propia estructura. Al finalizar esta iteración solo se almacenan los resultados para el usuario del dispositivo móvil, sin posibilidad de hacer los tests a cualquier persona mayor que se deseé.
- **Iteración 8: Desarrollo del subsistema de registro de usuarios/personas mayores:** En esta tarea se implementa el sistema de registro de personas mayores en la aplicación, añadiendo una pantalla de registro básica para la obtención, validación y almacenamiento de los datos de cada persona en el sistema. También se implementa la funcionalidad para acceder a la cámara del dispositivo por si se quiere tomar una foto de la persona durante el registro, o bien elegir una imagen de la galería.
- **Iteración 9: Desarrollo del subsistema de gestión de personas mayores:** En esta tarea se implementa la gestión de personas mayores registradas en la app. En la pantalla principal de la aplicación se muestra un listado de las personas registradas en la aplicación (con la información básica para identificar cada una de ellas, así como una foto de la persona en caso de habérsela tomado con el dispositivo móvil durante el registro). Si se toca sobre una persona del listado, se accede a la pantalla de detalle de usuario donde se da la posibilidad de eliminar al usuario del sistema. También se mostrarán las sesiones realizadas por el usuario.
- **Iteración 10: Desarrollo del subsistema de gestión de sesiones:** En esta tarea se implementa la gestión de sesiones, esto es, la posibilidad de conocer qué tests ha realizado cada usuario y cuales les queda pendiente por realizar para dar por completada una sesión, mostrando en todo momento los resultados obtenidos en los tests ya realizados.
- **Iteración 11: Desarrollo del subsistema de resultados y cálculos de estadísticas:** En esta iteración se implementa la pantalla de resultados que se muestra una vez que el usuario ha completado una sesión (es decir, ha realizado todos los tests de la SFT). Además, también se hace la implementación del cálculo de estadísticas mostradas cuando se accede a las sesiones realizadas por el usuario desde la pantalla de detalle del mismo.
- **Iteración 12: Desarrollo de posibles mejoras de la aplicación:** En este punto se procederá a solventar los errores y mejoras que pudieran demandar las últimas pruebas aplicadas sobre todo el sistema, así como la inclusión de nuevas funcionalidades o tests que se soliciten.
- **Iteración 13: Documentación del proyecto.**

Para tener una referencia del tiempo estimado para el desarrollo del proyecto y el tiempo real empleado, se muestra a continuación una tabla que representa dichos datos.

Etapa	Tiempo estimado	Tiempo real
Fase previa	30 días	28 días
Iteración 1	15 días	20 días
Iteración 2	5 días	7 días
Iteración 3	5 días	4 días
Iteración 4	10 días	6 días
Iteración 5	10 días	3 días
Iteración 6	5 días	3 días
Iteración 7	15 días	18 días
Iteración 8	15 días	10 días
Iteración 9	10 días	12 días
Iteración 10	15 días	19 días
Iteración 11	15 días	10 días
Iteración 12	10 días	8 días
Iteración 13	20 días	22 días
Total	180 días	170 días

Tabla 3.1: Tiempo estimado y tiempo real empleado en el proyecto

En el último punto de este capítulo se encuentra el diagrama de Gantt inicial diseñado para el proyecto, en el cual se representa de manera gráfica el tiempo estimado que se le dedicará a cada una de las fases del desarrollo del proyecto.

3.5. Pruebas y correcciones

Una de las etapas más importantes del desarrollo de cualquier proyecto. Esta etapa se realizó en paralelo a la de desarrollo ya que, conforme se implementaron nuevas funcionalidades, se iban probando exhaustivamente bajo el contexto de las distintas situaciones que pudieran darse, hasta obtener el comportamiento esperado.

Además, conforme se iban entregando versiones funcionales de la aplicación en cada iteración a Raquel Ureña, ella también ejercía de tester, dándome feedback muy importante para mejorar el proyecto.

3.6. Redacción de la memoria

La redacción de la memoria se ha realizado conforme se iba avanzando en el desarrollo del proyecto pero, tras la finalización de éste, se le ha dedicado más tiempo a su cumplimentación, corrigiendo puntos que han requerido modificaciones o que finalmente no se han adecuado al producto final.

3.7. Diagrama de Gantt

Para la documentación del proyecto y en particular para lo concerniente al tiempo en que se realizarán cada una de las tareas descritas en cada iteración, se hace uso del modelo de diagrama de Gantt. Este modelo representa el tiempo de dedicación prevista para diferentes tareas o actividades a lo largo de un tiempo total determinado.

A continuación se mostrará el diagrama de Gantt inicial con el que se estimó el tiempo que iban a ocupar cada una de las fases e iteraciones del presente proyecto, tal y como se indicó en la sección 3.4.2 del presente capítulo.

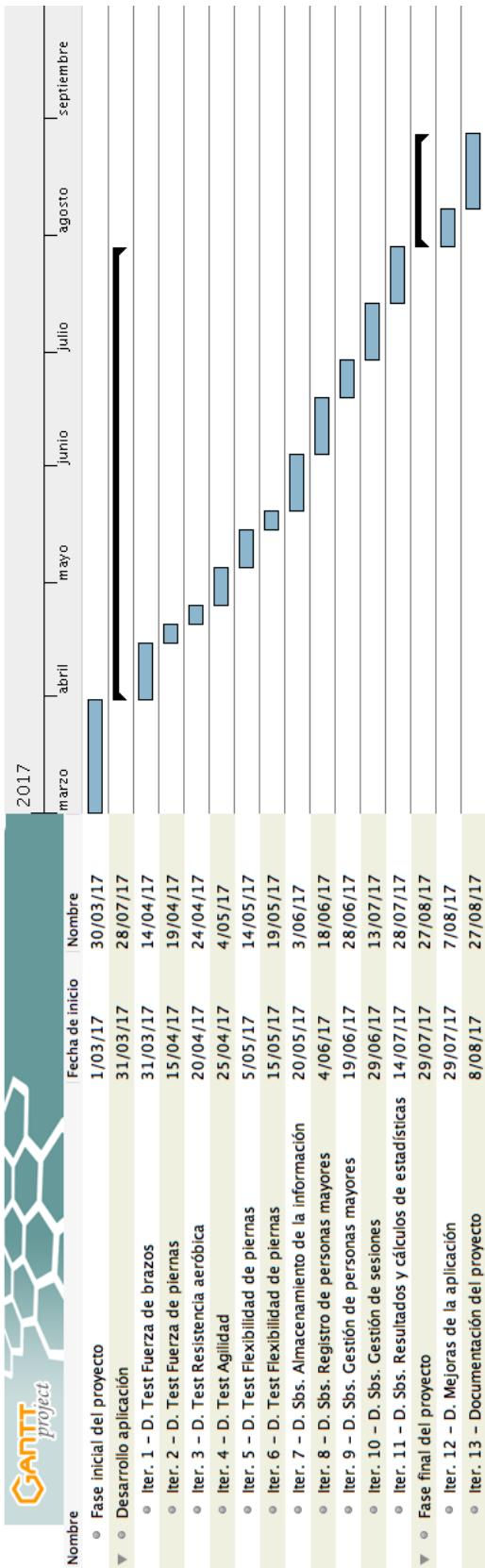


Figura 3.1: Diagrama de Gantt inicial del proyecto

Capítulo 4

Análisis de requisitos

En este capítulo se describen todos los aspectos relacionados con el análisis de requisitos del sistema: catálogo de requisitos, modelo conceptual, así como la solución propuesta.

4.1. Especificación de requisitos del sistema

En esta sección se detallan todos los tipos de requisitos necesarios para satisfacer los objetivos y características descritas en las secciones anteriores del presente documento.

4.1.1. Requisitos de interfaces externas

En este apartado se describirán los requisitos de conexión del software y el hardware, así como la interfaz de usuario.

Sobre la conexión entre el software y el hardware se encarga el SDK de Android, por lo que al ser un sistema preestablecido, no será necesario realizar el diseño ni el análisis; tan solo haremos uso de él.

A continuación, pasamos a definir la interfaz de la aplicación. Todas las ventanas de la aplicación se adaptarán a la resolución nativa del dispositivo en que se ejecute.

El usuario utilizará la pantalla táctil para interactuar con la interfaz. Por defecto, bastará con una pulsación típica para activar el evento correspondiente. En caso contrario, se mencionará explícitamente.

Las diferentes pantallas que modelarán la aplicación serán:

- **Pantalla principal:** Es la pantalla inicial, y tiene como función ofrecer al usuario el listado de personas mayores que ha registrado en la aplicación, así como la posibilidad de acceder a las siguientes pantallas:
 - Pantalla de registro de personas
 - Pantalla de selección de persona para comenzar o continuar con una sesión
 - Pantalla de detalle de persona
- **Registro de personas:** Permite al usuario registrar a personas en el sistema. La información introducida será validada y almacenada en la base de datos. Además dará la posibilidad de tomar una foto con la cámara o bien seleccionar una imagen de la galería para asignarla al usuario. Una

vez seleccionada la imagen, debe mostrar al usuario una pantalla de edición que permita recortar la imagen con la proporción de aspecto usada por la aplicación.

- **Selección de persona para comenzar o continuar con una sesión:** Permite al usuario seleccionar una persona de entre todas las registradas en la aplicación. Una vez seleccionada, llevará a la pantalla de selección de test.
- **Detalle de persona:** Muestra al usuario los datos de la persona que se está consultando, así como un listado de las sesiones realizadas por el mismo, encabezado por aquella sesión que esté en progreso en caso de existir. Además da la posibilidad al usuario de eliminar a la persona, así como de acceder a la pantalla de estadísticas de cualquier sesión ya realizada o a la pantalla de selección de test para aquella sesión que esté en progreso.
- **Selección de test:** Muestra al usuario el listado de tests que la persona seleccionada debe realizar. Se indicarán aquellos tests que ya se hayan completado y el resultado obtenido, diferenciándolos de aquellos que estén pendientes de hacer. Además, al lado de cada test aparecerá un botón de información, que de ser pulsado mostrará una pantalla de información sobre el test a realizar.
- **Información de test:** Muestra al usuario una breve explicación que indica en qué consiste el ejercicio que se debe realizar para completar el test, así como instrucciones para su correcta realización con el dispositivo móvil en caso de ser necesario. A esta pantalla se accederá desde la pantalla de selección de test.
- **Estadísticas de sesión:** Muestra al usuario los resultados obtenidos en los diferentes tests de una sesión ya realizada por una persona, así como estadísticas sobre la misma. A esta pantalla se accederá desde la pantalla de detalle de persona.
- **Resultados:** Esta pantalla es la que se muestra al usuario inmediatamente después de que la persona que esté realizando una sesión haya terminado el último test de la misma que tuviese pendiente. En la pantalla se indicarán los resultados obtenidos en la sesión para cada uno de los tests realizados.
- **Fuerza de piernas:** En esta pantalla se lleva a cabo la realización del test Fuerza de piernas (F_Pna). Permite al usuario especificar el tiempo (en segundos) que durará la prueba y unos botones que servirán para iniciar/parar/resetear la prueba. Durante la realización del ejercicio irá mostrando en pantalla el número de repeticiones realizadas, así como una cuenta atrás con el tiempo restante. Una vez haya finalizado el tiempo, mostrará un botón que permitirá almacenar el resultado obtenido para la persona que ha realizado el ejercicio. A esta pantalla se accederá desde la pantalla de selección de test.
- **Fuerza de brazos:** En esta pantalla se lleva a cabo la realización del test Fuerza de brazos (F_Br). Permite al usuario especificar el tiempo (en segundos) que durará la prueba y unos botones que servirán para iniciar/parar/resetear la prueba. Durante la realización del ejercicio irá mostrando en pantalla el número de repeticiones realizadas, así como una cuenta atrás con el tiempo restante. Una vez haya finalizado el tiempo, mostrará un botón que permitirá almacenar el resultado obtenido para la persona que ha realizado el ejercicio. A esta pantalla se accederá desde la pantalla de selección de test.
- **Resistencia aeróbica:** En esta pantalla se lleva a cabo la realización del test Resistencia aeróbica (Resist). Permite al usuario especificar el tiempo (en segundos) que durará la prueba y unos botones que servirán para iniciar/parar/resetear la prueba. Durante la realización del ejercicio irá mostrando en pantalla el número de repeticiones realizadas, así como una cuenta atrás con el tiempo restante. Una vez haya finalizado el tiempo, mostrará un botón que permitirá almacenar el

resultado obtenido para la persona que ha realizado el ejercicio. A esta pantalla se accederá desde la pantalla de selección de test.

- **Flexibilidad de piernas:** En esta pantalla se debe introducir la distancia en centímetros obtenida tras la realización del test Flexibilidad de piernas (Flex_Pna), y debe contener un botón que tras pulsarse almacenará en el sistema el valor para la persona que esté realizando la sesión. A esta pantalla se accederá desde la pantalla de selección de test.
- **Flexibilidad de brazos:** En esta pantalla se debe introducir la distancia en centímetros obtenida tras la realización del test Flexibilidad de brazos (Flex_Br), y debe contener un botón que tras pulsarse almacenará en el sistema el valor para la persona que esté realizando la sesión. A esta pantalla se accederá desde la pantalla de selección de test.
- **Agilidad:** Esta pantalla mostrará al usuario un cronómetro con los controles correspondientes para iniciarla, pausarla y resetearla. Además mostrará un botón que tras pulsarse almacenará en el sistema el tiempo obtenido (en segundos) para la persona que esté realizando el ejercicio. A esta pantalla se accederá desde la pantalla de selección de test.

4.1.2. Requisitos funcionales

Los requisitos funcionales se han agrupado en diferentes subsistemas, cada uno de éstos contienen los requisitos establecidos que permitirán cumplir los objetivos y características descritas en las secciones anteriores y que se detallarán más adelante.

Para describir los distintos comportamientos que tendrá el sistema, usaremos el lenguaje de modelado de sistemas UML, que representa los requisitos funcionales del sistema, centrado en qué hace y no cómo lo hace.

4.1.2.1. Diagramas de casos de uso

Para no sobrecargar el diagrama de casos de uso, se ha optado por dividirlos de la siguiente forma:

- Un diagrama de casos de uso que mostrará de forma global las funcionalidades de la aplicación.
- Un diagrama de casos de uso para cada uno de los diferentes tests que componen la Senior Fitness Test.

A continuación se muestran los diagramas de casos de uso comentados:

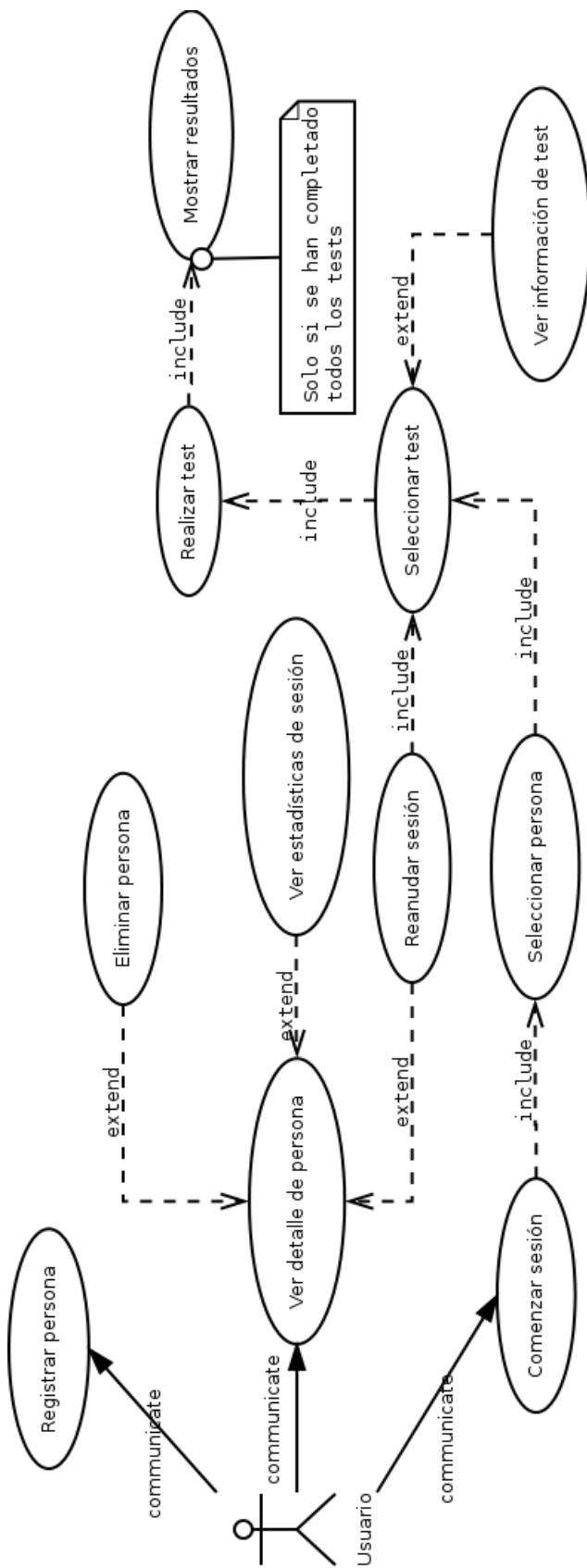


Figura 4.1: Diagrama UML: Casos de uso de la aplicación

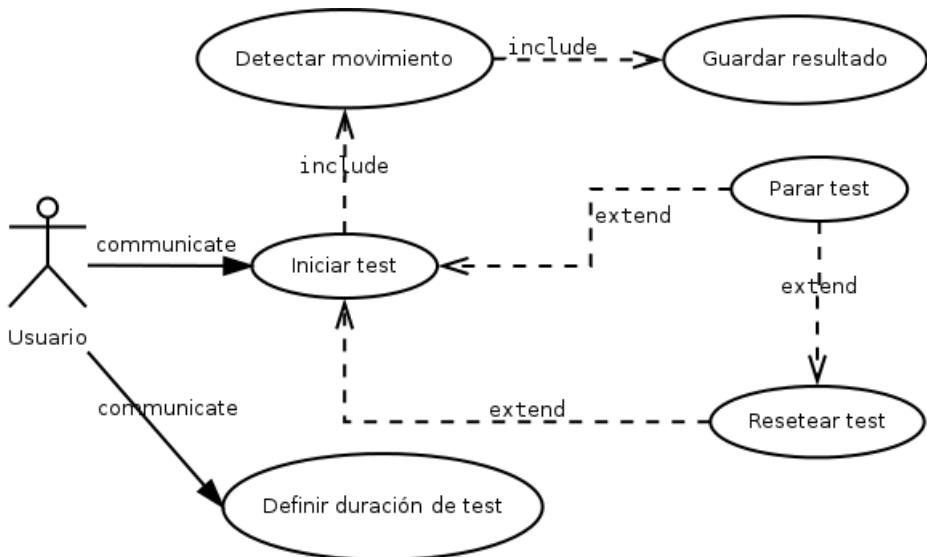


Figura 4.2: Diagrama UML: Casos de uso test fuerza de piernas

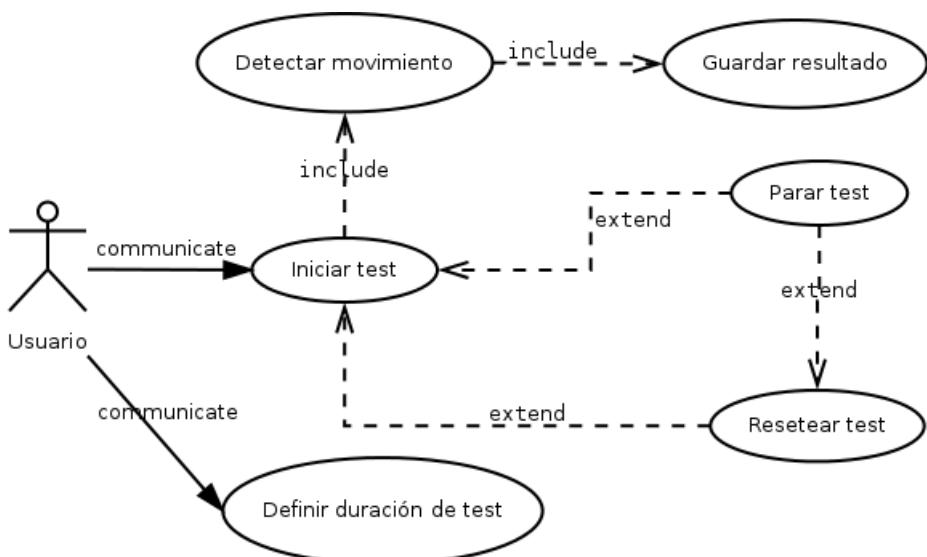


Figura 4.3: Diagrama UML: Casos de uso test fuerza de brazos

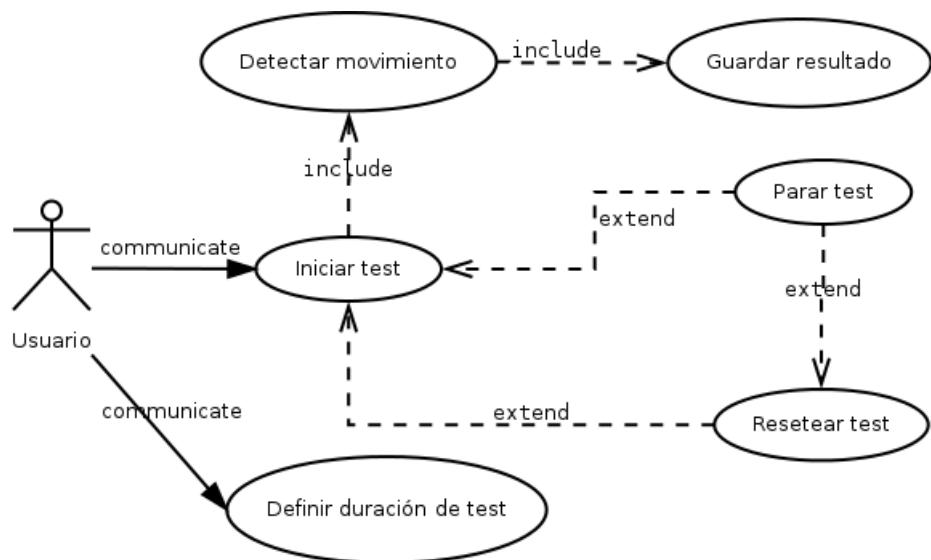


Figura 4.4: Diagrama UML: Casos de uso test resistencia aeróbica

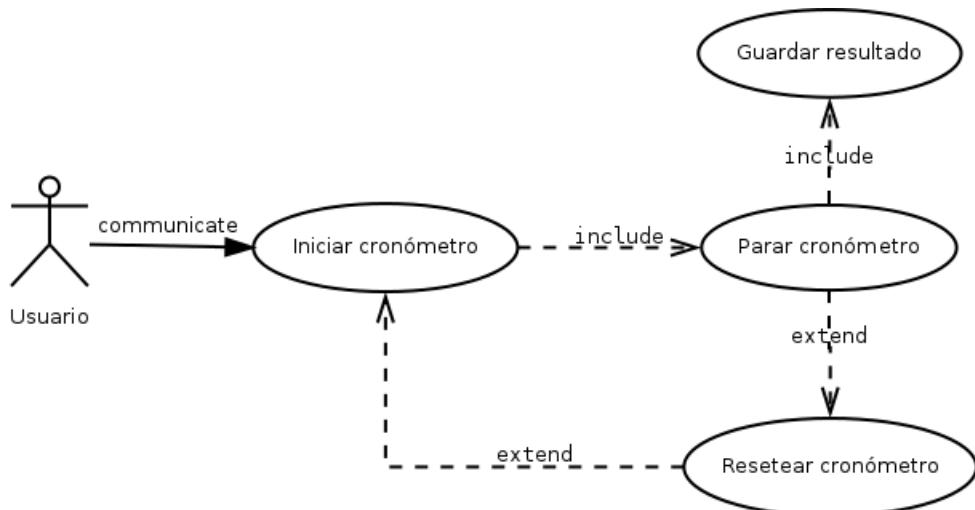


Figura 4.5: Diagrama UML: Casos de uso test agilidad



Figura 4.6: Diagrama UML: Casos de uso test flexibilidad de piernas



Figura 4.7: Diagrama UML: Casos de uso test flexibilidad de brazos

4.1.2.2. Descripción de casos de uso

A continuación se detallarán los casos de uso empleados en cada uno de los subsistemas.

4.1.2.2.1. Subsistema de registro de personas mayores

CU01	Registrar persona.
Descripción	La persona se registra en el sistema indicando sus datos.
Precondición	El usuario debe encontrarse en la pantalla de registro de personas.
Secuencia normal	1. El usuario pulsa sobre el botón de registro de persona desde la pantalla principal. 2. Se muestra al usuario la pantalla de registro de persona. 3. El usuario hace o selecciona una foto de la persona e introduce el DNI, su nombre, apellidos, fecha de nacimiento y género y pulsa sobre Guardar. 4. El sistema registra la persona y redirige a la pantalla principal de la aplicación.
Secuencia alternativa	4a. Se ha dejado uno o varios campos vacíos. Vuelve al paso 3. 4b. Se ha introducido un DNI de persona existente. Vuelve al paso 3.
Postcondición	Se registra a la persona en el sistema exitosamente y redirige a la pantalla principal de la aplicación.
Importancia	Vital.

Tabla 4.1: CU01

4.1.2.2.2. Subsistema de gestión de personas mayores

CU02	Ver detalle de persona.
Descripción	El usuario visualiza los datos y sesiones realizadas por la persona seleccionada.
Precondición	Debe existir al menos una persona registrada en la aplicación y el usuario debe haber pulsado sobre una persona del listado de personas registradas de la pantalla principal.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre una de las personas listadas en la pantalla principal. 2. Se muestra al usuario la pantalla de detalle de persona, donde se encuentran los datos de la misma y un listado de las sesiones realizadas.
Secuencia alternativa	No hay.
Postcondición	Se visualizan los datos de la persona seleccionada y se muestra el listado de sesiones realizadas y la sesión que actualmente tenga en progreso la persona, en caso de que tuviese alguna.
Importancia	Vital.

Tabla 4.2: CU02

CU03	Eliminar persona.
Descripción	El usuario elimina la persona seleccionada del sistema, así como todos los datos y resultados asociados a la misma.
Precondición	El usuario debe estar en la pantalla de detalle de usuario, por lo que debe existir al menos una persona registrada en la aplicación y el usuario debe haber pulsado sobre una persona del listado de personas registradas de la pantalla principal.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre una de las personas listadas en la pantalla principal. 2. Se muestra al usuario la pantalla de detalle de persona, donde se encuentran los datos de la misma y un listado de las sesiones realizadas. 3. El usuario pulsa sobre el ícono de la papelera, situado en la esquina superior izquierda. 4. Se muestra al usuario un popup de confirmación, donde el usuario puede confirmar o cancelar la acción. 5. El usuario confirma la acción y el sistema elimina la persona seleccionada y retorna al usuario a la pantalla principal de la aplicación.
Secuencia alternativa	5a. El usuario cancela la acción. Vuelve al paso 2.
Postcondición	Se elimina la persona seleccionada del sistema exitosamente, así como los datos y resultados asociados a la misma.
Importancia	Vital.

Tabla 4.3: CU03

4.1.2.2.3. Subsistema de gestión de sesiones

CU04	Comenzar sesión.
Descripción	El usuario pulsa sobre el botón de la pesa situado en la esquina superior derecha de la pantalla principal y la aplicación muestra el listado de personas registradas en el sistema para que el usuario seleccione la persona que comenzará la sesión.
Precondición	Debe existir al menos una persona registrada en la aplicación y el usuario debe haber pulsado sobre el botón de la pesa situado en la esquina superior derecha de la pantalla principal.
Secuencia normal	1. El usuario pulsa sobre el icono de la pesa situado en la esquina superior derecha de la pantalla principal. 2. Se muestra la pantalla el listado de personas registradas en la aplicación.
Secuencia alternativa	2a. Si no hay ninguna persona registrada en la aplicación, se muestra un mensaje indicando que debe existir al menos una persona registrada en el sistema.
Postcondición	Se muestra el listado de personas registradas en la aplicación para que el usuario seleccione la persona que comenzará la sesión.
Importancia	Vital.

Tabla 4.4: CU04

CU05	Seleccionar persona.
Descripción	El usuario pulsa sobre uno de los usuarios listados y, en caso de que el usuario no tenga ninguna sesión en progreso, se creará la sesión en el sistema para el usuario seleccionado. Por último se mostrará la pantalla de selección de test.
Precondición	El usuario se debe encontrar en la pantalla de selección de usuario y debe existir al menos una persona registrada en la aplicación.
Secuencia normal	1. El usuario selecciona una de las personas listadas. 2. Se crea la sesión en el sistema para la persona seleccionada en caso de que el usuario no tuviese ninguna sesión en progreso. 3. Se muestra la pantalla de selección de test.
Secuencia alternativa	2a. La persona seleccionada ya tiene una sesión creada y en progreso. Salta al punto 3, mostrándose los resultados obtenidos en aquellos tests que ya hayan sido completados para la sesión que tiene en progreso.
Postcondición	Se crea la sesión en el sistema satisfactoriamente para la persona seleccionada en caso de que no tuviese ninguna sesión en progreso y se muestra al usuario la pantalla de selección de test.
Importancia	Vital.

Tabla 4.5: CU05

CU06	Seleccionar test.
Descripción	El usuario pulsa sobre uno de los test listados que no estén ya completados y se mostrará la pantalla de realización del test correspondiente.
Precondición	El usuario debe encontrarse en la pantalla de selección de test y la persona seleccionada debe tener una sesión creada y no completada.
Secuencia normal	1. El usuario selecciona uno de los tests que no estén ya completados. 2. Se muestra la pantalla de realización del test correspondiente.
Secuencia alternativa	1a. El usuario selecciona un test que ya estaba completado. Se muestra un mensaje en pantalla indicando que el test ya ha sido realizado y el usuario permanece en la pantalla de selección de test.
Postcondición	Se accede satisfactoriamente a la pantalla de realización del test.
Importancia	Vital.

Tabla 4.6: CU06

CU07	Ver información de test.
Descripción	El usuario pulsa sobre el ícono de información situado a la derecha de cada test en la pantalla de selección de test. El sistema entonces muestra una pantalla con explicaciones sobre cómo realizar el test para el que se está consultando la información.
Precondición	El usuario debe encontrarse en la pantalla de selección de test y la persona seleccionada debe tener una sesión creada y no completada.
Secuencia normal	1. El usuario pulsa sobre el botón de información que aparece a la derecha de los tests no completados en la pantalla de selección de test. 2. Se muestra la pantalla de información del test, con las explicaciones pertinentes sobre como realizar el ejercicio.
Secuencia alternativa	No hay.
Postcondición	Se accede satisfactoriamente a la pantalla de información del test.
Importancia	Vital.

Tabla 4.7: CU07

CU08	Reanudar sesión
Descripción	El usuario pulsa sobre la sesión en progreso que aparece listada en la pantalla de detalle de persona, mostrándose la pantalla de selección de test.
Precondición	El usuario debe encontrarse en la pantalla de detalle de usuario y la persona seleccionada debe tener una sesión creada y no completada.
Secuencia normal	1. El usuario pulsa sobre la sesión en progreso listada para el usuario del que estamos consultando el detalle. 2. Se muestra la pantalla de selección de test, diferenciando aquellos que ya estén completados de los pendientes y mostrando los resultados para los que ya están realizados por el usuario para esa sesión.
Secuencia alternativa	No hay.
Postcondición	Se accede satisfactoriamente a la pantalla de selección de test y se muestran los resultados de aquellos que ya estuviesen completados para la sesión, en caso de que los hubiese.
Importancia	Vital.

Tabla 4.8: CU08

4.1.2.2.4. Subsistema de resultados y cálculos de estadísticas

CU09	Ver estadísticas de sesión
Descripción	El usuario pulsa sobre cualquier sesión ya completada de las que aparecen listadas en la pantalla de detalle de persona, mostrándose la pantalla de estadísticas de la sesión.
Precondición	El usuario debe encontrarse en la pantalla de detalle de usuario y la persona seleccionada debe tener al menos una sesión ya completada.
Secuencia normal	1. El usuario pulsa sobre cualquiera de las sesiones listadas como completadas para el usuario del que estamos consultando el detalle. 2. Se muestra la pantalla de estadísticas de la sesión completada seleccionada en el paso anterior.
Secuencia alternativa	No hay.
Postcondición	Se accede satisfactoriamente a la pantalla de estadísticas de la sesión completada seleccionada en la pantalla de detalle de usuario.
Importancia	Vital.

Tabla 4.9: CU09

CU10	Mostrar resultados
Descripción	Cuando el usuario realiza el último test que quede por completar de la sesión, se mostrará al usuario la pantalla de resultados, que mostrará los resultados obtenidos en cada uno de los tests durante la sesión que se acaba de completar.
Precondición	El usuario debe haber completado todos los tests de la sesión que en ese momento tuviese en progreso.
Secuencia normal	1. El usuario completa el último test que tuviese pendiente de la sesión que tuviese en progreso en el momento de la realización del test indicado. 2. Una vez obtenido el resultado del último test, se muestra la pantalla de resultados de todos los tests completados para la sesión que recién se ha acabado de completar.
Secuencia alternativa	1a. El test que se ha completado no era el último y había algún test más pendiente para la sesión actual. Se vuelve a la pantalla de selección de test.
Postcondición	Se muestra satisfactoriamente la pantalla de resultados de la sesión, mostrándose los valores obtenidos para cada uno de los tests que se han completado.
Importancia	Vital.

Tabla 4.10: CU10

4.1.2.2.5. Tests de valoración de la condición física

CU11	Definir duración de test
Descripción	El usuario define el tiempo (en segundos) que durará el test que se va a realizar. Este caso de uso solo aplica para aquellos tests que usan una cuenta atrás.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: fuerza de brazos, fuerza de piernas o resistencia aeróbica.
Secuencia normal	1. El usuario pulsa sobre el tiempo predefinido por defecto (30 segundos). 2. Se abre el teclado numérico del dispositivo móvil. 3. El usuario borra el valor que ya esté especificado, introduce el número de segundos deseado y acepta.
Secuencia alternativa	No hay.
Postcondición	Se actualiza satisfactoriamente el tiempo de duración de la prueba, mostrándose en la pantalla el número de segundos introducido por el usuario.
Importancia	Vital.

Tabla 4.11: CU11

CU12	Iniciar test
Descripción	El usuario pulsa sobre el botón START, activándose la detección de movimiento en el dispositivo móvil y comenzando la cuenta atrás. Además aparecen en pantalla los botones STOP y RESET.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: fuerza de brazos, fuerza de piernas o resistencia aeróbica.
Secuencia normal	1. El usuario coloca el dispositivo móvil en el brazo o en la pierna (según sea necesario) de la persona que va a realizar el test. 2. El usuario pulsa sobre el botón START.
Secuencia alternativa	No hay.
Postcondición	Se activa la detección de movimiento en el dispositivo móvil y comienza la cuenta atrás. El botón START desaparece y en su lugar aparecen los botones STOP y RESET.
Importancia	Vital.

Tabla 4.12: CU12

CU13	Detectar movimiento
Descripción	El dispositivo móvil, colocado en el brazo o en la pierna con el/la que se realizará los ejercicios, detectará el movimiento realizado y contará las repeticiones que se hayan realizado de forma correcta, emitiendo un sonido por cada repetición contabilizada.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: fuerza de brazos, fuerza de piernas o resistencia aeróbica. Debe tener colocado el móvil en el brazo o en la pierna, según el test a realizar y debe haber pulsado sobre el botón START, encontrándose el test en progreso y sin que haya finalizado la cuenta atrás.
Secuencia normal	1. La persona que esté realizando el ejercicio y tenga el dispositivo móvil colocado en la extremidad con la que esté realizando la actividad, hace el movimiento requerido por el test mientras la cuenta atrás siga en progreso.
Secuencia alternativa	1a. El movimiento realizado no es correcto. El sistema no contabiliza la repetición y no emite ningún sonido, manteniéndose inalterado el número de repeticiones que se muestra en pantalla. 1b. La persona realiza el movimiento cuando la cuenta atrás ha finalizado. El sistema no contabiliza la repetición y no emite ningún sonido, dado que al haber acabado la cuenta atrás el dispositivo móvil ya no detecta el movimiento realizado.
Postcondición	El sistema contabiliza la repetición, aumentando en 1 el número de repeticiones mostradas en pantalla y emitiendo un sonido.
Importancia	Vital.

Tabla 4.13: CU13

CU14	Guardar resultado
Descripción	El usuario pulsa sobre el botón GUARDAR y el sistema almacena en la base de datos el resultado obtenido en el test para la sesión en progreso y para la persona que lo haya realizado.
Precondición	El usuario debe encontrarse en la pantalla de realización del test. Para que el botón GUARDAR aparezca, el test debe haberse completado. Según el tipo de test, se considerará completado cuando se cumplan una de las siguientes condiciones: que la cuenta atrás haya terminado, que el cronómetro se haya parado por el usuario o que se haya introducido la distancia medida para los tests de flexibilidad.
Secuencia normal	<ol style="list-style-type: none"> 1. Se finaliza el ejercicio o se completa la información requerida por el test. 2. El sistema muestra el botón GUARDAR. 3. El usuario pulsa sobre el botón GUARDAR. 4. El sistema almacena en la base de datos el resultado del test para la sesión en progreso y la persona que lo haya realizado y lleva a la pantalla de selección de test, donde se muestra el resultado obtenido junto al test recién completado.
Secuencia alternativa	<ol style="list-style-type: none"> 1a. El test aún no se ha completado. Volver al paso 1. 4a. El test recién completado era el último que quedaba pendiente de la sesión. En lugar de llevar a la pantalla de selección de test, lleva a la pantalla de resultados de la sesión.
Postcondición	El sistema almacena satisfactoriamente en la base de datos el resultado del test para la sesión en progreso y la persona que lo haya realizado y lleva a la pantalla de selección de test, donde se muestra el resultado recién almacenado.
Importancia	Vital.

Tabla 4.14: CU14

CU15	Parar test
Descripción	La realización del test se para, de forma que el dispositivo móvil deja de detectar el movimiento y la cuenta atrás se detiene. Se muestra el botón RESET para volver los contadores a 0 si así se desea.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: fuerza de brazos, fuerza de piernas o resistencia aeróbica y el test debe estar en progreso para que el botón STOP se muestre.
Secuencia normal	1. El usuario pulsa sobre el botón STOP.
Secuencia alternativa	No hay.
Postcondición	Tanto la cuenta atrás como la detección de movimiento del dispositivo móvil se paran. Se permanece en la pantalla de realización de test, mostrando valores obtenidos hasta el momento y apareciendo el botón RESET.
Importancia	Vital.

Tabla 4.15: CU15

CU16	Resetear test
Descripción	Restablece todos los contadores y temporizadores a sus valores iniciales, volviéndose a mostrar la pantalla como en su estado inicial.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: fuerza de brazos, fuerza de piernas o resistencia aeróbica. El test debe estar en progreso o parado (tras haber estado en progreso) para que el botón RESET se muestre.
Secuencia normal	1. El usuario pulsa sobre el botón RESET.
Secuencia alternativa	No hay.
Postcondición	La cuenta atrás se inicializa y las repeticiones contabilizadas hasta el momento se ponen a 0. La detección de movimiento del dispositivo móvil se deshabilita. Se permanece en la pantalla de realización de test, con todos los contadores mostrando los valores iniciales y apareciendo únicamente el botón START.
Importancia	Vital.

Tabla 4.16: CU16

CU17	Iniciar cronómetro
Descripción	El usuario pulsa sobre el botón START, activándose el cronómetro. Además aparecen en pantalla los botones STOP y RESET.
Precondición	El usuario debe encontrarse en la pantalla de realización del test de Agilidad y el cronómetro no debe estar funcionando.
Secuencia normal	1. El usuario pulsa sobre el botón START.
Secuencia alternativa	No hay.
Postcondición	El cronómetro comienza a funcionar y se va mostrando el tiempo transcurrido en pantalla. El botón START desaparece y en su lugar aparecen los botones STOP y RESET.
Importancia	Vital.

Tabla 4.17: CU17

CU18	Parar cronómetro
Descripción	El cronómetro se para, mostrando en pantalla el tiempo transcurrido hasta el momento. Se muestra el botón RESET para volver los contadores a 0 si así se desea y el botón GUARDAR.
Precondición	El usuario debe encontrarse en la pantalla de realización del test de Agilidad y el cronómetro debe estar funcionando.
Secuencia normal	1. El usuario pulsa sobre el botón STOP.
Secuencia alternativa	No hay.
Postcondición	Se para el cronómetro y se muestra en pantalla el tiempo registrado hasta el momento. Además aparecen en pantalla el botón RESET, por si se quiere resetear el cronómetro, y el botón GUARDAR, por si se quiere guardar el tiempo obtenido durante la realización del ejercicio.
Importancia	Vital.

Tabla 4.18: CU18

CU19	Resetear cronómetro
Descripción	Restablece el cronómetro a 0 y se ocultan todos los botones, volviéndose a mostrar la pantalla como en su estado inicial.
Precondición	El usuario debe encontrarse en la pantalla de realización del test de Agilidad y el cronómetro debe estar en funcionamiento o parado (tras haber estado en funcionamiento) para que el botón RESET se muestre.
Secuencia normal	1. El usuario pulsa sobre el botón RESET.
Secuencia alternativa	No hay.
Postcondición	El cronómetro se inicializa a 0. Se permanece en la pantalla de realización de test pero volviendo a su estado inicial y apareciendo únicamente el botón START.
Importancia	Vital.

Tabla 4.19: CU19

CU20	Introducir distancia
Descripción	El usuario introduce la medida obtenida tras realizar el test de flexibilidad de brazos o de piernas a la persona que realiza la sesión.
Precondición	El usuario debe encontrarse en la pantalla de realización de alguno de los siguientes tests: flexibilidad de brazos o flexibilidad de piernas.
Secuencia normal	1. El usuario pulsa sobre el input donde introducirá la distancia en centímetros. 2. Se abre el teclado numérico del dispositivo móvil. 3. El usuario introduce la distancia obtenida y acepta.
Secuencia alternativa	No hay.
Postcondición	Se muestra satisfactoriamente en la pantalla la distancia introducida y se muestra el botón GUARDAR para almacenar en el sistema el valor introducido para la sesión y la persona que ha realizado el test.
Importancia	Vital.

Tabla 4.20: CU20

4.1.3. Requisitos no funcionales

En este apartado se detallarán los requisitos no funcionales del sistema.

RNF01	Usabilidad.
Descripción	El sistema debe ser usable y disponer de una interfaz intuitiva, adaptable y fácil de manejar para un usuario de nivel básico.
Importancia	Vital.

Tabla 4.21: RNF01

RNF02	Portabilidad.
Descripción	Debe ser posible usar la aplicación en diferentes dispositivos móviles Android, independientemente del tamaño de la pantalla de los mismos, gracias al diseño adaptable empleado en las interfaces de usuario.
Importancia	Vital.

Tabla 4.22: RNF02

RNF03	Escabilidad.
Descripción	El sistema debe responder de manera óptima y eficiente a futuras mejoras en el desarrollo de la aplicación, sin que éstas comprometan el estado de la misma. El código debe ser mantenible y fácilmente ampliable para futuras versiones.
Importancia	Vital.

Tabla 4.23: RNF03

RNF04	Seguridad
Descripción	Cuando la aplicación necesite hacer uso de algunas características del dispositivo móvil que puedan comprometer la privacidad/seguridad de los datos del usuario, como puede ser la cámara o el acceso de lectura/escritura a los archivos de la memoria del dispositivo, siempre solicitará permiso al usuario antes de hacer uso de las mismas. En caso de que no se conceda el permiso, la aplicación seguirá siendo funcional aunque no se pueda hacer uso de las características para las que no se ha concedido el permiso.
Importancia	Vital.

Tabla 4.24: RNF04

RNF05	Rendimiento
Descripción	El rendimiento de la aplicación de ser tal que permita un desempeño agradable y suave durante el uso de la misma. Los tiempos de respuesta al insertar/recuperar información en la base de datos serán cortos y deberá minimizarse la utilización de recursos cuando sea posible para ahorrar batería.
Importancia	Vital.

Tabla 4.25: RNF05

RNF06	Diseño
Descripción	Como se ha indicado en el requisito anterior, minimizar la utilización de recursos y el tiempo de respuesta debe primar sobre cualquier factor. Sin embargo, se debe tener en cuenta que los dispositivos móviles Android (Java) pueden perder el contexto de la aplicación que ejecutan, por lo que el contenido debe ser recuperable. Será de valor añadido que el paquete .APK final tenga el menor tamaño posible para abarcar aquellos teléfonos con menos memoria secundaria.
Importancia	Vital.

Tabla 4.26: RNF06

4.1.4. Reglas de negocio

El sistema en su totalidad está desarrollado para ser software libre bajo licencia GNU GPL. Ello permite que el código sea accesible a cualquier desarrollador que quiera incrementar su funcionalidad y contribuir a la libre creación de contenidos. Así mismo, las herramientas y tecnologías empleadas también son de libre uso.

4.1.5. Requisitos de información

A continuación se detallarán los requisitos de información, que describen cómo gestiona el sistema la información que se va a almacenar, y son los siguientes:

RIN01	Información de la persona.
Descripción	El sistema debe almacenar la información sobre las personas mayores registradas en la aplicación.
Datos específicos	DNI de la persona. Nombre de la persona. Apellidos de la persona. Género de la persona. Fecha de nacimiento de la persona. Foto asociada a la persona.
Importancia	Vital.

Tabla 4.27: RIN01

RIN02	Información de la sesión.
Descripción	El sistema debe almacenar la información correspondiente a las sesiones de ejercicio que realizan las personas registradas en la aplicación.
Datos específicos	Identificador de la persona que realiza la sesión. Estado de la sesión. Fecha de inicio de la sesión.
Importancia	Vital.

Tabla 4.28: RIN02

RIN03	Información del test.
Descripción	El sistema debe almacenar la información correspondiente a los diferentes tests que realizarán las personas y que componen la batería de pruebas Senior Fitness Test (y por tanto cada una de las sesiones).
Datos específicos	Nombre del test. Descripción del test.
Importancia	Vital.

Tabla 4.29: RIN03

RIN04	Información del resultado.
Descripción	El sistema debe almacenar los resultados que las personas obtienen en cada uno de los tests que componen la batería de pruebas Senior Fitness Test, y que estarán asociados a la sesión que la persona haya realizado en una fecha concreta.
Datos específicos	Identificador de la persona que ha realizado el test. Identificador de la sesión en la que la persona ha realizado el test. Identificador del test realizado. Resultado obtenido en el test. Fecha de obtención del resultado.
Importancia	Vital.

Tabla 4.30: RIN04

4.2. Modelo conceptual de datos

En esta sección se muestra el diagrama conceptual de datos UML de la aplicación, con el que se visualizarán a modo de primer vistazo las clases que posteriormente se diseñarán e implementarán para cubrir las funcionalidades impuestas en el proyecto.

En la siguiente imagen podemos ver el diagrama Entidad-Relación que representa las entidades relevantes del sistema de información implementado, así como sus interrelaciones:

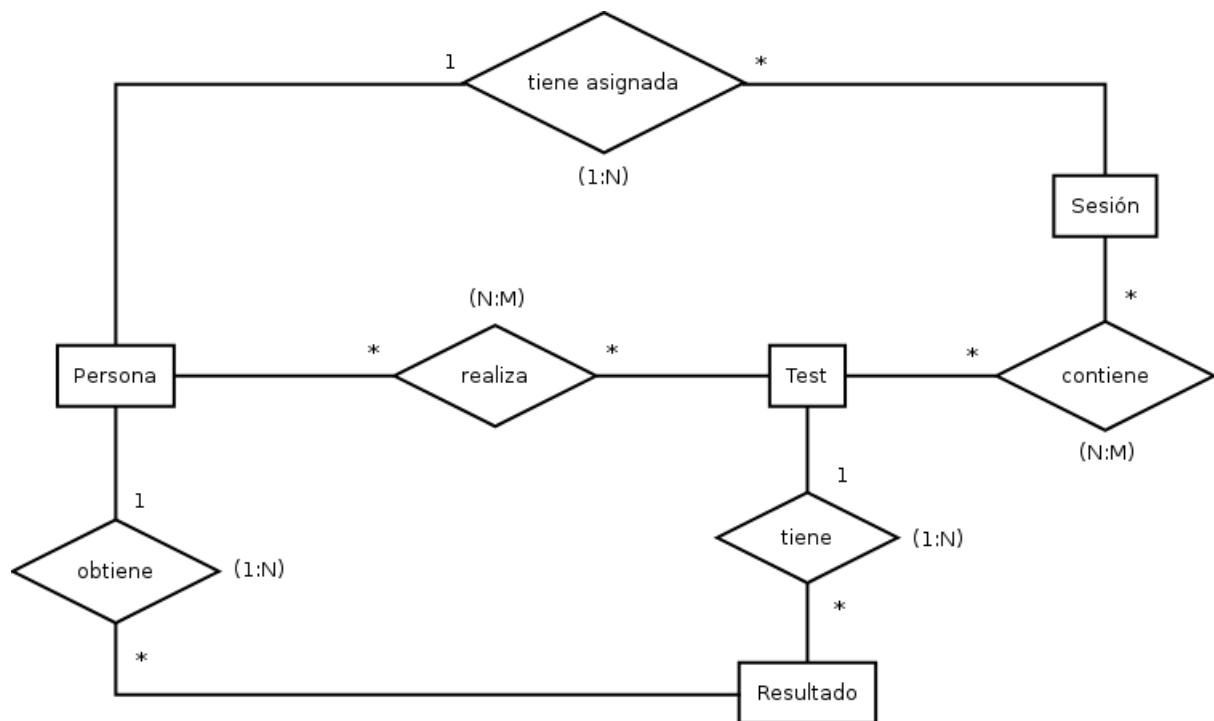


Figura 4.8: Análisis: Modelo conceptual Entidad-Relación

Capítulo 5

Diseño del sistema

En este capítulo se describen todos los aspectos relacionados con el diseño del sistema: arquitectura del sistema, patrones de diseño, diseño físico de datos, así como el diseño de la interfaz de usuario.

5.1. Arquitectura física del sistema

A continuación se presentan los elementos hardware que componen la arquitectura física del sistema. Para definirla, se detallarán los componentes a nivel de hardware requeridos para el correcto funcionamiento de la aplicación.

Cuando hablamos de un dispositivo móvil o smartphone, nos referimos a los sucesores de los teléfonos móviles sencillos, que han ido evolucionando hasta ser prácticamente ordenadores que caben en nuestros bolsillos o en nuestra palma de la mano, y que además incluyen las funciones básicas de sus antecesores, como pueden ser las llamadas, envío y recibo de sms, mms, etc.

En los últimos años los dispositivos móviles han empezado a cobrar importancia debido a su relación entre la potencia y su reducido tamaño, llegando incluso a ser indispensables en ciertos sectores de la sociedad hoy en día.

Junto con la aparición de los smartphones se presentaron a su vez los sistemas operativos que éstos dispositivos llevarían integrados como pueden ser Android o IOS, cada uno con características propias, ofreciendo así un entorno de fácil manejo para el usuario final y un sinfín de aplicaciones desarrolladas tanto por empresas como por particulares.

Dado que la presente aplicación ha sido desarrollada en Android, los requisitos a nivel de hardware se reducen a los siguientes:

- Dispositivos: Dispositivo móvil basado en Android.
- Requisitos mínimos:
 - La versión de Android instalada en el dispositivo debe ser superior o igual a la versión 4.0.3 (ICE_CREAM SANDWICH_MR1).
 - El dispositivo móvil debe disponer de Acelerómetro y Giroscopio.

Es conveniente destacar la necesidad de que el dispositivo móvil disponga de Acelerómetro y Giroscopio, dado que la combinación de estos dos sensores hardware son los que permiten que exista el sensor de Gravedad que, a diferencia de los dos sensores anteriormente citados, se trata de un sensor software.

5.2. Arquitectura lógica del sistema

En la arquitectura lógica del sistema se detallan los componentes a nivel de software empleados a lo largo del proyecto, que abarcan tanto al conjunto de aplicaciones, bibliotecas y librerías como al propio software desarrollado para cumplir con los objetivos y requisitos establecidos.

5.2.1. Android

Introducción

Android [3] puede entenderse como una plataforma software cuya misión consiste en abstraer el hardware subyacente. Inicialmente fue desarrollado para dispositivos táctiles con recursos limitados, con el fin de facilitar el desarrollo de aplicaciones para dichos dispositivos.

La gran diferencia de Android respecto al resto de sistemas operativos para dispositivos móviles es su núcleo basado en GNU/Linux. Esto hace que Android adquiera algunas de las principales características de Linux, convirtiéndose en un software libre, gratuito y multiplataforma.

Inicialmente fue desarrollado por Android IC, pero en 2005 fue comprado por Google, y hasta entonces era un sistema operativo muy poco conocido. En el 2007 se fundó la Open Handset Alliance, una agrupación de empresas de desarrollo de software, hardware y telecomunicaciones con el propósito de avanzar en los estándares abiertos para el desarrollo de software y hardware para dispositivos móviles, y fue entonces cuando se produjo la presentación de Android por parte de Google, liberando gran parte de su código bajo una licencia Apache.

El éxito tras la presentación fue escaso, debido a que el sistema operativo se presentó antes de que se comercializara ningún dispositivo que lo incluyese. En la actualidad es el sistema operativo más utilizado.

Arquitectura

La arquitectura del sistema Android puede verse como una arquitectura por capas o niveles, de forma que cada nivel puede utilizar servicios ofrecidos por los niveles anteriores y éste, a su vez, proporciona nuevas funciones a los niveles superiores. En la siguiente imagen podemos ver los niveles que componen la arquitectura:

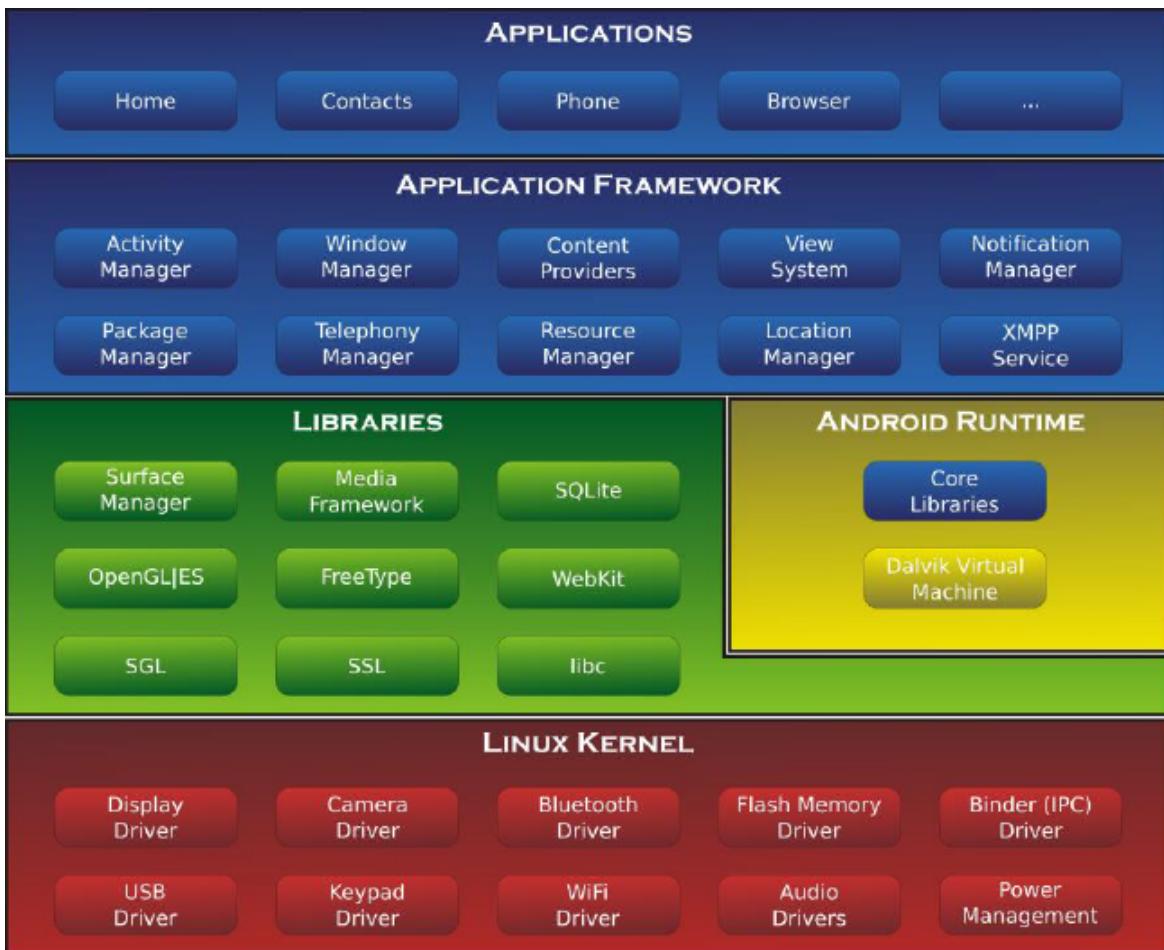


Figura 5.1: Arquitectura Android

- **Aplicaciones:** Constituye el conjunto de aplicaciones presentes en un dispositivo, ya sean las instaladas por el usuario o por defecto, las nativas y las administradas.
- **Framework de aplicaciones:** Plataforma de desarrollo que facilita la reutilización de componentes, permite el acceso a los diversos servicios ofrecidos y al hardware de los dispositivos. Los más importantes son:
 - Activity Manager: Conjunto de APIs encargadas de gestionar el ciclo de vida de las aplicaciones.
 - Window Manager: Gestiona las ventanas de una aplicación mediante la librería Surface Manager.
 - Telephone Manager: Compendio de APIs que gestionan las funciones básicas de los teléfonos (llamadas, mensajería, etc).
 - Content Provider: Proporciona los mecanismos necesarios para la comunicación entre aplicaciones.
 - View System: Ofrece los elementos básicos necesarios para la construcción de interfaces.
 - Location Manager: Posibilita a las aplicaciones el acceso a la ubicación del dispositivo.
 - Notification Manager: Permite a las aplicaciones notificar al usuario información asociada a ciertos eventos que ocurrán durante la ejecución de una aplicación.

- XMPP Service: Colección de APIs para el uso de este protocolo de intercambio basado en XML.
 - Resource Manager: Encargado de gestionar todos los elementos que forman parte de una aplicación y son externos al código.
 - Package Manager: Gestor de todos los paquetes instalados en un dispositivo Android y que permite la instalación de nuevos paquetes.
- **Bibliotecas:** Hacen referencia al conjunto de librerías presentes en Android y proporcionan la mayoría de las características más representativas de esta plataforma. Las principales librerías que podemos encontrar son las siguientes:
- Surface Manager: Gestión de la pantalla.
 - SQLite: Motor de bases de datos relacionales. Es el motor de bases de datos usado en la presente aplicación.
 - Media Framework: Reproducción de imágenes, vídeo y audio.
 - WebKit: Navegación web.
 - SGL: Gráficos 2D
 - Open GL/ES: Gráficos 3D.
 - FreeType: Renderizado de fuentes.
 - SSL: Comunicación segura mediante sockets.
 - Libcr: Variante optimizada de C.
- **Android Runtime:** Al mismo nivel que las librerías encontramos el entorno de ejecución, que está constituido por las librerías Java que forman el núcleo del lenguaje y la máquina virtual Dalvik o ART para las versiones más modernas.
- **Núcleo Linux:** Se utiliza como una capa de abstracción para el hardware subyacente y por tanto contiene los drivers y controladores necesarios para el correcto funcionamiento del mismo.

Componentes de una aplicación Android

Los componentes de una aplicación Android son los elementos esenciales que la componen. Cada uno de ellos supone un punto de entrada mediante el cual el sistema puede acceder a la aplicación, aunque no todos suponen puntos de entrada para el usuario.

Cada componente es un bloque básico que desempeña un papel específico en el funcionamiento de nuestra aplicación. Existen cuatro tipos diferentes de componentes, cada uno de ellos con un ciclo de vida y propósito único. Los componentes son los que siguen a continuación:

- **Actividades o Activities:** Cada actividad o activity representa una pantalla de nuestra aplicación. Las actividades trabajan en conjunto para dar una visión coherente de la aplicación, sin embargo, la vida de cada actividad es independiente del resto. El ciclo de vida que sigue una actividad se muestra en la siguiente imagen:

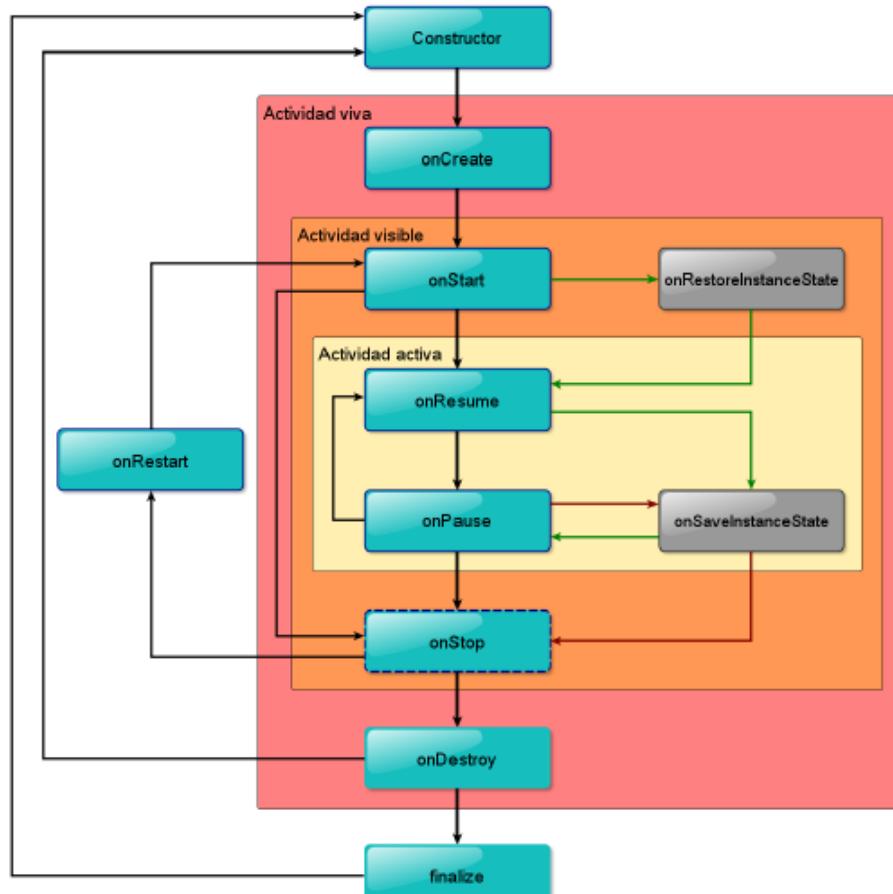


Figura 5.2: Arquitectura Android: Ciclo de vida de una Activity

- **Servicios o Services:** Son componentes que se ejecutan en segundo plano con el fin de realizar operaciones de larga duración o procesos remotos. Los servicios carecen de interfaz y generalmente son llamados por las activities para realizar tareas costosas sin bloquear la interfaz, es decir, se pueden dejar en ejecución en segundo plano o lograrlos a una actividad para interactuar con ella. En la siguiente imagen veremos el ciclo de vida de un servicio. En la izquierda está actuando en segundo plano y en la derecha está ligado a una activity:

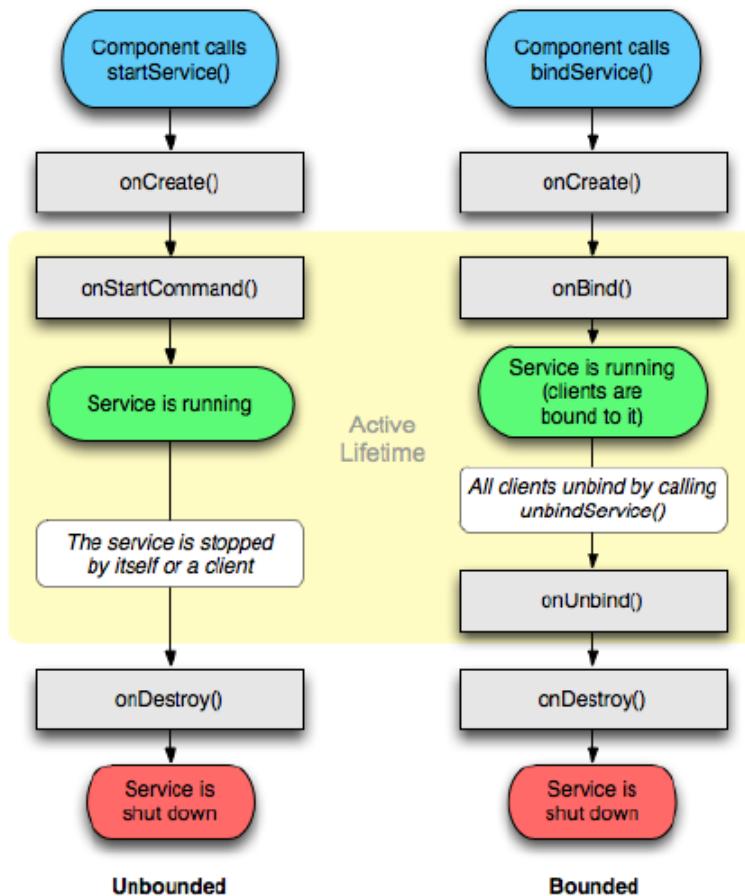


Figura 5.3: Arquitectura Android: Ciclo de vida de un Servicio

- **Proveedor de contenido o Content provider:** El proveedor de contenido gestiona los datos que maneja una aplicación. Controla el acceso a archivos, bases de datos SQLite, etc. Mediante el proveedor de contenidos las aplicaciones pueden consultar y modificar datos de otras aplicaciones siempre que posean los permisos adecuados. La siguiente imagen ilustra cómo una activity puede consultar datos de otra aplicación:

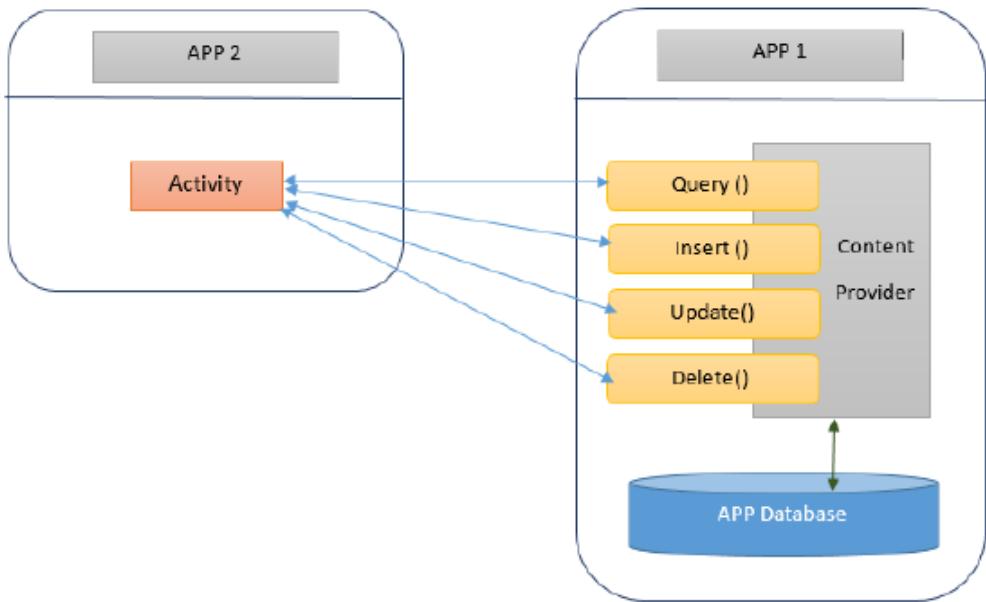


Figura 5.4: Arquitectura Android: Content Provider

- **Receptor de difusiones o Broadcast receiver:** Es el componente encargado de responder a las difusiones o anuncios del sistema (por ejemplo, un anuncio del sistema que indica que queda poca batería). Carece de interfaz, pero debe crear una barra de estado para notificar al usuario cuando detecta una difusión. Una aplicación debe registrar un broadcast receiver para indicar qué difusiones le interesan.

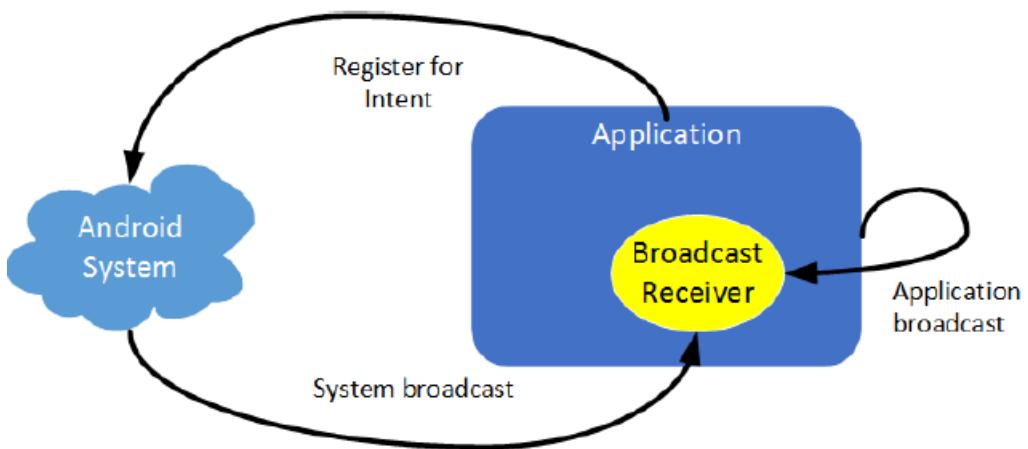


Figura 5.5: Arquitectura Android: Broadcast Receiver

Recursos de una aplicación Android

Los recursos de una aplicación corresponden con todos los ficheros, imágenes, cadenas de texto, etc, que nuestra aplicación utiliza. Cuando trabajamos desarrollando aplicaciones para Android, las buenas prácticas nos dicen que este tipo de archivos debe mantenerse independientemente al código de la aplicación.

Este hecho, conocido como externalización de recursos, permite adaptar un mismo código a dispositivos con diferentes configuraciones y características físicas de una forma prácticamente automática. En la siguiente imagen podemos comprobar cómo quedaría una pantalla que no ha sido adaptada para diferentes dispositivos (parte superior) y una que sí (parte inferior):

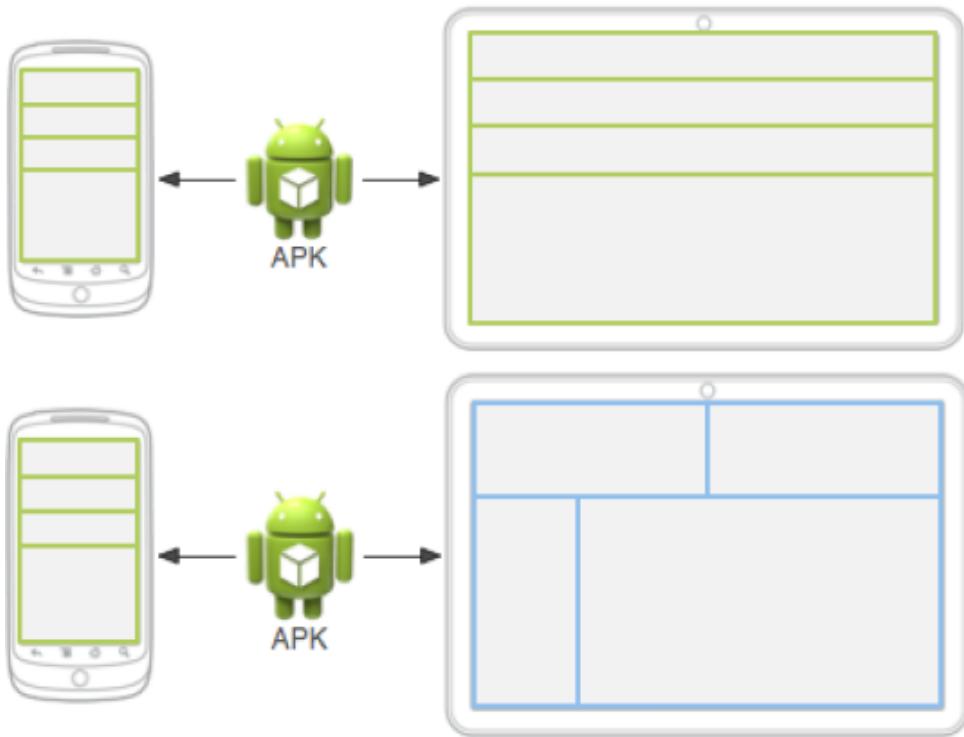


Figura 5.6: Arquitectura Android: Externalización de recursos

Manifest de una aplicación Android

El archivo Manifest es un archivo XML que contiene información esencial para Android acerca de la aplicación. Todas las aplicaciones Android deben tener un archivo `AndroidManifest.xml` en su directorio raíz.

Entre todas las funciones que realiza el archivo manifest, las más significativas son las siguientes:

- Da nombre al paquete Java de la aplicación.
- Describe los componentes y qué proceso los hospeda.
- Contiene la declaración de permisos.
- Lista las librerías necesarias.
- Declara el nivel mínimo de la API de Android requerido.

5.2.2. Git

Git [2] es el software de control de versiones que se ha elegido para el desarrollo del proyecto. Fue desarrollado por Linus Torvalds (creador del sistema operativo Linux) y a día de hoy se usa en grandes

proyectos, por ejemplo el propio núcleo de Linux. Como es costumbre en el desarrollo de éste proyecto, Git es software libre distribuido bajo la licencia GPL, lo que lo hace una alternativa libre de usar.

Git permite tener el código generado en la implementación del sistema en un repositorio (GitHub) y poder acceder a él remotamente desde cualquier PC, así como poder restaurar versiones anteriores en caso que la actual deba ser reemplazada.

Como para cualquier proyecto informático, el uso de un repositorio como GitHub para el alojamiento de la aplicación que nos ocupa se antojó necesario, no sólo por el simple hecho de tener todo el sistema alojado en la nube y disponible para cualquier persona, sino por la seguridad que transmite tener todo el desarrollo en un lugar desde donde sea posible descargarlo en cualquier PC y desarrollarlo en éste, por lo que pudiera suceder a lo largo del desarrollo del proyecto.

5.2.3. SQLite

La plataforma Android proporciona dos herramientas principales para el almacenamiento y consulta de datos estructurados, que ya hemos nombrado en los puntos anteriores:

- Bases de Datos SQLite.
- Content providers.

En este punto nos centraremos en la primera opción, que es la que se usará en la aplicación implementada para el presente proyecto, y que abarcará todas las tareas relacionadas con el almacenamiento de los datos propios de nuestra aplicación.

SQLite [7] es un sistema gestor de base de datos relacional (RDBMS) basado en el Lenguaje Estructurado de Consultas (SQL), tal y como sugiere su nombre. Lo que hace único a SQLite es que se considera una solución embebida. La mayoría de los sistemas de gestión de bases de datos como Oracle, MySQL y SQL Server son procesos de servidor autónomos que se ejecutan independientemente, mientras que SQLite es en realidad una librería que está enlazada dentro de las aplicaciones.

Se trata de un motor de bases de datos muy popular en la actualidad por ofrecer características tan interesantes como su pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto ser de código libre.

Android incorpora de serie todas las herramientas necesarias para la creación y gestión de bases de datos SQLite, y entre ellas una completa API para llevar a cabo de manera sencilla todas las tareas necesarias.

En Android, la forma típica de crear, actualizar y conectar con una base de datos SQLite será a través de una clase auxiliar llamada SQLiteOpenHelper, o para ser más exactos, de una clase propia que derive de ella y que debemos personalizar para adaptarnos a las necesidades concretas de nuestra aplicación.

5.3. Patrones de diseño

Los patrones de diseño se pueden definir como los esqueletos de las soluciones a problemas comunes en el desarrollo de software. Éstos brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

Para que una solución sea considerada un patrón, debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores, condensando conocimientos ya existentes.

Asimismo, no pretenden:

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.

Para el presente proyecto se han usado patrones de diseño pertenecientes a diferentes subconjuntos de patrones de diseño. Éstos son el patrón Singleton, perteneciente al subconjunto de patrones de diseño creacionales, y el patrón Adapter, perteneciente al subconjunto de patrones de diseño estructurales. A continuación se detallarán ambos para dar conocimiento de cómo funcionan y cómo resuelven determinados problemas del sistema que los implementan.

5.3.1. Patrones de diseño creacionales

Corresponden a patrones de diseño de software que solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.

Patrón de diseño singleton

El patrón singleton o patrón de instancia única es un patrón de diseño diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor haciendo que éste sea privado y que sea a través de un método específico la manera de crear la primera y única instancia que se va a usar a lo largo del desarrollo.

En el desarrollo del presente proyecto, el uso de este patrón se usa para la clase que se ocupa de la definición de la base de datos, ya que a lo largo del proceso de ejecución del sistema sólo es necesario crear una única instancia de dicha clase.

5.3.2. Patrones de diseño estructurales

Corresponden a patrones de diseño de software que solucionan problemas de composición (agregación) de clases y objetos.

Patrón de diseño adapter

El patrón adapter, también conocido como wrapper, es un patrón de diseño que se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar otra clase cualquiera, haga uso de ella a través de una segunda. Dicho de otro modo, convierte la interfaz de una clase en otra interfaz que el cliente espera.

Este patrón de diseño permite a clases con interfaces incompatibles trabajar juntas mediante un adaptador intermedio, que se encargará de realizar la conversión de una interfaz a otra.

En el desarrollo de nuestra aplicación, el uso de este patrón se antoja necesario debido a los listados que se manejan en las diferentes pantallas de la aplicación, dado que, una vez que hemos obtenido de la base de datos el listado de usuarios, tests o sesiones, necesitaremos convertir dichas colecciones de objetos específicos en una colección de vistas (objeto View), que será sobre la que finalmente se iterará en la capa de presentación para pintar cada elemento en la layout correspondiente.

5.4. Diseño físico de datos

En este apartado se detalla el diseño físico de datos usado por nuestra aplicación.

Para que se cumplan con los objetivos y los requisitos impuestos en el proyecto, es necesario implementar una estructura capaz de almacenar la información, de forma que se pueda llevar un control y hacer un estudio de los datos obtenidos durante la realización de los tests que componen la batería de pruebas Senior Fitness Test.

La información almacenada en la base de datos consistirá tanto en las personas que realizarán los tests, como las sesiones que realizan y los resultados obtenidos en cada una de las pruebas completadas en dichas sesiones.

5.4.1. Base de datos

A continuación se detalla a través del diagrama Entidad-Relación cómo se estructura la base de datos SQLite diseñada para albergar la información citada en el punto anterior.

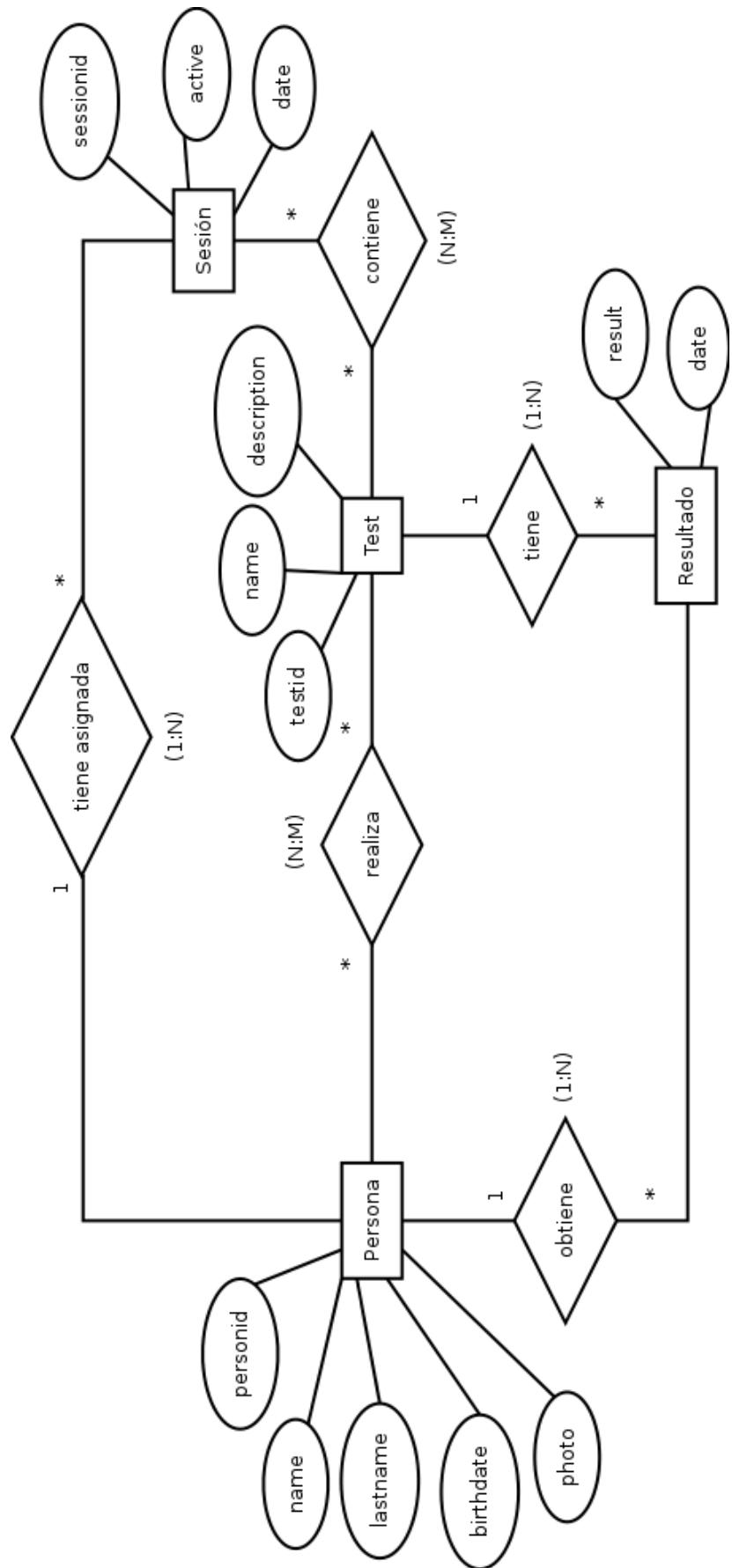


Figura 5.7: Diagrama Entidad-Relación de la base de datos

Los atributos identificadores de cada tabla son enteros a excepción del identificador de persona, que se corresponde con el DNI de la misma. Las fechas de los campos date y birthdate son de tipo String pero parseadas desde tipo Date con el siguiente formato dd/MM/yyyy.

En el diagrama no se han representado como atributos de las entidades las claves foráneas, aunque sí aparecen en el fichero Contract de la aplicación, en el que se definen la base de datos y así mismo las tablas del proyecto.

5.5. Diseño de la interfaz de usuario

En este punto se especificarán los elementos que componen las distintas interfaces de usuario que existen para la aplicación, detallándose el contenido de cada pantalla representada.

Será necesario desarrollar una interfaz sencilla y agradable para el usuario de la aplicación. En el diseño de la misma, se intentará en todo momento que el flujo de ejecución sea lo más intuitivo posible.

A continuación se listan las pantallas que componen la aplicación:

- Pantalla principal.
- Alta de persona.
- Detalle de persona.
- Estadísticas de sesión.
- Selección de persona.
- Selección de test.
- Información de test.
- Fuerza de piernas.
- Fuerza de brazos.
- Resistencia aeróbica.
- Flexibilidad de piernas.
- Flexibilidad de brazos.
- Agilidad.
- Resultados.

No se entrará en detalles sobre el objetivo y funcionalidad de cada pantalla dado que ya fueron descritas en el punto 4.1.1. del presente documento.

En el siguiente diagrama podemos ver las distintas opciones de navegación dentro de la aplicación:

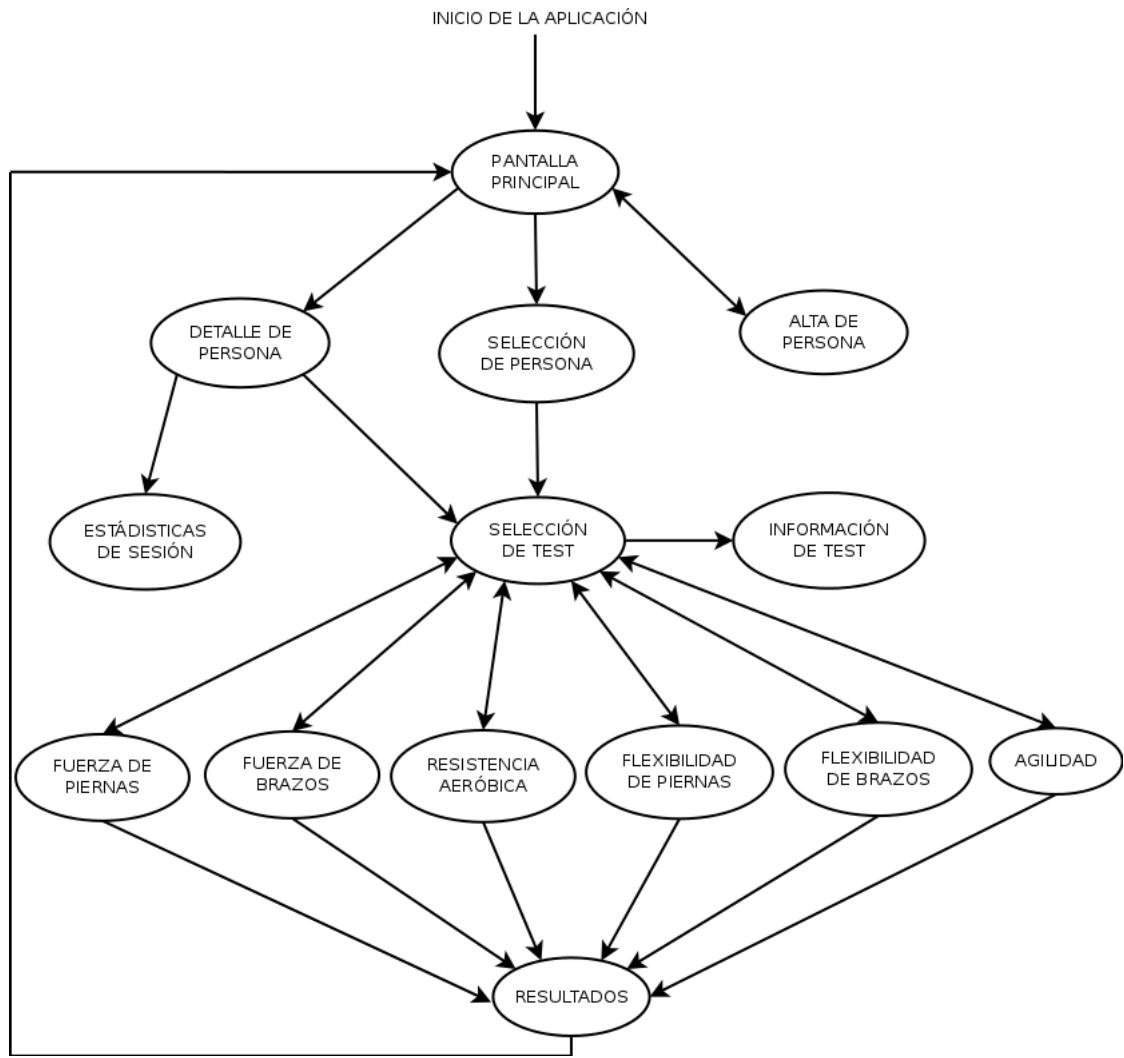


Figura 5.8: Navegación entre pantallas

5.5.1. Pantalla principal

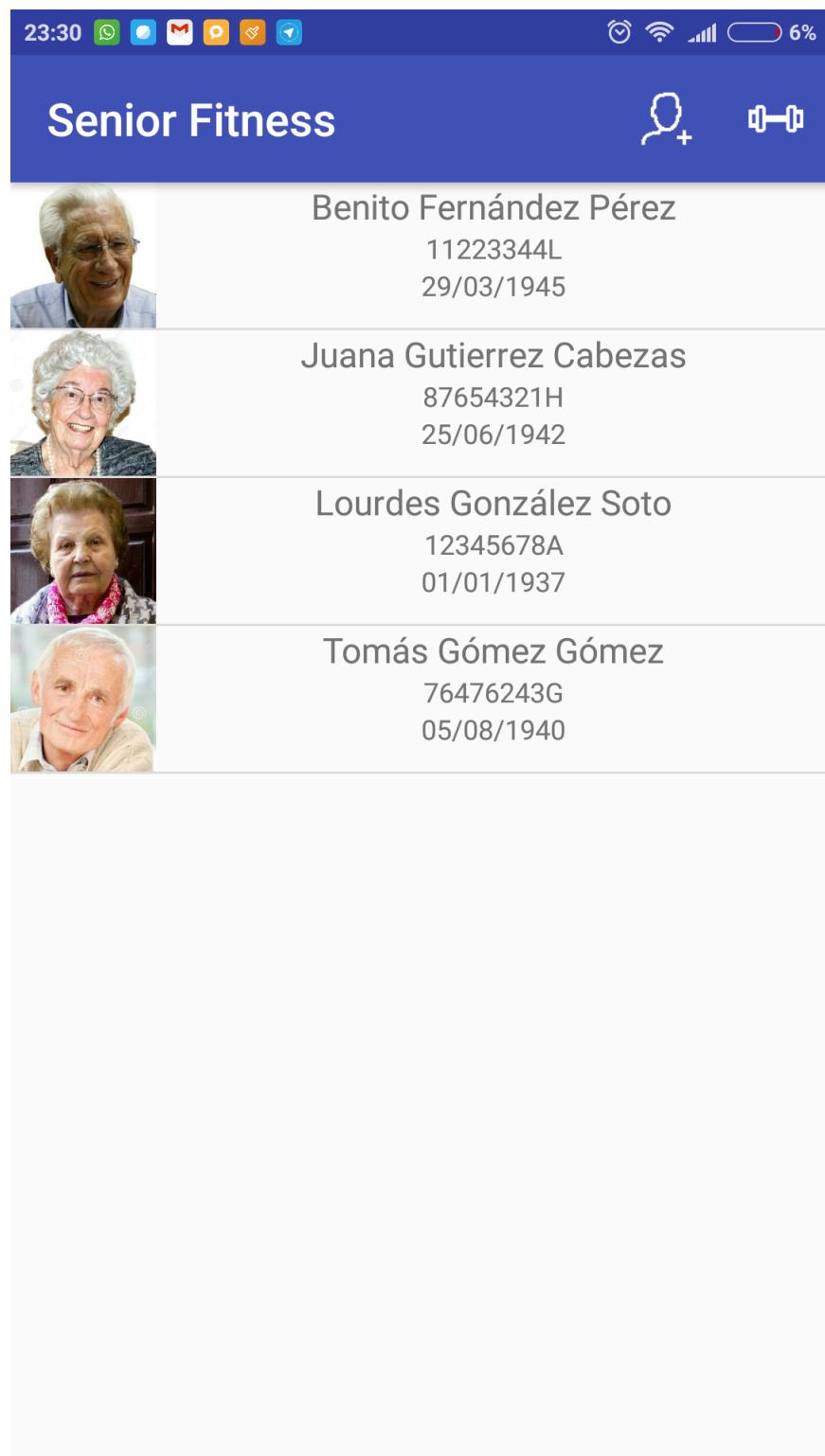


Figura 5.9: Pantalla principal

5.5.2. Alta de persona

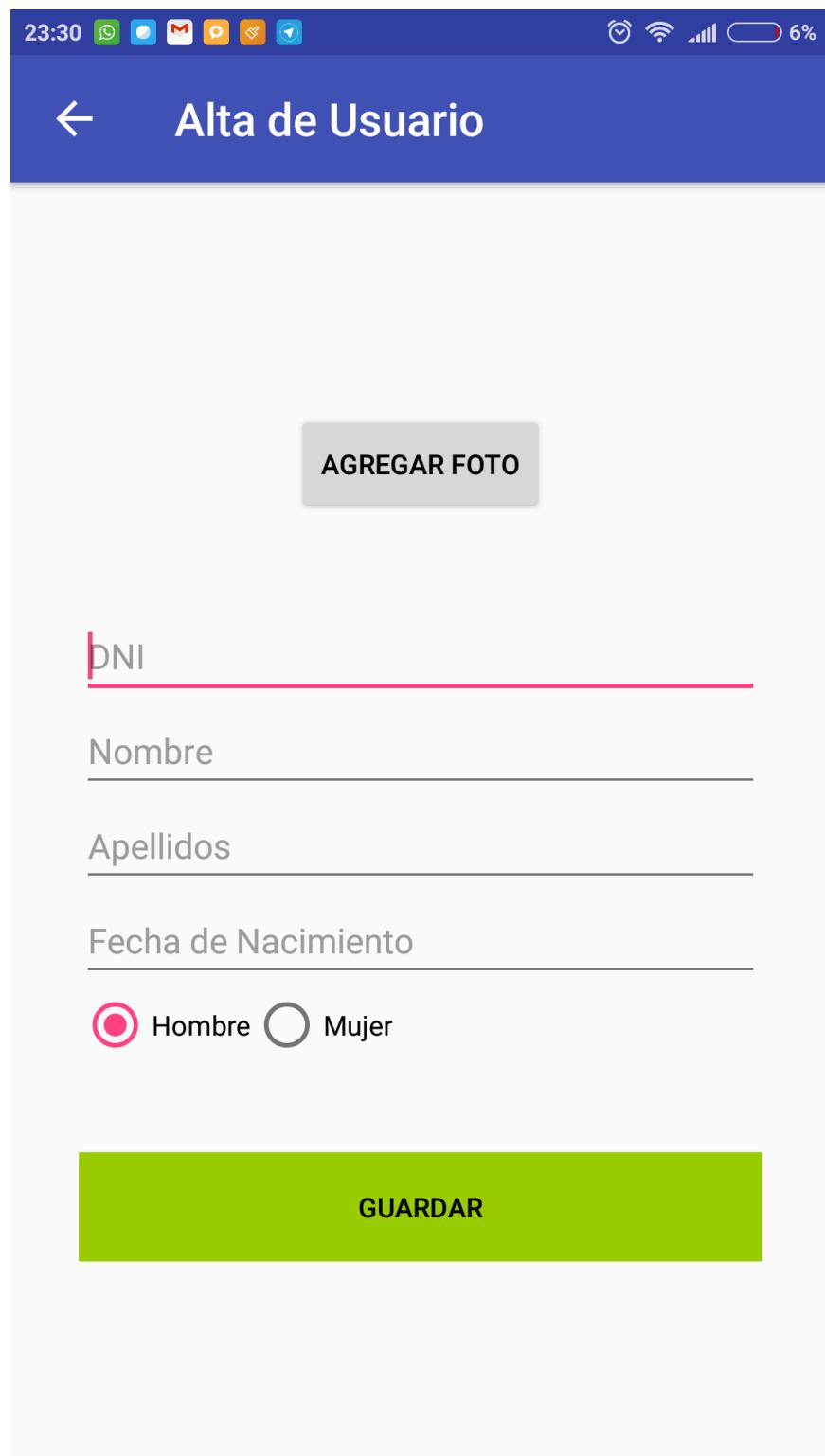


Figura 5.10: Pantalla de alta de persona

5.5.3. Detalle de persona

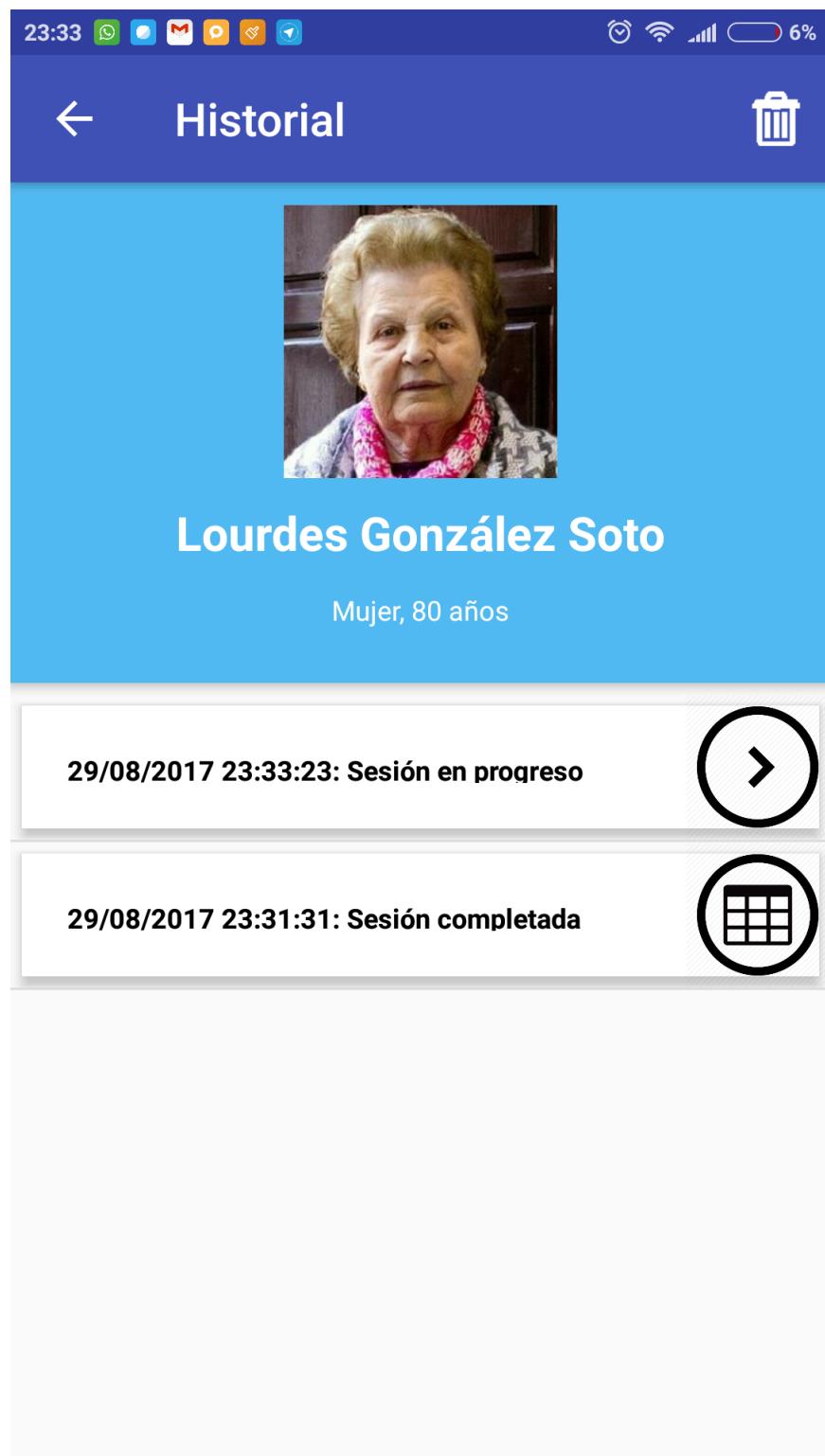


Figura 5.11: Pantalla de detalle de persona

5.5.4. Estadísticas de sesión

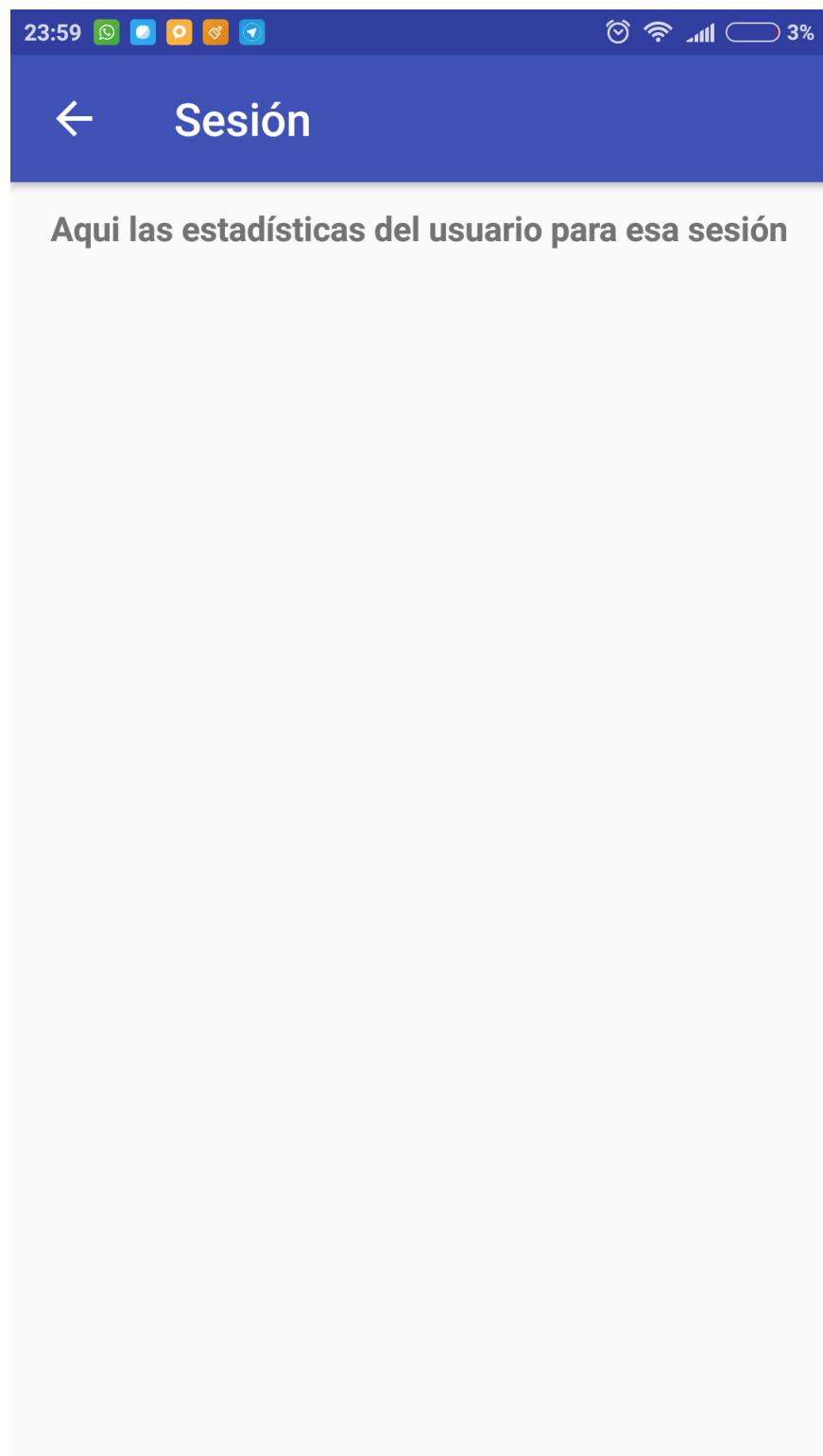


Figura 5.12: Pantalla de estadísticas de sesión

5.5.5. Selección de persona

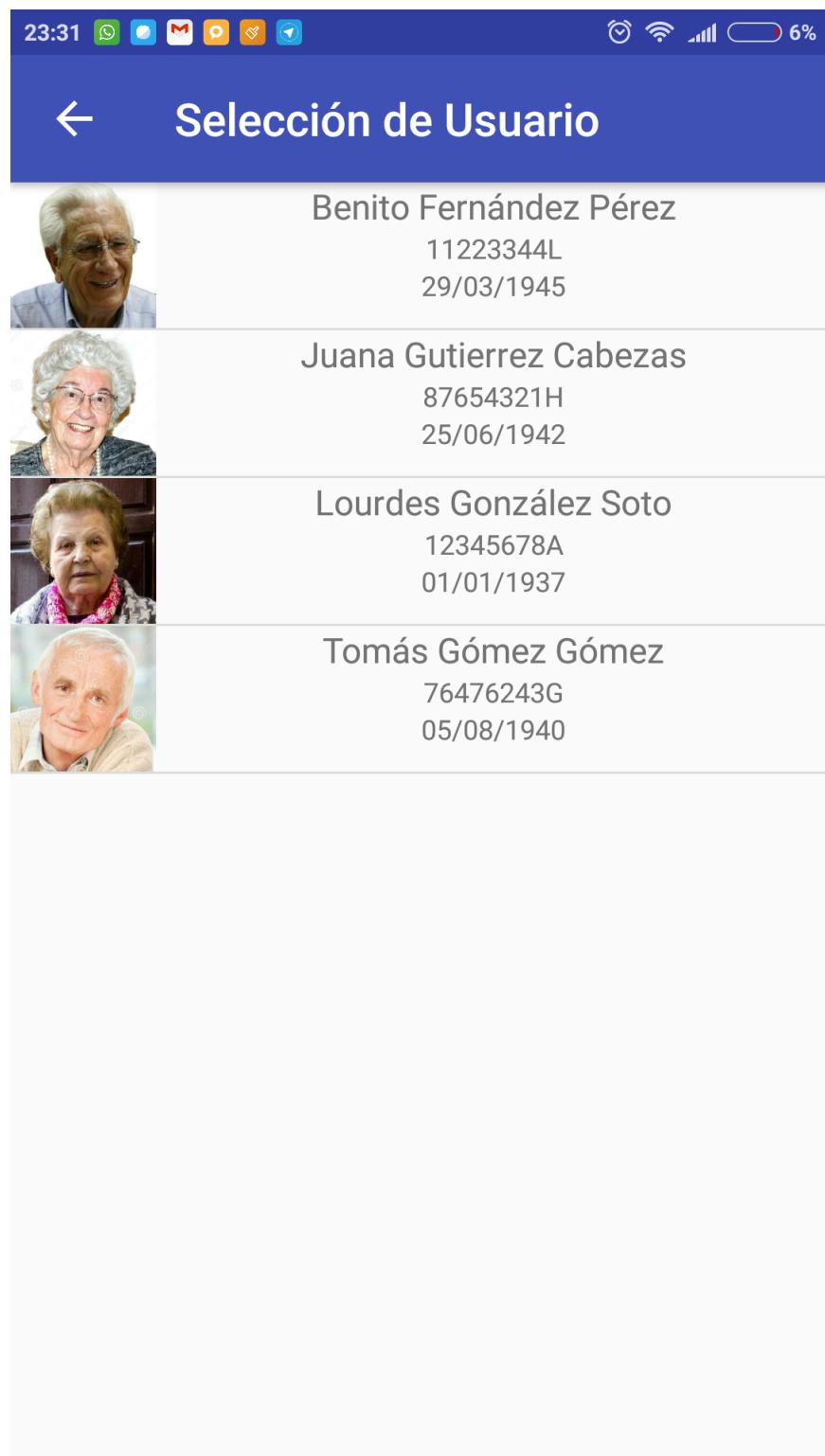


Figura 5.13: Pantalla de selección de persona

5.5.6. Selección de test



Figura 5.14: Pantalla de selección de test

5.5.7. Información de test

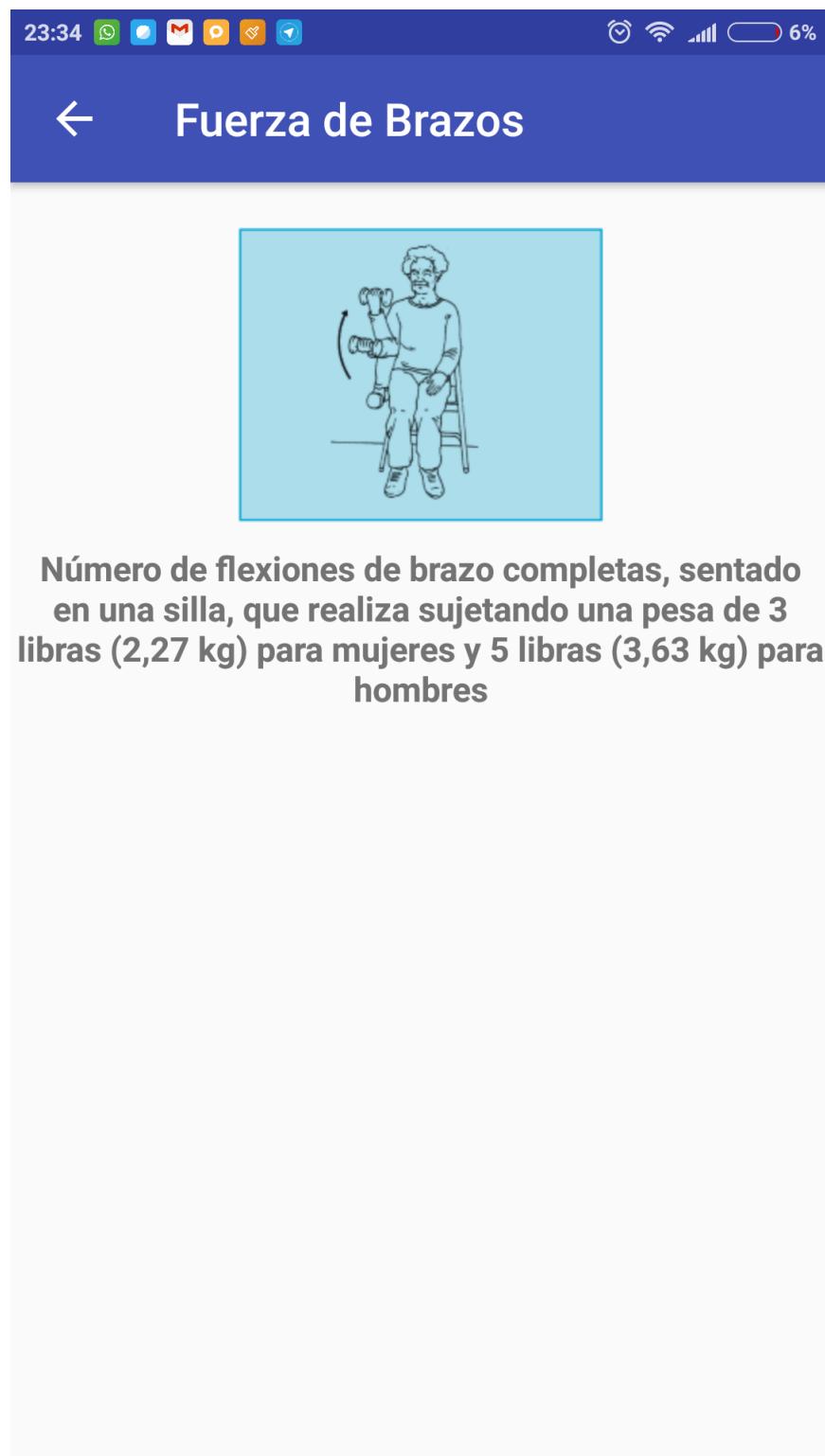


Figura 5.15: Pantalla de información de test

5.5.8. Fuerza de piernas



Figura 5.16: Pantalla de test de fuerza de piernas

5.5.9. Fuerza de brazos



Figura 5.17: Pantalla de test de fuerza de brazos

5.5.10. Resistencia Aeróbica



Figura 5.18: Pantalla de test de resistencia aeróbica

5.5.11. Flexibilidad de piernas



Figura 5.19: Pantalla de test de flexibilidad de piernas

5.5.12. Flexibilidad de brazos



Figura 5.20: Pantalla de test de flexibilidad de brazos

5.5.13. Agilidad

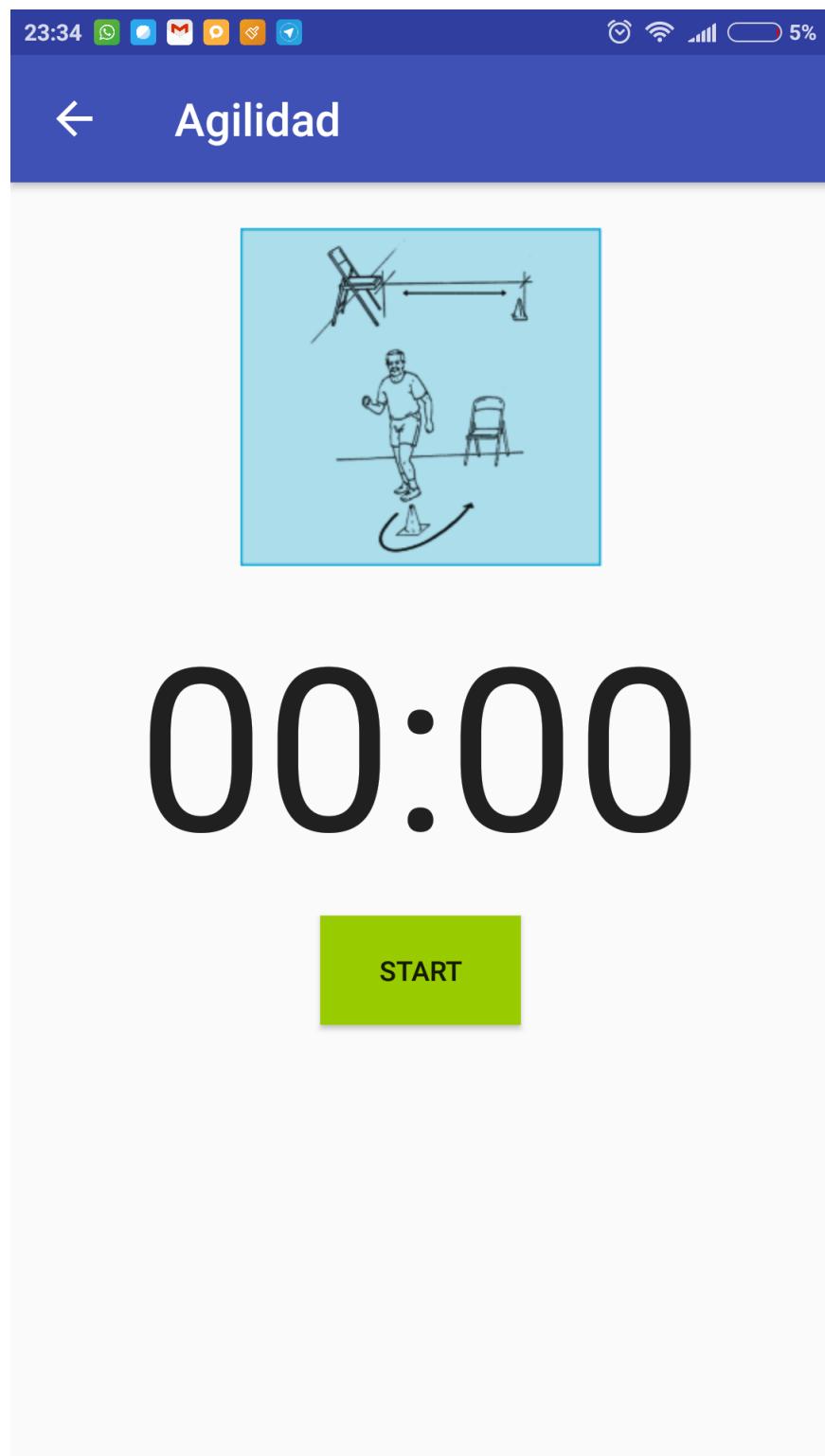


Figura 5.21: Pantalla de test de agilidad

5.5.14. Resultados

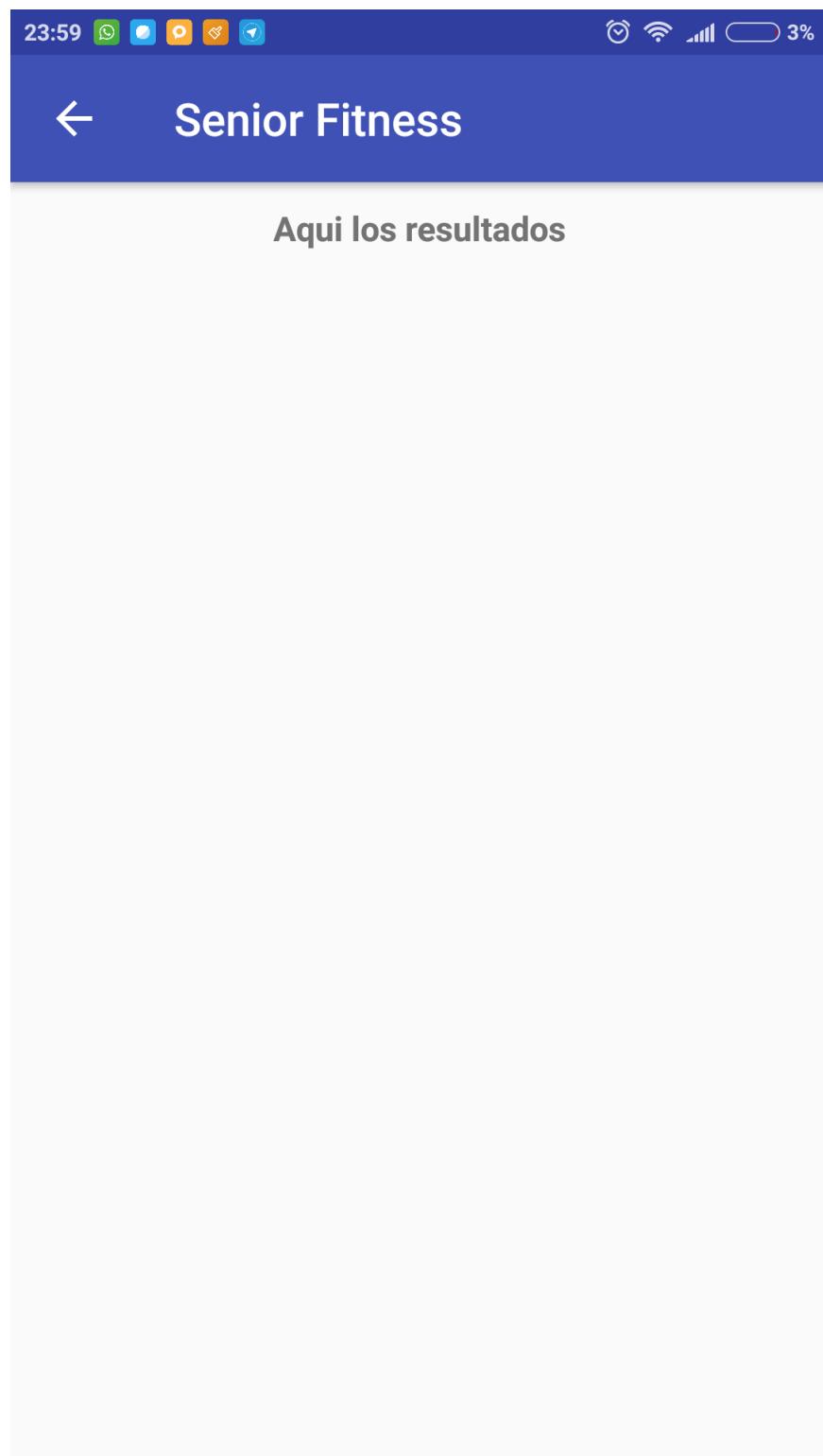


Figura 5.22: Pantalla de resultados

5.6. Uso del dispositivo móvil en la realización de ejercicios

En este punto se definirá cómo debe usarse el dispositivo móvil en aquellos tests de la batería de pruebas Senior Fitness Test en los que podremos usar el dispositivo para que, con ayuda de los diferentes sensores de los que dispone, detecte el movimiento realizado durante los ejercicios, contabilizando de forma autónoma aquellas repeticiones que se realicen de forma satisfactoria.

No será hasta el próximo capítulo (Implementación) donde se entre en detalles sobre cómo el dispositivo móvil hace uso de los sensores de movimiento para realizar los cálculos que determinarán si debe contabilizarse una repetición o no.

De las seis pruebas que componen la batería Senior Fitness Test, solo tres de ellas harán uso de los sensores de movimiento del dispositivo:

- Fuerza de piernas.
- Fuerza de brazos.
- Resistencia aeróbica.

5.6.1. Fuerza de piernas (F_Pna)

El dispositivo móvil debe ser capaz de contabilizar las repeticiones durante la realización del ejercicio estando éste colocado sobre el muslo de la persona que realiza la prueba. Quedará colocado con la pantalla hacia arriba y de forma que sea legible por la persona cuando ésta se encuentra en el punto de partida del ejercicio, esto es, estando sentada. En la siguiente imagen se ilustra la colocación del dispositivo móvil durante la realización del test:



Figura 5.23: Colocación del dispositivo móvil para el test Fuerza de piernas

5.6.2. Fuerza de brazos (F_Br)

El dispositivo móvil debe ser capaz de contabilizar las repeticiones durante la realización del ejercicio estando éste colocado sobre el antebrazo de la persona que realiza la prueba. Quedará colocado con la pantalla hacia arriba y de forma que sea legible por la persona cuando ésta se encuentra en el punto de

partida del ejercicio, esto es, con el brazo extendido. En la siguiente imagen se ilustra la colocación del dispositivo móvil durante la realización del test:



Figura 5.24: Colocación del dispositivo móvil para el test Fuerza de brazos

5.6.3. Resistencia aeróbica (Resist)

Al igual que para el test Fuerza de piernas, el dispositivo móvil debe ser capaz de contabilizar las repeticiones durante la realización del ejercicio estando éste colocado sobre el muslo de la persona que realiza la prueba, quedando colocado con la pantalla hacia arriba y de forma que sea legible por la persona cuando ésta se encuentra en el punto de partida del ejercicio (estando de pie en este caso), tal y como puede observarse en la figura 5.23.

Cabe destacar que, aunque el dispositivo quede colocado de igual forma que para el ejercicio Fuerza de piernas tal y como ya hemos comentado, diferirá la lectura que el dispositivo móvil tendrá que realizar sobre el movimiento ejecutado, dado que el sentido del movimiento (posición inicial y posición final) no será el mismo para ambos tests.

5.7. Diseño de clases UML

En esta sección se va a mostrar el diseño empleado para el diagrama de clases UML del sistema, que mostrará el diseño de las clases que componen el sistema con sus atributos y métodos necesarios para cumplir con los requisitos y funcionalidades del proyecto expuestas en la etapa de análisis.

Dada la naturaleza de las aplicaciones Android, la mayor parte de las clases empleadas en el desarrollo de la aplicación serán activities, y cada una de ellas contendrá la lógica específica de una pantalla de la aplicación.

De entre todas las pantallas de la aplicación, hay seis que se corresponden con los tests o pruebas que componen la batería de pruebas Senior Fitness Test (una pantalla para cada test). Al tratarse de pantallas con una finalidad y objetivo similar, hay funcionalidades que comparten entre ellas, por lo que se ha optado por la creación de una activity padre llamada `ExerciseActivity` de la que extenderán las activities asociadas a cada uno de los tests: fuerza de brazos, fuerza de piernas, resistencia aeróbica, agilidad, flexibilidad de piernas y flexibilidad de brazos, simplificando de esta forma drásticamente el desarrollo de los tests que se vayan implementando sucesivamente tras el primero.

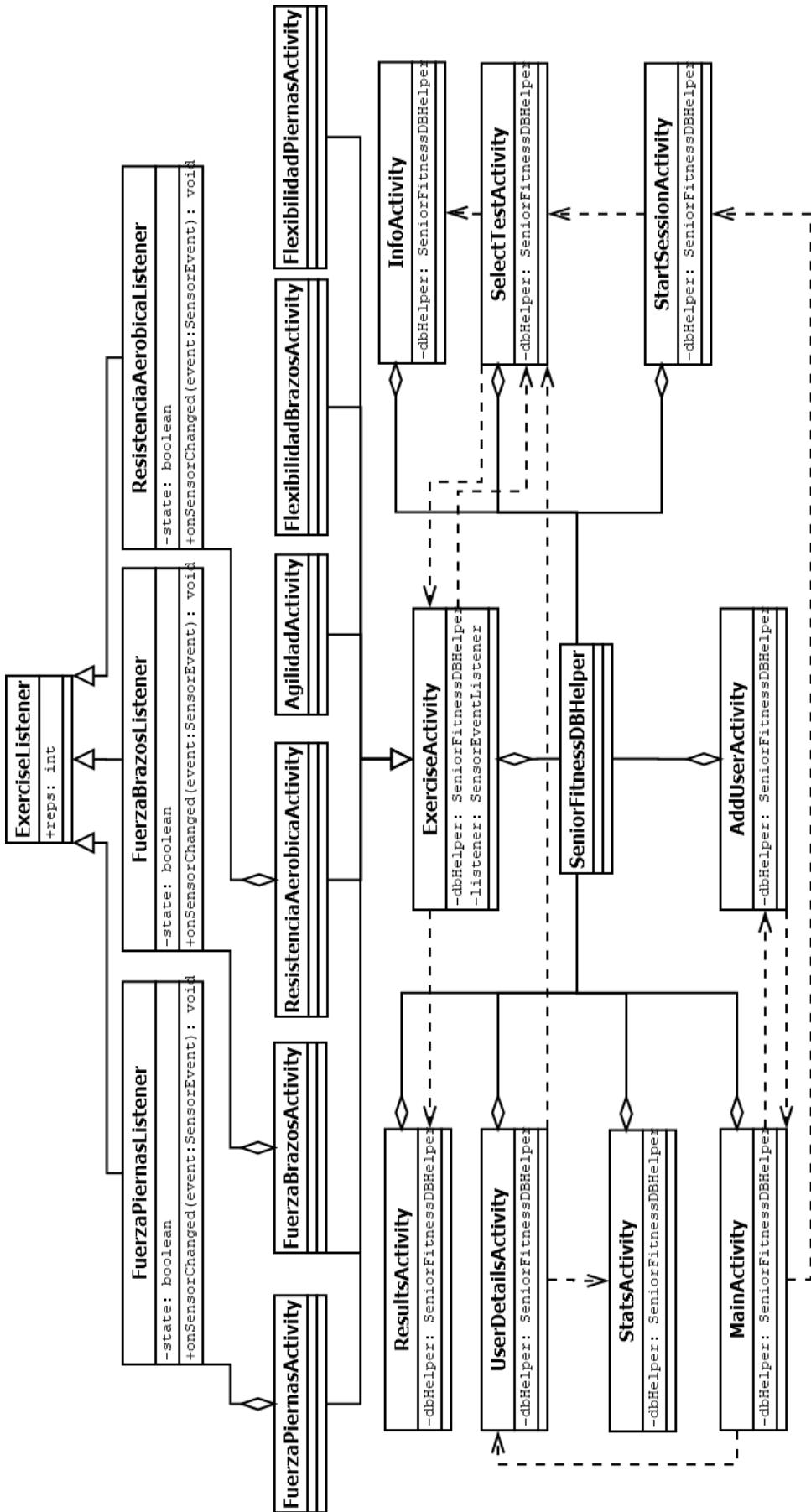


Figura 5.25: Diagrama UML con el diseño de clases de la aplicación

5.8. Diseño de componentes

Para el diseño de componentes se realizarán los diseños de los diagramas de secuencia que muestran el flujo de información existente entre el usuario y los componentes del sistema.

5.8.1. Diagramas de secuencia

Como es sabido de los diagramas de secuencia, éstos representan los flujos de comunicación que se dan para un determinado caso de uso del sistema, por lo que, dados los casos de uso desarrollados en el capítulo de análisis de requisitos para la aplicación que nos ocupa, se procederá a diseñarlos en base a estos diagramas.

Cabe destacar que no todos los posibles diagramas de secuencia aparecerán, sino que nos centraremos en aquellos casos de uso de la aplicación que se consideren más relevantes, de forma que solo se mostrarán aquellos que impliquen algún tipo de cambio en el sistema.

Diagrama de secuencia de registro de persona

El diagrama de secuencia para el registro o alta de persona en la aplicación constituye el flujo de comunicación que comienza una vez el usuario pulsa sobre el botón de registro de persona, que lleva a la pantalla donde se encuentra el formulario de datos de registro. En esta página, el usuario rellena los datos y seguidamente pulsa el botón Guardar.

Al hacer esto, se ejecutará la función de la actividad en la que nos encontramos encargada de validar los datos introducidos por el usuario. Seguidamente, consultará en la base de datos del sistema si el usuario ya existe en el sistema.

Tras la respuesta obtenida, se procede con la inserción de los datos en la base de datos y finalmente se redirige al usuario a la pantalla principal de la aplicación.

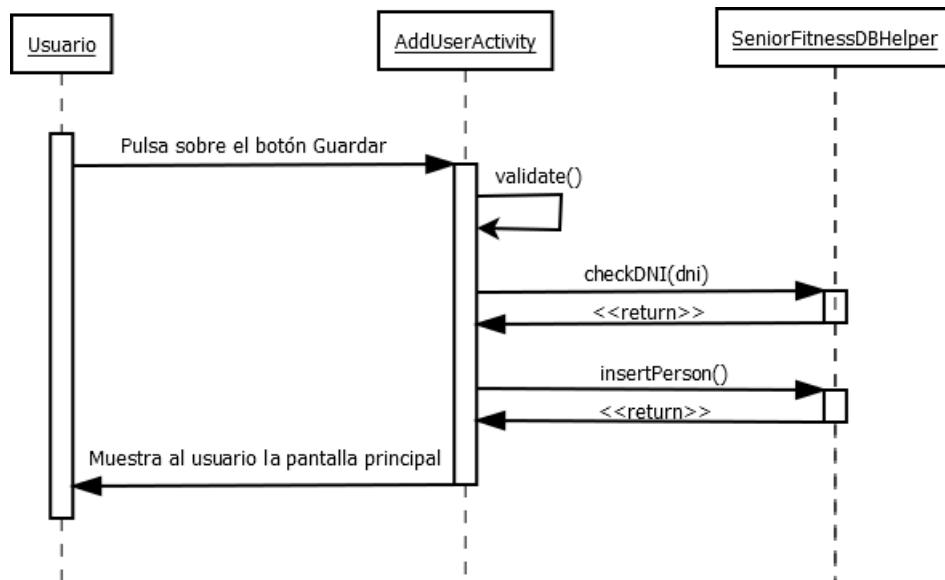


Figura 5.26: Diagrama de secuencia de registro de persona

Diagrama de secuencia de comienzo de sesión

El diagrama de secuencia para el comienzo de sesión en la aplicación constituye el flujo de comunicación que comienza una vez el usuario pulsa sobre el botón correspondiente al ícono de las pesas, que se encuentra en la pantalla principal y que lleva a la pantalla de selección de persona.

Cuando se accede a la pantalla de selección de persona, el sistema hace una consulta a la base de datos para recuperar el listado de personas que están dadas de alta en la aplicación, y se mostrarán al usuario para que se proceda con la selección de la persona para la cual se creará la sesión en el siguiente paso.

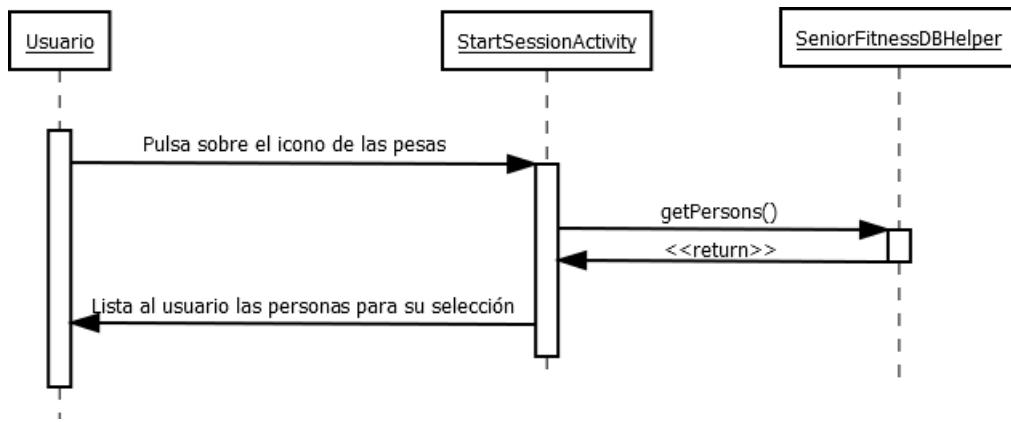


Figura 5.27: Diagrama de secuencia de registro de comienzo de sesión

Diagrama de secuencia de selección de la persona que comenzará la sesión

Partiendo de la base del diagrama de secuencia anterior, en el cual se ha obtenido el listado de personas dadas de alta en el sistema y se han mostrado al usuario, el siguiente paso consistirá en la selección de la persona para la cual se creará la sesión en el sistema.

El sistema asociará a la persona una nueva sesión en caso de que no tuviese ninguna otra en progreso. En el caso de que el usuario ya tuviese una sesión en progreso, se continuará con la misma, mostrando la pantalla de selección de tests al usuario con el progreso y los resultados obtenidos hasta el momento.

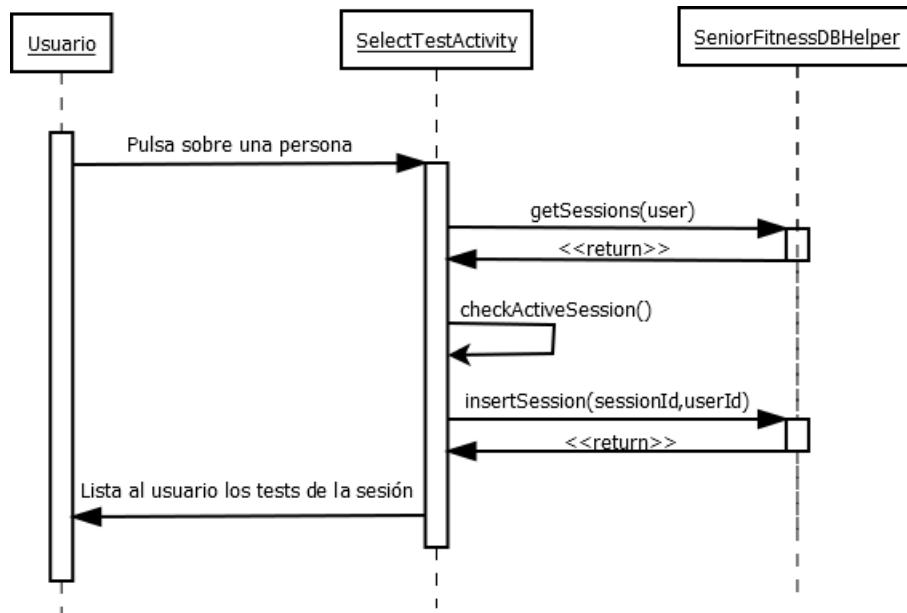


Figura 5.28: Diagrama de secuencia de selección de la persona que comenzará la sesión

Diagrama de secuencia de detección de movimiento

El diagrama de secuencia para la detección de movimiento constituye el flujo de comunicación que comienza una vez el usuario pulsa sobre el botón START de cualquiera de los siguientes tests de la batería de pruebas Senior Fitness Test:

- Fuerza de piernas.
- Fuerza de brazos.
- Resistencia aeróbica.

En ese momento en el que la persona ya puede comenzar a hacer los movimientos requeridos por el ejercicio, la aplicación será capaz de ir contabilizando las repeticiones hechas de forma satisfactoria.

Cuando se pulsa sobre el botón START, se activará el listener que permanece a la escucha de los sensores del dispositivo móvil, capturando el movimiento y haciendo los cálculos necesarios para determinar si el ejercicio se ha realizado correctamente o no. También se irá actualizando en pantalla el número de repeticiones contabilizadas, y se emitirá un sonido cada vez que se realice un incremento de las mismas.

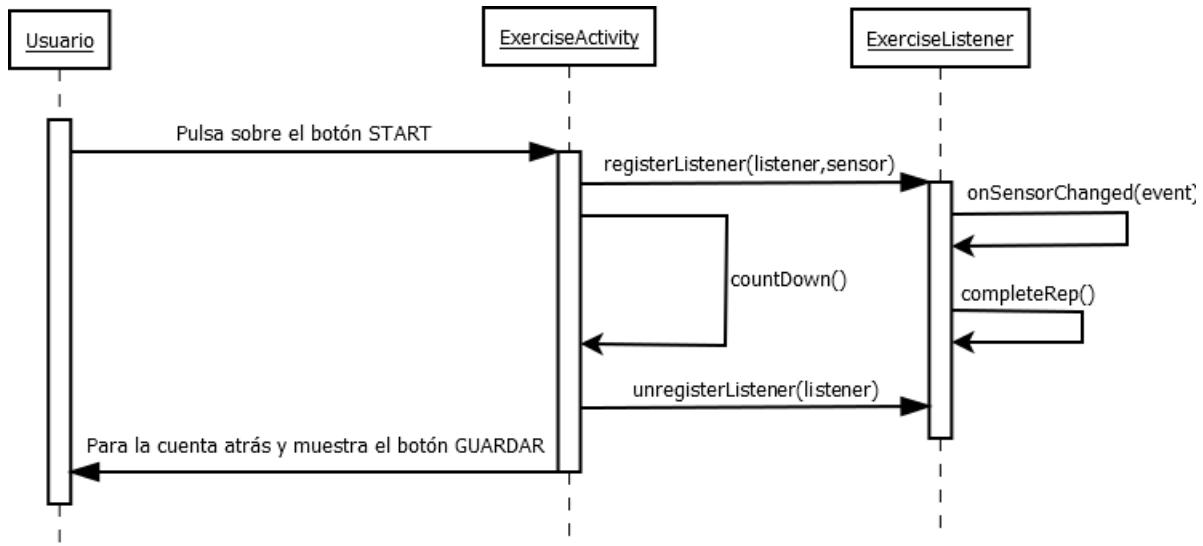


Figura 5.29: Diagrama de secuencia de detección de movimiento

Diagrama de secuencia de guardado de resultado

El diagrama de secuencia para el guardado de resultado constituye el flujo de comunicación que comienza una vez el usuario pulsa sobre el botón GUARDAR, que aparece en las pantallas de test una vez se ha finalizado la realización del mismo o se ha completado la información requerida.

Al hacer esto, se insertará en el sistema el resultado asociado al test, a la sesión y al usuario que acaba de completarlo y, seguidamente, llevará al usuario a la pantalla de selección de test, donde se mostrará el estado actualizado de la prueba que se acaba de completar junto al resultado guardado.

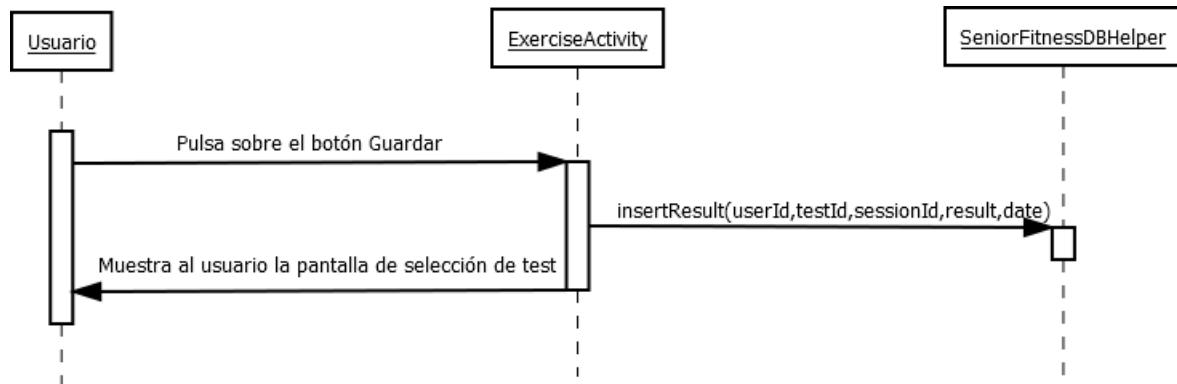


Figura 5.30: Diagrama de secuencia de guardado de resultado

Capítulo 6

Implementación del sistema

En este capítulo vamos a describir y explicar los detalles de la implementación del sistema software y de la solución propuesta.

6.1. Solución propuesta

Debido a los objetivos y requisitos definidos en las anteriores secciones del documento, ya desde un primer momento se antojaba necesario realizar la implementación de una aplicación móvil, por lo que disponíamos de dos opciones a considerar: hacer una aplicación Android o IOS.

Bajo esta disyuntiva, los factores que nos llevaron a tomar la decisión de implementar una aplicación móvil Android fueron los siguientes:

- Como bien es sabido, la base de Android es Java. Java es uno de los lenguajes más universales y extendidos en el mundo del desarrollo, así que un grandísimo número de desarrolladores lo encontrarán más amigable. Esto, sumado a que desde 2008 me dedico profesionalmente al desarrollo de aplicaciones web con Java, fue un factor determinante a la hora tomar la decisión.
- Android es un sistema más abierto y "libre" para el desarrollador, permitiéndole tener en una app un control mucho más amplio del terminal que en IOS.
- La disponibilidad de las herramientas para desarrollar la aplicación también influyó en la decisión. Desarrollar una aplicación IOS requiere hacer uso de un ordenador Mac y de un iPhone para el testeo de la aplicación en el hardware real; herramientas que, teniendo en cuenta que necesitamos hacer uso de los sensores hardware del dispositivo, habría sido necesario adquirir para acometer el desarrollo.
- Android copa actualmente el 92 % del mercado español [5], por lo que por estadística podría hacer uso de la aplicación un público más amplio.

Todos estos factores, sumados a la baja curva de aprendizaje necesaria debido principalmente a mi experiencia previa con Java, nos llevaron a seleccionar Android como plataforma de desarrollo.

6.2. Entorno de construcción

Una vez que se ha llegado a la conclusión anterior de implementar la aplicación móvil haciendo uso de Android, se procede a analizar el entorno de construcción que permitirá el desarrollo de todo el sistema.

6.2.1. Android Studio

Para el desarrollo de la aplicación haremos uso del entorno de desarrollo integrado (IDE) Android Studio [1]. Android Studio es el entorno de desarrollo integrado oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA.

Entre las muchas ventajas que nos ofrece podemos contar con una vista ordenada y modular de los archivos que componen nuestro proyecto. La vista mostrada no se corresponde con la forma original de los archivos en disco, sino que está pensada para optimizar el trabajo.

Integrado en nuestro entorno de trabajo, podemos encontrar un sistema para el control de versiones, que incluye soporte para los repositorios más conocidos (Git, Subversion, etc). En nuestro caso se integró con Git, que es el software de control de versiones seleccionado para el presente proyecto, como ya se indicó anteriormente en el documento.

La interfaz gráfica está dividida de forma que se muestra una gran cantidad de información de forma eficiente. Además es totalmente personalizable, permitiendo al usuario mostrar y ocultar secciones según se requiera. Android Studio sigue el contexto de trabajo del usuario, mostrando las ventanas con las herramientas que considera oportunas en cada instante.

A continuación se muestra un ejemplo de interfaz gráfica:

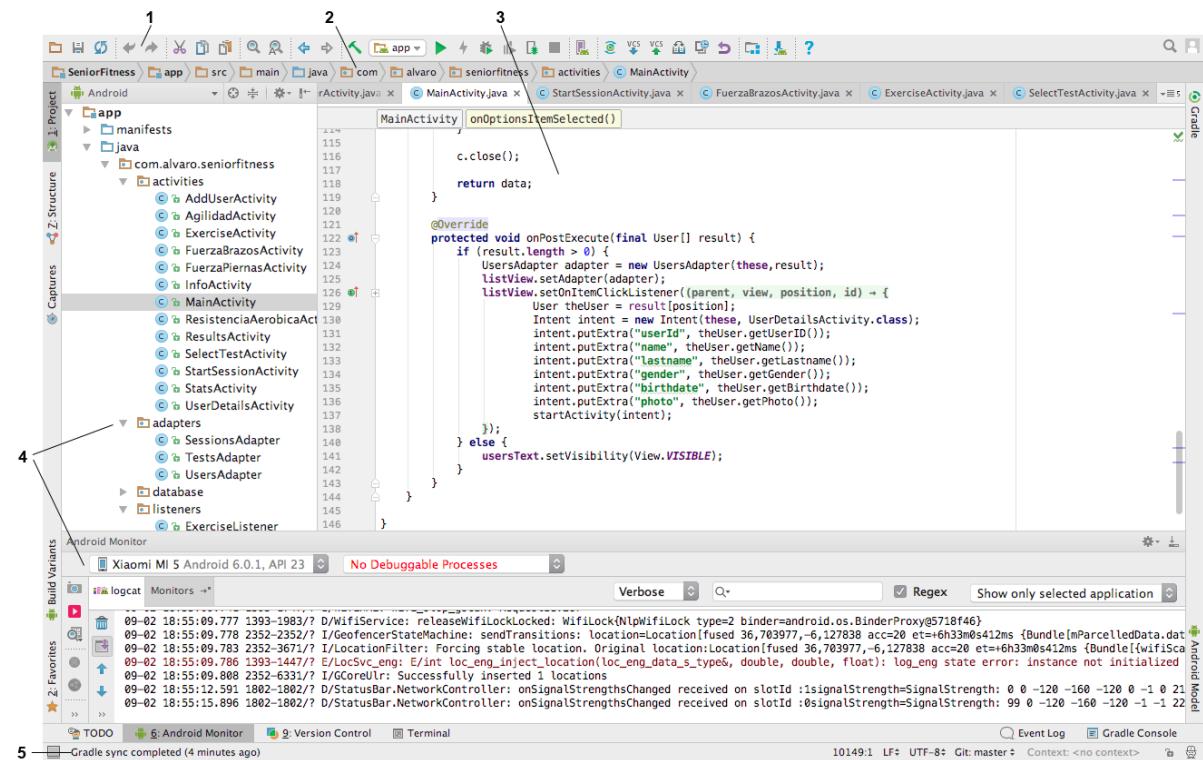


Figura 6.1: Interfaz de Android Studio

1. Barra de herramientas: Permite realizar una amplia gama de acciones: edición, búsqueda, iniciar aplicación, debug, etc.

2. Barra de navegación: Permite navegar entre los archivos abiertos para facilitar su edición.
3. Ventana de edición: Permite modificar los diferentes archivos.
4. Ventana de herramientas: Proporciona acceso a la gestión del proyecto, control de versiones, consola, etc.
5. Barra de estado: Muestra el estado del proyecto, así como mensajes y avisos.

Para finalizar nuestra visión de Android Studio, finalizaremos hablando de Gradle. Gradle es un sistema abierto que automatiza la compilación del código. Utiliza un grafo dirigido acíclico para determinar en qué orden se pueden ejecutar las tareas. Android Studio incorpora esta tecnología para facilitar la reutilización de código permitiendo generar APKs con distintas funcionalidades partiendo de un mismo proyecto.

6.2.2. Android Software Development Kit

Android SDK o Software Development Kit es un conjunto de herramientas para desarrollar, compilar y depurar aplicaciones para el sistema operativo Android, en definitiva, es una API que además de herramientas para el desarrollo, proporciona soporte técnico, ejemplos y una buena documentación.

El SDK incluye un emulador de dispositivos Android, lo que permite probar las aplicaciones de forma rápida y eficiente. Los parámetros de nuestro emulador pueden configurarse de forma que podemos elegir las características que tendrá nuestro dispositivo, como la memoria principal, el tamaño del dispositivo o la versión de Android, lo que permite al programador probar su código en una infinidad de dispositivos sin necesidad de adquirirlos.

6.3. Código fuente

El código fuente desarrollado de la aplicación está alojado en el repositorio GitHub:

<https://github.com/alvarogonzalezalvarez/SeniorFitness>

A continuación se muestra una lista con las diferentes clases Java implementadas durante la realización de la aplicación. Junto a cada una de las clases habrá una pequeña descripción sobre la labor que desempeña cada una.

6.3.1. Activities

Una Activity es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción. A cada actividad se le asigna una ventana en la que se pinta la interfaz de usuario asociada. A continuación se listan las activities que se han implementado:

- **AddUserActivity:** Activity que contiene la lógica de negocio referente a la pantalla de registro/alta de personas.
- **AgilidadActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Agilidad de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **ExerciseActivity:** Activity que contiene toda la lógica de negocio común a todos los tests de la batería de pruebas Senior Fitness Test. Provee los métodos y funciones listas para su uso o sobrescritura por cada uno de los tests.

- **FlexibilidadBrazosActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Flexibilidad de brazos de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **FlexibilidadPiernasActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Flexibilidad de piernas de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **FuerzaBrazosActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Fuerza de brazos de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **FuerzaPiernasActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Fuerza de piernas de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **InfoActivity:** Activity que contiene la lógica de negocio referente a la pantalla de información de los diferentes tests de la batería de pruebas Senior Fitness Test.
- **MainActivity:** Activity que contiene la lógica de negocio referente a la pantalla principal de la aplicación. Esta actividad es la que se lanza por defecto cuando se abre la aplicación.
- **ResistenciaAerobicaActivity:** Activity que contiene la lógica de negocio referente a la pantalla del test Resistencia aeróbica de la batería de pruebas Senior Fitness Test. Esta actividad extiende a ExerciseActivity.
- **ResultsActivity:** Activity que contiene la lógica de negocio referente a la pantalla de resultados mostrada tras finalizar todos los tests de una sesión.
- **SelectTestActivity:** Activity que contiene la lógica de negocio referente a la pantalla de selección de los diferentes tests que componen la batería de pruebas Senior Fitness Test.
- **StartSessionActivity:** Activity que contiene la lógica de negocio referente a la pantalla de selección de la persona que comenzará la sesión de ejercicios.
- **StatsActivity:** Activity que contiene la lógica de negocio referente a la pantalla de estadísticas de una sesión concreta ya completada por la persona para la cual se está visualizando el detalle.
- **UserDetailsActivity:** Activity que contiene la lógica de negocio referente a la pantalla de detalle/historial de usuario.

6.3.2. Listeners

Los listeners o escuchadores son clases usadas para recibir notificaciones procedentes del SensorManager (clase que nos permite acceder a los sensores del dispositivo). A continuación listamos los listeners que se han implementado.

- **ExerciseListener:** Listener que implementa la clase SensorEventListener. Cada vez que hay un evento del sensor usado se invoca el método onSensorChanged, que será el método donde se harán los cálculos necesarios para detectar si el ejercicio se ha realizado de forma satisfactoria. Éste método debe ser implementado por los listeners que extienden de ExerciseListener.
- **FuerzaBrazosListener:** Listener encargado de detectar y leer los movimientos realizados durante la realización del test Fuerza de brazos de la batería de pruebas Senior Fitness Test. Este listener extiende a ExerciseListener.

- **FuerzaPiernasListener:** Listener encargado de detectar y leer los movimientos realizados durante la realización del test Fuerza de piernas de la batería de pruebas Senior Fitness Test. Este listener extiende a ExerciseListener.
- **ResistenciaAerobicaListener:** Listener encargado de detectar y leer los movimientos realizados durante la realización del test Resistencia aeróbica de la batería de pruebas Senior Fitness Test. Este listener extiende a ExerciseListener.

6.3.3. Base de Datos

A continuación se listarán las clases encargadas de la definición, gestión y uso de la base de datos:

- **SeniorFitnessContract:** Clase java donde se definen las tablas usadas y los nombres y tipos de los campos de cada una de las mismas.
- **SeniorFitnessDBHelper:** Clase java que extiende a SQLiteOpenHelper. Se encarga de gestionar la creación y manipulación de la base de datos, así como del versionado de la misma.

6.3.4. Modelo

Se listarán a continuación las clases Java que representan el modelo de la aplicación:

- **Result:** Clase java que representa los resultados obtenidos, asociado a un test, a una sesión y a una persona.
- **Session:** Clase java que representa a las sesiones de tests que realizan las personas dadas de alta en la aplicación.
- **Test:** Clase java que representa a los diferentes tests que componen la batería de pruebas Senior Fitness Test.
- **User:** Clase java que representa a las personas dadas de alta en la aplicación.

6.3.5. Adapters

Un adapter es un objeto que funciona como puente entre un AdapterView y los correspondientes datos para esa vista, proporcionando acceso a los datos del elemento. También es el encargado de construir una vista (View) por cada elemento del conjunto de datos. A continuación listamos los adapters implementados para la aplicación:

- **SessionsAdapter:** Adapter utilizado para convertir en una View independiente cada elemento contenido en el listado de sesiones recuperadas para una persona concreta y mostradas en la pantalla de detalle de persona.
- **TestsAdapter:** Adapter utilizado para convertir en una View independiente cada elemento contenido en el listado de tests mostrados en la pantalla de selección de test.
- **UsersAdapter:** Adapter utilizado para convertir en una View independiente cada elemento contenido en el listado de personas dadas de alta en la aplicación, y que se muestran en la pantalla principal y en la pantalla de selección de persona.

6.4. Detección de movimiento

Una de las características principales de la aplicación es la capacidad de leer y detectar los movimientos realizados durante la realización de determinados tests. A continuación entraremos en los detalles sobre cómo se ha implementado el cálculo y la lectura de los parámetros necesarios para determinar si las repeticiones del ejercicio se han realizado de forma satisfactoria o no, para lo cual se ha hecho uso del sensor de gravedad.

6.4.1. Sensor de gravedad

El sensor de gravedad mide el efecto de la aceleración de la gravedad de la tierra sobre el dispositivo. No se trata de un sensor hardware, sino que es un sensor software o virtual derivado del acelerómetro, donde otros sensores hardware como el giroscopio ayudan a eliminar la aceleración lineal de los datos. La gravedad se mide en m/s^2 como el caso del acelerómetro, y se mide sobre los ejes X, Y y Z.

Supongamos que tenemos el dispositivo móvil colocado sobre una mesa, tumbado sobre su parte trasera y de forma totalmente paralela al suelo y con la pantalla mirando hacia el techo; en este caso la gravedad estaría ejerciendo una aceleración de $9,8 m/s^2$ sobre la pantalla del dispositivo de forma perpendicular a la misma, o lo que es lo mismo, estaría ejerciendo una aceleración de $9,8 m/s^2$ sobre el eje Z tal y como podemos observar en el siguiente gráfico:

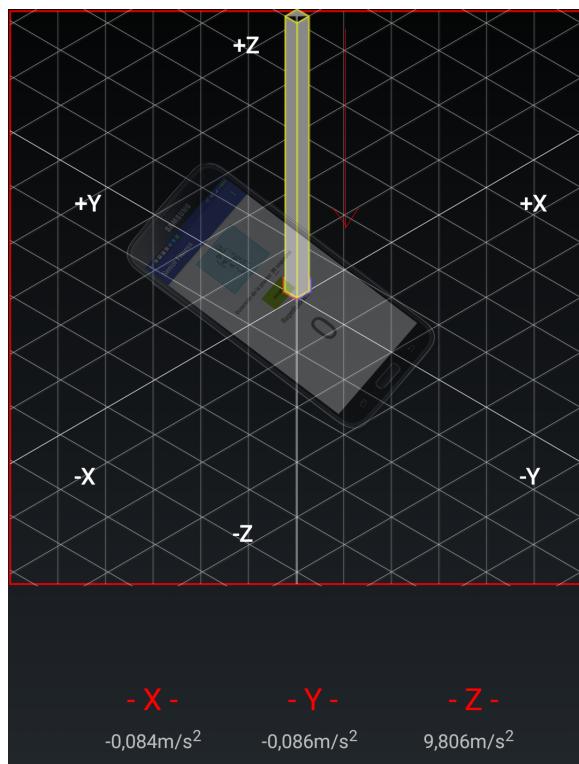


Figura 6.2: Efecto de la gravedad sobre el eje Z con el dispositivo colocado de forma paralela sobre el suelo y con la pantalla hacia arriba

Partiendo de este gráfico podemos definir cada eje como sigue:

- **Eje Z:** Es la línea que atraviesa el dispositivo móvil de forma perpendicular a su pantalla.

- **Eje Y:** Es la línea paralela a la pantalla que atraviesa el dispositivo desde su parte superior hasta la inferior o viceversa.
- **Eje X:** Es la línea paralela a la pantalla que atraviesa el dispositivo desde su lado izquierdo hasta el derecho o viceversa.

Por lo que, teniendo en cuenta la situación de los ejes respecto al dispositivo, podríamos decir entonces que si sostenemos el móvil de forma perpendicular al suelo, como por ejemplo en el supuesto de que estuviésemos de pie realizando una llamada telefónica y tuviésemos el altavoz sobre la oreja, la gravedad estaría ejerciendo sobre el eje Y del dispositivo (es decir, atravesándolo desde la parte superior hasta la inferior), por lo que el gráfico mientras sujetamos el dispositivo de la forma comentada sería el siguiente:

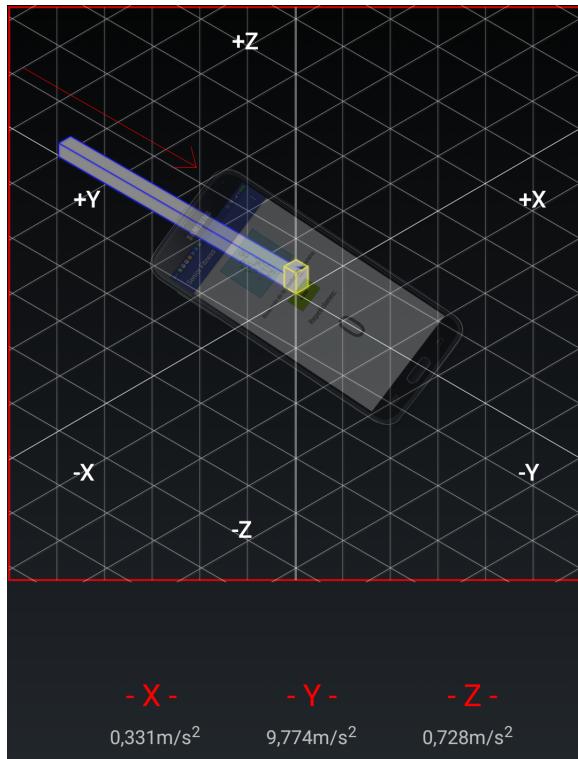


Figura 6.3: Efecto de la gravedad sobre el eje Y con el dispositivo colocado de forma perpendicular (de pie) sobre el suelo

Una vez hemos entendido como se mide la gravedad sobre cada uno de los ejes del dispositivo móvil, podemos hacer uso de las medidas obtenidas sobre cada eje para determinar la posición en la que se encuentra el móvil conforme se va realizando el movimiento de cada ejercicio.

6.4.2. Lectura para el test Fuerza de brazos

Teniendo en cuenta que el dispositivo móvil debe colocarse sobre el antebrazo tal y como se ilustró en la figura 5.24, sabremos que la aceleración ejercida sobre el dispositivo en la posición inicial del ejercicio será negativa sobre el eje Y (dado que la persona tendrá el brazo totalmente extendido con el puño hacia abajo, quedando el dispositivo bocabajo) tal y como puede observarse en el estado de la siguiente gráfica mientras el dispositivo se encuentra en la posición comentada:

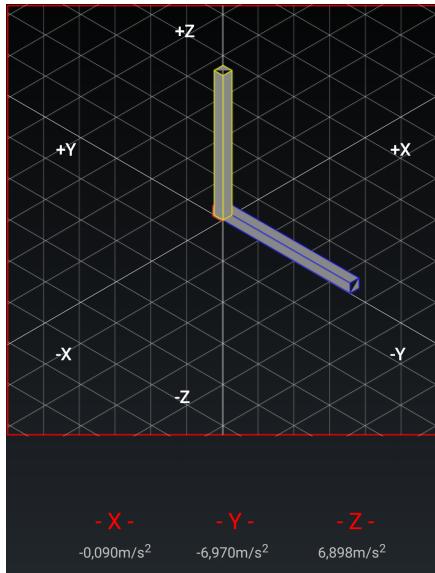


Figura 6.4: Fuerza de brazos: Efecto de la gravedad sobre el eje Y en la posición inicial

Por tanto, en la posición final del ejercicio (cuando el brazo queda flexionado con el puño hacia arriba) la gravedad ejercida sobre el eje Y será positiva:

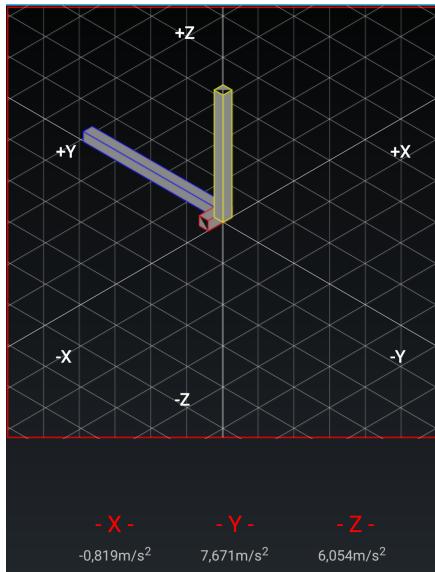


Figura 6.5: Fuerza de brazos: Efecto de la gravedad sobre el eje Y en la posición final

De esta forma, sabremos que la repetición se ha completado de forma satisfactoria cuando la gravedad ejercida en el eje Y pasa de ser negativa, a ser positiva y mayor que $7,5 \text{ m/s}^2$ (para asegurarnos de que se ha flexionado el brazo lo suficiente) en la posición final.

6.4.3. Lectura para el test Fuerza de piernas

Teniendo en cuenta que el dispositivo móvil debe colocarse sobre la pierna tal y como se ilustró en la figura 5.23, sabremos que la aceleración ejercida sobre el dispositivo en la posición inicial del ejercicio

será normalmente cercano a 0 m/s^2 (dos unidades arriba o abajo) sobre el eje Y (dado que el punto de partida del ejercicio es estando sentado, y por tanto el dispositivo móvil queda prácticamente paralelo al suelo) tal y como puede observarse en el estado de la siguiente gráfica mientras el dispositivo se encuentra en la posición comentada:

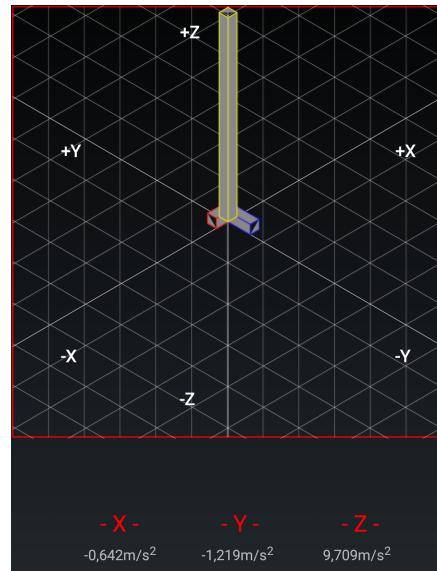


Figura 6.6: Fuerza de piernas: Efecto de la gravedad sobre el eje Y en la posición inicial

Sin embargo, en la posición final del ejercicio, al tenerse que levantar la persona de la silla y quedar prácticamente de pie (y por tanto quedar el dispositivo móvil bocabajo), la gravedad ejercida sobre el eje Y será negativa y se acercará a los $-9,8 \text{ m/s}^2$, tal y como se puede ver en la siguiente gráfica:

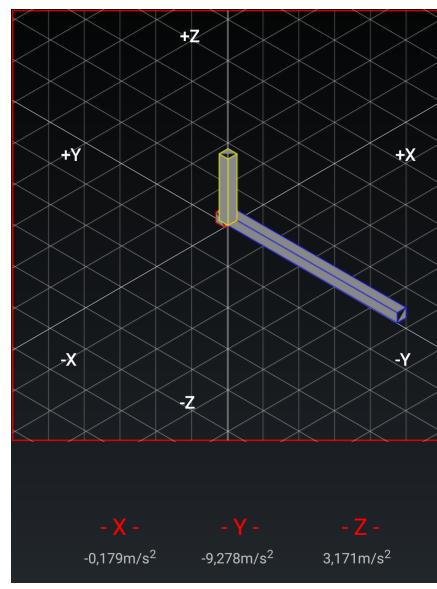


Figura 6.7: Fuerza de piernas: Efecto de la gravedad sobre el eje Y en la posición final

Así, sabremos que la repetición se ha completado de forma satisfactoria cuando la gravedad ejercida en el eje Y pasa de estar rondando el 0, a ser negativa y menor que $-8,5 \text{ m/s}^2$ en la posición final.

6.4.4. Lectura para el test Resistencia aeróbica

La lectura para este caso es exactamente la misma que para el test de Fuerza de piernas, pero de forma inversa; en este ejercicio la posición inicial será estando la persona de pie con el dispositivo móvil colocado en la pierna tal y como se ilustra en la figura 5.23, por lo que en este caso la gráfica sería la siguiente en la posición inicial:

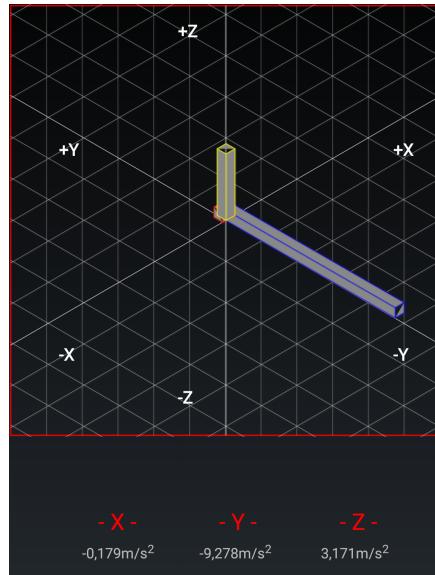


Figura 6.8: Resistencia aeróbica: Efecto de la gravedad sobre el eje Y en la posición inicial

Por tanto, para la posición final, al quedar la pierna flexionada con la rodilla levantada hasta una altura equivalente al punto medio entre la rótula y la cresta ilíaca, el dispositivo móvil quedará prácticamente paralelo al suelo, por lo que la gravedad ejercida sobre el eje Y estará rondando el 0, tal y como se puede observar a continuación:

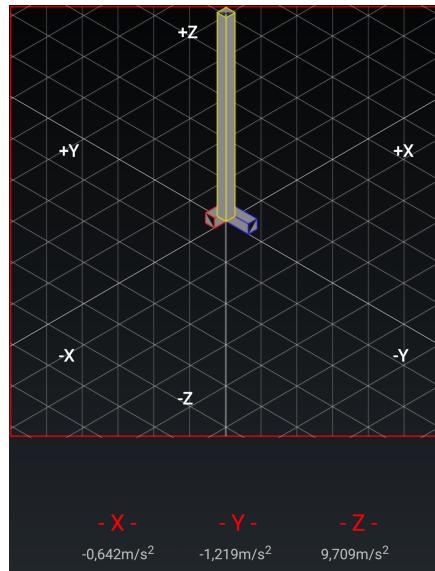


Figura 6.9: Resistencia aeróbica: Efecto de la gravedad sobre el eje Y en la posición final

De esta forma, sabremos que la repetición se ha completado de forma satisfactoria cuando la gravedad ejercida en el eje Y pasa de ser negativa y menor a $-8,5 \text{ m/s}^2$, a ser mayor que $-4,5 \text{ m/s}^2$ en la posición final (para asegurarnos de que se ha levantado la rodilla lo suficiente).

Capítulo 7

Pruebas y validaciones

En este capítulo se presenta el plan de pruebas del sistema, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales o con ayuda de algún software específico de pruebas para la depuración de la aplicación.

7.1. Estrategia

La estrategia que se siga en el devenir de la ejecución de pruebas sobre el sistema abarcará tanto al alcance de éstas como el objetivo, interpretación y evaluación de los resultados que éstas reflejen.

Para el proyecto se han realizado pruebas unitarias, de integración, funcionales y no funcionales. Las dos primeras se basan en las pruebas empleadas de manera automática, mientras que las otras dos se realizan atendiendo a los requerimientos funcionales y no funcionales especificados para el proyecto.

7.2. Entorno de pruebas

En esta sección se procede a especificar las características de los dispositivos empleados para la realización de las pruebas. La aplicación se ha ido depurando haciendo uso del mismo ordenador portátil que se ha empleado para el desarrollo del sistema y sus características son:

- MacBook Pro (13 pulgadas, mediados 2010)
- Procesador Intel Core 2 Duo, 2,4 GHz
- 4 GB RAM
- HDD 164GB
- Gráfica NVIDIA GeForce 320M 256 MB
- Monitor 13 pulgadas

En cuanto a las características del dispositivo móvil usado para las pruebas funcionales y no funcionales realizadas sobre el hardware real, son las siguientes:

- Xiaomi MI5
- Versión de Android 6.0.1 MXB48T
- Procesador Quad-core Max 1,80 GHz

- 3 GB RAM
- 32 GB de memoria interna
- Pantalla de 5,15 pulgadas

7.3. Pruebas unitarias

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad o módulo de código, de manera que sirva para asegurar que cada módulo funciona por separado antes de integrarlo con el resto de módulos del sistema.

Al dividirse el desarrollo del proyecto en iteraciones como se mencionó en el capítulo 3 Planificación en la sección y se detalló concretamente en la sección 3.4.2. del mismo, se han realizado las pruebas en cada una de las iteraciones antes de continuar con la siguiente.

Por tanto, se realizaron pruebas unitarias en cada módulo desarrollado de forma independiente tras cada iteración haciendo uso del depurador de código que lleva integrado Android Studio, y que nos permite ir haciendo debug a la aplicación línea a línea, comprobando que el comportamiento de la lógica de negocio de cada uno de los módulos era el esperado y que los valores de las variables y devueltos por los diferentes métodos y funciones eran consistentes y ajenos a fallos, corrigiéndolos si estos aparecían.

7.4. Pruebas de integración

Una prueba de integración se basa en la comprobación del correcto funcionamiento de un subsistema completo compuesto por varios módulos que, aunque funcionen correctamente por separado, pueden provocar errores a la hora de integrar unos con otros.

Al igual que para las pruebas unitarias, las pruebas de integración se han realizado a medida que se avanzaron en las iteraciones del proyecto cuando los módulos de cada iteración se integraban con los anteriormente desarrollados, comprobando que no se produjesen por ejemplo errores de dependencia entre clases.

La realización de pruebas de integración se llevó a cabo con las herramientas de debug integradas en Android Studio, tal y como se ha comentado en el punto anterior para el caso de las pruebas unitarias.

7.5. Pruebas funcionales

Las pruebas funcionales se realizan en base a los requerimientos y requisitos establecidos para el desarrollo del proyecto, de modo que será necesario ir comprobando que se cumplen cada una de las funcionalidades especificadas.

Se ha establecido por tanto una serie de pruebas concretas para probar que los resultados que proporcione el sistema son los que se esperan de cada funcionalidad, que son las que pasamos a listar a continuación:

- Alta de persona
- Listar todas las personas dadas de alta en la aplicación y verificación de los datos mostrados
- Visualizar el detalle de la persona

- Eliminar persona
- Mostrar detalles de sesión realizada
- Continuar sesión de ejercicios en progreso
- Iniciar nueva sesión de ejercicios
- Listar los diferentes tests y sus datos
- Seleccionar test para su realización
- Realizar el test Fuerza de piernas
- Realizar el test Fuerza de brazos
- Realizar el test Resistencia aeróbica
- Realizar el test Agilidad
- Introducir la medida obtenida tras la realización del test Flexibilidad de piernas
- Introducir la medida obtenida tras la realización del test Flexibilidad de brazos
- Guardar el resultado obtenido/introducido en los tests
- Visualizar la pantalla de resultados tras completar todos los tests

7.6. Pruebas no funcionales

Las pruebas no funcionales se encargan de verificar que se cumplen los requisitos no funcionales establecidos en el capítulo 4 Análisis de requisitos, en el apartado 4.1.3.

En cuanto al ordenador portátil empleado para el desarrollo de la aplicación y al dispositivo móvil usado para las pruebas en el hardware, se ha verificado que han respondido correctamente a las pruebas realizadas a través de ambos.

Referente al rendimiento del sistema con respecto a la gestión y persistencia de los datos almacenados, se ha verificado que la información se almacena correctamente a través de la aplicación y se recupera sin incidencias, manteniéndose estable en todo momento el comportamiento de la aplicación.

Para finalizar, se realizaron pruebas sobre la aplicación con respecto a la visualización del contenido en dispositivos móviles con pantallas de diferentes tamaños y formatos, corroborando que el diseño responsive empleado en las pantalla de la aplicación Android realizan correctamente su función.

Capítulo 8

Herramientas utilizadas

A continuación se listarán las herramientas software que se han usado durante la realización del proyecto, excluyendo aquellas que ya se nombraron y detallaron en los apartados:

- 5.2. Arquitectura lógica del sistema
- 6.2. Entorno de construcción

8.1. Cliente del sistema de control de versiones

Como ya se comentó anteriormente en el presente documento, el software de control de versiones que se ha usado durante la realización del proyecto ha sido Git, que se integra perfectamente con Android Studio.

Aun así, también se ha hecho uso del cliente GitHub para la manipulación del código fuente alojado en el repositorio, así como para versionar ficheros a los que no se accedían directamente desde Android Studio, como por ejemplo aquellos correspondientes a la memoria.

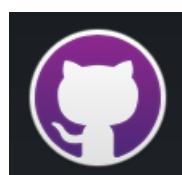


Figura 8.1: Logo de GitHub

8.2. Redacción de la memoria

Para la completa realización de la memoria se ha usado LATEX. Se trata de un sistema de composición de textos, orientado específicamente a la creación de libros, documentos científicos y técnicos que pudiesen contener fórmulas matemáticas.

LATEX

Figura 8.2: Logo de LATEX

LATEX es un sistema de composición de textos que está formado mayoritariamente por órdenes (márcos) construidas a partir de comandos de TEX. LATEX es una herramienta práctica y útil pues, a su facilidad de uso, se une toda la potencia de TEX.

8.3. Gestión del proyecto

Gantt Project es un programa gratuito multiplataforma desarrollado en Java y especializado en la gestión de proyectos.



Figura 8.3: Logo de Gantt Project

Cuenta con la posibilidad de generar fácilmente diagramas de Gantt para la gestión de los tiempos de las tareas de un proyecto, y gestión de recursos usando gráficos de carga de los mismos. Es muy usado gracias a sus numerosas opciones de reportes (MS Project, HTML, PDF, hojas de cálculo, etc).

8.4. Realización de diagramas

Para la realización de todos los diagramas necesarios que aparecen a lo largo de toda la memoria se ha usado el creador de diagramas llamado Dia.



Figura 8.4: Logo de Dia

Dia es un programa de creación de diagramas en GNU/Linux, MacOS X, Unix y Windows, bajo licencia GPL. Puede ser utilizado para dibujar diferentes tipos de diagramas. Actualmente cuenta con herramientas para dibujar diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de red y muchos otros tipos.

8.5. Ilustración y retoque

Para la ilustración y retoque de imágenes se ha usado GIMP. GIMP es un programa libre y gratuito de edición de imágenes digitales en forma de mapa de bits. Forma parte del proyecto GNU y está disponible bajo la licencia pública general de GNU. Su interfaz está disponible en varios idiomas.



Figura 8.5: Logo de Gimp

En una gran cantidad de casos, GIMP es una alternativa sólida, potente y rápida a Photoshop, aunque no ha sido desarrollado como un clon de éste y por tanto posee una interfaz diferente. Con él se han realizado iconos, botones y edición y retoque de algunos de los recursos usados en la aplicación y en la memoria.

Capítulo 9

Manual de instalación

Capítulo 10

Manual de usuario

Capítulo 11

Conclusiones

En este capítulo se detallan los objetivos alcanzados, las enseñanzas obtenidas por parte del desarrollador del proyecto y las posibles mejoras futuras que puedan implementarse en el sistema.

11.1. Objetivos alcanzados

Bibliografía

- [1] Android Developers. Android Studio. https://developer.android.com/studio/intro/index.html#project_structure.
- [2] GIT. Documentation. <https://git-scm.com/documentation>.
- [3] Google. Android Developers. <https://developer.android.com/index.html>.
- [4] INE. Instituto Nacional de Estadística. <http://www.ine.es>.
- [5] Kantar Worldpanel ComTech. Cuota de mercado de smartphones en España. <http://es.kantar.com/tech/móvil/2017/junio-2017-cuota-de-mercado-de-smartphones-en-espa~na-2017/>.
- [6] C. Rikli, R. y Jones. *Senior fitness test manual*. Human Kinetics, 2001.
- [7] SQLite. Documentation. <https://www.sqlite.org/docs.html>.

GNU Free Documentation License

Version 1.3, 3 November 2008
Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with … Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.