

ALGORITMOS DE ADAPTACIÓN DINÁMICA DEL EJERCICIO PARA MEJORA DEL RENDIMIENTO DEPORTIVO



TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE COMPUTADORES

CURSO 2013-1014

Jesús Mariano García-Bravo Fernández

Director

José Luis Ayala Rodrigo

Departamento de Arquitectura de Computadores y Automática

Facultad de Informática

Universidad Complutense de Madrid.

Madrid, 23 de Junio de 2014

Autorización de difusión y utilización

Jesús Mariano García-Bravo Fernández autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales, y mencionando expresamente a su autor, tanto la propia memoria, como el código, la documentación y/o la aplicación desarrollada.

Agradecimientos

A todos mis familiares y amigos que han estado apoyándome todo este tiempo, aguantando mi mal genio cuando no me salían las cosas, y que me han ayudado de la mejor manera que podían.

Agradecer a José L. Ayala, director del TFG, la confianza mostrada en mí y su ayuda durante todo el TFG. También quiero agradecer al entrenador Raúl Pérez, su ayuda tanto en conceptos sobre rendimiento deportivo, como por la prueba de la aplicación y posterior análisis de los resultados, con usuarios reales.

Índice

Índice de figuras	IX
Índice de abreviaturas.....	XI
Resumen.....	XIII
Abstract	XV
Capítulo 1	1
Introducción	1
1.1 Conceptos.....	1
1.1.1. Rendimiento deportivo	1
1.1.2. Zonas de trabajo o entrenamiento	1
1.1.3. Variables de entrenamiento.....	2
1.1.4. Test de evaluación inicial	2
1.1.5 Ejercicios de resistencia	2
1.1.6. Sistemas de recomendación	3
1.1.7. Ítem	3
1.1.8. Aprendizaje automático	3
1.1.9. Android.....	3
1.1.10. Android SDK	3
1.2 Motivaciones	4
1.3 Objetivos	4
1.4 Visión general del documento. Estructura.....	5
Capítulo 2	7
2.1 Estado del arte	7
2.1.1 Adidas MiCoach.....	7
2.1.2 Entrenamiento Flanax	7
2.1.3 Last.fm.....	7
2.1.4. Manual del Entrenador Personal [11]	8
Capítulo 3	9
Especificaciones y Métodos	9
3.1 Requisitos	9
3.2. Funcionamiento de la aplicación.....	10
3.3. Investigación preliminar.....	12
3.3.1. Algoritmos de Filtrado Colaborativo basados en memoria [13]	13

3.3.1. Sistemas de Recomendación híbridos basados en memoria [13].....	14
3.3.2. Enfoque de sistemas basados en memoria.....	14
3.3.3. ¿Qué es Mahout?.....	15
3.3.4 ¿Cómo realizar una valoración de un ejercicio?.....	17
3.4. Algoritmos usados.....	20
3.4.1. Algoritmo del Método 1 de recomendación de la aplicación.....	21
3.4.2 Algoritmo del Método 2 de recomendación de la aplicación.....	23
3.4.3. Algoritmo del Método 3 de recomendación de la aplicación.....	25
3.5. Entorno de desarrollo y banco de pruebas.....	26
Capítulo 4.....	27
Arquitectura.....	27
4.1. Introducción.....	27
4.2. Módulo de interfaz de Usuario.....	29
4.3. Módulo de gestión de Usuarios.....	29
4.4. Módulo de BB.DD.....	30
4.5. Módulo de recomendación.....	33
Capítulo 5.....	35
Resultados.....	35
5.1. Introducción.....	35
5.2. Metodología de evaluación.....	35
5.3. Análisis de resultados.....	37
Capítulo 6.....	41
Conclusiones.....	41
6.1.1 Conclusiones.....	41
6.1.2 Conclusions.....	42
6.2. Trabajo Futuro.....	42
6.3. Seguimiento del Proyecto.....	43
Capítulo 7.....	45
Manual de uso.....	45
7.1 Introducir usuario en la aplicación.....	45
7.2 Modificar datos de un usuario.....	46
7.3 Test de valoración inicial.....	49
7.4 Añadir entrenamientos y ejercicios.....	50
ANEXO 1.....	57
ANEXO 2.....	63
Bibliografía.....	65

Índice de figuras

Figura 3.1 Esquema Funcionamiento General Aplicación.....	10
Figura 3.2 Arquitectura algoritmo Recommender, de Mahout	16
Figura 3.3 Valores de similaridad, usando coeficiente de Tanimoto	21
Figura 3.4. Código algoritmo, coeficiente de <i>Tanimoto</i>	22
Figura 3.5 Distancia euclídea entre Usu1 y otros usuarios	23
Figura 3.6. Código algoritmo, usando distancia euclídea para similaridad.....	24
Figura 4.1. Esquema MVC	27
Figura 4.2. Esquema MVC en Android	28
Figura 4.3. Diagrama mostrando dependencias entre paquetes de la aplicación.....	28
Figura 4.4. Diagrama Controlador de la aplicación.	29
Figura 4.5. Diagrama clases referente a la BB.DD.....	31
Figura 4.6. Clases que intervienen en la recomendación	33
Figura 5.1. Relación ejercicios recomendados VS ejercicios rechazados.....	38
Figura 5.2. Porcentaje de ejercicios rechazados por cada método.	38
Figura 7.1 Menú principal	45
Figura 7.2 Introducir datos.....	45
Figura 7.3 Botón menú principal.....	45
Figuras 7.4 Buscador de usuarios.....	46
Figura 7.5 Perfil de un usuario	47
Figura 7.6 Cambio datos en perfil de un usuario	48
Figura 7.7 Cuestionario	48
Figura 7.8 Test valoración inicial	49
Figura 7.9 Ejemplo test valoración inicial.....	50
Figura 7.10 Añadir entrenamiento.....	51
Figura 7.11 Añadir entrenamiento por método de recomendación.....	52
Figura 7.12 Ventana entrenamientos de usuario	53

Figura 7.13 Eliminar entrenamientos de usuario.....	54
Figura 7.14 Añadir ejercicio.....	54
Figura 7.15 Ejercicios de un entrenamientos.....	54
Figuras 7.16 Ventana introducir valores de ejercicio.....	55
Figuras 7.17 Ventana para ver valores de ejercicio realizado.....	56
Figura A.1.1. Tabla valores test O'neill (Mujeres)	57
Figura A.1.2. Tabla valores test O'neill (Hombres).....	58
Figura A.1.3. Tabla valores test O'neill (Junior)	59
Figura A.1.4. Tabla clasificación y valores normalizados O'neill	59
Figura A.1.5. Tabla clasificación y valores normalizados de test de cinta y remo	60
Figura A.1.6. Tabla clasificación test de VO ₂ Máx	60
Figura A.1.7. Tabla clasificación normalizada de test de VO ₂ Máx	60
Figura A.1.8 Nomograma de Astrand y Ryhming cicloergómetro	61
Figura A.1.9. Nomograma de Astrand y Ryhming escalón	62
Figura A.2.1. Tabla zonas de entrenamiento.....	63

Índice de abreviaturas

- APPs: Aplicaciones
- BB.DD: Bases de Datos
- TFG: Trabajo de Fin de Grado.
- XML: Extensible Markup Language
- MVC: Modelo Vista Controlador

Resumen

En la actualidad, la optimización del rendimiento deportivo es una necesidad que atañe tanto a los deportistas de élite, a los entrenadores, y a cualquiera que realice una práctica deportiva frecuente en mayor o menor medida.

Este trabajo consiste en una aplicación en Android en la que, partiendo de una serie de ejercicios categorizados, vamos a tratar de ayudar a un entrenador a trazar un plan de entrenamiento recomendando aquéllos ejercicios que mejor se pueden ajustar a un individuo. Para ello, se va a hacer uso de tres algoritmos de recomendación basados en resultados anteriores que tanto el propio individuo como otros, han experimentado realizando dichos ejercicios.

Hoy en día los sistemas de recomendación se están convirtiendo en una herramienta de gran importancia para muchos sitios web que tratan de ofrecer a un usuario un contenido que se ajuste a sus necesidades. En estos sitios se puede presentar al usuario por ejemplo una serie de ítems que puede que le interesen o quizás la recomendación de algún ítem específico que podría gustarle porque a otros usuarios similares les ha gustado. La aplicación va a estar enfocada con la misma idea que tienen estos sitios web, sólo que en este caso recomendando ciertos ejercicios de entrenamiento que mejor se adapten a un usuario ya que a otros les ha ido bien realizarlos.

Palabras clave:

Android, sistema de recomendación, rendimiento deportivo, entrenamiento

Abstract

Nowadays, the optimization of sport performance is an issue that concerns elite athletes, coaches, and anyone who practices sports in varying strength.

This project consists of an Android application that, starting from a set of pre-defined exercises, will help a coach to develop a training plan by recommending the best exercises adjusted to a person. For that purpose, it will make use of three recommendation algorithms that use the previously obtained results by the individual himself as other users of the application.

Nowadays, recommendation systems are becoming an important tool for many websites in order to give the user a content that will fit his/her needs. This proposed content can be presented to the user, for example, as a number of items that may interest him/her or also recommend any other specific item that the user might like, since other similar users did. The approach of this application will offer the same information as shown on these websites. However, this application will not only provide the user information about exercises, but also it will recommend the user those exercises adjusted to her/his need; since their practice has already helped others with the same need as well.

Keywords:

Android, recommendation system, sport performance, training

Capítulo 1

Introducción

Este Trabajo de Fin de Grado, de aquí en adelante TFG, “Algoritmos de adaptación dinámica del ejercicio para mejora del rendimiento deportivo” consiste en la realización de una aplicación en Android en la que un monitor o entrenador va a poder gestionar un entrenamiento compuesto de una serie de ejercicios a ciertas personas que sean introducidas en la aplicación.

Para ello, la aplicación se basará en unos métodos de recomendación que ayudarán al entrenador a elegir los ejercicios de los que tiene que componerse un entrenamiento. Es en este punto es cuando hay que resaltar el uso de tres algoritmos de recomendación, dos de los cuales hacen uso de la librería Mahout y un tercero que está inspirado en el funcionamiento de éstos y desarrollado específicamente para el entrenamiento deportivo.

1.1 Conceptos

1.1.1. Rendimiento deportivo

Parece existir un amplio acuerdo en considerar que el rendimiento deportivo es el resultado de una actividad deportiva.

A veces el rendimiento deportivo no se considera sólo el resultado de una actividad, sino que en su valoración y en la definición de su concepto se incluyen también el método y el esfuerzo individual que han conducido a ese resultado, es decir, el proceso de rendir [1].

1.1.2. Zonas de trabajo o entrenamiento

“La resistencia física viene determinada por la potencia y la capacidad de las fuentes de energía para mantener dicha potencia” (Zhelyazkov, 2001).

Se entiende por zonas de entrenamiento o áreas funcionales a la aplicación de cargas de trabajo que provocan modificaciones específicas. La experiencia

permite dividir las zonas de entrenamiento según las capacidades biomotoras y definir, de forma más precisa, las variables de las tareas (Raczek, 1990). La estructura de un modelo basado en zonas de potencia puede estar formada por los objetivos funcionales de la magnitud de las cargas y sus respuestas fisiológicas dentro de cada una de esas zonas [2].

En este trabajo nos vamos a centrar en cuatro zonas de entrenamiento: Aeróbica, umbral anaeróbico, potencia aeróbica y capacidad anaeróbica.

1.1.3. Variables de entrenamiento

Vamos a referirnos a variables de entrenamiento como aquéllos parámetros que nos ayudan a determinar el rendimiento de un individuo. Estas variables pueden indicarnos si se está trabajando en una zona de entrenamiento o no, dependiendo de sus valores obtenidos tras la realización de un ejercicio físico.

1.1.4. Test de evaluación inicial

Los test de evaluación inicial son una serie de ejercicios que sirven al entrenador para ver el estado físico de una persona y así poder trazar su plan de entrenamiento. En este trabajo, además de información para el monitor, nos va a servir para poder crear grupos de individuos con una condición física parecida. En el Anexo 1 se pueden consultar los test utilizados por la aplicación.

1.1.5 Ejercicios de resistencia

La resistencia busca que puedas hacer un esfuerzo físico durante el mayor tiempo posible.

Los ejercicios de resistencia, aumentan el ritmo cardíaco y de respiración. Mantienen en forma el corazón, los pulmones y el sistema circulatorio, además de mejorar la aptitud física general.

Y es que la resistencia del corazón, los pulmones y el sistema circulatorio para aportar oxígeno y nutrientes a los músculos, serán las claves de nuestra resistencia [3].

1.1.6. Sistemas de recomendación

Los sistemas de recomendación son herramientas que generan recomendaciones sobre un determinado objeto o ítem de estudio, a partir de las preferencias y opiniones dadas por los usuarios. El uso de estos sistemas se está poniendo cada vez más de moda en Internet debido a que son muy útiles para evaluar y filtrar la gran cantidad de información disponible en la Web con objeto de asistir a los usuarios en sus procesos de búsqueda y recuperación de información [4].

1.1.7. Ítem

Un ítem es cualquier entidad empleada como objeto de recomendación. Pueden ser películas, libros, música...dependiendo del dominio del sistema [5]. En el caso del TFG, se refiere a un ejercicio de entrenamiento.

1.1.8. Aprendizaje automático

“El aprendizaje automático es parte de la inteligencia artificial relacionado con las técnicas que permiten a las computadoras mejorar sus salidas de acuerdo a la experiencia previa. Para ello el sistema debe de tener la habilidad de aprender en un medio cambiante” [6].

1.1.9. Android

Android es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, como smartphones y tablets. Inicialmente fue desarrollado por Android Inc pero Google más tarde lo compró en 2005. Android fue presentado en 2007 junto con la fundación del Open Handset Alliance. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008 [7].

1.1.10. Android SDK

Software Development Kit (SDK) proporciona las APIs y herramientas de desarrollo necesarias para construir, testear y depurar aplicaciones para Android [8].

1.2 Motivaciones

Hasta donde nos alcanza el conocimiento, no tenemos constancia de la existencia de APPs desarrolladas en Android en la que los ítems que se recomiendan a los usuarios sean ejercicios para un entrenamiento. Sí que conocemos que existen APPs como por ejemplo *Last.fm* para recomendar canciones musicales, entre otras.

Por este motivo, la principal motivación para la realización de este TFG es la de desarrollar una aplicación novedosa a la hora de realizar ejercicio físico que esté orientada para un entrenador en la que pueda controlar los entrenamientos de varios usuarios a la vez.

Sabemos que existen multitud de APPs que monitorizan nuestros entrenamientos o nos muestran cómo realizar ciertos ejercicios, como puede ser el caso de *Endomondo*, *Adidas miCoach*, *Entrenamiento Flanax*, etc. Pero estas APPs, aparte de estar orientadas para un único usuario, no nos indican qué ejercicio realizar para favorecer nuestro rendimiento deportivo en función de nuestros resultados y los de otras personas, de las variables de entrenamiento. Esta es la gran diferencia que encontraremos en esta aplicación con las demás.

Además existe la posibilidad de crear un entrenamiento, compuesto por ejercicios que no sean recomendados sino que se añadan manualmente. Esto aparte de ser una prestación más para el entrenador que le da total libertad a la hora de crear entrenamientos, es necesario para el sistema de recomendación que necesita de resultados de ejercicios ya realizados para poder recomendar.

1.3 Objetivos

Uno de los principales objetivos de este TFG es el desarrollo de una aplicación en Android que sea capaz de monitorizar las sesiones de entrenamiento de un deportista donde un entrenador puede tener almacenada toda la información relacionada con él, necesaria para tener en cuenta a la hora de trazar un plan de entrenamiento. De este modo el monitor puede crear entrenamientos para una persona compuestos por ejercicios que él añade manualmente a ese entrenamiento, basándose en los datos que tiene almacenados en la aplicación sobre el deportista en concreto.

Otro objetivo de este trabajo es identificar y clasificar las variables de entrenamiento que intervienen en el rendimiento deportivo de una persona. La aplicación ofrece una BB.DD en la que están almacenados un gran número de ejercicios de resistencia categorizados en las cuatro zonas de trabajo indicadas anteriormente, donde se encuentran estas variables de entrenamiento.

Además, este trabajo pretende evaluar, mediante la opinión de un experto en la materia, los resultados de tres algoritmos de recomendación que recomiendan unos ejercicios para un entrenamiento personalizado. Dos de ellos son algoritmos de recomendación genéricos existentes en la librería Mahout y utilizados en cualquier contexto de aplicación que requiera de una recomendación automatizada. Por otro lado, el tercer algoritmo es una propuesta específica de este TFG, que pretende diseñar un algoritmo de recomendación adaptado específicamente a las necesidades de nuestro objetivo (sistema automatizado de entrenamiento deportivo).

1.4 Visión general del documento. Estructura

Este documento está estructurado de la siguiente manera:

En primer lugar, Capítulo 1, se hace una introducción del trabajo. Se enumeran los conceptos generales para comprender el resto de lectura del documento, se exponen las motivaciones que han llevado al desarrollo del TFG, y finalmente se muestran los objetivos que se pretenden con su realización.

En el Capítulo 2, se exponen las referencias que han servido para tomar ideas y asentar conceptos para la realización del TFG.

Seguidamente, se explican los requisitos y métodos utilizados para el desarrollo de la aplicación en el Capítulo 3. Se muestra una descripción del funcionamiento del sistema y una explicación de los tipos de algoritmos de recomendación utilizados. Además de una explicación detallada de los tres algoritmos implementados.

En el capítulo 4, se expone la arquitectura llevada a cabo para el desarrollo de la aplicación, distinguiendo las partes principales.

Finalmente, en los capítulos 5 y 6, se muestran los resultados de los algoritmos de recomendación, comparando sus resultados, y las conclusiones con posibles mejoras de la aplicación en un futuro. Y en el capítulo 7, un pequeño manual de uso de la aplicación.

Capítulo 2

2.1 Estado del arte

Como referencias para la realización del TFG y obtener ideas de APPs enfocadas en el campo del rendimiento deportivo y sistemas de recomendación de ítems a un usuario, podemos mencionar las siguientes:

2.1.1 Adidas MiCoach

La aplicación *Adidas miCoach* ofrece la posibilidad de realizar entrenamientos predefinidos y el entrenamiento eligiendo los ejercicios. Una vez realizado un ejercicio, evalúa los resultados indicando si vamos o no por el buen camino para conseguir nuestro objetivo.

2.1.2 Entrenamiento Flanax

Flanax es una aplicación que ofrece rutinas de entrenamiento de carrera con recomendaciones, retos y consejos saludables para mejorar el rendimiento de nuestro cuerpo al hacer las actividades físicas manteniendo sanos nuestro cuerpo y mente. [9]

Esta aplicación sólo solicita la edad y el sexo, y a partir de ahí recomienda rutinas de entrenamiento.

2.1.3 Last.fm

Last.fm [10] es una red social, una radio vía Internet y además un sistema de recomendación de música que construye perfiles y estadísticas sobre gustos musicales, basándose en los datos enviados por los usuarios registrados.

Un usuario de Last.fm puede construir un perfil musical escuchando su colección musical personal. Estas canciones escuchadas son añadidas a un registro desde donde se calcularán las recomendaciones musicales. Estas recomendaciones son calculadas usando un algoritmo de filtrado colaborativo, así los usuarios pueden explorar una lista de artistas no listados en su propio perfil pero que sí aparecen en otros usuarios con gustos similares.

La característica más usada por la comunidad de Last.fm es la formación de grupos de usuarios con algo en común (por ejemplo, militancia en otro foro de Internet). Last.fm generará un perfil de grupos similar a los perfiles de los usuarios, mostrando una amalgama de datos y mostrando listas con los gustos globales del grupo.

2.1.4. Manual del Entrenador Personal [11]

En este artículo se muestran indicaciones como que las funciones de un entrenador son evaluar, planificar y ajustar los programas de entrenamiento. Estas tareas pertenecen a un continuo proceso sin fin, en el que permanentemente se está evaluando, ajustando y planificando los entrenamientos del deportista.

Cuando un deportista acude a un entrenador, primero se le realiza una valoración mediante una recogida de datos y una serie de ejercicios de acondicionamiento y test, que muestran el estado físico en que se encuentra el deportista. Esto ayudará al entrenador en las planificaciones y creación de entrenamientos.

Capítulo 3

Especificaciones y Métodos

En este capítulo vamos a describir el diseño del proyecto, centrándonos en el análisis y especificación del sistema. Por ello, se detallarán los requisitos que definen la funcionalidad de la aplicación, para más tarde explicar con detalle los algoritmos utilizados para llevar a cabo el principal objetivo de la aplicación.

3.1 Requisitos

En este punto se describen las funcionalidades de la aplicación. Este trabajo ha sido desarrollado con la ayuda de un entrenador conocedor del campo del rendimiento deportivo, y se han consensuado una serie de requisitos:

1. Guardar los datos personales y físicos (estatura, peso, cintura...) de multitud de usuarios para que puedan ser evaluados posteriormente. Un nombre de usuario es único en la BB.DD de la aplicación.
2. Poder consultar y editar los datos de un usuario cuando sea necesario.
3. Poder evaluar la condición física de un usuario realizando test de evaluación inicial.
4. Crear entrenamientos para un usuario compuestos de ejercicios con total libertad de elección de ejercicios. En este caso, se puede añadir los ejercicios manualmente y también se puede elegir que los ejercicios sean recomendados por alguno de los tres métodos de recomendación que se explicarán más adelante. Aunque los ejercicios hayan sido recomendados, el entrenador podrá eliminar del entrenamiento aquéllos que estime oportunos y añadir otros que considere que debe realizar el usuario.
5. Introducir los valores de las variables de entrenamiento correspondientes a un ejercicio.
6. En la BB.DD se almacenan todos los ejercicios realizados por todos los usuarios, junto con sus resultados. De un mismo ejercicio ejecutado por un usuario, se van a almacenar los resultados de las últimas tres veces que lo ha practicado, para poder valorar la tendencia de ese ejercicio y saber si es favorable o no para el usuario.

7. Poder consultar los resultados de los ejercicios realizados por un usuario pudiendo editar el último ejercicio realizado.
8. Todos los test y algoritmos deberán realizar los cálculos de tal forma que garanticen que los resultados obtenidos son reales.
9. Realizar una interfaz de usuario amigable y funcional en Android.
10. Añadir una opción de evaluación de la recomendación (debug) mediante el trazado de rechazos de ejercicios que hace el entrenador personal. Esto es, que mediante un etiquetado en la BB.DD podemos saber, del número de ejercicios que ha generado una recomendación para un entrenamiento, con cuantos se queda el entrenador para que los realice el usuario.

3.2. Funcionamiento de la aplicación

A continuación se muestra un diagrama y se explica el funcionamiento de la aplicación desde que los datos son recogidos hasta que se muestran los resultados, pasando por todos los procesos en que consiste la aplicación:

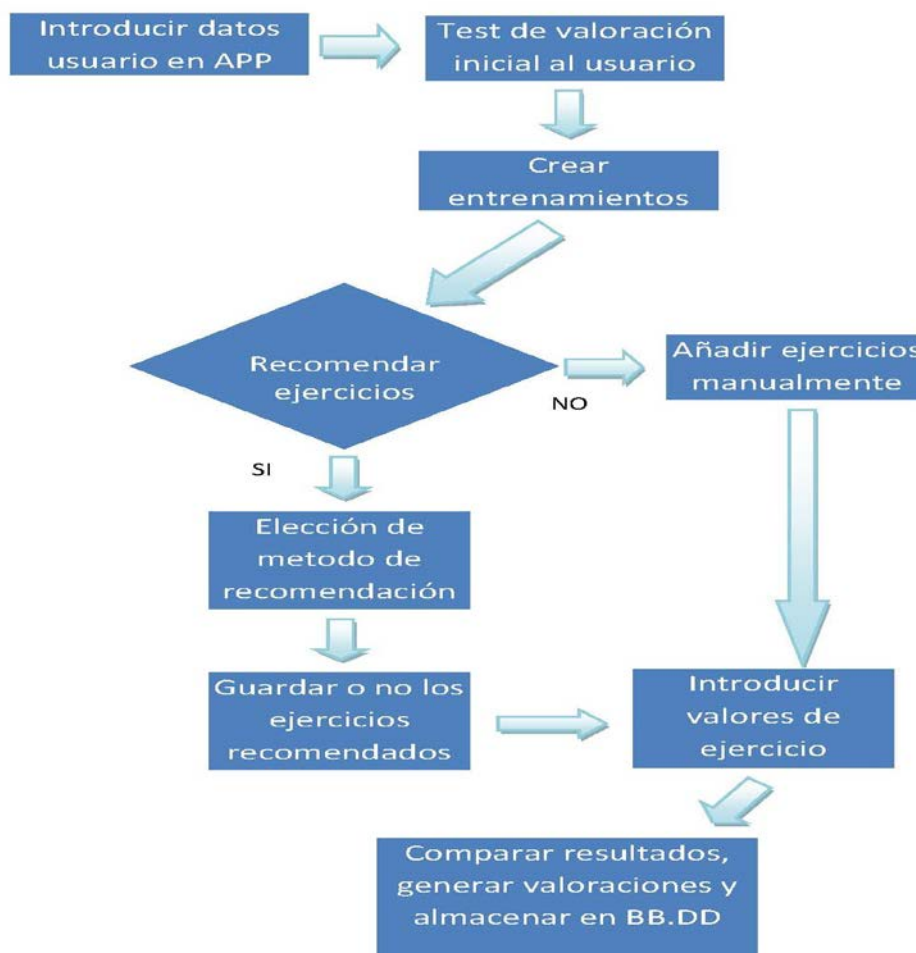


Figura 3.1. Esquema Funcionamiento General

Para poder recomendar ejercicios a un usuario, éste debe estar almacenado en la BB.DD de la aplicación. Por tanto hay que introducir a un usuario, cuyo nombre será único en el sistema, con sus datos personales y físicos de interés para el entrenador, y para poder realizar los cálculos de ejercicios y test de valoración inicial. Además se indicará el objetivo que busca el usuario para que el entrenador pueda conocer la zona de entrenamiento en la que podría trabajar el usuario.

Después, el entrenador pasará unos test de evaluación inicial a ese usuario para saber su condición física y que pueda ayudarle a la hora de indicar qué ejercicios realizar de una zona u otra de entrenamiento. Además el resultado de estos test servirá para agrupar al usuario en un perfil junto con otros usuarios, que serán sus vecinos, de condición física parecida.

Una vez que tenemos al usuario almacenado y clasificado en la aplicación, el monitor podrá crear los entrenamientos que estime oportunos, compuestos de los ejercicios elegidos en función de la zona de trabajo de ese usuario. El entrenador puede optar por 4 métodos de composición de entrenamientos:

- **Método personalizado:** en este método el entrenador elige manualmente desde el primero al último ejercicio que componen el entrenamiento.
- **Método 1:** este método origina 5 ejercicios, de las zonas de trabajo seleccionadas por el entrenador, recomendados por un algoritmo de recomendación que hace uso de la librería Mahout. En el punto 3.4 lo veremos en detalle.
- **Método 2:** este método, como el anterior, origina 5 ejercicios pertenecientes a zonas de entrenamiento seleccionadas por el entrenador, recomendados por un algoritmo de recomendación que hace uso de la librería Mahout. En este caso el algoritmo utilizado es distinto y los resultados pueden serlo también. En el punto 3.4 lo veremos en detalle.
- **Método 3:** este método origina 6 ejercicios, de las zonas de trabajo elegidas por el entrenador, recomendados por un algoritmo de recomendación orientado al rendimiento deportivo, muy diferente a los dos anteriores. En el punto 3.4 lo veremos en detalle.

Aunque estos métodos recomienden ejercicios, el entrenador no estará obligado en ningún caso a elegirlos y puede eliminar los que considere oportunos cuando desee y añadir otro ejercicio manualmente.

Cuando un usuario ya tiene un entrenamiento compuesto por ejercicios, el entrenador pasará a introducir en la aplicación los valores que han tomado las variables del ejercicio una vez lo ha realizado el usuario. Después de almacenar esos valores, se procede a valorar esos resultados respecto a resultados anteriores, si los hubiera para ese mismo ejercicio por parte del usuario.

Cuanto más ejercicios valorados según resultados de usuarios se almacenen en la aplicación mejores recomendaciones harán los algoritmos.

3.3. Investigación preliminar

La idea principal de los sistemas de recomendación es la de recomendar a usuarios, ítems o productos concretos en base a sus preferencias. Una buena definición puede ser esta, de un artículo de Sergio Manuel Galán:

“La idea que subyace tras ellos es encontrar usuarios con gustos similares a los de otro determinado y recomendar a éste cosas que desconoce pero que gustan a aquellos con los que se tiene similitud. Es decir, un sistema de recomendación es un amigo virtual cuyos gustos son una mezcla de los gustos de miembros de la comunidad de usuarios con gustos similares a los de uno mismo” [12].

Hasta el momento, la principal aplicación que tienen los sistemas de recomendación se ve reflejado en sitios web de comercio electrónico donde a un usuario le presentan una serie de productos que pueden ser de su agrado, bien porque a otros usuarios les ha gustado o bien porque según sus preferencias le pueden gustar.

Así, la recopilación de datos es una tarea muy importante en los sistemas de recomendación. Se puede llevar a cabo de dos maneras [5]:

1. Explícita: el usuario expresa de forma clara sus preferencias. Puede ser:

- Pedir a un usuario que califique en base a una escala determinada algún tema en particular.

- Solicitar a un usuario que seleccione de una lista de temas, los que él considere dentro de sus preferencias.

- Pedir al usuario que ordene por preferencia un conjunto de temas de una lista de temas preferidos.

2. Implícita: el sistema es el encargado de recopilar información sin que el usuario tenga que intervenir de una manera continuada. Puede ser:

- Guardar un historial con los ítems visitados del usuario.
- Analizar el número de visitas que recibe un ítem.
- Analizar el comportamiento de un usuario respecto a un ítem, como es el caso de la aplicación desarrollada en este proyecto.

Esta idea aplicada a este trabajo consiste en recomendar a un usuario aquéllos ejercicios, de sus zonas de entrenamiento, que hayan sido mejor valorados tanto por él mismo (si ya lo ha realizado), como por otros usuarios. Estas valoraciones se van a generar a partir de los resultados obtenidos en los ejercicios realizados.

Existen distintas técnicas de procesamiento de recomendaciones que realizan esta labor, y en este trabajo se van a utilizar dos algoritmos de Filtrado Colaborativo basados en memoria, proporcionados por el proyecto Mahout de Apache, y un tercero desarrollado personalmente basado en la idea sistemas de recomendación híbridos.

3.3.1. Algoritmos de Filtrado Colaborativo basados en memoria [13]

“El filtrado colaborativo es una técnica empleada por los Sistemas de Recomendación que utiliza la información de preferencias y calificación de un grupo de usuarios respecto a los ítems de un repositorio con el fin de predecir o inferir la preferencia de un usuario en particular sobre un ítem y a partir de esto generar una recomendación acertada “.

Los algoritmos basados en memoria utilizan la BB.DD, que debe contener una relación entre usuarios, ítems y valoración para generar las predicciones. Se basan en que cada usuario es parte de un grupo con intereses similares y al identificar a usuarios similares de otro usuario, se pueden producir predicciones de preferencias de nuevos ítems. Emplean **métricas de similitud** como coeficientes, correlaciones...que determinan el grado de similitud existente entre dos ítems o usuarios.

Por tanto, debe existir un historial de valoraciones sobre ítems, similar al del usuario actual y así poder originar a partir de este historial ítems para el usuario actual.

Este tipo de sistemas de recomendación presenta problemas cuando hay escasez de datos referentes a calificaciones y preferencias. Esto afecta al rendimiento del algoritmo de recomendación siendo el problema más común el conocido como *arranque en frío* [14]. Este problema ocurre cuando un usuario o ítem entra nuevo en el sistema y no existe la suficiente información como para poder generar predicciones.

Por tanto, es necesario acumular el mayor número de valoraciones posibles de los usuarios con respecto a los ítems en la BB.DD si queremos que los algoritmos sean eficientes.

3.3.1. Sistemas de Recomendación híbridos basados en memoria [13]

Los sistemas híbridos combinan las técnicas de filtrado colaborativo y basado en contenido (que consiste en utilizar la información disponible tanto del ítem como del usuario para calcular la recomendación), con el fin de mejorar las recomendaciones resultantes o superar problemas como el “*arranque en frío*” [14], que pueden presentar los métodos de filtrado colaborativo y basado en contenido al ser utilizados por separado.

Este tipo de sistemas de recomendación parten de obtener información a partir de los datos de un usuario para crear perfiles de usuario. Un perfil de usuario indica la información necesaria de las preferencias relacionadas a los ítems que son de interés del usuario. Este perfil puede ser creado a partir de ponderaciones manuales en la que a cada ítem se le asigna un peso por parte del usuario de acuerdo a sus preferencias. También esta ponderación se puede asignar de manera automática, calculándose los valores de los atributos e información previa del usuario. Una vez creados los perfiles, se recomiendan al usuario en cuestión, los ítems mejor valorados por el grupo de usuarios al que pertenece.

3.3.2. Enfoque de sistemas basados en memoria

Los sistemas de recomendación basados en memoria pueden ser:

1. **Basado en el usuario**, que recomienda al usuario ítems que han sido valorados por otros usuarios con gustos similares.
2. **Basado en el ítem**, que se basa en el hecho de que a los usuarios les gustan ítems similares a los que ya les gusta, por lo que recomienda a un usuario los ítems similares a los que ha valorado antes.

3.3.3. ¿Qué es Mahout?

“Mahout es una biblioteca de aprendizaje automático en código abierto de Apache. Los algoritmos implementados realizan en lo fundamental las funciones de motores de recomendación (filtraje colaborativo), agrupamiento y clasificación” [6].

Mahout es usado como una herramienta de aprendizaje automático cuando el conjunto de datos que hay que procesar es muy grande, posiblemente demasiado grande para una sola máquina.

Esta implementación está realizada en Java y construida sobre el proyecto de computación distribuida Apache Hadoop.

3.3.3.1. Algoritmos recomendación de Mahout: Recommender

Una aplicación Java hace uso de un algoritmo de recomendación de Mahout, que a su vez hace uso de un **DataModel** que se construye a partir de un conjunto de preferencias que se encuentran en una BB.DD para realizar las recomendaciones [15].

Los pasos para generar una recomendación con filtrado colaborativo usando Mahout son los siguientes [13]:

1. Proporcionar el modelo de datos (DataModel) que puede ser un archivo de texto o CSV donde se muestran las preferencias de los usuarios respecto de los ítems. Este archivo se compone de líneas expresadas de la siguiente manera:

Id_usuario, id_ítem, valoración

2. Seleccionar el algoritmo de similaridad que proporciona Mahout como puede ser Coeficiente de Tanimoto, Distancia euclídea, correlación de Pearson, entre otros.
3. Si estamos evaluando la similaridad entre usuarios, hay que establecer el tamaño del vecindario del usuario U para determinar los usuarios similares a U, y así obtener mejores recomendaciones.
4. Inicializar el recomendador en el que se especifica el modelo de datos, el algoritmo de similaridad y el vecindario.

5. Por último se obtienen los ítems recomendados al indicar sobre cuál usuario realizar la recomendación.

A continuación se muestra una imagen de la arquitectura de un Algoritmo de recomendación de Mahout, para explicar su funcionamiento [15]:

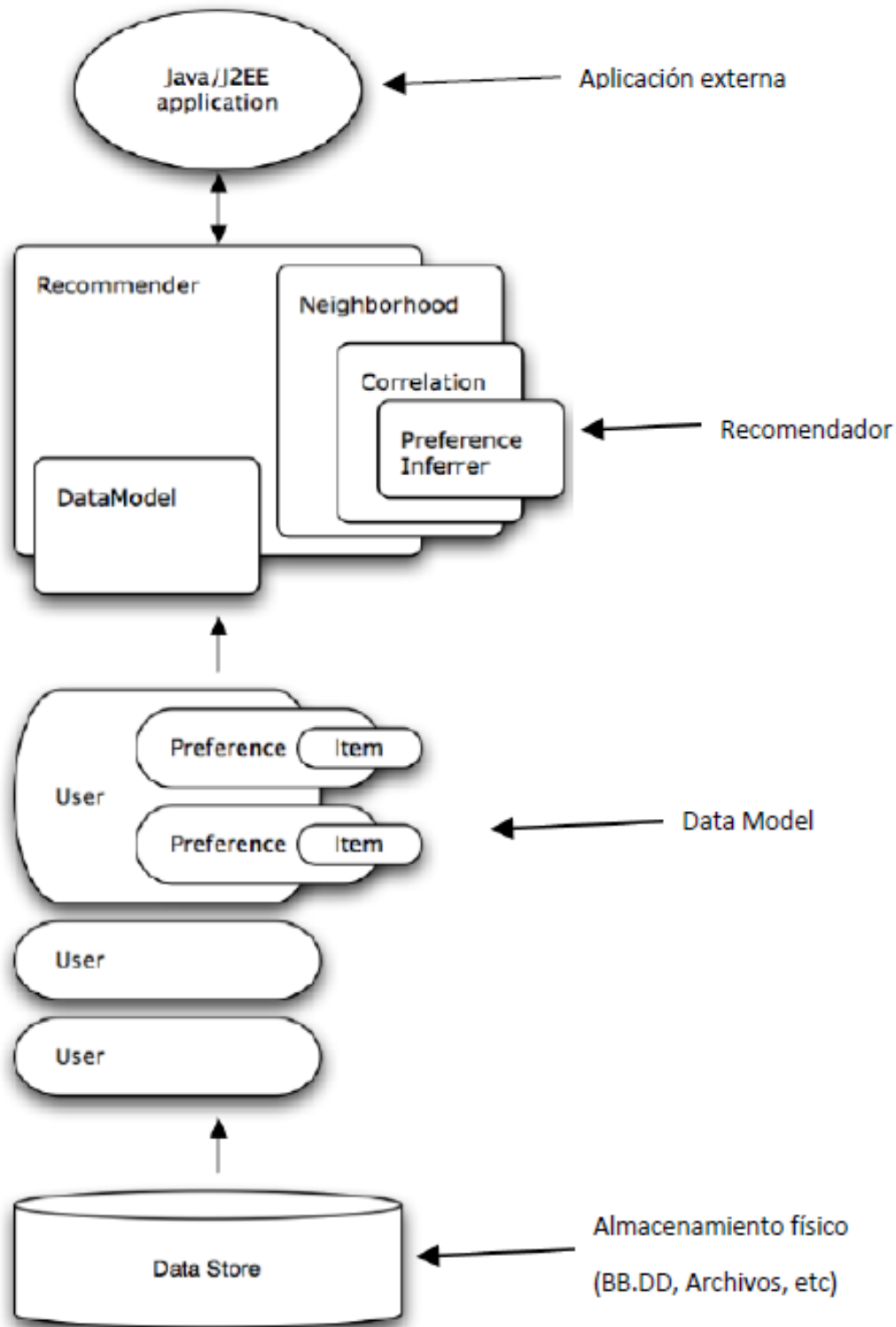


Figura 3.2. Arquitectura algoritmo Recommender, de Mahout.

3.3.4 ¿Cómo realizar una valoración de un ejercicio?

En este punto se va a explicar los criterios seguidos para realizar las valoraciones de un usuario respecto de un ejercicio de entrenamiento. Estas valoraciones van a ser calculadas de manera automática por la aplicación y en ningún momento interfieren el usuario y el entrenador en ellas.

Primero, hay que recordar que un ejercicio de entrenamiento se compone de varias variables de entrenamiento. Para poder valorar ese ejercicio, nos vamos a centrar en las variables que son resultado de la realización del ejercicio, ya que otras sólo sirven como indicador para saber cómo realizar el ejercicio en cuestión (variables antes de realizar el ejercicio).

Las variables de entrenamiento que son consideradas para realizar estas valoraciones se pueden clasificar en tres grupos: variables de intensidad, de volumen y de densidad.

Comenzamos explicando las variables de entrenamiento que establecen la zona de trabajo de un deportista. Estas variables pertenecen al grupo de variables de intensidad y son cuatro:

1. Frecuencia Cardíaca Máxima (FcMáx)

La frecuencia cardíaca (Fc) es el número de veces que el corazón se contrae en un minuto, y es importante conocer su comportamiento entre otras razones por su utilidad práctica como un índice de intensidad para dosificar el ejercicio físico [16].

La FcMáx es la frecuencia máxima (teórica) que puedes alcanzar en un ejercicio de esfuerzo sin poner en riesgo tu salud, siempre y cuando te encuentres en óptima condición física. Al alcanzar la frecuencia cardíaca máxima, teóricamente, se ha alcanzado la máxima capacidad de trabajo [16].

La FcMáx de una persona se calcula mediante la fórmula 220-Edad.

2. Frecuencia Cardíaca de Reserva (FcReserva)

La FcReserva es la diferencia entre la FcMáx y la cardíaca de reposo (FcReposo). La FcReserva permite un cálculo más ajustado de la frecuencia cardíaca de entrenamiento o instantánea (FCE), puesto que tiene en cuenta la de FcReposo [16]:

$$\text{FcReserva} = \text{FcMáx} - \text{FcReposo}$$

3. Volumen Máximo de Oxígeno (VO₂Máx)

Es la cantidad máxima de oxígeno que el organismo puede absorber, transportar y consumir por unidad de tiempo determinado, es decir, el máximo volumen de oxígeno en la sangre que nuestro organismo puede transportar y metabolizar. Es la manera más eficaz de medir la capacidad aeróbica de un individuo. Cuanto mayor sea el VO₂Máx, mayor será la capacidad cardiovascular [17].

4. Velocidad Aeróbica Máxima (VAM)

Es la velocidad de carrera alcanzada por un atleta cuando su consumo de oxígeno es máximo.

Para un deportista, conocer su VAM es indispensable para planificar su entrenamiento y establecer las velocidades del entrenamiento. Todas las velocidades de entrenamiento se trabajan a un cierto porcentaje del VAM del deportista.

Existen diferentes pruebas para calcular el VAM de un deportista. En esta aplicación el entrenador ya conoce el VAM del usuario y lo introduce directamente en sus datos de perfil.

Una vez que tenemos definidas estas cuatro variables, para conocer si un usuario está trabajando en la zona de entrenamiento a la que pertenece el ejercicio realizado, los resultados se comparan con una escala (ver en anexo 2) que indica en qué porcentaje del valor de estas variables debe trabajar el deportista para pertenecer a una zona de trabajo. Según la información del entrenador experto, no es necesario tomar el valor de todas las variables cada vez que se realiza un ejercicio. Por tanto, bastaría con que el valor de una de estas cuatro variables cumpla los valores de la escala.

Ahora vamos a pasar a explicar las demás variables que son producto de la realización del ejercicio y se deben tener en cuenta para saber si un ejercicio es favorable o no para el usuario que lo realiza.

En las cuatro zonas de entrenamiento, dentro del grupo de las variables de intensidad, tenemos las kilocalorías y las sensaciones que son resultado de la realización de un ejercicio:

- **Kilocalorías:** Es una unidad de energía térmica, que es usada para expresar el valor energético de los alimentos. Cuantas más calorías se consuma en la práctica del ejercicio, mejor será para el deportista.

- **Sensaciones:** son un indicador del esfuerzo que realiza un deportista. Se miden por la *escala de Borg* [18], que es una escala de valores de 0 a 10 en el que el 0 indica que el esfuerzo es mínimo y el 10 que es máximo. Por tanto cuanto mayor sea el valor de esta variable, peor será para el deportista.

Luego, dependiendo de la zona de entrenamiento y tipo de ejercicio realizado podemos encontrar variables de tiempo, distancia recorrida, y más variables relacionadas con éstas como pueden ser el número de brazadas, remadas, tiempo medio de unas series, etc. En el caso de las variables relacionadas con el tiempo, cuanto menor sea ese valor mejor para el deportista, al contrario de lo que ocurre con las variables relacionadas con la distancia.

Al tener definidas las variables que van a ser consideradas a la hora de realizar la valoración del ejercicio, ya se puede pasar a explicar en qué consiste dicha valoración. Los pasos son los siguientes:

- Un ejercicio será valorado en una escala de 1 a 5, siendo las puntuaciones de la siguiente manera:

- ✓ **1:** los resultados de las variables VAM, FcReserva, FcMáx, y VO₂Máx indican que el usuario no está trabajando en la zona de trabajo correspondiente al ejercicio, y además en el cómputo global de resultados de las demás variables empeora respecto a resultados anteriores del mismo ejercicio realizado por el usuario.
- ✓ **2:** los resultados de las variables VAM, FcReserva, FcMáx, y VO₂Máx indican que el usuario sí está trabajando en la zona de trabajo correspondiente al ejercicio, pero los resultados en el cómputo global del resto de variables empeoran respecto a resultados anteriores del mismo ejercicio realizado por el usuario.
- ✓ **3:** los resultados de las variables VAM, FcReserva, FcMáx, y VO₂Máx indican que el usuario sí está trabajando en la zona de trabajo correspondiente al ejercicio, pero los resultados en el cómputo global del resto de variables no alteran los resultados.
- ✓ **4:** los resultados de las variables VAM, FcReserva, FcMáx, y VO₂Máx indican que el usuario no está trabajando en la zona de trabajo correspondiente al ejercicio, pero sin embargo los resultados en el cómputo global del resto de variables mejoran respecto a resultados anteriores.
- ✓ **5:** los resultados de las variables VAM, FcReserva, FcMáx, y VO₂Máx indican que el usuario sí está trabajando en la zona de trabajo

correspondiente al ejercicio, y los resultados en el cómputo global del resto de variables mejoran respecto a resultados anteriores.

- Y ahora bien, ¿qué método se sigue para saber si los resultados de las variables mejoran con respecto a los anteriores?

Como se ha comentado en esta memoria en puntos anteriores, la aplicación almacena hasta un máximo de tres, los resultados de las veces que un usuario practica el mismo ejercicio. Por tanto, se pueden llegar a comparar hasta 4 resultados para una variable (los tres ya almacenados, más el que el monitor está introduciendo), y hallando la ecuación de la tendencia de esos valores por el **Método de los Mínimos Cuadrados** [19], se puede saber si esos valores tienen una progresión alta o baja fijándonos en la pendiente de la recta, si es positiva o negativa respectivamente.

Una vez sabido si cada una de las variables mejora o empeora pueden ocurrir tres casos:

1. Si la suma de las variables que mejoran es **mayor** que las que empeoran, se considera que en el cómputo global las variables del último ejercicio realizado con respecto a los anteriores mejoran.

2. Si la suma de las variables que mejoran es **menor** que las que empeoran, se considera que en el cómputo global las variables del último ejercicio realizado con respecto a los anteriores empeoran.

3. Si la suma de las variables que mejoran es **igual** que las que empeoran, se considera que en el cómputo global las variables del último ejercicio realizado se quedan igual que la vez anterior.

3.4. Algoritmos usados

En este apartado se van a explicar los tres algoritmos de recomendación que ofrece la aplicación. Los dos primeros se utilizan para comparar sus resultados con un tercer algoritmo que es el principal de la aplicación. Hay que hacer especial hincapié en que para que los algoritmos generen resultados, deben existir previamente un número de ejercicios valorados en la aplicación, que deben haber sido añadidos de forma manual por el entrenador.

3.4.1. Algoritmo del Método 1 de recomendación de la aplicación

Este algoritmo hace uso de la librería de Mahout, y es un algoritmo basado en usuario con filtrado colaborativo, y que usa la métrica del coeficiente de *Tanimoto* y el método *NearestUserNeighborhood* para definir la vecindad.

El coeficiente de *Tanimoto* [20] ignora las preferencias de los usuarios y solamente se fija en que un usuario exprese preferencia por un ítem, que en nuestro caso es que un ejercicio haya sido realizado por un usuario y tenga una valoración.

TanimotoCoefficientSimilarity [20] es una de las implementaciones basadas en el coeficiente de Tanimoto. Es el número de ítems por el que dos usuarios expresan preferencias dividido por el número de ítems para el que cada usuario expresa preferencias.

Cuando los ítems valorados por dos usuarios se superponen completamente, el resultado del coeficiente de Tanimoto es 1.0 y cuando no lo hacen, en ningún ítem, es 0.0. Un ejemplo es la siguiente tabla:

	Ejer1	Ejer2	Ejer3	Ejer4	Ejer5	Ejer6	Ejer7	Similaridad con Usu1
Usu1	x	x	x					1.0
Usu2	x	x	x	x				0.75
Usu3	x			x	x		x	0.17
Usu4	x		x	x		x		0.4
Usu5	x	x	x	x	x	x		0.5

Figura 3.3 [20]. Valores de similitud entre Usu1 y otros usuarios, usando coeficiente de Tanimoto.

En esta tabla se aprecian los valores del coeficiente de similitud de Tanimoto entre Usu1 y los otros 4. Aplicado a nuestro trabajo, las valoraciones de los ejercicios no importan, sólo importa que hayan sido realizados por los usuarios y coincidan los mismos ejercicios realizados lo máximo posible.

Una vez que se calcula el coeficiente de similitud, se hace uso del método *NearestUserNeighborhood* para definir la vecindad, en el que los primeros N usuarios con mayor coeficiente de similitud respecto al usuario valorado, serán seleccionados como vecinos. En el algoritmo de esta aplicación se seleccionan 10 vecinos.

En la siguiente imagen se muestra el código, usado en el desarrollo de la aplicación, de este algoritmo y la explicación:

```
if(metodo==1){
    DataModel dm = new FileDataModel(file);

    TanimotoCoefficientSimilarity simT = new TanimotoCoefficientSimilarity(dm);

    UserNeighborhood neighborhood = new NearestNUserNeighborhood(10, simT, dm);

    Recommender recommender = new GenericUserBasedRecommender(dm, neighborhood, simT);

    List<RecommendedItem> recommendations = recommender.recommend(idUsuario, 5);
    listaIdEjercicio=new ArrayList<Long>();

    for(RecommendedItem recommendation : recommendations) {
        listaIdEjercicio.add(recommendation.getItemID());
    }
}
```

Figura 3.4. Algoritmo basado en usuario con filtro colaborativo, coeficiente de *Tanimoto* para similitud y método *NearestUserNeighborhood* para definir vecindad.

1. Se obtiene el DataModel que es un archivo CSV, que ha sido generado a partir de las valoraciones que existan de los ejercicios en la BB.DD. En este algoritmo se van a coger todas las valoraciones de todos los usuarios, de aquéllas zonas seleccionadas por el entrenador antes de generar la recomendación.
2. Se calcula el coeficiente de similitud de Tanimoto entre el usuario en cuestión y los demás.
3. Se hace uso del método *NearestUserNeighborhood* para obtener 10 usuarios vecinos del usuario.
4. Se crea el motor de recomendación (clase *Recommender*) haciendo uso del método *GenericUserBasedRecommender* que usando el DataModel, la definición del vecindario y la métrica de similitud, va a contener aquéllas líneas del DataModel que cumplen las condiciones.
5. Añade a una lista las 5 recomendaciones que mayor similitud tienen con el usuario.
6. Se obtiene el idEjercicio (identificador del ejercicio) de esas recomendaciones y se añaden a la lista de ejercicios del entrenamiento.

3.4.2 Algoritmo del Método 2 de recomendación de la aplicación

Este algoritmo, como el anterior, hace uso de la librería de Mahout y es un algoritmo basado en ítem con filtrado colaborativo, y que usa como métrica de similitud la distancia euclídea.

La distancia euclídea mide la distancia entre dos usuarios $u1$ y $u2$, pensando en los ítems como las dimensiones y las valoraciones como puntos a lo largo de esas dimensiones [21]. Esta distancia se calcula utilizando todos los ítems (dimensiones) donde ambos usuarios han expresado su preferencia por ese ítem. Se calcula con la raíz cuadrada de la suma de los cuadrados de las diferencias entre las valoraciones del mismo ítem por parte de los usuarios:

Distancia $(u1,u2)= \sqrt{a^2 + b^2 + \dots + n^2}$, siendo a, b .. n, las diferencias existentes de las valoraciones entre el mismo ítem valorado por $u1$ y por $u2$.

La similitud se calculada como $1 / (1 + \text{la distancia})$, por lo que los valores resultantes están en el rango (0,1]. Un ejemplo es la siguiente tabla:

	Ejer1	Ejer2	Ejer3	Distancia	Similitud con Usu1
Usu1	5	3	2	0	1.0
Usu2	2	2	5	4.358	0.186
Usu3	5		3	1	0.5
Usu4	4	3	3	1.4	0.41

Figura 3.5 [20]. Distancia euclídea entre Usu1 y otros usuarios, y valor de similitud

En la siguiente imagen se muestra el código, usado en el desarrollo de la aplicación, de este algoritmo y la explicación:

```
DataModel dm = new FileDataModel(file);

ItemSimilarity simE = new EuclideanDistanceSimilarity(dm);

GenericItemBasedRecommender recommender = new GenericItemBasedRecommender(dm, simE);

List<RecommendedItem> recommendations = recommender.recommend(idUsuario, 5);

listaIdEjercicio=new ArrayList<Long>();

for (RecommendedItem recommendation : recommendations) {
    listaIdEjercicio.add(recommendation.getItemID());
}
```

Figura 3.6. Algoritmo basado en ítem con filtro colaborativo y distancia euclídea para similitud.

1. Se obtiene el DataModel que es un archivo CSV, que ha sido generado a partir de las valoraciones que existan de los ejercicios en la BB.DD. En este algoritmo se van a coger todas las valoraciones de todos los usuarios, de aquéllas zonas seleccionadas por el entrenador antes de generar la recomendación.
2. Se calcula la similitud, entre los ejercicios de los usuarios, con la distancia euclídea.
3. Se crea la recomendación (Recommender) haciendo uso del método *GenericItemBasedRecommender* que usando el DataModel, y la métrica de similitud, va a contener aquéllas líneas del DataModel que cumplen las condiciones.
4. Añade a una lista las 5 recomendaciones que mayor similitud tienen con los ítems del usuario.
5. Se obtiene el idEjercicio de esas recomendaciones y se añaden a la lista de ejercicios del entrenamiento.

3.4.3. Algoritmo del Método 3 de recomendación de la aplicación

El algoritmo de recomendación que se encuentra en este método es un algoritmo desarrollado específicamente para la recomendación en la práctica deportiva, sin hacer uso de la librería de Mahout. Se puede clasificar como un algoritmo de recomendación híbrido, basado en usuario, ya que obtiene información tanto de los test de valoración inicial, para crear grupos de usuarios, como de las valoraciones existentes sobre ejercicios, para posteriormente generar recomendaciones de ejercicios que sólo están valorados por los usuarios de ese grupo y por él mismo.

Cuando el usuario realiza un test, el resultado del mismo puede ser un valor numérico o un valor de clasificación que indica su estado de forma (por ejemplo muy bajo, bajo, medio, etc). Estos resultados se encuentran dentro de una escala (ver anexo 1) de 6 valores que identifican 6 estados de forma de una persona.

A la hora de crear los grupos de usuarios con condiciones físicas similares, los resultados de los test se van a normalizar de tal forma que el estado de forma más bajo le corresponde un 1 y el mayor estado de forma un 6. De esta manera, se puede hallar la media de los resultados de los test y usarla para establecer el factor de similaridad entre dos usuarios.

Por tanto, para considerar que dos usuarios o más pertenecen al mismo grupo se tienen que cumplir dos condiciones:

1. Todos los usuarios del mismo grupo deben haber realizado los mismos test de valoración o test cuya finalidad sea la misma. Esto último ocurre con los test de Mader, Hollman, Astrand y Rhyming (submáximo), 1 milla, y Astrand y Rhyming de escalón, que todos ellos miden el $VO_2Máx$ del deportista.
2. Los usuarios cuya media de test similares realizados difiera como máximo en un 10% (factor de similaridad), se consideran que son usuarios de un perfil de condición física similar y pertenecerán al mismo grupo (serán vecinos).

Una vez explicado lo anterior vamos a pasar a explicar los pasos que sigue el algoritmo:

1. Se obtienen los resultados de los test del usuario en cuestión, y se normalizan para adecuarlos a la escala de valores de 1 a 6 comentada anteriormente, y calcular su media.

2. Se crea la lista de vecinos del usuario que consiste en aquéllos usuarios que han realizado los mismos test o similares, y que la media entre ellos no difiera en más de 0.6. En este grupo se incluye al propio usuario.
3. Se obtienen todos los ejercicios realizados por todos los usuarios del grupo, y posteriormente se seleccionan los 6 ejercicios mejor valorados de entre todos los del grupo de usuarios creado anteriormente.

Por último, hay que dejar constancia que este algoritmo genera recomendaciones si existen en la aplicación usuarios que han realizado los mismos test de valoración inicial y la media de sus resultados muestra que tienen una condición física parecida difiriendo como mucho en un 10%. Si no existen vecinos, el grupo sólo está compuesto por el usuario en cuestión y sólo recomendaría los ejercicios que éste hubiera realizado.

3.5. Entorno de desarrollo y banco de pruebas

La aplicación ha sido realizada bajo el sistema operativo Windows 8 de 64 bits.

La herramienta de desarrollo utilizada ha sido Android Developer Tool, que incluye el SDK y la plataforma Eclipse con entorno de desarrollo para el lenguaje Java.

A continuación, se especifican los datos del software utilizado:

- JDK: versión 1.6 de 64 bits.
- ADT: Android Developer Tools versión 22.2.1.
- SDK: versión Android-sdk revisión 22.2.1 para Windows
- Android: versión de desarrollo Android 4.3 (API 18). Se optó por esta versión al ser la versión más reciente en el mercado a la hora de comenzar a desarrollar la aplicación. Se ha comprobado que la aplicación es compatible con versiones de Android 4.0 en adelante y funciona correctamente.

Las pruebas de depuración de la aplicación han sido realizadas en dos dispositivos que tienen distintas resoluciones y especificaciones para observar así el impacto de la aplicación en distintos dispositivos: un móvil y una Tablet.

- Móvil POMP w88s, con Android 4.2.2, Jelly Beans.
- Samsung GALAXY Tab 2 GT-P3110, con Android 4.0, Ice Cream Sandwich.

Capítulo 4

Arquitectura

4.1. Introducción

Para la realización de esta aplicación se ha seguido el patrón MVC (Modelo-Vista-Controlador) aplicado a Android [22]. La estructura general de este patrón, divide la aplicación en tres niveles:

1. La Vista, que representa la interfaz gráfica del usuario.
2. El Modelo, que representa el tratamiento de datos.
3. El Controlador, que se encarga de toda la lógica que se tiene que llevar a cabo por la aplicación.



Figura 4.1. Esquema MVC [22]

Esta división del sistema se hace para facilitar la labor a la hora de realizar cambios en la aplicación. Por ejemplo, si se quiere modificar la interfaz de usuario, sólo se debe modificar la Vista, o si se quiere cambiar de sistema de almacenamiento de datos, sólo hay que modificar el Modelo.

Ahora bien, este patrón en Android tiene alguna peculiaridad. Las vistas en Android corresponden a los archivos XML (Extensible Markup Language) creados, que cada uno de ellos está asociado con una Activity que gestiona ese XML. Estas Activities van a pertenecer al nivel del controlador, y si necesitan conectar con los datos de la aplicación que se encuentran en el modelo, todas ellas van a pasar por un controlador intermedio.

Este controlador intermedio va a ser una clase encargada de la gestión de toda la aplicación a la que llamaremos *Aplicacion.java*, que será la que controle las Activities, de tal forma que la comunicación entre el Modelo y las Activities se

hará a través de esta clase. Por esto, la estructura del MVC en Android, y seguida por esta aplicación, quedaría así:

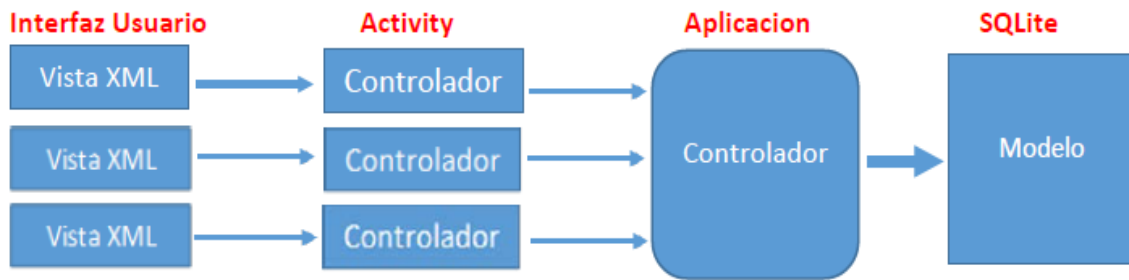


Figura 4.2. Esquema MVC en Android [22]

Tal y como se ha mencionado anteriormente, el Modelo hace referencia al tratamiento de datos de la aplicación. En este sistema, el gestor de BB.DD es SQLite, y en la imagen se ve que se encuentra en el Modelo, estando integrado como una parte más de la aplicación. Dentro de este nivel, se implementa una clase llamada *Modelo.java*, que es la que va a contener todas las implementaciones de los métodos de la aplicación.

En la siguiente imagen se muestra el diagrama de dependencias entre paquetes de la aplicación:

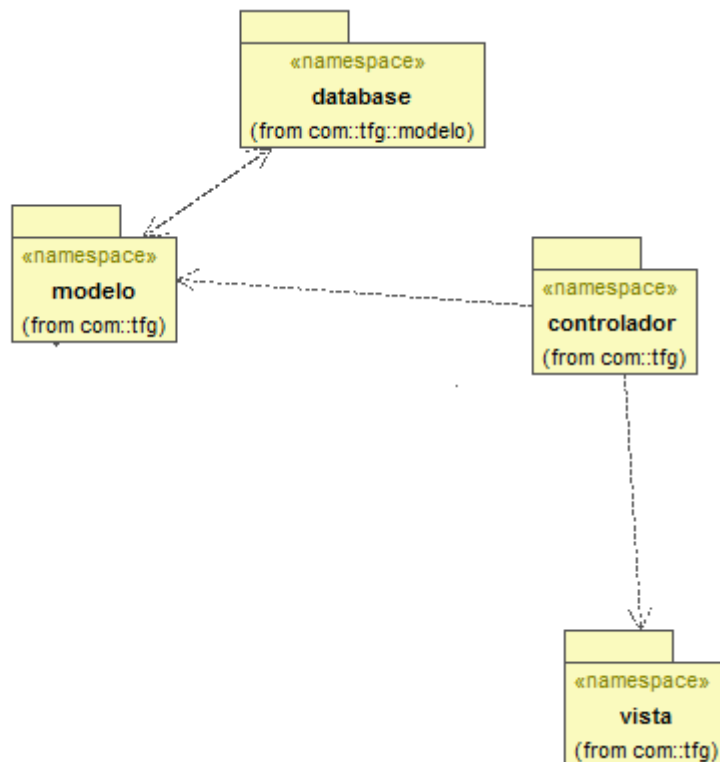


Figura 4.3. Diagrama mostrando dependencias entre paquetes de la aplicación.

En los siguientes puntos se va a describir el funcionamiento de esta arquitectura centrándonos en los 4 módulos funcionales de la aplicación, indicando en qué nivel de la arquitectura se encuentra cada uno de ellos.

4.2. Módulo de interfaz de Usuario

Este módulo se compone de todos los XML que representan la interfaz gráfica del usuario. Cada XML está asociado con una Activity, que gestiona esa interfaz.

En el desarrollo de la aplicación, se ha considerado crear un paquete vista, que contiene una única clase que hace referencia a la funcionalidad de introducir la fecha de un usuario. Esa es la clase *DatePickerFragment.java*. Todas las demás vistas son archivos XML, y están fuera de los paquetes del código fuente del proyecto, concretamente en el directorio *res.layout*.

4.3. Módulo de gestión de Usuarios

La funcionalidad de gestionar todo lo relacionado con usuarios se encuentra en el Controlador, son las Activities las encargadas de hacerlo por medio de la clase *Aplicación.java*. El diagrama de clases es el siguiente:

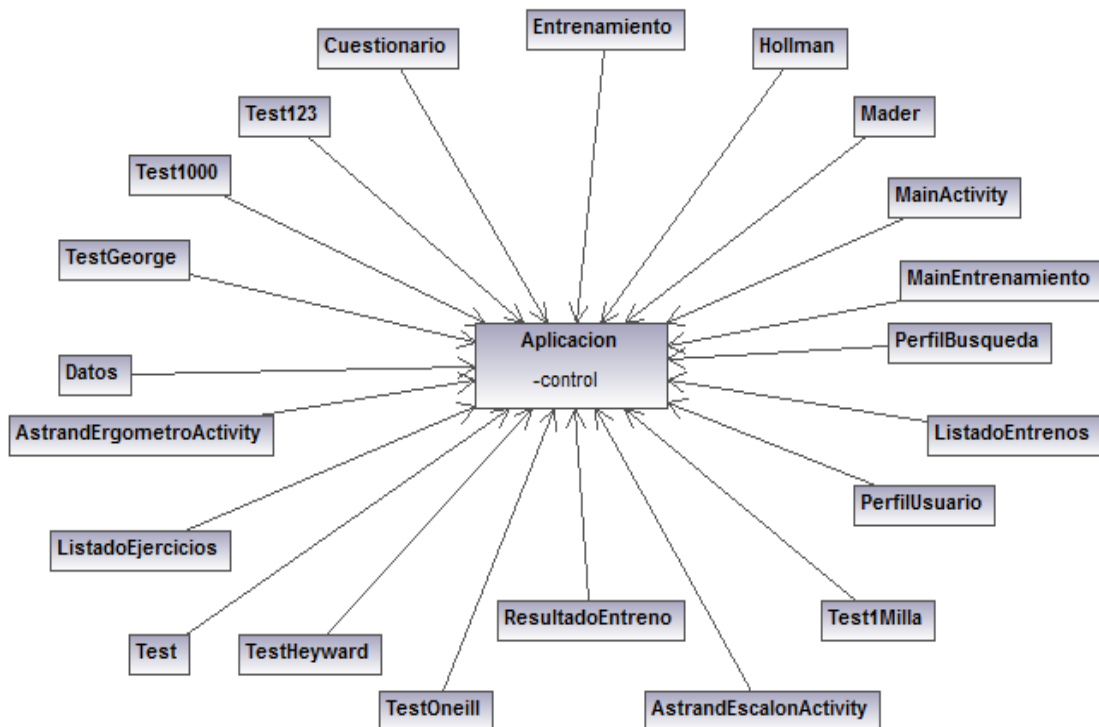


Figura 4.4. Diagrama Controlador de la aplicación.

La clase *Aplicacion.java* es la encargada de comunicar todas las Activities con el Modelo, mediante el atributo *control*. En ella se encuentran todos los métodos necesarios para la gestión de un usuario. Esto es, los métodos que tratan su información física, datos personales, cálculos de test de evaluación inicial, valoraciones a un ejercicio, etc.

Como clases destacables, tenemos:

- *MainActivity.java*: es la clase principal con la que se inicia la aplicación. En ella se encuentra un menú con las opciones que ofrece la aplicación, para ir accediendo a las demás Activities.

- *ListadoEntrenos.java*: esta Activity se encarga de eliminar o añadir entrenamientos de un usuario. Además, en ella se pueden ver las zonas en las que trabaja un usuario.

- *ListadoEjercicios.java*: esta Activity se encarga de eliminar o añadir ejercicios a un entrenamiento.

- *MainEntrenamiento.java*: en ella, se encuentran todas las variables de un ejercicio, en las que el entrenador va a introducir valores después de la realización del ejercicio. De este modo, se obtiene una valoración para dicho ejercicio por parte del usuario. También se muestran los dos últimos resultados de ese mismo ejercicio realizados por el usuario. Los resultados del último ejercicio pueden ser modificados.

4.4. Módulo de BB.DD

El sistema de Gestión de BB.DD empleado en la aplicación es SQLite. Este gestor es el más conocido de Android. SQLite [23] es un gestor de BB.DD de código libre que tiene unas características como que es de pequeño tamaño, no necesita servidor, necesita poca configuración, y es transaccional.

Todo lo referente a la BB.DD. se encuentra dentro del nivel Modelo. En la Figura 4.2, se muestra su diagrama de clases donde se pueden observar las tablas que componen la BB.DD, además de una clase utilizada para la gestión de esas tablas, llamada *DBAdapter.java*. Esta clase es la encargada de crear las conexiones con la BB.DD, acceso, consulta y actualización de datos. Las demás clases ofrecen las consultas necesarias para la gestión de las tablas a las que representan: creación, actualización y consulta de datos.

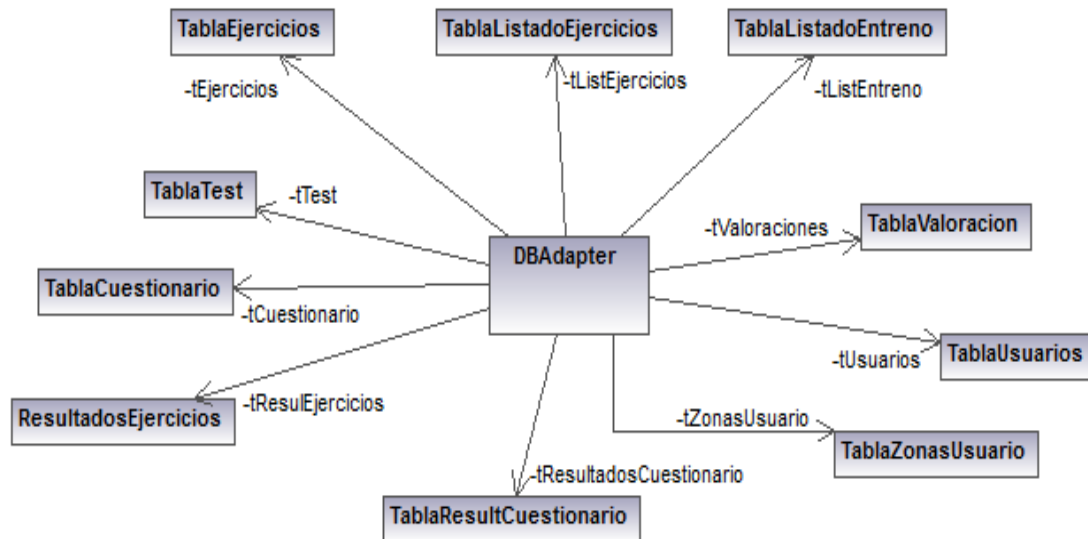


Figura 4.5. Diagrama clases referente a la BB.DD.

La BB.DD. se compone de 10 tablas. Todas ellas tienen un identificador numérico que se genera automáticamente que es la clave primaria de cada tabla. Las tablas son las siguientes:

- **Usuarios:** esta tabla es creada por *DBAdapter.java*, a través de la clase *TablaUsuarios.java*. Contiene toda la información referente a un usuario, tanto datos personales, como características físicas (altura, peso, etc). El nombre de usuario es único en el sistema. Los usuarios no se eliminan de la BB.DD.
- **Test:** esta tabla es creada por *DBAdapter.java*, a través de la clase *TablaTest.java*. En ella se almacenan los resultados de los test de valoración inicial realizados por los usuarios. Estos resultados clasifican al usuario en 6 niveles de estado de forma, mediante el uso de las tablas y nomograma recogidos en el Anexo 1. El cálculo de estos resultados se encuentra implementado en el nivel de Modelo en la clase *Modelo.java*, que utiliza a su vez otras clases como *AstrandErgometro.java*, *AstrandEscalon.java*, *Punto.java*, *Ecuacion.java* y *Oneill.java*.

Esta tabla contiene el nombre de usuario que realiza los test, y sus resultados. Los resultados de esos test, se van actualizando según los realizan los usuarios, guardando siempre el último resultado. No se pueden eliminar los test realizados por un usuario.

- **Preguntas:** tabla creada por *DBAdapter.java*, a través de la clase *TablaCuestionario.java*. Almacena un cuestionario que realiza cada usuario, para proporcionar información al entrenador. Es una tabla de sólo lectura. Mediante el identificador de pregunta se relaciona con la tabla **Cuestionario**.

- **Cuestionario:** esta tabla es creada por *DBAdapter.java*, a través de la clase *TablaResultCuestionario.java*. Contiene las respuestas de un usuario a las preguntas del cuestionario. Contiene el nombre de usuario y el identificador de pregunta, juntos con las respuestas del usuario a cada pregunta.

-**Ejercicios:** tabla creada por *DBAdapter.java*, a través de la clase *TablaEjercicios.java*. En esta tabla se almacenan todos los ejercicios que puede recomendar la aplicación. Son ejercicios de resistencia y se clasifican en 4 zonas de entrenamiento. Contiene, además del identificador de ejercicio, el nombre de todas las variables de entrenamiento de cada ejercicio acompañado del método de entrenamiento, actividad, y la zona de trabajo a la que pertenece el ejercicio. Es una tabla de sólo lectura.

- **ListadoEntreno:** esta tabla es creada por *DBAdapter.java*, a través de la clase *TablaListadoEntreno.java*. Almacena los entrenamientos que realizan los usuarios. Contiene el identificador del entrenamiento, el nombre de usuario, el nombre del entrenamiento, y el método de creación de entrenamiento elegido. Al eliminar un entrenamiento, se eliminan a su vez los ejercicios de los que se compone, pero no se eliminan los resultados de los ejercicios (si un usuario ha realizado un ejercicio, se mantienen sus resultados para posteriores comprobaciones).

-**ListadoEjercicios:** tabla creada por *DBAdapter.java*, a través de la clase *TablaListadoEjercicio.java*. Almacena los ejercicios que pertenecen a un entrenamiento. Contiene el identificador de ejercicio, el identificador de entrenamiento, y si el ejercicio es añadido de forma manual o automática.

-**ResultadosEntreno:** tabla creada por *DBAdapter.java*, a través de la clase *ResultadoEjercicios.java*. Almacena los resultados de las variables de los ejercicios realizados por los usuarios. Se distingue con tabla ejercicios, en que además de tener todos los campos de ésta, contiene el identificador del usuario y la fecha de realización del ejercicio. Se almacenan los 3 últimos resultados de un mismo ejercicio realizado por el mismo usuario, y solo se pueden modificar los últimos resultados de un mismo ejercicio

-**tabla_valoracion:** tabla creada por *DBAdapter.java*, a través de la clase *TablaValoración.java*. Almacena la valoración que recibe un ejercicio al ser realizado por un usuario. Contiene el identificador del usuario, el identificador del ejercicio y la valoración. A partir de esta tabla se obtiene el archivo CSV, necesario para las recomendaciones generadas por los algoritmos de Mahout. Además de ser usada para la recomendación por el método 3. Un mismo ejercicio no está valorado dos veces por un mismo usuario, siempre se actualiza a la última valoración conseguida.

-tabla_zonas: esta tabla es creada por *DBAdapter.java*, a través de la clase *TablaZonasUsuario.java*. Almacena las zonas de trabajo en las que cada usuario ha realizado ejercicios. Contiene el nombre de usuario y la zona de entrenamiento. Si al eliminar ejercicios de un entrenamiento, desaparecen todos los de una zona de entrenamiento, en la tabla se elimina esa zona para el usuario en cuestión.

4.5. Módulo de recomendación

La funcionalidad de este módulo se encuentra dentro del nivel de Modelo. En la siguiente imagen se muestra el diagrama de clases, indicando las clases que interactúan para generar una recomendación:

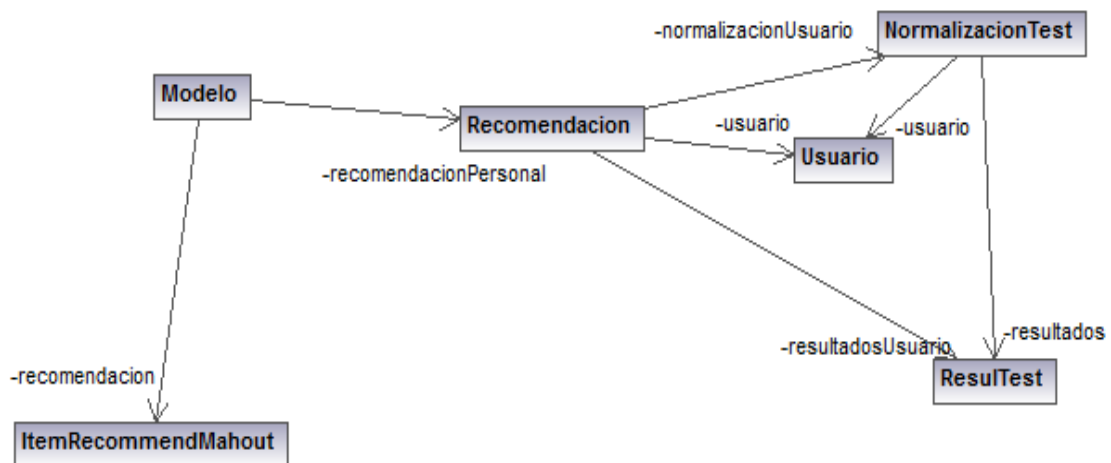


Figura 4.6. Clases que intervienen en la recomendación.

En lo referente a la recomendación, la clase *Modelo.java* contiene dos atributos: *recomendación* y *recomendacionPersonal*, que son los encargados de obtener una lista de ejercicios recomendados. En esta clase se crea el archivo CSV necesario para la recomendación, con las valoraciones de los ejercicios.

La clase *ItemRecommendMahout.java*, contiene los métodos de recomendación 1 y 2, que son los que utilizan los algoritmos de Mahout. Como se ha mencionado antes, en la clase *Modelo.java* se encuentra el atributo *recomendacion*, el cual va a ser el encargado de generar la lista de los ejercicios recomendados dependiendo del método de recomendación (el 1 o el 2) elegido por el entrenador.

La clase *Recomendacion.java* es la encargada de gestionar la recomendación por el método 3. Contiene el algoritmo desarrollado para la optimización del rendimiento deportivo. El atributo *recomendacionPersonal* de la clase *Modelo.java* es quien se va a encargar de generar la lista de ejercicios recomendados.

Capítulo 5

Resultados

5.1. Introducción

En este capítulo se van a explicar y analizar la metodología seguida para la evaluación y análisis de la aplicación desarrollada. La aplicación utiliza 3 algoritmos de recomendación, pero sólo uno de los tres (Método 3) está creado específicamente para lo que nos atañe en este TFG, la optimización del rendimiento deportivo. Los otros dos, al ser algoritmos ya implementados pertenecientes a la librería de Mahout, son utilizados para comparar resultados y ayudar a comprender la base del desarrollo del algoritmo del Método 3.

5.2. Metodología de evaluación

Para poder evaluar los resultados obtenidos en el uso de esta aplicación, se hizo entrega de la misma a un entrenador conocedor de la materia. El entrenador, al cabo de unos días nos entrega la BB.DD generada de las operaciones que ha realizado con la aplicación. Estas son las características que encontramos en esa BB.DD:

- Introdujo hasta 8 usuarios en el sistema, pero con dos de ellos (Usuario2 y Usuario7) no realizó ningún entrenamiento ni test de evaluación inicial, y la aplicación no les recomendaba nada por lo que no se van a tener en cuenta a la hora de analizar los resultados.
- Según los test de evaluación inicial que han realizado los 6 usuarios se observa que:
 - ✓ El Usuario1 y Usuario3 tienen el mismo perfil según condición física, y por tanto son vecinos. Por esto, el algoritmo del método 3 les recomendará los ejercicios mejor valorados realizados por ambos.
 - ✓ El Usuario5 y Usuario8 tienen el mismo perfil según condición física (vecinos). Por esto, el algoritmo del método 3 les recomendará los ejercicios mejor valorados realizados por ambos.

- ✓ Los otros 2 usuarios (Usuario4 y Usuario6) no serían vecinos de ninguno, según sus test realizados, por lo que el algoritmo del método 3 sólo les recomendará los ejercicios más valorados, que hayan realizado ellos mismos.
- El entrenador creó un total de 11 entrenamientos personalizados y añadió un total de 20 ejercicios a esos entrenamientos (que un ejercicio pertenezca a un entrenamiento no quiere decir que esté realizado, y por tanto no tiene valoración).
- El total de ejercicios realizado por los usuarios es de 19, por lo que la BB.DD contaba con 19 valoraciones sobre ciertos ejercicios (algún ejercicio estaba valorado por más de un usuario).
- Además, el entrenador generó 8 entrenamientos utilizando los métodos de recomendaciones automáticas, obteniendo un total de 8 ejercicios recomendados entre todos los entrenamientos generados.

Con el objetivo de ampliar los casos de pruebas de la aplicación, después de analizar la BB.DD. generada, se amplía el conjunto de entrenamientos de manera sintética para los usuarios dados. A partir de esta BBDD, vamos a generar más entrenamientos compuestos de ejercicios de manera automática. Para ello, se va a seguir el siguiente procedimiento:

- ✓ Para el usuario1, se van a generar valoraciones aleatorias totalmente indicando que los resultados de los ejercicios de sus entrenamientos son muy dispares. Estos valores estarán comprendidos entre 1 y 5.
- ✓ Para el Usuario2, como se ha comentado antes, al no tener ejercicios no se hace nada con él.
- ✓ Para el Usuario3, se generan valoraciones de manera creciente, de tal forma que siempre va mejorando en el ejercicio o se mantiene.
- ✓ Para el Usuario4, se generan valoraciones aleatorias, entre 1 y 3, indicando que este usuario no es muy bueno en la realización de sus ejercicios. Nunca mejora los resultados de las variables de entrenamiento. Como mucho, alguna vez los resultados indicarán que trabaja en la zona adecuada de entrenamiento.
- ✓ Para el Usuario5, se generarán valoraciones de tal forma que muestre una progresión en sus resultados decreciente, de forma que siempre empeora en la realización de sus ejercicios.

- ✓ Para el Usuario6, se generan valoraciones aleatorias, entre 3 y 5, indicando que es un usuario con buenos resultados en todos los ejercicios que realiza. Sus resultados muestran que siempre trabaja en la zona de entrenamiento adecuada, y el valor del resto de variables siempre mejora.
- ✓ Para el usuario7, como se ha comentado antes, al no tener ejercicios no se hace nada con él.
- ✓ Para el usuario8, se generan valoraciones de forma aleatoria totalmente como en el caso del Usuario1. Esos valores estarán comprendidos entre 1 y 5.
- ✓ Y por último, para cada uno de estos usuarios, se han realizado tres iteraciones, de tal forma que se generan automáticamente 9 entrenamientos nuevos para cada uno. Es decir, tres entrenamientos (uno por cada método de recomendación) por iteración. Para realizar estas recomendaciones automáticas, sólo se tienen en cuenta aquéllas zonas de entrenamiento del usuario, a las que pertenecen sus ejercicios asignados.
- ✓ Para hacer todo lo anterior, se ha creado una clase *BancoPruebas.java* en la que se implementan las funciones necesarias para conseguirlo. Después de realizar las tres iteraciones, al ver la BB.DD vemos que se han generado 51 valoraciones de ejercicios, o lo que es lo mismo, el total de ejercicios realizados por todos los usuarios es de 51.

5.3. Análisis de resultados

Tras las pruebas realizadas creando más entrenamientos de manera sintética, se muestran al entrenador los ejercicios recomendados para que valore las decisiones tomadas por los distintos métodos de recomendación de la aplicación. De esta manera, el entrenador, sin conocer lo que cada algoritmo hace, indica aquéllos ejercicios que él considera que no deberían ser recomendados en los entrenamientos de ciertos usuarios.

En este apartado se van a mostrar el total de ejercicios recomendados por cada uno de los métodos de entrenamiento, y el número de ejercicios rechazados por el entrenador cada vez que se crea un entrenamiento por un método de recomendación. En la siguiente tabla y gráfico se pueden observar estos datos:

	Método 1	Método 2	Método 3
Nº Total ejercicios propuestos	16	24	87
Nº Total rechazados por el entrenador	7	8	8
% Rechazo	43.75 %	33.33%	9.19%

Figura 5.1. Relación ejercicios recomendados VS ejercicios rechazados.

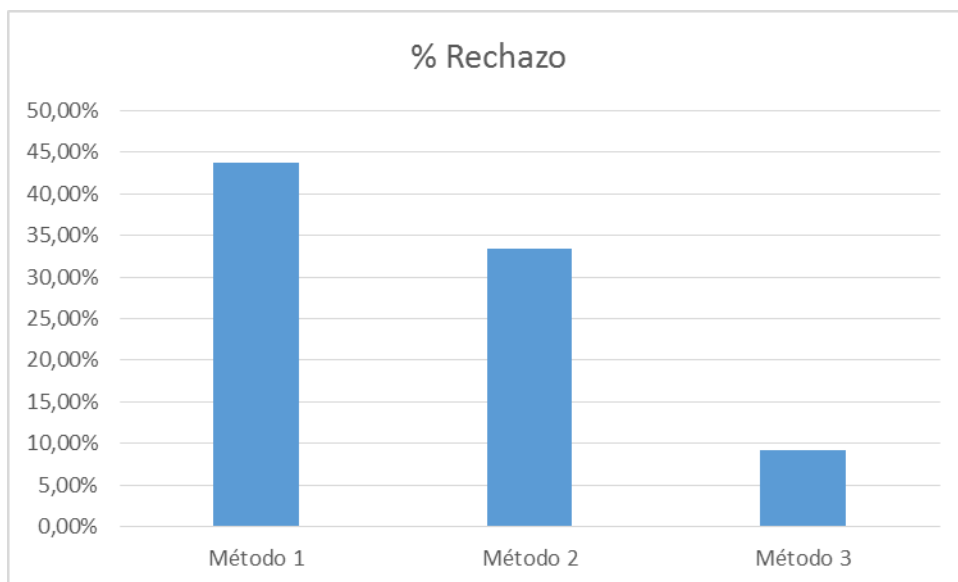


Figura 5.2. Porcentaje de ejercicios rechazados por cada método.

Como se puede observar, el porcentaje de ejercicios rechazados por el experto es superior en el caso de los algoritmos de recomendación 1 y 2, ambos pertenecientes a la librería de Mahout. Sin embargo, podemos ver cómo el algoritmo, del Método 3, propuesto en este TFG alcanza resultados sensiblemente mejores (menos de un 10% de rechazos).

Al conocer la valoración del entrenador sobre los ejercicios propuestos, tenemos que decir que esos resultados podrían ser aún mejores si los algoritmos tuvieran en cuenta el conocimiento propuesto por el entrenador a la hora de decidir los ejercicios. Esto es, añadir una serie de restricciones como las siguientes a los algoritmos:

- No poder recomendar en el mismo entrenamiento, el mismo ejercicio aunque pertenezca a una zona de entrenamiento diferente.
- No repetir ejercicios que sean igual en la actividad, pero distintos en el método de realización.

- Para evitar que el algoritmo recomiende ninguno o pocos ejercicios, tener asignados unos ejercicios de cada zona por defecto, para recomendar. Estos ejercicios deben ser de actividades distintas.

Otro aspecto que no queremos dejar sin comentar es la falta de homogeneidad en el número de ejercicios recomendados por parte de los distintos métodos de recomendación. Esto es debido a que varios entrenamientos recomiendan 0 ejercicios. Estos entrenamientos son generados por los métodos 1 y 2, que pertenecen a los algoritmos de recomendación de la librería Mahout. Estos métodos hacen uso de los coeficientes de Tanimoto y distancia Euclídea respectivamente, como coeficientes de similaridad. Para que se generen recomendaciones, un requisito de estos algoritmos es que debe existir un buen volumen de ítems valorados para crear el CSV, esto es, que en la tabla de valoraciones existan muchos ejercicios valorados.

Estos algoritmos se han probado con un “dataset” de MovieLens [24] que contiene 100.000 ítems valorados y sí se obtienen resultados. Y en nuestra BB.DD, después de tres iteraciones, se ha conseguido llegar tan sólo a las 51 valoraciones. Se tienen que ir añadiendo manualmente más ejercicios a los usuarios por parte del entrenador, además de utilizar los métodos de recomendación, y a la larga se deberían ir obteniendo más ejercicios recomendados. Una posible manera de poder solucionar la falta de datos en nuestra aplicación es, como se ha comentado en las restricciones anteriores, recomendar una lista de ejercicios por defecto si la lista de ejercicios recomendados es vacía. Así cada vez se van valorando más ejercicios por parte de los usuarios, y el número de valoraciones irá creciendo.

Capítulo 6

Conclusiones

6.1.1 Conclusiones

El objetivo principal de este TFG es, tal y como se ha explicado en el presente documento, la creación de un mecanismo computacional capaz de optimizar el rendimiento deportivo de un deportista. El rendimiento deportivo de una persona está sujeto a una serie de condicionantes, como las variables de entrenamiento y las características físicas de la persona, que dificultan su optimización.

Las herramientas computacionales existentes permiten ayudar a un entrenador a decidir sobre el plan de entrenamiento que va a implantar a un deportista, pero no garantizan la optimización del rendimiento del deportista que sigue el plan de entrenamiento, debido a los condicionantes antes mencionados.

En este TFG se ha propuesto una aplicación, portada para Android, que realiza un seguimiento de una serie de usuarios a través de unos entrenamientos compuestos por varios ejercicios de entrenamiento. El desarrollo de esta aplicación se puede decir que es bastante satisfactorio para ser la primera vez que el autor tiene contacto con el desarrollo de programas para esta plataforma.

Se han analizado tres algoritmos de recomendación diferentes, uno de los cuáles ha sido propuesto específicamente en este proyecto para la práctica deportiva, y a la vista de los resultados, se puede ver como la tasa de acierto de éste algoritmo mejora, y cada vez se acerca más a la decisión que tomaría el entrenador experto en la elección de ejercicios de entrenamiento. Además, si a dicho algoritmo se le añaden una serie de restricciones ya mencionadas en el apartado de resultados, elevaría más la tasa de aciertos en la elección de los ejercicios recomendados.

Los resultados de los otros dos algoritmos demuestran que no están orientados para la recomendación de la práctica deportiva, ya que están orientados para sistemas en los que existen un conjunto amplio de usuarios e ítems valorados, y así generar recomendaciones.

6.1.2 Conclusions

As it has already been explained throughout this project, the aim of this study is to create an application that can optimize athletic performance of an sportsman. The performance in the practice of sports is constrained by a set of factors, such as training variables and physical characteristics of the person, which complicate their optimization.

Computational tools help a coach decide on the training plan that will be applicable to an athlete. However, according to the aforementioned factors, they do not guarantee the performance optimization of the athlete who follows the training plan.

In this TFG, it has been proposed an application for Android O.S which keeps track of the users who develop a set of training sessions composed of several exercises. It can be said that the development of this application is quite satisfactory, since it is the first time that the author of this document has contact with development of programs for this platform.

For that purpose, three different recommendation algorithms have been evaluated. One of them has been developed in this Project specifically for the case of sports practice. It can be seen how the success rate for this proposed algorithm is the best for the three evaluations and how it is closer to the recommendation done by the expert human trainer. Moreover, if we apply a set of restrictions already mentioned in point 5 to this algorithm, the hit rate will rise.

The results of the other two algorithms show that they are not intended for the recommendation of the sport performance. These algorithms are oriented to systems where there are a high number of users and items valued in order to get the results.

6.2. Trabajo Futuro

Un posible trabajo futuro podría ser ampliar los ejercicios de la BBDD, para que la aplicación no esté solamente orientada a ejercicios de resistencia. Por tanto, se podrían añadir test de valoración inicial de Fuerza y ejercicios para recomendar de Fuerza. Además se podría sincronizar con dispositivos de monitorización de las variables de entrenamiento, como pulsómetros, etc.

También existe la posibilidad que sea una aplicación colaborativa de tal forma que se puedan compartir los resultados, con resultados de otros usuarios que los esté evaluando un entrenador diferente en cualquier otro punto de la geografía.

Otras mejoras podrían ser traducir la aplicación a diversos idiomas o portarla a otras plataformas móviles como iOS.

6.3. Seguimiento del Proyecto

El código fuente de la aplicación móvil se puede conseguir para uso exclusivo de investigación mandando un correo a:

`jayala@ucm.es`

Para contactar con el autor del proyecto se puede hacer a través de la siguiente dirección:

`jesusmagar@hotmail.com`

Capítulo 7

Manual de uso

7.1 Introducir usuario en la aplicación

Para introducir a un usuario en la aplicación, en el menú principal se pulsa sobre “Datos”, y nos aparecerá una pantalla para introducir los datos de un usuario. Es imprescindible saber que el nombre de usuario es único en la aplicación:



Figura 7.1 Menú principal

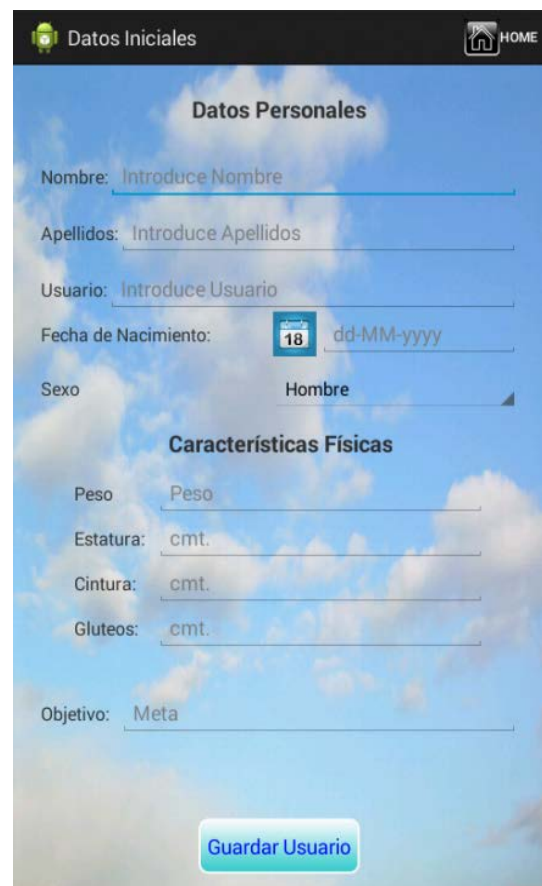


Figura 7.2 Introducir datos

Una vez introducidos los datos pulsamos en “Guardar Usuario” y el usuario quedará almacenado en la BB.DD de la aplicación.

El botón de “HOME” nos lleva al menú principal:



Figura 7.3 Botón menú principal

7.2 Modificar datos de un usuario

Para modificar los datos de un usuario, en el menú principal pulsamos en “Perfil”, y ahí nos aparecerá un buscador para cargar los datos del usuario que seleccionemos. Se puede introducir directamente el nombre de usuario que queremos modificar los datos, o buscarlo:



Figuras 7.4 Buscador de usuarios

Una vez que tenemos seleccionado al usuario en cuestión, pulsamos en “Cargar Usuario”, y nos aparecerán los datos del usuario:



Figura 7.5 Perfil de un usuario

Para modificar cualquier dato, bastaría con pinchar sobre el valor que se quiere cambiar y aparecerá una ventana emergente para poder introducir el nuevo valor:



Figura 7.6 Cambio datos en perfil de un usuario

Una vez modificado todos los valores deseados, pulsamos en “Guardar cambios” y se almacenarán los nuevos datos.

Como se observa en las imágenes también en el apartado de “Perfil” se encuentra el cuestionario que sirve al entrenador para tener más información sobre el usuario, a la hora de crear los entrenamientos. Para contestar a las preguntas, se pulsa sobre cualquiera de ellas, y se introduce la respuesta:

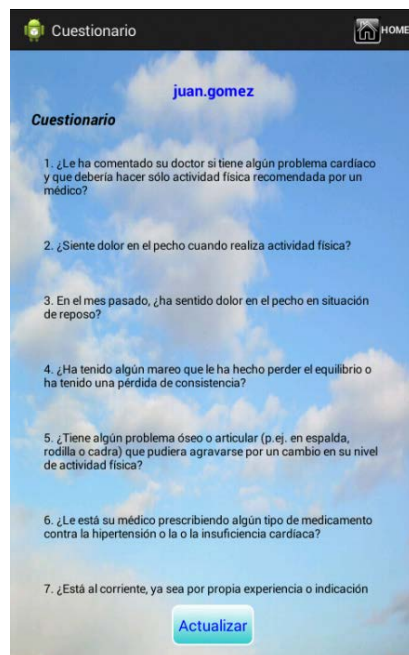


Figura 7.7 Cuestionario

7.3 Test de valoración inicial

Para realizar test de evaluación inicial a un usuario, en el menú inicial se pulsa en “Test Iniciales”, y se selecciona el usuario al que se va a realizar el test, como en el apartado anterior. Si el usuario no se encuentra en el sistema, hay que crearlo. Una vez seleccionado nos aparece una ventana con un menú de “pestañas” en las que hay distintos test de evaluación inicial:

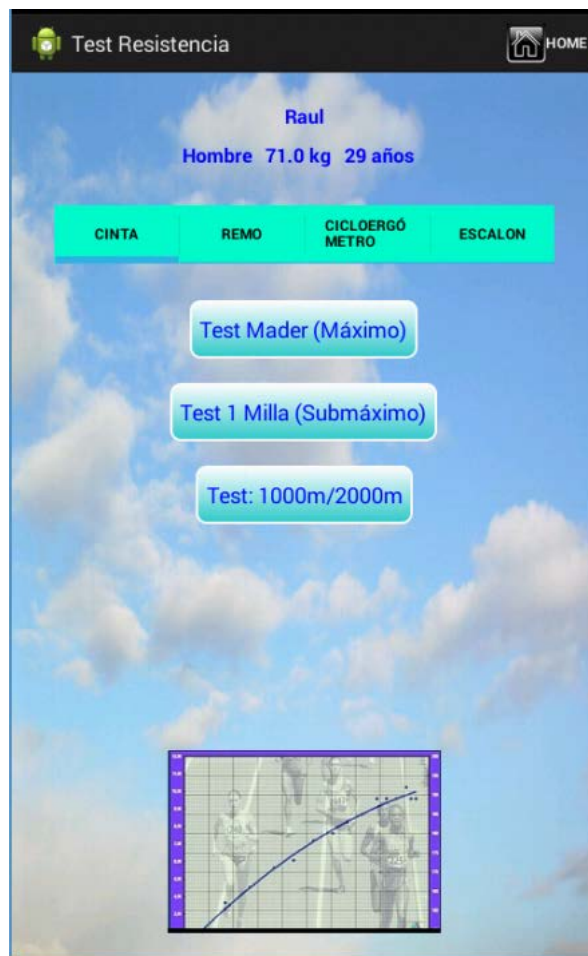


Figura 7.8 Test valoración inicial

Seleccionamos el test que queremos que evalúe al usuario, e introducimos los parámetros que nos solicite el test. Después de obtener la clasificación del test, pulsamos en “Guardar” para almacenarlo en la BB.DD. Por ejemplo para el test de O’neill:



Figura 7.9 Ejemplo test valoración inicial

7.4 Añadir entrenamientos y ejercicios

Una vez que tenemos usuarios con test realizados para saber su condición física (necesario para poder recomendar ejercicios por el método 3), podemos crear entrenamientos compuestos por ejercicios de entrenamiento. Para ello, en el menú principal pulsamos en “Entrenamientos” y seleccionamos el usuario como en los apartados anteriores. Después, nos aparecerá la ventana para crear entrenamientos. Pulsamos en “añadir” y aparece la ventana para elegir la forma de creación del entrenamiento:



Figura 7.10 Añadir entrenamiento

Existen 4 maneras de crear un entrenamiento: personalizado, método 1, método 2, y método 3.

Personalizado:

Se introduce un nombre al entrenamiento y se selecciona el método de recomendación "Personalizado".

Método 1:

Se introduce un nombre al entrenamiento y se selecciona el "Método 1" de recomendación. Este método utiliza un algoritmo de Mahout en el que el factor de similitud es el coeficiente de Tanimoto. Además, se seleccionan las zonas de trabajo para recomendar ejercicios sólo de esas zonas:



Figura 7.11 Añadir entrenamiento por método de recomendación

Para que este método recomiende ejercicios, el usuario debe haber realizado ejercicios anteriormente, y en la BB.DD, debe existir un volumen de ejercicios valorados considerable. La aplicación lo avisa con un mensaje cuando no se recomiendan ejercicios.

Método 2:

Se introduce un nombre al entrenamiento y se selecciona el “Método 2” de recomendación. Este método utiliza un algoritmo de Mahout en el que el factor de similitud es la distancia Euclídea. Además, se seleccionan las zonas de trabajo para recomendar ejercicios sólo de esas zonas, como en el método anterior. Para que este método recomiende ejercicios, el usuario debe haber realizado ejercicios anteriormente, y en la BB.DD, debe existir un volumen de ejercicios valorados considerable. La aplicación lo avisa con un mensaje cuando no se recomiendan ejercicios.

Método 3:

Se introduce un nombre al entrenamiento y se selecciona el “Método 3” de recomendación. Este método utiliza un algoritmo propuesto específicamente para la práctica deportiva, en el que factor de similaridad se basa en los resultados obtenidos en los test de valoración inicial de los usuarios. Recomienda ejercicios siempre y cuando el usuario haya realizado ejercicios previamente, o existan otros usuarios en el sistema con una condición física parecida y que tengan ejercicios realizados. Como en los métodos anteriores, se seleccionan las zonas de entrenamiento de las que quiere recomendar ejercicios.

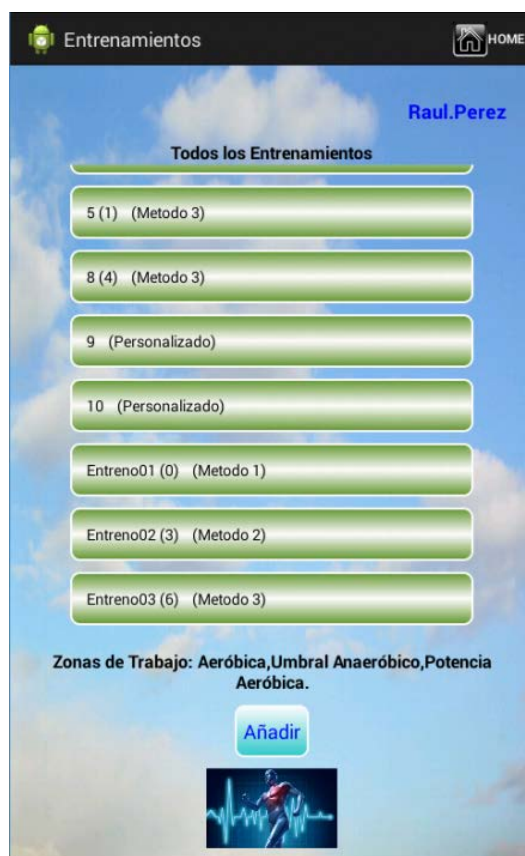


Figura 7.12 Ventana entrenamientos de usuario

El número entre paréntesis al lado del nombre del entrenamiento, hace referencia al número de ejercicios que ha recomendado ese método. Esto se hizo para seguir un trazado de lo que va recomendando la aplicación para después poder ser evaluada. El entrenador, después de ver los ejercicios recomendados, puede eliminar los que considere oportunos y añadir otros que crea conveniente.

Los entrenamientos se pueden borrar simplemente manteniendo pulsado el entrenamiento o ejercicio, y nos aparecerá un mensaje si deseamos eliminar:



Figura 7.13 Eliminar entrenamientos de usuario

Para añadir ejercicios a un entrenamiento se pulsa en el entrenamiento donde se quieren añadir y pulsando en el botón "Añadir", se añaden los ejercicios deseados seleccionando la zona, método y actividad:



Figura 7.14 Añadir ejercicio

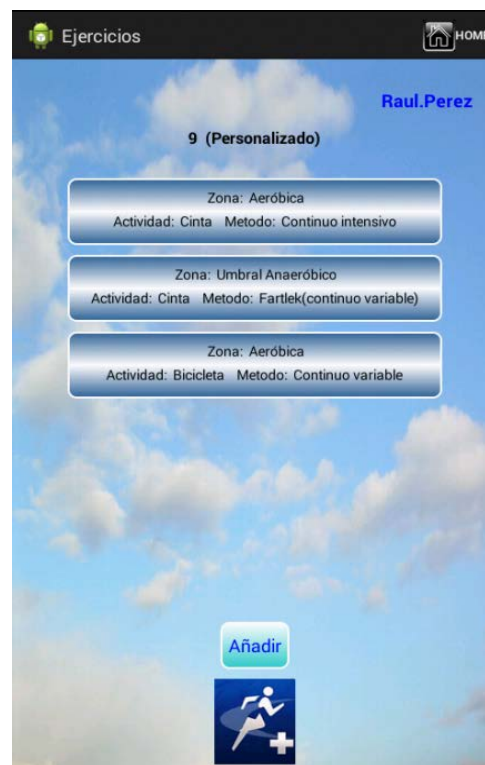
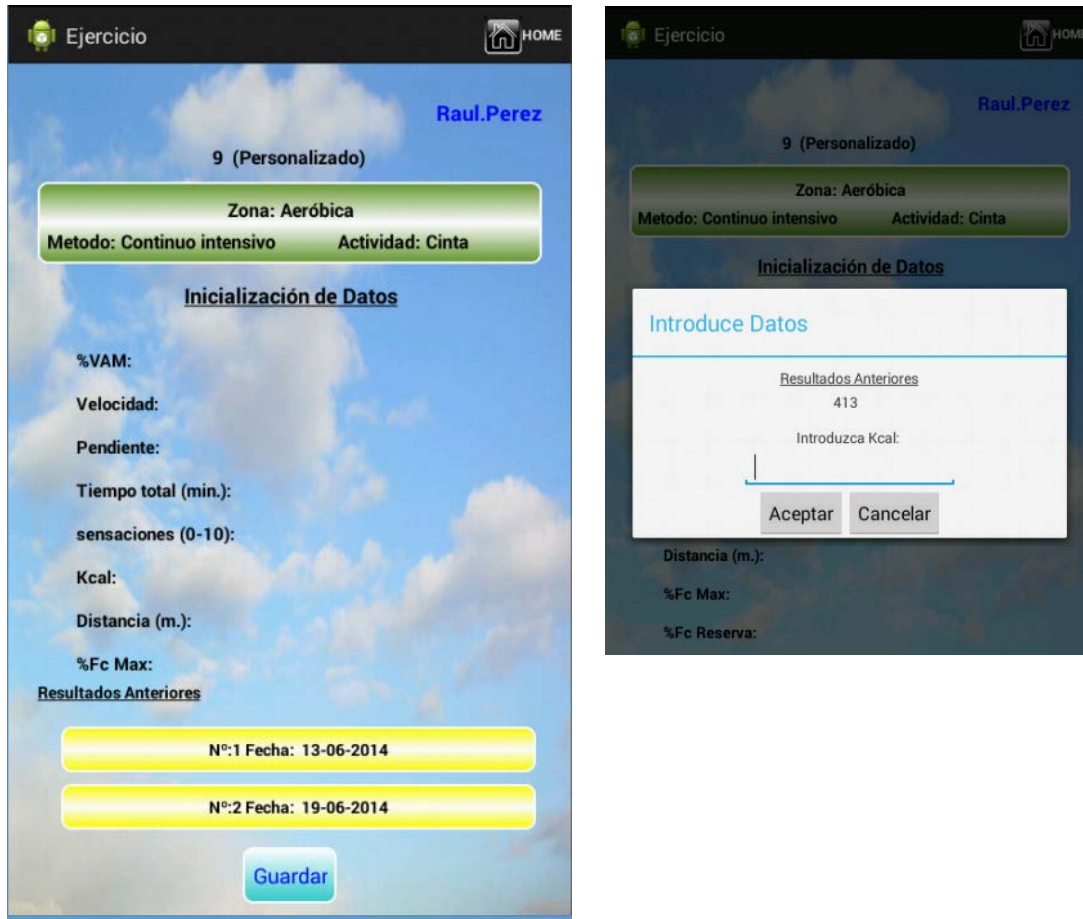


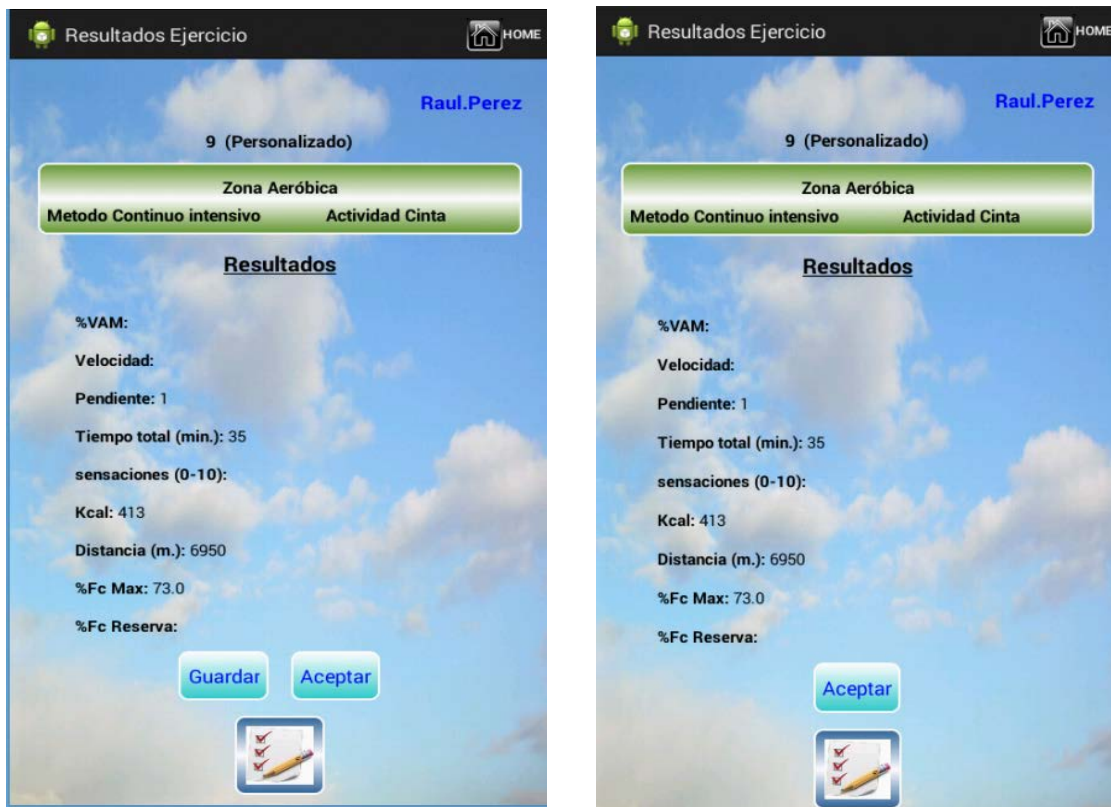
Figura 7.15 Ejercicios de un entrenamientos

Por último nos queda introducir los valores, de las variables de entrenamiento, en los ejercicios realizados por el usuario, para que puedan ser valorados. Pulsamos sobre el ejercicio y nos aparecen las variables para introducir los valores pulsando sobre ellas:



Figuras 7.16 Ventana introducir valores de ejercicio

También se puede ver en la parte inferior, que existen resultados anteriores de ese ejercicio para ese usuario. Se van a mostrar los resultados de las dos últimas veces que el usuario ha realizado el ejercicio, y se puede editar los resultados del último solamente. Esto es por razones a la hora de evaluar la tendencia de los resultados. Pulsando sobre la fecha, podemos ver los valores de las variables cuando se realizó el ejercicio:



Figuras 7.17 Ventana para ver valores de ejercicio realizado

Si es el último ejercicio realizado, como se ha dicho anteriormente, la aplicación ofrece la posibilidad de editar los valores. Eso se ve si aparece el botón de “Guardar”. Si no aparece, quiere decir que no es el último ejercicio realizado y es sólo de consulta para el entrenador.

Una vez introducidos los valores en ciertas variables de un ejercicio (no es necesario rellenar todas), se pulsa en “Guardar” y ese ejercicio obtiene una valoración por parte del usuario.

ANEXO 1

Test de valoración inicial: Todos los test se normalizan con valores de 1 a 6, haciendo el 1 al peor estado de forma y el 6 al mejor.

- 1- **Test O'neill:** Es un test de remo que consiste en recorrer durante 4 minutos la máxima distancia posible. Los valores que puede tomar el test de O'Neill, dependiendo de la distancia recorrida, el sexo, peso y edad del deportista, son los siguientes [25]:

Mujeres	Excelente	Bueno	Superior a la media	Media	Bajo
19-29 Lwt	1078	1038	958	878	798
30-39 Lwt	1050	1010	929	849	769
40-49 Lwt	1030	990	909	829	749
50-59 Lwt	1011	971	891	811	730
60-69 Lwt	992	951	871	791	711
70-79 Lwt	973	933	852	772	692
19-29 Hwt	1105	1065	985	905	824
30-39 Hwt	1057	1017	936	856	776
40-49 Hwt	1044	1004	923	843	763
50-59 Hwt	1037	997	917	836	756
60-69 Hwt	1023	983	903	823	743
70-79 Hwt	944	904	823	743	663

Lwt: 61.5 kg o menos

Hwt: más de 61.5 kg

Figura A.1.1. Tabla valores test O'neill (Mujeres) [25].

Hombres	Excelente	Bueno	Superior a la media	Media	Bajo
19-29 Lwt	1243	1203	1122	1042	962
30-39 Lwt	1227	1187	1107	1026	946
40-49 Lwt	1208	1168	1087	1007	927
50-59 Lwt	1172	1132	1051	971	891
60-69 Lwt	1131	1091	1011	931	850
70-79 Lwt	1052	1012	931	851	771
80-89 Lwt	953	912	832	752	672
19-29 Hwt	1281	1241	1161	1080	1000
30-39 Hwt	1237	1197	1117	1037	957
40-49 Hwt	1219	1178	1098	1018	938
50-59 Hwt	1182	1142	1062	982	901
60-69 Hwt	1141	1101	1021	940	860
70-79 Hwt	1061	1020	940	860	780
80-89 Hwt	993	953	872	792	712

Lwt: 61.5 kg o menos

Hwt: más de 61.5 kg

Figura A.1.2. Tabla valores test O'neill (Hombres) [25].

Junior	Excelente	Bueno	Superior a la media	Media	Bajo
Mujer 12	886	846	766	685	605
Mujer 13	956	916	835	755	675
Mujer 14	999	955	885	795	725
Mujer 15	1042	1001	921	841	761
Mujer 16	1074	1034	954	874	793
Mujer 17	1109	1069	988	908	828
Mujer 18 Lwt	1046	1006	926	846	765
Mujer 18 Hwt	1100	1060	980	899	819
Hombre 12	888	848	768	687	607
Hombre 13	1008	967	887	807	727
Hombre 14	1095	1055	974	894	814
Hombre 15	1171	1130	1050	970	890
Hombre 16	1212	1172	1092	1011	931
Hombre 17	1251	1211	1130	1050	970
Hombre 18 Lwt	1221	1180	1100	1020	940
Hombre 18 Hwt	1281	1241	1161	1081	1000

Lwt: 61.5 kg o menos

Hwt: más de 61.5 kg

Figura A.1.3. Tabla valores test O'neill (Junior) [25].

La normalización de este test es la siguiente:

EXCELENTE	6
BUENO	5
SUPERIOR A LA MEDIA	4
MEDIO	3
BAJO	2
NO CALIFICADO ≈ MUY BAJO	1

Figura A.1.4. Tabla clasificación y valores normalizados O'neill.

2- **Test de cinta y remo:** Según los tiempos o distancias recorridas, los resultados de los test para saber el estado de forma del deportista se normalizan como se indica en la tabla (valores proporcionados por el entrenador experto):

	6	5	4	3	2	1
	EXCELENTE	MUY BUENO	BUENO	REGULAR	MALO	MUY MALO
CINTA 1000	<=2.3'	3.3'	4,3'	5.3'	6,3'	>7,3'
CINTA 2000	<=5'	7'	9'	11'	13'	>15'
REMO 1000	<=3'	4'	5'	6'	7'	>8'
REMO 2000	<=6.3'	8.3	10.3'	12.3'	14,3	>16,3'
REMO 1'	>=300 (m)	250 (m)	200 (m)	150 (m)	100 (m)	<50 (m)
REMO 2'	>=600 (m)	500 (m)	400 (m)	300 (m)	200 (m)	<100 (m)
REMO 3'	>=900 (m)	750 (m)	600 (m)	450 (m)	300 (m)	<150 (m)

Figura A.1.5. Tabla clasificación y valores normalizados de test de cinta y remo.

3- **Test de Mader, Astrand y Ryhming (Submáximo), Hollman, 1 milla y Astrand y Ryhming (escalón).** Todos estos test calculan el $VO_{2Máx}$ del deportista y la escala de valores es como se indica a continuación [26]:

EDAD	VARONES VO_{2max} ($ml \cdot min^{-1} \cdot kg^{-1}$)					MUJERES VO_{2max} ($ml \cdot min^{-1} \cdot kg^{-1}$)				
	BAJO	ALGO BAJO	MEDIO	ALTO	MUY ALTO	BAJO	ALGO BAJO	MEDIO	ALTO	MUY ALTO
20-29	<38	39-43	44-51	52-56	>56	<28	29-34	35-43	44-48	>48
30-39	<34	35-39	40-47	48-51	>51	<27	28-33	34-41	42-47	>47
40-49	<30	31-35	36-43	44-47	>47	<25	26-31	32-40	41-45	>45

Figura A.1.6. Tabla clasificación test de $VO_{2Máx}$ [26].

Normalizando los resultados de la siguiente manera:

MUY ALTO	6	ALGO BAJO	3
ALTO	5	BAJO	2
MEDIO	4	NO CALIFICADO ≈ MUY BAJO	1

Figura A.1.7. Tabla clasificación normalizada de test de $VO_{2Máx}$.

Es importante mencionar, que para calcular el resultado de los test de **Astrand y Ryhming (escalón y cicloergómetro)**, se hace uso de los siguientes Nomogramas de Astrand y Ryhming:

FC	VARONES VO _{2max} (litros/min)					MUJERES VO _{2max} (litros/min)				
	50 w	100 w	150 w	200 w	250 w	50 w	100 w	150 w	200 w	250 w
120	2.2	3.5	4.8			2.6	3.4	4.1	4.8	
125	2.0	3.2	4.4	5.9		2.3	3.0	3.7	4.4	
130	1.9	3.0	4.1	5.5		2.1	2.7	3.4	4.0	4.7
135	1.7	2.8	3.8	5.1		2.0	2.6	3.1	3.7	4.3
140	1.6	2.6	3.6	4.8	6.0	1.8	2.4	2.8	3.4	4.0
145		2.4	3.4	4.5	5.6	1.6	2.2	2.7	3.2	3.7
150		2.3	3.2	4.2	5.3		2.0	2.5	3.0	3.5
155		2.2	3.0	4.0	5.0		1.9	2.4	2.8	3.2
160		2.1	2.8	3.8	4.8		1.8	2.2	2.6	3.0
165		2.0	2.7	3.6	4.5		1.7	2.1	2.5	2.9
170		1.8	2.6	3.4	4.3		1.6	2.0	2.4	2.7

Figura A.1.8 Nomograma de Astrand y Ryhming cicloergómetro [26]

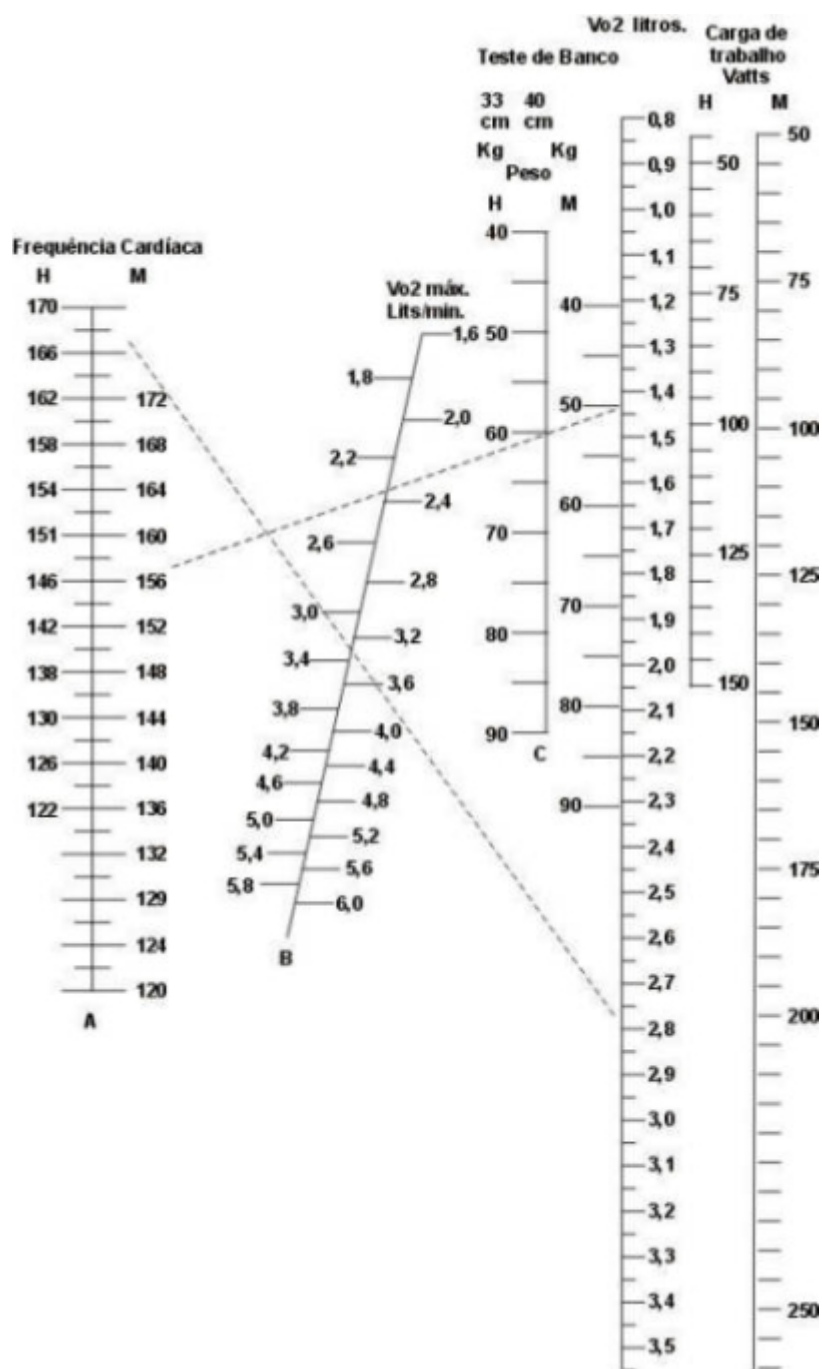


Figura A.1.9. Nomograma de Astrand y Ryhming escalón (imagen proporcionada por el entrenador)

El resultado del test es el punto de corte de la recta fija VO₂Máx, con la recta formada que une el peso con el número de pulsaciones de la persona en el minuto 5 de comenzar a realizar el ejercicio.

ANEXO 2

Zonas y métodos de entrenamiento de la resistencia cardiorrespiratoria [27]:

ZONA O RITMO	ABREV.	OBJETIVO	INTENSIDAD				MÉTODOS DE ENTRENAMIENTO PARA SU DESARROLLO*
			%VO _{2max} %FC _{reserva}	%FC _{max}	%VAM	[Lact] (mmol·L ⁻¹)	
Aeróbico lipolítico	R1	<ul style="list-style-type: none"> • Aumentar la eficiencia aeróbica. • Mejorar la capacidad de soportar esfuerzos aeróbicos prolongados mediante la mejora de la oxidación de grasas y el aumento de sus depósitos. 	65-75%	70-80%	65-75%	1-2	<ul style="list-style-type: none"> • Continuo Uniforme Extensivo • Continuo Uniforme Intensivo • Continuo Variable 1
Umbral anaeróbico	R2	<ul style="list-style-type: none"> • Aumentar la capacidad de soportar esfuerzos aeróbicos prolongados en condiciones de umbral anaeróbico. • Mejora de la oxidación del glucógeno y sus depósitos. • Adaptaciones centrales: ↑Afinidad por la hemoglobina, ↑Difusión pulmonar, ↑Volumen sistólico, ↑Gasto cardíaco, ↑Volemia. 	75-85%	80-90%	75-85%	2-4	<ul style="list-style-type: none"> • Continuo Variable 1 • Continuo Variable 2 • Interválico Extensivo Largo
Potencia aeróbica	R3	<ul style="list-style-type: none"> • Aumentar la capacidad de soportar esfuerzos en condiciones próximas e iguales al VO_{2max}. • Adaptaciones periféricas: ↑densidad capilar, ↑densidad mitocondrial, ↑enzimas oxidativas, ↑reservas de glucógeno. 	90-95%	95-98%	90-95%	4-6	<ul style="list-style-type: none"> • Interválico Extensivo Largo • Interválico Extensivo Medio
	R3+	<ul style="list-style-type: none"> • Adaptaciones periféricas: ↑densidad capilar, ↑densidad mitocondrial, ↑enzimas oxidativas, ↑reservas de glucógeno. 	100%	100%	100%	6-8	<ul style="list-style-type: none"> • Interválico Extensivo Medio • Interválico Intensivo Corto
Capacidad anaeróbica	R4	<ul style="list-style-type: none"> • Mejorar la capacidad de tolerar elevadas concentraciones de acidosis metabólica. Sistema Buffer. • Aumentar la capacidad glucolítica. Enzimas glucolíticas. 	-	-	105-120%	8-14	<ul style="list-style-type: none"> • Interválico Extensivo Medio • Interválico Intensivo Corto • Repeticiones Largo

Figura A.2.1. Tabla zonas de entrenamiento.

Bibliografía

- [1] *Manual de Metodología del Entrenamiento Deportivo*, Dietrich Martin:
http://books.google.es/books?id=ehXzkJzpQIC&pg=PT25&lpg=PT25&dq=definir+rendimiento+deportivo&source=bl&ots=1LCg8yljz2&sig=ntALnHgc_v-Nb611XROqUZBTLgY&hl=es&sa=X&ei=TAWKU_S_FOMu0QWH5YG4DQ&ved=0CG0Q6AEwCA#v=onepage&q=definir%20rendimiento%20deportivo&f=false
- [2] *Entrenamiento por zonas: Un modelo sencillo y asequible*: <http://g-se.com/es/org/garciaverdugo-com/blog/el-entrenamiento-de-resistencia-por-zonas-un-modelo-sencillo-y-asequible>
- [3] *Guia Fitness*: <http://guiafitness.com/ejercicios-resistencia.html>
- [4] *Universitat Pompeu Fabra*: <http://www.upf.edu/hipertextnet/numero-2/recomendacion.html>
- [5] *Escuela Superior Politécnica del Litoral*:
http://www.cib.espol.edu.ec/Digipath/D_Tesis_PDF/D-92363.pdf
- [6] *Suárez I., Plasencia A., Aprendizaje automático*:
http://www.icimaf.cu/revista_cca/sites/default/files/2M_0.pdf
- [7] *Wikipedia Android*: <http://es.wikipedia.org/wiki/Android>
- [8] *Software Development Kit*: <http://developer.android.com/sdk/index.html>
- [9] *Spain Movil*: <http://www.spainmovil.es/android/apps/n165/entrenamiento-flanax-apps-gratis-android.html>
- [10] *Wikipedia Last.fm*: <http://es.wikipedia.org/wiki/Last.fm>
- [11] *Manual del Entrenador Personal*:
http://www.felipeisidro.com/articles/manual/manual_del_entrenador_personal.pdf
- [12] *Sergio Manuel Galán S., Filtrado Colaborativo y Sistemas de Recomendación*:
<http://www.it.uc3m.es/~jvillena/irc/practicas/06-07/31.pdf>
- [13] *Carrillo G., Ochoa X., Universidad Oberta de Catalunya*:
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18962/6/gcarrillobTFM0113memoria.pdf>
- [14] *Su, X., Khoshgoftaar, T., A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence Vol 2009. (2012).*

- [15] *Cataldo Musto, Apache Mahout - Tutorial (2014)*
<http://www.slideshare.net/Cataldo/apache-mahout-tutorial-recommendation-20132014>
- [16] *La frecuencia cardiaca:*
<http://www.elromeralcastejon.com/archivos/FORMACION/LA%20FRECUENCIA%20CARDIACA.pdf>
- [17] *Wikipedia VO₂Max:* http://es.wikipedia.org/wiki/VO2_m%C3%A1x
- [18] *Escalas de Borg:*
<http://www.bkool.com/content/es/Asesoramiento/Entrenamiento/ESFUERZO%20Y%20ESCALAS%20DE%20BORG/ESCALAS-DE-BORG/>
- [19] *Análisis de tendencia para series de tiempo:*
<http://www.monografias.com/trabajos87/analisis-tendencia-series-tiempo/analisis-tendencia-series-tiempo.shtml>
- [20] *Owen S. Anil R., Dunning T., Friedman E., Mahout in Action. 2012 by Manning Publications.*
- [21] <https://builds.apache.org/job/Mahout-Quality/javadoc/index.html?org/apache/mahout/cf/taste/impl/similarity/EuclideanDistanceSimilarity.html>
- [22] *Proyecto Simio: Programación en Android:*
<http://www.proyectosimio.com/programacion-android-base-de-datos-i-modelo-vista-controlador/>
- [23] *Sgoliver.net BB.DD en Android:* <http://www.sgoliver.net/blog/?p=1611>
- [24] *Datasets MovieLens GroupLens:* <http://grouplens.org/datasets/movielens/>
- [25] *Test O'neill:* <https://sites.google.com/site/nrctrainingprogram/training-information/o-neil-fitness-test>
- [26] <http://articulos.forociclista.com/prueba-de-astrand-y-ryhming/>
- [27] *Zonas y Métodos de entrenamiento, García Pallarés J., Morán Navarro R., Pérez Caballero C.:*
http://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCkQFjAA&url=http%3A%2F%2Fdigitum.um.es%2Fxmlui%2Fbitstream%2F10201%2F25373%2F6%2FZonas%2520y%2520M%25C3%25A9todos%2520de%2520Entrenamiento%2520Cardiorrespiratorio.pdf&ei=LuaQU67iBZSysATrjoHwCw&usg=AFQjCNGf-8a_0YG5W5kBwmQWajGwiuqoEQ&bvm=bv.68445247,d.b2k