



**Universidad
Zaragoza**

Trabajo Fin de Grado

Sistema de Recomendación de Ejercicios de Entrenamiento de
Voleibol

Recommendation System for Volleyball Training Exercises

Autor

Israel Francisco Gimeno Franco

Director

Sergio Ilarri Artigas

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2023

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer la labor de Sergio Ilarri, director de este Trabajo Fin de Grado. Su implicación, ayuda y paciencia han sido indispensables para poder llevar a cabo el proyecto. Sin las directrices e indicaciones proporcionadas no habría sido capaz de superar los desafíos que iban surgiendo en el camino.

Agradecer a mi familia su continuo apoyo, que me ha servido de fuente de motivación para seguir trabajando en los peores momentos. Agradezco especialmente a mi madre por sus palabras de ánimo y comprensión, y a mi hermana y cuñado por la confianza depositada.

También querría agradecer a mis compañeros de trabajo y a los jugadores de los distintos equipos del Club Voleibol Barrio Santa Isabel. Vuestro apoyo y vuestros ánimos me han permitido seguir adelante tanto en el grado como en la realización del TFG.

Por último, debo agradecer el apoyo brindado a este trabajo por parte del proyecto Next-gEnerATion dAta Management to foster suitable Behaviors and the resilience of cltizens against modErN ChallEnges (NEAT-AMBIENCE)(PID2020-113037RB-I00 / AEI / 10.13039/501100011033) y por parte del grupo COSMOS (última referencia: T64_23R), subvencionado por el Departamento de Ciencia, Universidad y Sociedad del Conocimiento del Gobierno de Aragón.

Sistema de Recomendación de Ejercicios de Entrenamiento de Voleibol

RESUMEN

En la actualidad, la mayor parte de la población utiliza aplicaciones y plataformas que emiten recomendaciones de contenido. Por ejemplo, *Youtube* permite ver vídeos relacionados con aquellos que un usuario ya ha visualizado, y *Netflix* recomienda películas o series que ha calculado que al usuario le pueden interesar.

Dentro del mundo deportivo, el uso de herramientas tecnológicas ha avanzado en los últimos años, permitiendo obtener datos precisos rápidamente con los que poder realizar multitud de tareas, como hacer análisis del juego o programar entrenamientos, todo ello con el fin de aumentar el rendimiento de los jugadores, prevenir lesiones y aumentar el conocimiento de la población en lo relacionado con el deporte.

Este trabajo busca aunar los sistemas de recomendación con el voleibol, permitiendo a un entrenador generar una rutina de entrenamiento con varios ejercicios que le puedan resultar de interés tanto a él como a los jugadores a su cargo. Para ello, se ha generado un conjunto de datos recopilando más de 300 ejercicios a través de documentos relacionados y de entrenadores reales, que se han ofrecido a aportar las rutinas que siguen a diario en sus entrenamientos.

Se han utilizado diferentes enfoques en la implementación del sistema recomendador. Por un lado, se ha desarrollado un recomendador basado en conocimiento, que utiliza las preferencias indicadas por el entrenador y las características del ejercicio de voleibol. Por otro lado, se ha ampliado el recomendador con técnicas basadas en contenido, las cuales utilizan las descripciones de los ejercicios para buscar otros similares que puedan interesar al entrenador. En este caso, los principales algoritmos usados en la minería de estas descripciones textuales han sido la Asignación Latente de Dirichlet (LDA) y la Frecuencia de Término-Frecuencia Inversa de Documento (TF-IDF).

Para conseguir este objetivo, se ha creado una aplicación informática de escritorio y portable que permite a un entrenador generar rutinas de entrenamiento, consultar los distintos ejercicios que se encuentran en la base de datos, añadir los que vea necesario y eliminarlos. También se ha creado una aplicación Android que permite a los jugadores consultar esos ejercicios y puntuarlos, con el fin de generar información útil para las siguientes recomendaciones.

Para evaluar las distintas técnicas empleadas en el sistema de recomendación, se han comparado los diferentes algoritmos usados para determinar cuál se ajusta mejor al perfil de usuario del entrenador. Por último, estas aplicaciones desarrolladas han sido proporcionadas a jugadores y entrenadores reales para que validen su funcionamiento, usabilidad y utilidad en un entorno deportivo.

Índice de contenidos

1. Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Herramientas utilizadas	3
1.4 Diagrama de Gantt	3
2. Investigación sobre sistemas de recomendación	5
2.1 Tipos o técnicas de recomendación.....	5
2.2 Sistemas recomendadores colaborativos	6
2.3 Sistemas recomendadores basados en contenido.....	7
2.4 Sistemas recomendadores basados en conocimiento	8
2.5 Sistemas recomendadores en el deporte.....	9
2.6 Recomendaciones grupales	10
3. Análisis del sistema.....	11
3.1 Requisitos funcionales.....	11
3.2 Requisitos no funcionales	12
3.3 Diagrama de casos de uso	13
4. Diseño del sistema.....	15
4.1 Arquitectura Modelo-Vista-Controlador.....	15
4.2 Diseño de la base de datos	16
4.2.1 Esquema entidad-relación.....	16
4.2.2 Esquema relacional.....	17
4.3 Diseño de las interfaces gráficas de usuario.....	20
4.3.1 Prototipado de la interfaz de usuario de la aplicación de escritorio	20
4.3.2 Prototipado de la interfaz de usuario de la aplicación móvil	22
5. Implementación del recomendador del sistema	24
5.1 Población de la base de datos	24
5.2 Recomendador utilizado en la aplicación.....	25
5.3 Algoritmos utilizados en el recomendador	26
5.3.1 Algoritmo por pesos	26
5.3.2 Algoritmo TF-IDF.....	27
5.3.3 Algoritmo LDA.....	28
6. Pruebas realizadas.....	30
6.1 Pruebas de utilidad de la aplicación.....	30

6.2 Resultados de las pruebas realizadas	30
7. Conclusiones y Trabajo Futuro.....	35
7.1 Dificultades técnicas encontradas	35
7.2 Líneas futuras.....	35
7.3 Valoración personal.....	37
Referencias	38
Anexos.....	42
A. Manual de usuario	42
A.1 Manual de usuario de la aplicación de escritorio	42
A.2 Manual de usuario de la aplicación móvil	46
B.1 Diagrama de actividades del caso de uso CU01 (Obtener recomendación)	48
B.2 Diagrama de actividades del caso de uso CU02 (Consultar ejercicios)	49
B.3 Diagrama de actividades del caso de uso CU03 (Puntuar ejercicio).....	50
B.4 Diagrama de actividades del caso de uso CU04 (Añadir ejercicio).....	51
B.5 Diagrama de actividades del caso de uso CU05 (Eliminar ejercicio)	52
C. Esquema físico en PostgreSQL.....	53
C.1 Tabla Ejercicio de la base de datos.....	53
C.2 Tabla Usuario de la base de datos.....	53
C.3 Tabla Rutina de la base de datos.....	54
C.4 Tabla RutinaEjercicio de la base de datos	54
C.5 Tabla UsuarioEjercicio de la base de datos	55
D. Ventanas y cuadros de diálogo de la aplicación de escritorio.....	56
E. Pantallas o actividades de la aplicación móvil.....	59
F. Formularios	61
G. Funcionamiento de LDA	64
H. Pruebas con perfiles de entrenador.....	65
H.1 Perfil de entrenador con preferencia por ejercicios de recepción	65
H.2 Perfil de entrenador con preferencia por ejercicios grupales	66
H.3 Perfil de entrenador con preferencia por ejercicios de remate, individualizados, recomendados para personas jóvenes y físicos	67
H.4 Perfil de entrenador con preferencia por ejercicios tácticos complejos, de defensa y recepción, de larga duración y recomendados para personas adultas	68

1. Introducción

En la Sección 1.1 se introduce la motivación del proyecto. En la Sección 1.2 se mencionan los objetivos del proyecto. En la Sección 1.3 se mencionan las herramientas utilizadas para la elaboración del TFG. Por último, en la Sección 1.4 se muestra el diagrama de Gantt que se ha seguido a lo largo del proyecto.

1.1 Motivación

En los últimos años ha surgido un gran interés por deportes con bajo nivel mediático como el voleibol, y esto se ha visto reflejado en un aumento de los equipos, tanto federados como escolares. Por ello, se ha creado esta herramienta de recomendación de ejercicios de voleibol, con el fin de facilitar el trabajo de los entrenadores y motivar el ingreso de los mismos en equipos que, ahora mismo, no salen a competición por no disponer de entrenador.

En el contexto del proyecto de investigación NEAT-AMBIENCE (<http://webdiis.unizar.es/~silarri/NEAT-AMBIENCE/>) se plantea el desarrollo de técnicas de gestión de datos novedosas para ayudar a las personas a tomar mejores decisiones ante los desafíos a los que nos enfrentamos como sociedad. En particular, la pandemia del COVID-19 ha tenido un impacto en muchas áreas de nuestra vida, incluyendo el deporte en general [22] y la práctica de voleibol en particular [33]. Por ello, y dado mi interés por el voleibol, en este TFG se pretende contribuir a dicho proyecto mediante el desarrollo de herramientas que faciliten la toma de decisiones a entrenadores/as de voleibol. De esta forma se pretende ayudar especialmente a entrenadores/as noveles para que puedan incorporarse más fácilmente a las actividades de entrenamiento y así incentivar la práctica de este deporte.

La idea surge en una conversación con el entrenador de voleibol Juan Chicón, en la cual se comentó la falta de entrenadores en varios equipos de todo Aragón, y el gran aumento de jugadores en los mismos. El interés era tan grande que se estaba comenzando a abrir la posibilidad de empezar competiciones de voleibol en municipios de la periferia como La Muela o Cadrete.

La Federación Aragonesa de Voleibol (FAVB) [18], con su presidente David Lechón, se mostró interesada en la herramienta de recomendación y se dispuso a facilitar toda la información y el apoyo que se requiriese para su elaboración.

1.2 Objetivos

El objetivo de este proyecto es poder gestionar fácilmente la información de entrenamientos de voleibol en un contexto deportivo y poder generar recomendaciones de ejercicios siguiendo las necesidades del entrenador. Para ello, se ha desarrollado una aplicación informática de escritorio y portable, la cual almacena la información en una base de datos relacional y genera recomendaciones de ejercicios de voleibol dentro de una rutina de entrenamiento. Se busca además que los entrenadores que utilicen esta aplicación puedan evaluar ellos mismos los ejercicios, teniendo en cuenta también la opinión de los jugadores, para de ese modo mejorar el algoritmo de recomendación y conseguir una herramienta útil en la labor de elaborar la organización de los entrenamientos deportivos futuros. Los jugadores pueden puntuar los distintos ejercicios que han realizado a través de sus dispositivos móviles mediante una aplicación Android, con el fin de conseguir una recomendación que se adecúe a las preferencias del grupo en conjunto y no solamente del entrenador.

Es importante recalcar el objetivo global del proyecto, que es conseguir facilitar el trabajo del entrenador de voleibol. De ese modo, se fomenta que personas interesadas en el deporte puedan hacer el curso de entrenador y conseguir dirigir alguno de los muchos equipos que están surgiendo en la actualidad.

Estos objetivos se relacionan con el Objetivo de Desarrollo Sostenible (ODS) a lograr en 2030 número 4, destacando las metas:

- **ODS 4.4** Aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento.
- **ODS 4.c** Aumentar considerablemente la oferta de docentes calificados, incluso mediante la cooperación internacional para la formación de docentes en los países en desarrollo, especialmente los países menos adelantados y los pequeños Estados insulares en desarrollo.

Se considera que el TFG se relaciona con las metas del ODS 4 debido a que se busca elaborar una aplicación que fomente el empleo de entrenador en jóvenes. Además, se busca que la formación de entrenadores sea profesional y pueda traspasar fronteras, llegando a países menos adelantados.

1.3 Herramientas utilizadas

Para la realización del sistema recomendador, se van a utilizar las siguientes herramientas:

- **IDE Eclipse [16]:** plataforma de software de código abierto y gratuita que proporciona funcionalidades concretas con determinados lenguajes de programación. Aporta gran utilidad al permitir utilizar conjuntamente el resto de herramientas con sus respectivas librerías.
- **PostgreSQL y pgAdmin [34]:** sistema de gestión de bases de datos orientado a objetos y de código abierto. La herramienta de gestión pgAdmin ayuda a crear, diseñar y editar la base de una manera rápida y sencilla.
- **WindowBuilder [23]:** sistema de diseño de interfaces gráficas de usuario acoplado como plug-in a Eclipse. Destaca su facilidad de uso y no requerir librerías adicionales para compilar y ejecutar de las incluidas por la plataforma.
- **Apache Mahout [13]:** proyecto de Apache Software Foundation que permite crear algoritmos de aprendizaje automático basados principalmente en álgebra lineal.
- **Android Studio [4]:** entorno de desarrollo para la creación de aplicaciones Android. Se ha elegido esta herramienta debido a su anterior uso en asignaturas de la carrera, como Ingeniería del Software.
- **Clever Cloud [10]:** servicio de almacenamiento en la nube con posibilidad de usar bases de datos. Se ha utilizado para alojar la base de datos de PostgreSQL cuando se realizaron pruebas con usuarios reales.

1.4 Diagrama de Gantt

En la Figura 1 se muestra el diagrama de Gantt que se ha utilizado para organizar la creación del proyecto de fin de grado. Cabe destacar el tiempo invertido en la recopilación de ejercicios e investigación, con la labor de ponerme en contacto con entrenadores facilitados por la FAVB y obtener parte de los distintos ejercicios de voleibol usados.

La elaboración de la memoria se ha hecho a lo largo de todo el proyecto, aunque focalizándose en la parte final del mismo.

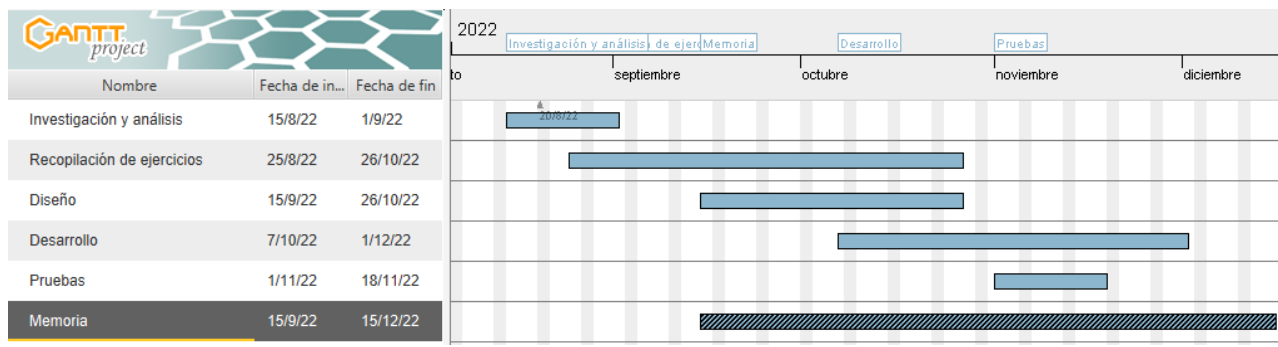


Figura 1: Diagrama de Gantt usado para la organización del proyecto

Se comenzó utilizando una metodología de trabajo en cascada siguiendo el diagrama anterior pero, debido a varias revisiones por parte del tutor en el mes de diciembre, el proyecto se amplió con una aplicación móvil y otras funcionalidades extra, lo cual requirió volver a las fases de análisis y diseño. Por tanto, a partir de ese mes se cambió a una metodología en espiral o incremental, que permitió añadir la funcionalidad Android de la aplicación al mismo tiempo que se revisaba todo el trabajo anterior y se corregían fallos existentes. Finalmente en 2023 se fueron realizando mejoras en el código, añadiendo funciones a la aplicación y siguiendo en la elaboración de la memoria.

2. Investigación sobre sistemas de recomendación

La definición de sistema de recomendación ha estado en constante evolución, pero el término ahora describe a todo sistema que genera recomendaciones a un individuo como salida, o que ayuda a ese individuo de una manera personalizada en la selección de un objeto existente que le sea útil o interesante entre un amplio catálogo. En la Sección 2.1 se introducen los principales tipos o técnicas de recomendación y se profundiza en los sistemas colaborativos en la Sección 2.2, en los basados en contenido en la Sección 2.3 y en los basados en conocimiento en la Sección 2.4. En la Sección 2.5 se estudian los sistemas recomendadores en el ámbito del deporte y se termina en la Sección 2.6 con el estudio de las recomendaciones grupales.

2.1 Tipos o técnicas de recomendación

La mayoría de sistemas de recomendación que buscan un buen rendimiento, mezclan dos o más técnicas de recomendación, entre las que destacan (ver Figura 2):

- **Colaborativas [35]:** agregan puntuaciones o recomendaciones de ítems, reconociendo similitudes entre usuarios, o entre los propios ítems, basándose en esas puntuaciones dadas. Dependiendo entre quién reconozca las similitudes, existen los recomendadores colaborativos user-user o ítem-ítem.
- **Basadas en contenido [29]:** los ítems en este sistema se definen en base a sus características o rasgos, y las recomendaciones generadas se basan en el interés del usuario por ítems con características similares a los que ha valorado positivamente. Puede resultar más complejo que el filtrado colaborativo debido a que es necesario extraer conocimiento de los ítems.
- **Demográficas [32]:** categorizan al usuario basándose en sus atributos personales y crean recomendaciones que dependen de la demografía de la persona.
- **Basadas en utilidad [15]:** utilizan una función para calcular la utilidad que tiene un ítem para un usuario. La creación de esta función determinará la eficacia de esta técnica, además, obtenerla puede ser un gran problema.
- **Basadas en conocimiento [7]:** estas técnicas intentan llegar a una conclusión observando los intereses del usuario y sus preferencias, lo que es un rasgo común en todas las técnicas anteriores, aunque en este caso debe existir un conocimiento sobre cómo un ítem cumple una determinada necesidad del usuario.

Técnicas	Ventajas	Desventajas
Colaborativas	A. Puede identificar gustos de múltiples géneros cruzados B. No necesita que se conozca el área de conocimiento C. Se adapta: la calidad mejora con el tiempo D. Suficiente feedback implícito	I. Arranque en frío para nuevos usuarios J. Arranque en frío para nuevos ítems K. Problema “Grey sheep” L. Calidad dependiente de tener un gran volumen de datos M. Problema “Estabilidad vs. Plasticidad”
Basadas en contenido	B, C, D	I, L, M
Demográficas	A, B, C	I, K, L, M N. Se debe recopilar información demográfica
Basadas en utilidad	E. Sin arranque en frío F. Sensible a cambios de preferencia G. Puede incluir características que no sean del propio ítem	O. Los usuarios deben introducir la función de utilidad P. Habilidad de hacer sugerencias estática (no aprende)
Basadas en conocimiento	E, F, G H. Puede mapear las necesidades por los ítems de los usuarios	P Q. Requiere ingeniería del conocimiento

Figura 2: ventajas y desventajas de técnicas de recomendación. Adaptado de Hybrid Recommender Systems: Survey and Experiments (p. 6), por Robin Burke, 2002

Pero una correcta mezcla de las técnicas anteriores es lo que puede conseguir un alto rendimiento en la búsqueda de recomendaciones. Ahí entran los recomendadores híbridos. Su mayor inconveniente surge en el momento de pensar una manera óptima de unir las otras técnicas, debido a que existen varias maneras de hacerlo. Las más utilizadas son por pesos (sumando los votos de todas las recomendaciones usadas como si fuera una única recomendación), por situación (dependiendo el momento se utiliza una técnica u otra) y en cascada (usar una técnica después de la otra) [8][17].

2.2 Sistemas recomendadores colaborativos

Un sistema recomendador colaborativo es una herramienta que utiliza la información de usuarios y sus interacciones para proporcionar recomendaciones personalizadas. Estos sistemas son ampliamente utilizados en diversos dominios, como en el comercio electrónico, la música y las redes sociales, entre otros. Su objetivo principal es ayudar a los usuarios a descubrir y seleccionar elementos o ítems de su interés en entornos con gran cantidad de opciones. Poseen un funcionamiento que utiliza algoritmos

matemáticos que clasifican la información, la estudian y genera sugerencias ajustadas a las necesidades del usuario [21].

Estos sistemas se basan en la premisa de que las preferencias de un usuario pueden ser inferidas a partir de las opiniones o acciones de otros usuarios con gustos o comportamientos similares. En lugar de analizar los elementos individualmente, los sistemas recomendadores colaborativos se enfocan en las relaciones entre los usuarios y los elementos.

Si dos clientes han comprado CDs similares en el pasado y los han calificado de manera similar, se recomendarán los CDs comprados por uno de ellos al otro si además tienen una calificación positiva, aunque este límite puede variar dependiendo del recomendador. El filtrado colaborativo implementa perfectamente la idea de la promoción de boca en boca, ya que el comprar o no un producto está predominantemente influenciado por las opiniones de otros usuarios [3].

Este tipo de recomendadores ha sido utilizado en multitud de plataformas y webs, destacando:

1. **Amazon:** Amazon utiliza un sistema recomendador colaborativo para sugerir productos a sus usuarios. Examina el historial de compras y visualizaciones de los usuarios, y también considera las acciones de usuarios similares para generar recomendaciones de productos personalizadas [5].
2. **Spotify:** El sistema recomendador de Spotify utiliza el historial de escuchas de los usuarios para ofrecer recomendaciones musicales. Analiza los gustos musicales de los usuarios y encuentra canciones, artistas o listas de reproducción similares que puedan ser de su interés [9].
3. **YouTube:** YouTube emplea un sistema recomendador colaborativo para mostrar videos relacionados en función del historial de reproducciones y las interacciones de los usuarios. Considera factores como las suscripciones, las búsquedas y los "me gusta" para ofrecer recomendaciones relevantes [2].

2.3 Sistemas recomendadores basados en contenido

El filtrado basado en contenido es un tipo especial de filtrado de información que utiliza los elementos o ítems que al usuario le han gustado en el pasado para inferir nuevas recomendaciones, centrándose en las características y propiedades de estos elementos. En lugar de comparar las preferencias de usuarios similares como en el filtrado colaborativo, este enfoque se basa en el análisis de los atributos de los elementos que un usuario ha mostrado interés o preferencia en el pasado.

En estos sistemas, se crea un perfil de usuario basado en las características de los elementos que ha consumido o evaluado positivamente. Luego, se utilizan algoritmos para encontrar elementos similares en función de sus atributos y se recomiendan aquellos que compartan características con los elementos en que el usuario ha mostrado interés previamente. A diferencia de los enfoques

colaborativos, el filtrado basado en contenido tiene limitaciones en su capacidad para proporcionar recomendaciones totalmente novedosas para el usuario, ya que selecciona y recomienda todos los productos en función de la información disponible del usuario actual.

Aunque ambos enfoques no explotan un conocimiento profundo sobre el dominio del producto y son excelentes técnicas que respaldan los procesos de recomendación para productos simples como libros o películas. Una fortaleza importante de estos enfoques es que no se necesitan esfuerzos adicionales de adquisición de conocimiento, siempre y cuando se disponga de un historial de datos [3].

Un ejemplo de herramienta que utiliza un sistema recomendador basado en contenido es GoodReads, aunque la plataforma más exitosa conocida es Netflix. Aunque el sistema recomendador usado por Netflix es complejo y combina varias técnicas, la predominante es el análisis del contenido. El sistema de recomendación de películas y series de Netflix utiliza características como el género, el director, los actores y la trama para recomendar películas y programas de televisión similares a los que un usuario ha podido visualizar previamente [12].

2.4 Sistemas recomendadores basados en conocimiento

Los sistemas recomendadores basados en conocimiento utilizan información y reglas específicas del dominio para realizar recomendaciones personalizadas. Se diferencia de los anteriores en que realiza recomendaciones no basadas en el historial de calificaciones del usuario, sino en consultas específicas que realiza el usuario. Este tipo de sistema solicita unas determinadas reglas o pautas sobre cómo deben ser los resultados o un ejemplo de un elemento deseado. Luego, el sistema busca en su base de datos de elementos y devuelve resultados similares que se ajusten a las especificaciones del usuario.

Un ejemplo común de un sistema recomendador basado en conocimiento se encuentra en plataformas de búsqueda de casas o automóviles online. Los usuarios pueden ingresar criterios como el precio, número de habitaciones, metros cuadrados, etc., y el sitio web les muestra una lista de resultados basada en esas restricciones. La ventaja de utilizar un sistema recomendador basado en conocimiento es que permite filtrar y encontrar resultados en espacios de elementos complejos, donde un simple filtrado de datos podría no ser suficiente. Además, ofrece la posibilidad de personalizar las recomendaciones para cada usuario, brindando resultados más relevantes y adaptados a sus necesidades.

En un trabajo de investigación, es importante comprender la importancia y funcionalidad de los sistemas recomendadores basados en conocimiento, así como explorar ejemplos específicos de su aplicación en diferentes dominios, como la búsqueda de viviendas, automóviles u otros productos y servicios. También es relevante analizar las etapas del proceso de recomendación y cómo se pueden implementar estas soluciones para mejorar la experiencia del usuario y proporcionar resultados personalizados [17].

2.5 Sistemas recomendadores en el deporte

En el mundo del deporte ha llamado la atención la utilización de sistemas inteligentes para lograr objetivos como mejorar el rendimiento de los atletas, prevenir lesiones o simplemente obtener información de interés. En los últimos años se han creado numerosas aplicaciones, sensores y dispositivos para obtener, analizar y comunicar el estado de salud del cuerpo y la actividad deportiva que se ha realizado. Estos datos personales pueden ser muy útiles, pero solo si se desarrollan herramientas para descubrir y entender los patrones que los forman [36].

Un sistema recomendador es una herramienta de ese tipo, ya que utiliza la información para intentar mejorar la calidad de vida del usuario, mostrándole preferencias en multitud de aspectos. Las razones por las que la utilidad de estos sistemas ha despuntado en los últimos años son varias, destacando:

- Existe gran cantidad de atletas con ganas de hacer deporte pero con poca experiencia, y pueden necesitar consejos y guías en aspectos como los entrenamientos en su deporte favorito, recomendaciones de equipamiento y planificación de sesiones.
- Los usuarios generan una gran cantidad de datos detallados y pueden usarse para entender mejor sus habilidades, preferencias y objetivos.
- La gran mayoría de personas tienen un dispositivo móvil y este ofrece una oportunidad única para usar recomendadores y resolver problemas en tiempo real.

Por todas estas oportunidades, se han llevado a cabo varios estudios de caso [37] donde se han utilizado recomendadores en el deporte del atletismo, y es importante conocerlos para extrapolar ideas y poder pasarlas al voleibol. Cabe destacar el estudio que se realizó para recomendar sesiones de entrenamiento personalizadas para preparar una maratón. El problema principal que surgió fue la falta de puntuaciones de los atletas al realizar una serie de ejercicios, por lo que se quedó en un estudio teórico que usa como evaluación los datos de entrenamiento de corredores reales, para demostrar la utilidad de las recomendaciones de sesiones.

Para terminar, un caso de estudio en el ámbito del voleibol es SAETA, un asistente inteligente para el entrenamiento profesional de voleibol basado en técnicas de aprendizaje automático. SAETA realiza análisis al abordar dos aspectos principales del entrenamiento en jugadores de élite: los aspectos técnico-tácticos, con el objetivo de controlar el esfuerzo del ejercicio y los niveles de fatiga; y el nivel de calidad del ejercicio, que complementa lo anterior al controlar la ejecución de los movimientos de los jugadores. Estos análisis se informan a los entrenadores y jugadores en tiempo real para ayudarles a mejorar el entrenamiento [24].

2.6 Recomendaciones grupales

Es importante entender cómo funciona una recomendación grupal, pues en este TFG se va a abarcar un sistema de recomendación en el que interactúan varios usuarios, el entrenador y sus jugadores. Los ejercicios se le recomiendan al entrenador pero también se tienen en cuenta las preferencias de los jugadores, ya que pueden puntuarlos y hacer que un ejercicio pase a ser relevante o deje de serlo.

Los algoritmos de recomendación más usados se basan en los gustos y preferencias de un solo usuario, pero en determinados casos puede ser interesante realizar una recomendación a un grupo de usuarios, por ejemplo, una película puede ser vista por un grupo de amigos o una familia. Por esta razón, recientes estudios han intentado abarcar este problema. Para lograr una buena recomendación grupal se pueden utilizar dos estrategias, por un lado, obtener las evaluaciones de cada usuario del grupo por separado e integrarlas, o identificar las del grupo como una sola. Obtener evaluaciones concluyentes puede resultar muy problemático, ya que puede que la evaluación de un usuario afecte a las del resto, lo que se denomina contagio emocional [6].

La aplicación de recomendación de ejercicios de voleibol ofrece sugerencias de rutinas de ejercicios a un único individuo, el entrenador. Pero esto no es motivo para no poder utilizar técnicas de recomendación grupal, ya que estas pueden servir para utilizar la toma de decisiones mediante criterios múltiples, para solventar el problema de arranque en frío, o para tener en cuenta opiniones de otros usuarios [27].

3. Análisis del sistema

Se muestran a continuación los requisitos funcionales y no funcionales de la aplicación en las Secciones 3.1 y 3.2 respectivamente, y el diagrama de casos de uso en la Sección 3.3, explicando cada caso que lo compone. Todo ello con el fin de analizar el sistema y sentar las bases que definirán su posterior diseño.

3.1 Requisitos funcionales

- RF01: El usuario podrá introducir sus preferencias a la hora de elaborar la recomendación de ejercicios de voleibol.
- RF02: Las preferencias del usuario podrán ser variadas, pudiendo elegir la cantidad de jugadores, su edad, el objetivo a mejorar, el tipo de los ejercicios y la duración de la sesión.
- RF03: El sistema generará una recomendación con una serie de ejercicios a realizar.
- RF04: La recomendación se basará en el perfil del usuario, con sus preferencias descritas y en las puntuaciones de cada ejercicio.
- RF05: El sistema tendrá una función para puntuar los distintos ejercicios.
- RF06: Tanto los usuarios de tipo entrenador como los de tipo jugador podrán puntuar los ejercicios.
- RF07: Se podrá escoger el tipo de recomendación a realizar.
- RF08: Los ejercicios listados se presentarán en forma de tabla con sus características más importantes.
- RF09: El usuario podrá visualizar individualmente cada ejercicio que le haya sido recomendado.
- RF10: Cada ejercicio dispondrá de un nombre, descripción, características y puntuación.
- RF11: Existirán ejercicios con imagen asociada.
- RF12: Se podrán añadir ejercicios nuevos.
- RF13: Se podrán borrar los ejercicios.
- RF14: El usuario podrá borrar y actualizar las puntuaciones de los ejercicios que requiera.
- RF15: El usuario será capaz de visualizar un listado completo con todos los ejercicios.
- RF16: Se podrán visualizar los ejercicios y rutinas desde la aplicación Android.
- RF17: Se podrán puntuar los ejercicios desde la aplicación Android

3.2 Requisitos no funcionales

- RNF01: El sistema representará los ejercicios en una lista de fácil manejo, mostrando su nombre y duración, además, de alguna característica más que pueda ser relevante.
- RNF02: La creación de la lista de ejercicios será rápida, con un tiempo de respuesta máximo de 10 segundos.
- RNF03: El sistema será capaz de gestionar una recomendación hasta con 300 ejercicios diferentes.
- RNF04: Los datos de los ejercicios y usuarios se almacenarán en la base de datos.
- RNF05: El sistema será fácil de usar por un usuario sin necesidad de práctica en entornos informáticos.
- RNF06: Los textos mostrados se mostrarán en un color determinado que no interfiera con el del fondo.
- RNF07: Los mensajes de error serán fácilmente entendibles.
- RNF08: La estructura de las páginas de la aplicación será homogénea.
- RNF09: En el borrado de ejercicios, existirá una confirmación antes de realizar la eliminación.
- RNF10: La interfaz móvil será *responsive* para permitir un buen visionado.

3.3 Diagrama de casos de uso

En la Figura 3 se muestra el diagrama de casos de uso, con sus diferentes actores y varios casos que integran la aplicación. El diagrama se ha realizado con la herramienta Modelio [30].

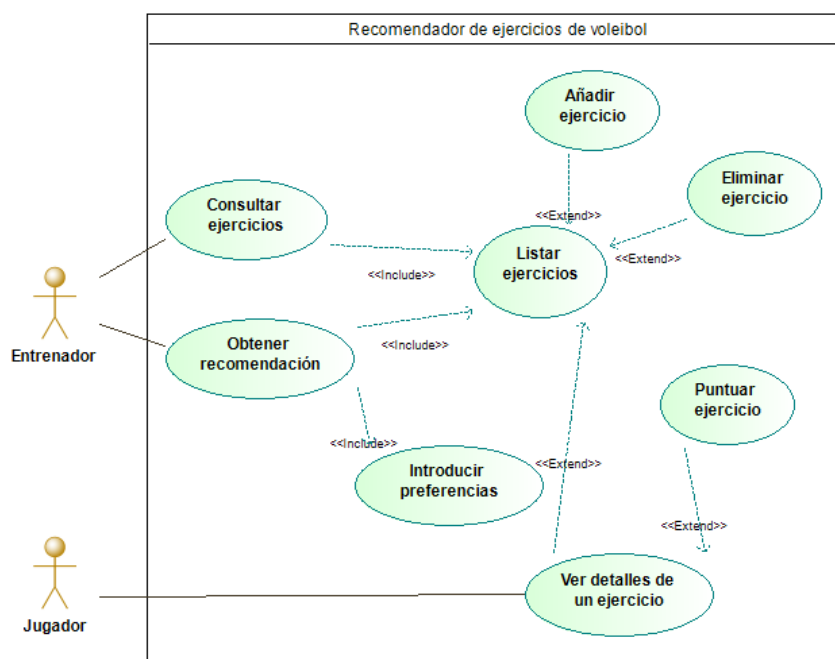


Figura 3: Diagrama de casos de uso usado en el análisis de la aplicación

El actor principal es el entrenador encargado del equipo de voleibol. Es la persona usuaria de la aplicación y quien utilizará sus funcionalidades. Existe otro actor secundario, el jugador, que solo podrá utilizar parte de las funcionalidades del entrenador.

- CU01: Caso de uso “Consultar ejercicios”:

El actor podrá observar todos los ejercicios incluidos en la base de datos. Estos ejercicios pueden ser consultados a modo de resumen, así como realizar futuras tareas con ellos. Incluye el caso de “Listar ejercicios”, ya que estos se mostrarán ordenados y organizados.

- CU02: Caso de uso “Obtener recomendación”:

Es la funcionalidad principal del programa. El entrenador podrá generar una recomendación de ejercicios a realizar en una futura sesión de entrenamiento. Incluye el caso de “Listar ejercicios”, ya que estos se mostrarán ordenados y organizados. También incluye el caso de “Introducir preferencias”, con el cual se podrán elegir las características más convenientes elegidas. Estas características son el número de jugadores de la sesión, su edad aproximada, la duración que se requiera, el objetivo a reforzar y el tipo de entrenamiento.

- CU03: Caso de uso “Añadir ejercicio”:

El entrenador podrá añadir un ejercicio, escribiendo sus datos obligatoriamente para que no existan ejercicios vacíos. Se podrá elegir una imagen explicativa del mismo.

- CU04: Caso de uso “Eliminar ejercicio”:

El entrenador podrá escoger un ejercicio existente y eliminarlo. Este pasará a desaparecer y no podrá ser consultado ni aparecerá recomendado.

- CU05: Caso de uso “Ver detalles de un ejercicio”:

Cada ejercicio posee unas características que podrán ser consultadas por el usuario si así lo requiriera. Estas características son el nombre del ejercicio, su descripción, una posible foto, el número de jugadores necesario, su edad recomendada y la duración. Estos detalles podrán consultarse por el entrenador y también por los distintos jugadores.

- CU06: Caso de uso “Puntuar ejercicio”:

En caso de que lo requiera, el entrenador podrá puntuar un ejercicio desde sus detalles. Además, también podrá especificar la puntuación que le ha dado el grupo de jugadores. Este dato se utiliza para mejorar el sistema de recomendación y adaptarlo a las preferencias del grupo. Tanto el entrenador como los jugadores podrán puntuar los ejercicios.

4. Diseño del sistema

Se explica a continuación el diseño del sistema, tanto de la base de datos como de la interfaz gráfica, además de una breve explicación de la arquitectura que sigue este proyecto. En la Sección 4.1 se menciona la arquitectura Modelo-Vista-Controlador. En la Sección 4.2 se presenta el diseño de la base de datos. En la Sección 4.3 se muestra el diseño de las interfaces gráficas de usuario de la aplicación de escritorio y móvil. Se profundiza en el diseño del sistema con los diagramas de actividades del Anexo B y con el esquema físico del Anexo C.

4.1 Arquitectura Modelo-Vista-Controlador

El diseño del sistema se ha basado en el patrón Modelo-Vista-Controlador para separar las distintas partes de la aplicación y facilitar el desarrollo de la aplicación y su futuro mantenimiento. Los datos utilizados, la interfaz de usuario y la lógica del negocio quedan separadas en los siguientes componentes, mostrados en la Figura 4:

- **Modelo:** Contiene los datos del sistema, organizados en la base de datos *PostgreSQL* y las funciones para acceder a ellos con sentencias *SQL*. Se comunica con la *Vista* enviando la información que necesita en cada momento.
- **Vista:** Contiene la interfaz gráfica mediante la cual se muestran los datos y funcionalidades para que el usuario pueda interactuar con ellos. Se ha realizado una interfaz de aplicación escritorio con *WindowBuilder* y una interfaz de aplicación móvil con *Android Studio*. Se comunica con el *Modelo* para obtener los datos necesarios para mostrar.
- **Controlador:** Contiene la interacción entre *Vista* y *Modelo*. Se encarga de responder a eventos y mandar peticiones para actualizar el estado de cualquiera de los otros dos componentes.

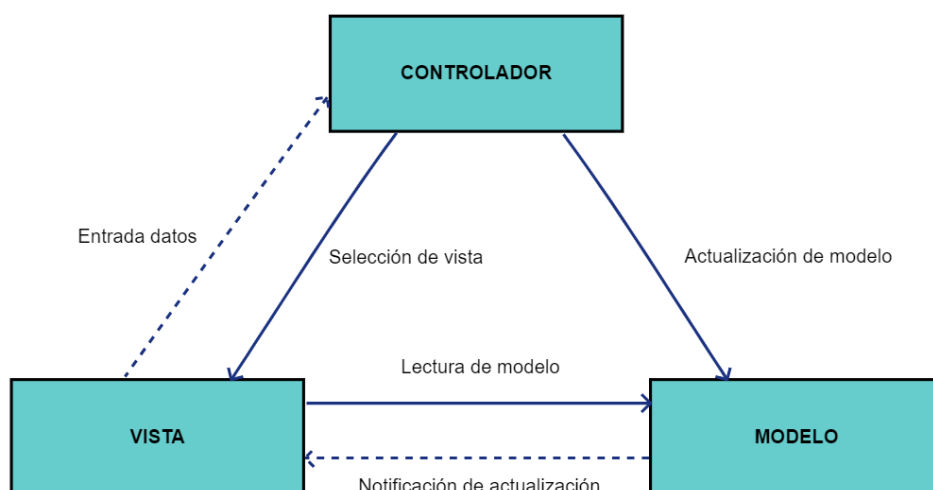


Figura 4: Modelo-Vista-Controlador

4.2 Diseño de la base de datos

El sistema trabaja con una base de datos relacional que se explica en los siguientes apartados y que contiene toda la información necesaria para su correcta implementación. En la Sección 4.2.1 se muestra el esquema entidad-relación y en la Sección 4.2.2 se presenta el esquema relacional.

4.2.1 Esquema entidad-relación

La base de datos se compone de tres entidades: *Ejercicio*, *Rutina* y *Usuario*. Cada uno contiene varios atributos para su distinción y se relaciona con las demás a través de distintas relaciones. *Rutina* y *Ejercicio* comparten la relación contiene, y *Usuario* y *Ejercicio* comparten *puntuá* (ver Figura 5).

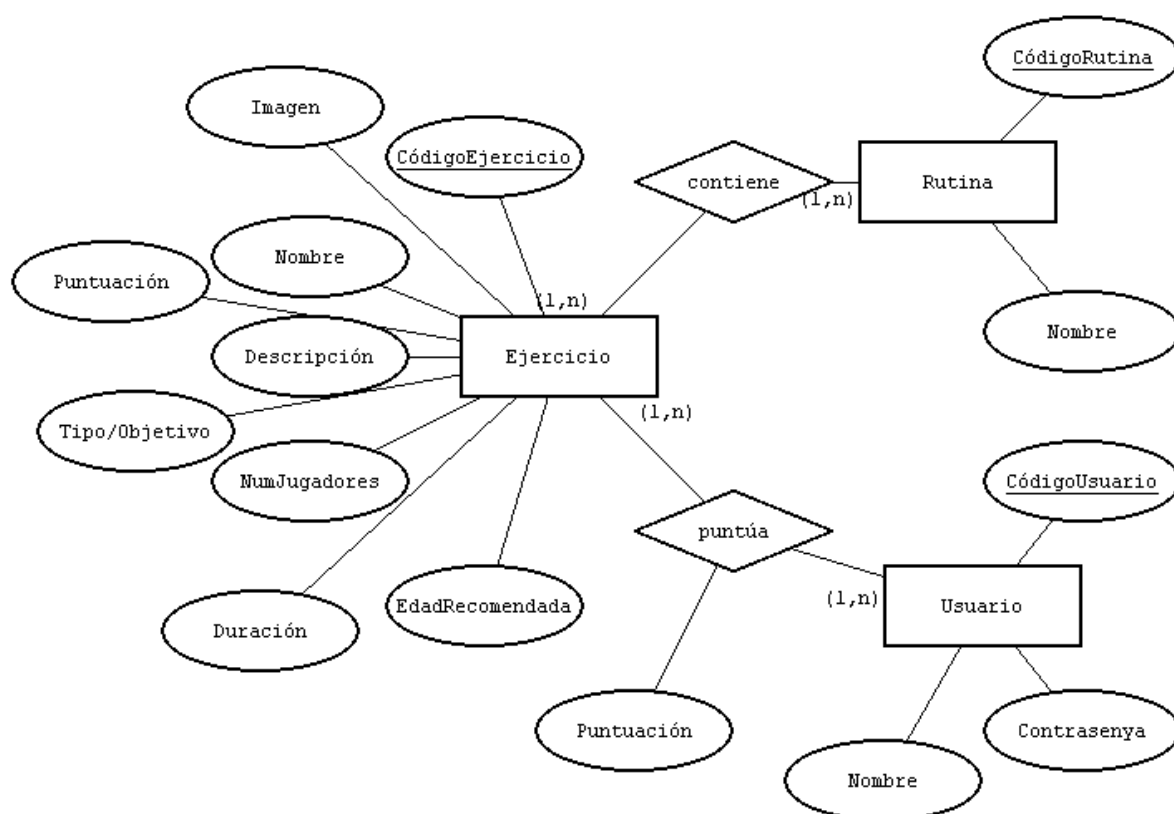


Figura 5: Esquema entidad-relación de la base de datos de ejercicios de voleibol

Destaca el uso de claves artificiales en lugar de claves naturales en las tres entidades. Esta característica se debe a que los nombres naturales de usuarios, ejercicios y rutinas se pueden repetir y de esta manera se consigue una clave que los identifica de manera única. Cabe la posibilidad de guardar el DNI de los usuarios y almacenarlo como clave primaria, aunque este dato posee un carácter muy personal y se ha decidido guardar la menor cantidad de información posible de los usuarios para mantener su privacidad.

4.2.2 Esquema relacional

En este apartado se muestran en detalle los diferentes campos que componen los tres tipos de entidad del apartado anterior, explicando su uso. En la Figura 6 se observa la tabla de Ejercicio, en la Figura 7 la tabla de Rutina y en la Figura 8 la tabla de Usuario.

Ejercicio

Campo	Tipo de datos	Restricciones	Integridad de los datos
CódigoEjercicio	Número entero	No nulo, único	Clave Primaria
Nombre	Cadena de texto	No nulo	
Descripción	Cadena de texto		
EdadMínima	Número entero	No nulo	Mayor que 0
EdadMáxima	Número entero	No nulo	Mayor que 0
Duración	Número entero	No nulo	Mayor que 0
NúmeroJugadoresMínimo	Número entero	No nulo	Mayor que 0
NúmeroJugadoresMáximo	Número entero	No nulo	Mayor que 0
Puntuación	Número real	No nulo	Mayor o igual que 0, menor o igual que 10
NúmUsos	Número entero	No nulo	Mayor que 0
Objetivo	Cadena de texto	No nulo	
Tipo	Cadena de texto	No nulo	
ImagenURL	Cadena de texto		

Figura 6: Tabla de tipos de datos de Ejercicio para el diseño de la aplicación

Los distintos campos que componen un ejercicio son:

- CódigoEjercicio: clave primaria que sirve para identificar cada ejercicio y es generada automáticamente.
- Nombre: apodo breve de cada ejercicio que intenta resumir su contenido.
- Descripción: explicación elaborada del ejercicio, con el fin de que el entrenador no tenga ninguna duda a la hora de mandarlo ejecutar.
- Edad mínima y máxima: edad para la cual está recomendada la realización del ejercicio. Hay ejercicios que no son adecuados para niños pequeños, y otros simples que apenas tendrían utilidad para jugadores más avanzados.
- Duración: tiempo medio que se tarda en realizar el ejercicio. Es cuestión del entrenador el cambiarlo para adecuarlo al resto del entreno y a las necesidades de los jugadores.
- Número de jugadores mínimo y máximo: cantidad de personas para las que está recomendado el ejercicio. Hay ejercicios que requieren una cantidad mínima de jugadores para poder hacerlos, y otros que pierden efectividad si la cantidad es demasiado grande.
- Puntuación: nota que el entrenador en sus distintos usos ha ido actualizando y que sirve para mejorar el algoritmo recomendador.
- Número de usos: cantidad de veces que el ejercicio ha sido seleccionado en una recomendación. Es importante para que ejercicios poco usados no queden eclipsados por ejercicios usados y bien puntuados en futuras recomendaciones.
- Objetivo: es el fin principal que se quiere fortalecer en los distintos jugadores. Se divide en ataque, defensa, recepción, colocación, remate, bloqueo, saque y equilibrado. Cada ejercicio es más útil para cubrir ciertas regiones del juego, ya sea las más ofensivas o las más defensivas.
- Tipo: clase del ejercicio. Se dividen en físico, táctico, técnico y equilibrado. Cada ejercicio se especializa en un tipo distinto y sirve para cubrir diferentes aspectos del jugador.
- Imagen URL: dirección de la imagen a la que se accede para obtener el icono que presenta el ejercicio. La imagen se guarda de manera local como recurso del proyecto para poder tener un acceso rápido a ella, aunque podría plantearse la opción de almacenarla en la base de datos como tipo binario.

Rutina

Campo	Tipo de datos	Restricciones	Integridad de los datos
CódigoRutina	Número entero	No nulo, único	Clave Primaria
Nombre	Cadena de texto		

Figura 7: Tabla de tipos de datos de Rutina para el diseño de la aplicación

Los distintos campos que componen un ejercicio son:

- CódigoRutina: clave primaria que sirve para identificar cada rutina y es generada automáticamente.
- Nombre: apodo breve que se le puede introducir a la rutina.

Usuario

Campo	Tipo de datos	Restricciones	Integridad de los datos
CódigoUsuario	Número entero	No nulo, único	Clave Primaria
Usuario	Cadena de texto	No nulo	
Contraseña	Cadena de texto	No nulo, cifrado	

Figura 8: Tabla de tipos de datos de Usuario para el diseño de la aplicación

Los distintos campos que componen un ejercicio son:

- CódigoUsuario: clave primaria que sirve para identificar cada usuario y es generada automáticamente.
- Usuario: nombre o apodo que se elige cada jugador para acceder a la aplicación.
- Contraseña: clave que elige cada jugador y se establece junto a su usuario para acceder a la aplicación.

Cabe destacar además la puntuación que cada jugador puede adjuntar a los ejercicios, la cual se guarda en la tabla UsuarioEjercicio.

4.3 Diseño de las interfaces gráficas de usuario

El sistema utiliza dos interfaces gráficas, una para interactuar con la aplicación de escritorio, y otra para poder usar la aplicación móvil. En la Sección 4.3.1 se explica el prototipado usado para crear la interfaz de usuario en la aplicación de escritorio. En la Sección 4.3.2 se explica el usado para crear la interfaz de usuario en la aplicación móvil.

4.3.1 Prototipado de la interfaz de usuario de la aplicación de escritorio

La aplicación se divide en las siguientes actividades, mostradas en la Figura 9:

- Inicio:
Comienza la aplicación y sirve como bienvenida para el usuario. Muestra dos opciones de cambio de ventana, la primera para usar el recomendador, y la segunda para mostrar la lista de ejercicios.
- Especificación del recomendador:
Muestra las distintas características requeridas al entrenador como especificación del entrenamiento a realizar. Dependiendo de las que se introduzcan, se generará un entrenamiento diferente posteriormente.
- Ejercicios del entrenamiento:
Aparece una lista de ejercicios que han sido elegidos siguiendo las necesidades descritas anteriormente. Además, el usuario puede ver allí las características principales de cada ejercicio, como el nombre y la duración del mismo.
- Lista de ejercicios totales:
Muestra todos los ejercicios añadidos en la base de datos. Además, el usuario puede ver allí las características principales de cada ejercicio, como el nombre y la duración del mismo. También hay disponibles las opciones de añadir ejercicio y borrarlo, siendo necesaria una confirmación por parte del usuario en caso de elegir un borrado.
- Detalles del ejercicio:
En ella, se pueden observar todos los datos de un ejercicio. Estos son el nombre del ejercicio, su descripción, una posible foto, el número de jugadores necesario, su edad recomendada y la

duración. Además, se podrá especificar una puntuación del entrenador para aumentar la calidad del algoritmo y realizar mejores recomendaciones en futuros usos de la aplicación.

- **Introducción de datos:**

Muestra los diferentes parámetros necesarios que conforman un ejercicio. El usuario debe introducirlos para incorporar un nuevo ejercicio, siendo no obligatorio incorporar una imagen.

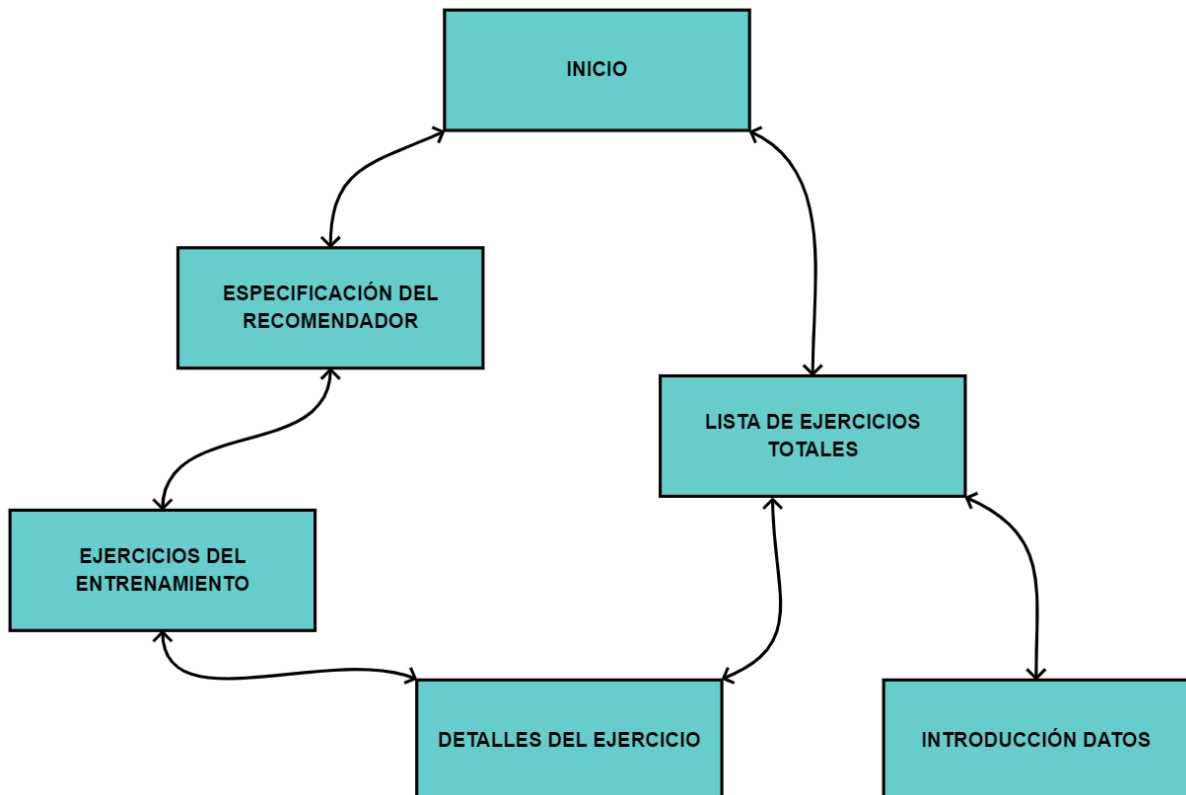


Figura 9: Diagrama de navegación para el diseño de la aplicación de escritorio

En el Anexo D se muestran las distintas ventanas y cuadros de diálogo de la aplicación de escritorio. Las ventanas se han elaborado a modo de boceto con la herramienta Pencil [14], con el fin crear un diseño visual básico que sirva para preparar las ventanas finales de aplicación en la programación del sistema recomendador.

4.3.2 Prototipado de la interfaz de usuario de la aplicación móvil

La aplicación se divide en las siguientes actividades, mostradas en la Figura 10:

- Inicio:

Comienza la aplicación y sirve como bienvenida para el usuario. Se le pide un usuario y contraseña al jugador, tras lo que muestra dos opciones para visualizar los ejercicios, una en la que salen todos y otra en la que se puede seleccionar una rutina y visualizar solamente sus ejercicios.

- Lista de ejercicios totales:

Muestra todos los ejercicios añadidos en la base de datos, únicamente mostrados por nombre.

- Lista de ejercicios de una rutina determinada:

Muestra los ejercicios añadidos en una rutina generada por el sistema recomendador del usuario. La rutina se muestra numerada para poder diferenciarla, y cada ejercicio muestra únicamente su nombre para evitar la saturación de información en la pantalla móvil.

- Detalles del ejercicio:

En ella, se pueden observar varios datos de un ejercicio. Estos son el nombre del ejercicio, su descripción y su duración. Se ha elegido no mostrar todos los datos del ejercicio a los jugadores para no saturarlos con demasiada información que no tiene demasiada relevancia para personas que no son entrenadores. Además, se podrá especificar una puntuación aportada por los jugadores para aumentar la calidad del algoritmo y realizar mejores recomendaciones en futuros usos de la aplicación.

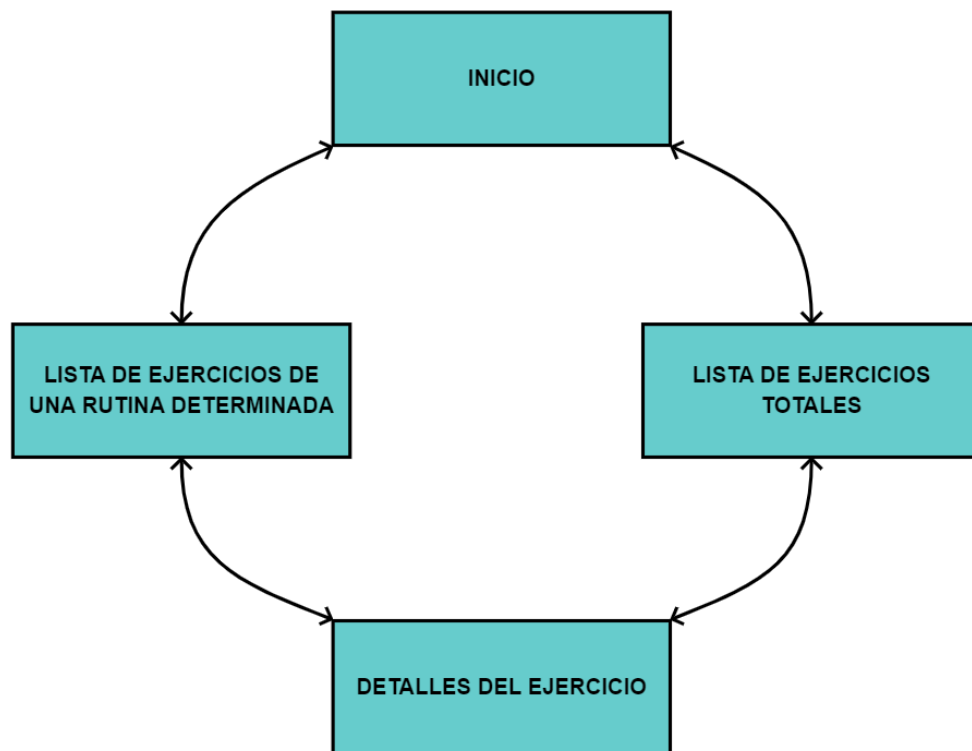


Figura 10: Diagrama de navegación para el diseño de la aplicación móvil

En el Anexo E se muestran las distintas pantallas de la aplicación móvil. Las pantallas se han elaborado a modo de boceto con la herramienta Pencil, con el fin crear un diseño visual básico que sirva para preparar las pantallas finales de aplicación en la programación del sistema recomendador.

5. Implementación del recomendador del sistema

En este apartado se comenta la implementación del sistema recomendador con sus algoritmos, así como la población de la base de datos y las pruebas realizadas. En la Sección 5.1 se comienza con la explicación de la población de la base de datos. En la Sección 5.2 se explica el recomendador usado en la aplicación. En la Sección 5.3 se presentan los distintos algoritmos utilizados en el recomendador.

5.1 Población de la base de datos

Al no existir una base de datos de ejercicios de voleibol de la que obtener la información necesaria, se ha realizado una búsqueda intensiva a través de libros especializados [25][39]. Del mismo modo, y para incluir ejercicios novedosos que se hayan realizado en entornos reales, se contactó con el Club Voleibol Barrio Santa Isabel, en el cual varios entrenadores accedieron a aportar las rutinas de ejercicios que realizaban para incluirlas en la base de datos.

Con ambas fuentes de datos, los distintos ejercicios se fueron introduciendo en una hoja Excel, adjudicándoles un id, un nombre y una descripción, así como las diferentes características necesarias para la recomendación. Con ello, ya se pudo realizar la transformación a csv y la población a la base de datos de *PostgreSQL*.

En los libros revisados aparecen más de 1000 ejercicios diferentes [25][39], pero se ha escogido realizar una selección de los mismos debido a varios motivos:

- Se incluían ejercicios de minivoley, una parte del voleibol para edades de iniciación al deporte a las que no está enfocada la aplicación.
- La similitud de varios ejercicios, diferenciándose en pequeños cambios de duración y movimiento, por lo que se ha escogido seleccionar ejercicios con suficientes diferencias entre ellos.
- Repetición de ejercicios entre los libros de consulta y los obtenidos de entrenadores reales.

Por estos motivos, la selección final consta de 300 ejercicios diferentes. Además, es posible el aumento de los mismos de manera manual, utilizando la opción de la aplicación de añadir un ejercicio.

5.2 Recomendador utilizado en la aplicación

Con el fin de utilizar un sistema recomendador que permita comparar dos ejercicios sin utilizar las puntuaciones de los usuarios, fijándose únicamente en las características de los mismos, se ha escogido aprovechar la información que se obtiene de las descripciones de los ejercicios.

Pero este tipo de recomendadores, basados en contenido, se presenta el gran problema del arranque en frío, que consiste en que el sistema no puede extraer inferencias para los usuarios sobre los que aún no ha reunido suficiente información [38]. En este caso, el sistema empieza sin que el usuario haya puntuado ningún ejercicio que sea de su agrado. Por ello, se ha utilizado como sistema recomendador base un algoritmo por pesos que pide las preferencias del entrenamiento al usuario antes de realizar la recomendación, y le muestra resultados que se adaptan a esas preferencias, aunque pueden no cumplir alguna de ellas. Una vez que el usuario ya ha utilizado la aplicación y tiene un perfil con ejercicios puntuados, se pasa al sistema recomendador basado en contenido, del cual se han realizado y comparado tres aproximaciones, que serán comentadas posteriormente en la Sección 5.3:

- Asignación Latente de Dirichlet (LDA en adelante)
- Frecuencia de Término-Frecuencia Inversa de Documento (TF-IDF en adelante)
- TF-IDF con preselección de palabras clave

Se ha realizado una comparación de las recomendaciones obtenidas con TF-IDF, TF-IDF con preselección de palabras clave y LDA para, en la versión final del recomendador, utilizar el algoritmo que mejor funcione. Además, los usuarios pueden elegir el tipo de recomendación que quieran en todo momento, eligiendo entre LDA, TF-IDF, TF-IDF con preselección de palabras clave, e incluso utilizando el algoritmo por pesos cuando ya se han introducido varias puntuaciones de ejercicios.

El verdadero problema reside a la hora de comparar dos ejercicios para que el sistema pueda realizar una buena recomendación, observando los ejercicios más similares a otros que el usuario haya puntuado satisfactoriamente. Para ello, se van a generar vectores para cada ejercicio dependiendo de la aproximación utilizada, y se usará la similitud coseno para compararlos, debido a su amplia utilización en diferentes algoritmos de aprendizaje automático.

En TF-IDF se ha creado un vector para cada ejercicio, el cual contiene la frecuencia de aparición de las palabras con mayor puntuación TF-IDF. Ejemplo: ([Palabra “Antebrazos”: 1], [Palabra “Dedos”: 0], [Palabra “Ejercicio”: 3], [Palabra “Correr”: 1]).

En LDA se ha creado también un vector para cada ejercicio, pero este contiene los porcentajes de cada categoría creada. Ejemplo: ([Categoría “Partes del cuerpo”: 0.08], [Categoría “Físico”: 0], [Categoría “Técnicas de voleibol”: 0.4], [Categoría “Elementos de juego”: 0.52]).

Una vez que se han obtenido los ejercicios más parecidos a aquellos puntuados positivamente por el entrenador, se realiza una ordenación de los mismos. Los ejercicios más relevantes quedan ordenados de mayor a menor por su puntuación media, la cual ha sido calculada utilizando todas las

puntuaciones de la base de datos, tanto del entrenador como de los jugadores. De esta manera se consigue tener en cuenta la preferencia de los jugadores y no solo la del entrenador.

5.3 Algoritmos utilizados en el recomendador

A continuación se detallarán los algoritmos que se han usado en la aplicación para conseguir realizar recomendaciones de ejercicios de voleibol: algoritmo por pesos, TF-IDF y LDA.

5.3.1 Algoritmo por pesos

En primer lugar, se necesita un algoritmo sencillo pero que no sea completamente aleatorio, ya que se busca aportar una utilidad al entrenador desde su primer uso de la aplicación, incluso cuando no se dispone de valoraciones previas a partir de las cuales puedan inferirse los gustos de la persona. Debido a ello, se ha utilizado un algoritmo que pide por pantalla una serie de datos sobre el entrenamiento a realizar (número de jugadores, edad aproximada de los mismos, objetivo a conseguir y tipo de ejercicio preferido).

Con esas características, se añade un peso igual a cada una y se ordenan los ejercicios de mayor a menor según cumplan o no las mismas. La aplicación mostrará al usuario todos los ejercicios posibles en ese orden hasta que se llegue a la duración establecida del entrenamiento. En la Figura 11 se muestra el código que va sumando el peso de cada ejercicio para obtener cada una de las puntuaciones de preferencia.

```
if(numJugMin <= rut.getNumJugadores() && numJugMax >= rut.getNumJugadores()) {  
    preferencia = preferencia + 1;  
}  
if(edadMin <= rut.getEdad() && edadMax >= rut.getEdad()) {  
    preferencia = preferencia + 1;  
}  
if(objetivo.equals(rut.getObjetivo())){  
    preferencia = preferencia + 1;  
}  
if(tipo.equals(rut.getTipo())){  
    preferencia = preferencia + 1;  
}  
if(rut.getNumJugadores() == 0) {  
    preferencia = preferencia + 0.5;  
}  
if(rut.getEdad() == 0) {  
    preferencia = preferencia + 0.5;  
}
```

Figura 11: código de la asignación de pesos en el algoritmo de recomendación basado en conocimiento

La puntuación de preferencia de cada ejercicio suma un punto si cumple una de las características introducidas, no suma nada si no la cumple, y suma la mitad de un punto si el entrenador ha preferido no introducir una determinada característica. Si no se ha especificado el número o la edad de los jugadores, estos valores se consideran 0 en el código y al realizar la comprobación se suma el medio punto.

5.3.2 Algoritmo TF-IDF

Comenzando con los algoritmos basados en contenido, se ha realizado una aproximación utilizando TF-IDF [28] para expresar la importancia de una palabra en la descripción de un ejercicio. Esta frecuencia es un valor que se calcula como el producto de dos medidas: la frecuencia de término y la frecuencia inversa de documento. En la Figura 12 se presentan las fórmulas para el cálculo del coeficiente TF-IDF.

$$tf(t, d) = \frac{f(t, d)}{\max\{f(t, d): t \in d\}}$$

$$idf(t, D) = \log \frac{|D|}{|\{d \in D: t \in d\}|}$$

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Figura 12: fórmulas TF-IDF

En la aplicación, se ejecuta el algoritmo al generar un nuevo entrenamiento. Al momento de iniciarlo, se obtienen todos los ejercicios que ha puntuado el entrenador anteriormente y se genera un vector para cada uno con la frecuencia de aparición de cada palabra y su puntuación TF-IDF que sigue las fórmulas de la Figura 12 y el código de la Figura 13. Para cada ejercicio, se realiza una comparación mediante la similitud coseno con cada uno de los ejercicios de la base de datos excepto consigo mismo (que daría una similitud de 1) y se ordenan de mayor a menor similitud. La diferencia con el algoritmo de TF-IDF con preselección de palabras clave es que el número de palabras que se utilizan como elementos del vector de comparación ha sido reducido para utilizar las palabras que se han considerado más importantes para la descripción de un ejercicio de voleibol, según el criterio de varios entrenadores que han rellenado el primer formulario del Anexo F.

```

public double tf(List<String> doc, String term) {
    double result = 0;
    for (String word : doc) {
        if (term.equalsIgnoreCase(word))
            result++;
    }
    return result / doc.size();
}

public double idf(List<List<String>> docs, String term) {
    double n = 0;
    for (List<String> doc : docs) {
        for (String word : doc) {
            if (term.equalsIgnoreCase(word)) {
                n++;
                break;
            }
        }
    }
    if (n != 0) {
        return Math.log(docs.size() / n);
    } else {
        return 0;
    }
}

public double tfIdf(List<String> doc, List<List<String>> docs, String term) {
    return tf(doc, term) * idf(docs, term);
}

```

Figura 13: funciones del cálculo del índice TF-IDF [20]

Una vez que se han ordenado todos los ejercicios, la aplicación muestra los ejercicios de voleibol que encajen dentro de la duración de la rutina especificada por el entrenador. El entrenador puede seguidamente acceder a los datos de los mismos y puntuarlos.

5.3.3 Algoritmo LDA

Alternativamente, otra manera de observar la similitud entre ejercicios es LDA [31]. Es un modelo generativo que en este caso sirve para generar categorías para las distintas palabras de las descripciones, y determinar qué categorías y en qué medida contiene cada descripción. Su funcionamiento general se muestra en la Figura 14, y en más detalle en el Anexo G.

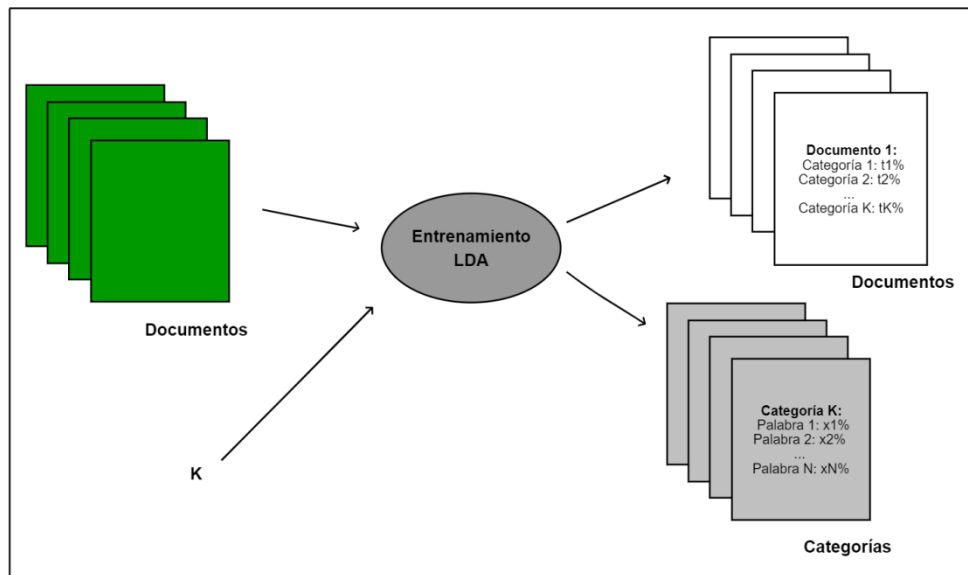


Figura 14: funcionamiento de LDA. Adaptado de Practical Recommender Systems (p. 264), por Kim Falk, 2019

En la aplicación, se utiliza LDA al generar un nuevo entrenamiento. Al momento de iniciarlo, se obtienen todos los ejercicios que ha puntuado el entrenador anteriormente y se genera un vector para cada uno. En este caso, cada vector contiene los porcentajes de la cantidad de palabras que tiene un ejercicio para cada categoría de palabras. Para cada ejercicio, se realiza una comparación mediante la similitud coseno con cada uno de los ejercicios de la base de datos excepto consigo mismo (que daría una similitud de 1) y se ordenan de mayor a menor similitud como en los algoritmos anteriores.

6. Pruebas realizadas

En este capítulo se van a mostrar las distintas pruebas realizadas en el sistema y se van a explicar los resultados obtenidos. En la Sección 6.1 se muestran las pruebas de utilidad. En la Sección 6.2 se comentan los resultados obtenidos.

6.1 Pruebas de utilidad de la aplicación

Para obtener una medida de la utilidad real de la aplicación, se han recogido las opiniones y comentarios aportados por los entrenadores que han utilizado la aplicación. Se les han realizado dos encuestas mediante formularios de Google, las cuales se pueden consultar en el Anexo F.

Un total de 6 entrenadores utilizaron la aplicación de escritorio para generar sus rutinas de entrenamiento. Solamente dos entrenadores tiene una experiencia de más de 2 años entrenando equipos, mientras que el resto pueden ser considerados entrenadores noveles, aunque todos tienen experiencia como jugadores. Se han obtenido unas notas aceptables de la aplicación, con una media de 3.17 sobre 4 en la adecuación de los ejercicios de voleibol, y una media de 2.67 sobre 4 en la utilidad real de la aplicación. Estas notas muestran una actitud positiva a la utilización de los sistemas de recomendación en la creación de rutinas de entrenamiento, aunque sería importante obtener una mayor cantidad de las mismas, priorizando encuestas de entrenadores noveles, que son el principal objetivo a cubrir.

Los comentarios recibidos destacan en cuanto a la apariencia de la aplicación, ya que se explica que el diseño no es nada estético y se debería cambiar para hacerla más atractiva al público. Se comenta además la gran importancia del listado total de ejercicios y el buscador, muy útiles para completar rutinas de ejercicios cuando te faltan uno o dos y no sabes cuál hacer.

Por último, se han mostrado respuestas dispares sobre la utilización de la aplicación durante una temporada entera de juego, con dos “Sí”, un “No” y tres “Tal vez”. Esta disparidad puede suponer un nivel de reticencia hacia el hecho de utilizar la aplicación durante largo tiempo, y quizá sería mejor focalizarla en ayudar a los entrenadores en entrenamientos puntuales. Es importante destacar el hecho de que el número de encuestas no es elevado y las conclusiones obtenidas pueden mostrar resultados que no son totalmente generales.

6.2 Resultados de las pruebas realizadas

Para evaluar los distintos algoritmos empleados en el recomendador, se han diseñado cuatro perfiles de entrenador:

1. Perfil de entrenador con preferencia por ejercicios de recepción.

2. Perfil de entrenador con preferencia por ejercicios grupales.
3. Perfil de entrenador con preferencia por ejercicios de remate, individualizados, físicos y recomendados para personas jóvenes.
4. Perfil de entrenador con preferencia por ejercicios tácticos complejos, de defensa y recepción, de larga duración y recomendados para personas adultas.

Seguidamente, para cada perfil de entrenador definido, se han calculado las medidas de precisión@k, recall@k, F-score y novedad en cada uno de los algoritmos utilizados en el recomendador (por pesos, TF-IDF, TF-IDF con preselección de palabras clave y LDA). La variable k viene determinada por la cantidad de ejercicios de cada rutina, que está acotada por la duración elegida por el entrenador. Para obtener los perfiles de entrenador se ha creado un nuevo usuario correspondiente a cada perfil y se han puntuado ejercicios que se consideran relevantes para ellos a través de la búsqueda de ejercicios de la aplicación. Los detalles de las pruebas y el cálculo de las métricas realizadas se presentan en el Anexo H.

En la Figura 15 se presentan los datos de precisión. Cabe destacar que todas las recomendaciones que usan TF-IDF cuentan con una precisión muy buena, aunque son datos que hay que cotejar a la par que con la novedad, ya que se realizan recomendaciones que en mayoría son de ejercicios que el entrenador ya ha puntuado positivamente. La precisión del algoritmo por pesos y LDA son menores pero, gracias a su alto coeficiente en novedad, aportan nuevos ejercicios al entrenador que ampliarán su perfil y conseguirán el principal objetivo de la aplicación, que es ayudar a obtener ejercicios que un entrenador no conoce y no hubiera pensado que le podrían ser de utilidad.

PRECISION@K	Pesos	TF-IDF	TF-IDF con preselección de palabras clave	LDA
Perfil 1	1	0.85	1	0.38
Perfil 2	0.5	1	1	0.57
Perfil 3	0.14	1	0.89	0.67
Perfil 4	0.6	1	1	0.71

Figura 15: Tabla resumen con los datos de precisión obtenidos en las pruebas de perfil de entrenador.

En la Figura 16 se muestran los datos de novedad. El algoritmo por pesos muestra en todo momento ejercicios que el usuario no ha puntuado y LDA solo repite ejercicios en el perfil 4. Ambos algoritmos TF-IDF repiten muchos ejercicios puntuados, por lo que este aspecto es negativo si el entrenador busca descubrir ejercicios muy diferentes a su perfil.

NOVEDAD	Pesos	TF-IDF	TF-IDF con preselección de palabras clave	LDA
Perfil 1	1	0.57	0.4	1
Perfil 2	1	0	0	1
Perfil 3	1	0.13	0.22	1
Perfil 4	1	0	0	0.71

Figura 16: Tabla resumen con los datos de novedad obtenidos en las pruebas de perfil de entrenador.

En la Figura 17 se presentan los datos de recall. El algoritmo por pesos presenta un alto recall en uno solo de los perfiles, pudiendo deberse simplemente a la casualidad, ya que en los perfiles más avanzados tiene la medida más baja comparada al resto de algoritmos. TF-IDF, TF-IDF con preselección de palabras clave y LDA presentan un recall intermedio. Que el recall sea más bajo que la precisión se puede deber a que existen muchos ejercicios relevantes y en una rutina de entrenamiento no se recomiendan todos los posibles, sino solo los que da tiempo a realizar en la duración de la misma.

RECALL@K	Pesos	TF-IDF	TF-IDF con preselección de palabras clave	LDA
Perfil 1	0.8	0.6	0.5	0.3
Perfil 2	0.4	0.4	0.4	0.4
Perfil 3	0.1	0.8	0.8	0.4
Perfil 4	0.3	0.4	0.4	0.5

Figura 17: Tabla resumen con los datos de recall obtenidos en las pruebas de perfil de entrenador. La métrica se explica en el Anexo H

En la Figura 18 se muestran los datos de F-score, que son muy útiles para comparar directamente los algoritmos usados en el recomendador, ya que es una medida que tiene en cuenta tanto la precisión como el recall. Se puede observar que lo más recomendable es utilizar cualquiera de los dos tipos de TF-IDF por tener esta medida más alta que el resto. El algoritmo LDA se suele recomendar en lugar de TF-IDF [17] porque es capaz de inferir conocimiento y obtener información, aunque en este caso

no es así debido posiblemente a la baja cantidad de ejercicios de la aplicación, a través de los cuales no puede obtener demasiada información importante.

F-SCORE	Pesos	TF-IDF	TF-IDF con preselección de palabras clave	LDA
Perfil 1	0.89	0.7	0.67	0.34
Perfil 2	0.44	0.57	0.57	0.47
Perfil 3	0.12	0.89	0.84	0.5
Perfil 4	0.4	0.57	0.57	0.59

Figura 18: Tabla resumen con los datos de F-score obtenidos en las pruebas de perfil de entrenador

Por lo tanto, los algoritmos con mejores resultados serían TF-IDF en cualquiera de sus dos versiones si el entrenador buscara ejercicios que le gustan y le pueden ser de utilidad. Los resultados que da el algoritmo LDA son peores, pero podría ser útil en caso de que el entrenador buscara nuevas posibilidades al realizar rutinas de entrenamiento novedosas. Estos malos resultados de LDA en lo relativo a recall pueden ser debidos a una cantidad insuficiente de ejercicios de los que se puede obtener información y crear categorías coherentes. Si se quisiera utilizar LDA para obtener recomendaciones más acertadas, se recomienda ampliar la base de datos de ejercicios y, tras ello, probar diferentes valores de distribución entre palabras, categorías y descripciones de ejercicios. Si se quisiera obtener un recall más alto, se podría disminuir el umbral de puntuación que se ha asignado en los cálculos, tomando como ejercicios relevantes aquellos que superen una nota menor. Al hacer este cambio, el recall aumentaría pero la precisión disminuiría, ya que están muy ligadas y es complicado obtener un umbral que dé como resultado valores óptimos tanto de precisión como de recall.

Se puede concluir que el algoritmo por pesos no recomienda ejercicios lo suficientemente relevantes al entrenador. Se puede deber a que la preferencia de un entrenador por un tipo de ejercicio no implica que le sean relevantes todos los ejercicios de ese tipo, y tiene mayor relevancia la información obtenida en las descripciones de los mismos. Otra posibilidad es que la cantidad de palabras en cada descripción no sea suficiente para poder inferir información y realizar una buena recomendación. Se ha calculado la longitud media de las descripciones y es de 41 palabras (12413 palabras entre 300 ejercicios), por lo que se podrían ampliar y comparar resultados como trabajo futuro.

Por último, se ha tenido en cuenta la posibilidad de comparar los algoritmos observando el Mean Squared Error (MSE) o el Mean Absolute Error (MAE). Se ha considerado que las medidas que se obtienen no son totalmente relevantes al utilizar un recomendador basado en contenido. Esto se debe a

que el cálculo de errores se ha realizado comparando el perfil del entrenador, y no existe una predicción de la puntuación de un ejercicio. Esta predicción en puntuaciones sí que se ha usado en casos de recomendadores colaborativos [26] y en estos casos el MSE y el MAE pueden mostrar conclusiones relevantes.

7. Conclusiones y Trabajo Futuro

Se van a comentar brevemente las ideas finales del proyecto, incluyendo las dificultades técnicas encontradas en la Sección 7.1, las líneas futuras en la Sección 7.2 y una valoración personal en la Sección 7.3. El código fuente de la aplicación se puede consultar en un repositorio de GitHub [11].

7.1 Dificultades técnicas encontradas

Han surgido algunos inconvenientes a la hora de crear la aplicación, algunos de los cuales se han subsanado sin mayor problema, mientras que otros han requerido de mayor esfuerzo y han cambiado la manera de proceder de la aplicación.

En primer lugar, se decidió utilizar una base de datos SQLite, que es embebida y local. Posteriormente, fue necesario incorporar a la aplicación la funcionalidad de que los jugadores pudieran puntuar los ejercicios de voleibol desde sus teléfonos móviles, y por esto se requirió cambiar a una base de datos de tipo servidor, como PostgreSQL. Hubo que invertir un tiempo en la conexión con esta base de datos (Android no permite usar localhost directamente al conectar, sino que se debe utilizar la dirección 10.0.2.2) y en la elaboración de las sentencias SQL del gestor (PostgreSQL adapta el SQL estándar a uno más específico propio del gestor).

Además, al pasar los datos de los ejercicios no se reconocían las tildes existentes, y hubo que volver a cargarlos con el formato de codificación de caracteres Unicode UTF-8. En el momento de utilizar el recomendador de Mahout, se empezó utilizando un recomendador colaborativo ítem-ítem, pero no generaba ninguna recomendación debido a la falta de puntuaciones de los distintos usuarios, ocurría un “arranque en frío”. Se terminó descartando este tipo de recomendador porque para obtener buenos resultados con filtrado colaborativo se requieren muchas valoraciones de usuarios para que la matriz de valoraciones usuario-ítem no sea demasiado dispersa y, en este caso, la cantidad de ejercicios de la base de datos siempre va a ser mayor que la cantidad de jugadores de un equipo de voleibol .

7.2 Líneas futuras

En la primera revisión de la aplicación, se añadió una versión móvil en Android con el objetivo de conectarse con el sistema para puntuar directamente la sesión de entrenamiento mediante ella. De este modo, todos los jugadores pasan a ser usuarios de la aplicación y pueden puntuar los ejercicios, facilitando la tarea del entrenador y mejorando notablemente el sistema recomendador al tener mayor cantidad de puntuaciones. La puntuación de los jugadores se tiene en cuenta al ordenar los ejercicios relevantes que son recomendados al entrenador.

Esta mejora se realizó cambiando la base de datos de una embebida con SQLite a una base de datos de tipo servidor, como PostgreSQL. Primero se intentó alojar la base de datos en el servidor de la Universidad de Zaragoza, pero esto no fue posible porque fue en época de Navidad y surgió un error de conexión genérico desde pgAdmin con un mensaje ambiguo que indicaba que el servidor de la base de datos estaba sin arrancar o que no aceptaba conexiones. Se optó por alojar la base de datos en Clever Cloud, ya que es un servidor de alojamiento en la nube gratuito y se necesitaba hacer pruebas con jugadores reales cuanto antes. Un posible cambio a futuro sería alojar la base en el servidor de la Universidad de Zaragoza.

En lo relativo a las recomendaciones grupales, se tienen en cuenta las puntuaciones de los jugadores que han realizado el ejercicio para recomendar una rutina al entrenador, en función de su perfil de usuario. Otra línea futura de trabajo consistiría en profundizar en las recomendaciones grupales e incluir una funcionalidad que consista en elaborar perfiles de equipo que se tengan en cuenta en las recomendaciones. No se ha podido incluir la funcionalidad de realizar recomendaciones grupales por falta de tiempo, aunque añadirla a futuro permitiría comprobar si una recomendación grupal resulta más útil a un entrenador que sus preferencias propias en una recomendación individualizada. Para lograr generar la recomendación grupal, habría que añadir una funcionalidad a la aplicación que permitiera al entrenador seleccionar a los diferentes jugadores de la base de datos e incluirlos en un equipo. Posteriormente, se realizarían las recomendaciones utilizando el perfil del equipo en lugar del perfil del entrenador, utilizando las puntuaciones de los ejercicios de cada individuo.

Además, se podría crear una funcionalidad de elaboración propia de rutinas, para que el entrenador sea él mismo quien elabore las sesiones de entrenamiento utilizando para ello todos los ejercicios de la base de datos. Esta idea no se ha seguido debido a que no es el objetivo de la aplicación. No se busca tener una base de datos estática de ejercicios para que distintas personas los utilicen, sino que los propios entrenadores deben ser capaces de cambiar los entrenamientos que son recomendados con el fin de realizar un trabajo individualizado para sus jugadores. En este caso, una opción de modificación de los ejercicios existentes tendría mucha utilidad y se acoplaría fácilmente a las ya creadas opciones de adición y borrado de ejercicios.

7.3 Valoración personal

En primer lugar, me gustaría destacar la rigurosa labor de obtención de los ejercicios. Comencé hablando con la FAVB y me dieron el contacto de dos entrenadores del club Voleibol Barrio Santa Isabel, Héctor Muñoz y Paula Moragrega. Se mostraron siempre muy amables y dispuestos a ayudarme, aunque la cantidad de ejercicios aportados no superaba los 30 y les costó un esfuerzo realizar la labor de obtención de mayor cantidad de los mismos. Los jugadores siempre se mostraron curiosos en lo relacionado con la aplicación, aunque en el momento de tener que descargarla y utilizarla, muchos lo olvidaban y puntuaban ejercicios desde casa, en lugar de después del entrenamiento que es el momento para el que estaba pensado.

En lo relacionado con los sistemas de recomendación, siempre me habían parecido interesantes y sentía curiosidad por investigarlos, aunque la poca experiencia en su terreno me hizo invertir muchas horas en estudiarlos bien y llegar a entenderlos en todas sus vertientes, ya que existen multitud de tipos distintos, con sus fortalezas y debilidades, y al pasarlos a mi problema, ninguno terminaba de encajar perfectamente. Siempre faltaba algún dato para que terminara de funcionar, y parecía que el bucle no se iba a cerrar nunca, atascado estudiando y buscando otra manera de franquear el problema. Es un tema muy amplio, y estoy interesado en seguir estudiándolo, ya que este trabajo tan solo utiliza una pequeña parte de las funcionalidades que tienen los sistemas de recomendación. Además, me ha resultado de interés colaborar con el proyecto de investigación NEAT-AMBIENCE, con el que he corroborado la importancia de ayudar a las personas en su vida diaria con técnicas de gestión de datos.

Para terminar, siento que al comienzo de la realización de este Trabajo de Fin de Grado lo veía como una asignatura más que había que superar, pero tras todo el tiempo y el esfuerzo invertidos, se ha convertido en un proyecto del que estoy orgulloso por lo muchísimo que he aprendido. Debido a ello, quiero seguir pensando en ideas para mejorarlo y refinarlo.

Referencias

- [1] Knowledge-Based Recommender Systems: An Overview. [En línea, último acceso en mayo de 2023]. <https://medium.com/@jwu2/knowledge-based-recommender-systems-an-overview-536b63721dba>
- [2] ¿Cómo funciona el sistema de recomendación de vídeos de Youtube? [En línea, último acceso en mayo de 2023]. <https://mrinformatica.es/como-funciona-el-sistema-de-recomendacion-de-videos-de-youtube/>
- [3] Felfernig, A., Friedrich, G., & Schmidt-Thieme, L. (2007). Guest Editors' Introduction: Recommender Systems. En IEEE Intelligent Systems (Vol. 22, Issue 3, pp. 18-21). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/mis.2007.52>
- [4] Android Studio. [En línea, último acceso en mayo de 2023]. <https://developer.android.com/studio>
- [5] Así funcionan las recomendaciones de Amazon. [En línea, último acceso en mayo de 2023]. <https://www.genbeta.com/web/asi-funcionan-las-recomendaciones-de-amazon>
- [6] Baltrunas, L., Makcinskas, T., & Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. En Proceedings of the fourth ACM conference on Recommender systems (pp. 1-2). RecSys '10: Fourth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/1864708.1864733>
- [7] Burke, R. (2000). Knowledge-based recommendation. En Recommender Systems (pp. 1-8). Cambridge University Press. <https://doi.org/10.1017/cbo9780511763113.006>
- [8] Burke, R. (2002). En User Modeling and User-Adapted Interaction (Vol. 12, Issue 4, pp. 331-370). Springer Science and Business Media LLC. <https://doi.org/10.1023/a:1021240730564>
- [9] Caso Spotify: cómo usa Advanced Analytics para recomendar canciones. [En línea, último acceso en mayo de 2023]. <http://isaacgonzalez.es/caso-spotify-como-usa-advance-analytics-para-recomendar-canciones/>
- [10] Clever Cloud. [En línea, último acceso en mayo de 2023]. <https://www.clever-cloud.com/>

- [11] Código fuente de la aplicación, en Github. [En línea, último acceso en mayo de 2023]. <https://github.com/697334/TFGRecomendacionEjerciciosVoleibol.git>
- [12] Cómo funciona el sistema de recomendaciones de Netflix. [En línea, último acceso en mayo de 2023]. <https://help.netflix.com/es-es/node/100639>
- [13] Complemento Apache Mahout para Eclipse. [En línea, último acceso en mayo de 2023]. <https://mahout.apache.org/>
- [14] Creación de diagramas con Pencil. [En línea, último acceso en mayo de 2023]. <https://pencil.evolus.vn/>
- [15] Deng, F. (2015). Utility-based Recommender Systems Using Implicit Utility and Genetic Algorithm. En Proceedings of the 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering. 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering. Atlantis Press. <https://doi.org/10.2991/meic-15.2015.197>
- [16] Eclipse Desktop IDE. [En línea, último acceso en mayo de 2023]. <https://www.eclipse.org/ide/>
- [17] Falk, K. (2019). *Practical Recommender Systems*. ISBN 9781617292705. (pp. 211-247). <https://www.manning.com/books/practical-recommender-systems>
- [18] Federación Aragonesa de Voleibol. [En línea, último acceso en mayo de 2023]. <https://www.favb.es/>
- [19] Funciones de cálculo de LDA. [En línea, último acceso en mayo de 2023]. <https://github.com/soberqian/TopicModel4J>
- [20] Funciones de cálculo de TF-IDF. [En línea, último acceso en mayo de 2023]. https://gist.github.com/guenodz/d5add59b31114a3a3c66?permalink_comment_id=2533317
- [21] Graph Everywhere, “Sistemas de recomendación | ¿Qué es el filtrado colaborativo?”. [En línea, último acceso en mayo de 2023]. <https://www.grapheverywhere.com/sistemas-de-recomendacion-que-es-el-filtrado-colaborativo/>

- [22] Grix, J., Brannagan, P. M., Grimes, H., & Neville, R. (2020). The impact of Covid-19 on sport. En *International Journal of Sport Policy and Politics* (Vol. 13, Issue 1, pp. 1-12). Informa UK Limited. <https://doi.org/10.1080/19406940.2020.1851285>
- [23] Herramienta Window Builder. [En línea, último acceso en mayo de 2023]. <https://www.eclipse.org/windowbuilder/>
- [24] Vales-Alonso, J., Chaves-Dieguez, D., Lopez-Matencio, P., Alcaraz, J. J., Parrado-Garcia, F. J., & Gonzalez-Castano, F. J. (2015). SAETA: A Smart Coaching Assistant for Professional Volleyball Training. En *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (Vol. 45, Issue 8, pp. 1138-1150). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/tsmc.2015.2391258>
- [25] Jara, B. (2015). *1000 Ejercicios y juegos de Vóleibol y Mini Vóleibol*. <https://dialnet.unirioja.es/servlet/articulo?codigo=4706544>
- [26] Marlin, B. (2004). *Collaborative Filtering: A Machine Learning Perspective*. (pp. 10-19) <https://hdl.handle.net/1807/119002>
- [27] Masthoff, J. (2010). Group Recommender Systems: Combining Individual Models. En *Recommender Systems Handbook* (pp. 677-702). Springer US. https://doi.org/10.1007/978-0-387-85820-3_21
- [28] Medida TF-IDF. TF-IDF/Term Frequency Technique. [En línea, último acceso en mayo de 2023]. <https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-textclassification-in-nlp-with-code-8ca3912e58c3>
- [29] Pazzani, M. J., & Billsus, D. (s. f.). Content-Based Recommendation Systems. En *The Adaptive Web* (pp. 325-341). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_10
- [30] Modelio. [En línea, último acceso en mayo de 2023]. <https://www.modelio.org/index.htm>
- [31] Modelo generativo LDA. Latent Dirichlet Allocation. [En línea, último acceso en mayo de 2023]. <https://medium.com/analytics-vidhya/latent-dirichelt-allocation-1ec8729589d4>
- [32] Pazzani, M. J. (1999). En *Artificial Intelligence Review* (Vol. 13, Issue 5/6, pp. 393-408). Springer Science and Business Media LLC. <https://doi.org/10.1023/a:1006544522159>

- [33] Piatti, M., Turati, M., Bigoni, M., & Gaddi, D. (2021). Volleyball and COVID-19 emergency: experience of a high-level Italian club team. En *Sport Sciences for Health* (Vol. 17, Issue 1, pp. 253-255). Springer Science and Business Media LLC. <https://doi.org/10.1007/s11332-020-00718-3>
- [34] PostgreSQL con el gestor pgAdmin. [En línea, último acceso en mayo de 2023]. <https://www.postgresql.org/ftp/pgadmin/pgadmin4/>
- [35] Mustafa, N., Ibrahim, A. O., Ahmed, A., & Abdullah, A. (2017). Collaborative filtering: Techniques and applications. En *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*. IEEE. <https://doi.org/10.1109/iccccee.2017.7867668>
- [36] Smyth, B. (2019). Recommender Systems: A Healthy Obsession. En *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, Issue 01, pp. 9790-9794). Association for the Advancement of Artificial Intelligence (AAAI). <https://doi.org/10.1609/aaai.v33i01.33019790>
- [37] Smyth, B., Lawlor, A., Berndsen, J., & Feely, C. (2021). Recommendations for marathon runners: on the application of recommender systems and machine learning to support recreational marathon runners. En *User Modeling and User-Adapted Interaction* (Vol. 32, Issue 5, pp. 787-838). Springer Science and Business Media LLC. <https://doi.org/10.1007/s11257-021-09299-3>
- [38] The Cold Start Problem for Recommender Systems. [En línea, último acceso en mayo de 2023]. <https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems89a76505a7>
- [39] Pozo, Y.P., Stiven, E.R., & Núñez, M.V. (2014). Ejercicios para el jugador líbero en el voleibol como deporte participativo. (pp. 19-27). ISSN 1989-8304