

**Sistema de recomendación para el apoyo de los procesos de entrenamiento y competencia
de los ciclistas del municipio de Tuluá**

Luis Daniel Mejía Quiroga

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2017**

**Sistema de recomendación para el apoyo de los procesos de entrenamiento y competencia
de los ciclistas del municipio de Tuluá**

Luis Daniel Mejía
Código 201356055
`luis.daniel.mejia@correounivalle.edu.co`

**Documento presentado como requisito parcial para la obtención de
grado de Ingeniero de Sistemas**

Director
Joshua David Triana Madrid, Ing
`joshua.triana@correounivalle.edu.co`

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2017

Trabajo de grado presentado por
Luis Daniel Mejía Quiroga
Como requisito parcial para la obtención del título de Ingeniero de Sistemas

Joshua David Triana Madrid
Director

Jurado

Jurado

A Dios por darme la vida, salud y permitirme lograr cumplir este sueño.
A mi familia por la compañía, el amor y por la gran confianza depositada en mi persona.
A mis profesores por su guía en mi aprendizaje y mi crecimiento personal.
Al Ingeniero Joshua Triana por ser mi guía para lograr desarrollar este trabajo.

Tabla de Contenido

1. Introducción	1
1.1. Formulación del problema	2
1.2. Descripción del problema	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Resultados esperados	4
1.5. Antecedentes	5
1.6. Justificación	7
1.7. Información sobre los capítulos	8
2. Alcance de la propuesta	9
3. Marco referencial	10
3.1. Conceptos básicos	10
3.2. Marco Teórico	10
3.2.1. Sistemas de recomendación	11
3.2.2. Tipos de sistemas de recomendación	12
3.2.3. Agrupamiento de datos	15
3.2.4. Modelos de agrupamiento de datos	15
3.2.5. Sistema de gamificación	18
3.2.6. Técnicas de gamificación	18
4. Modelo	21
4.1. Modelo de recomendador de rutas	21
4.1.1. Datos de entrada	21
4.1.2. Datos de salida	22
4.1.3. Algoritmos planteados	22
4.1.4. Pruebas y resultados experimentales	24
4.1.5. Justificación del modelo	26
4.2. Modelo de agrupamiento de datos	28
4.2.1. Algoritmos planteados	28
4.2.2. Justificación del modelo	30
4.3. Modelo para el sistema de gamificación mediante retos	31

5. Desarrollo del proyecto	32
5.1. Diseño e implementación del servicio web REST	32
5.1.1. Características del modelo REST	32
5.1.2. Entorno de desarrollo	33
5.1.3. Arquitectura	34
5.1.4. Implementación del recomendador de rutas	34
5.1.5. Implementación del modelo de sugerencia de eventos	35
5.2. Diseño e implementación de la aplicación Android	36
5.2.1. Entorno de desarrollo	36
5.2.2. Estructura del proyecto	36
5.2.3. Arquitectura	37
5.2.4. Módulos de la aplicación	37
6. Pruebas	45
6.1. Pruebas de software	45
6.1.1. Pruebas unitarias	45
6.2. Pruebas generadas a los algoritmos implementados	47
6.2.1. Recomendador de rutas	47
6.2.2. Creador de eventos	50
6.2.3. Sistema de gamificación mediante retos	53
7. Conclusiones y trabajos futuros	54
7.1. Conclusiones	54
7.2. Trabajos futuros	55
8. Bibliografía	56
Anexos	59
Diagrama de despliegue	59
Modelo entidad relación	60

Lista de Figuras

3.1. Entradas y salidas de un sistema de recomendación	11
3.2. Tipos de sistemas de recomendación	12
3.3. Proceso del algoritmo jerárquico	15
3.4. Agrupamiento basado en reubicación con K igual a 2	17
4.1. Diagrama de flujo del proceso de recomendación para un usuario	27
5.1. Patrón de desarrollo MVP	37
5.2. Ejemplos de pantallas del módulo de autenticación y usuarios	40
5.3. Ejemplo de una pantalla del módulo de rutas	41
5.4. Ejemplo de una pantalla del módulo de eventos	42
5.5. Ejemplo de una pantalla del módulo de retos	43
5.6. Ejemplo de una pantalla del módulo de entrenamiento	44
6.1. Proceso de alineación de ruta al mapa	46
6.2. Resultado de pruebas para usuarios incremental	47
6.3. Resultado de pruebas para rutas incremental	48
6.4. Resultado de pruebas para usuarios y rutas incremental	48
6.5. Resultado de pruebas para porcentaje de calificaciones	49
6.6. Resultado de pruebas con diferente número de usuarios	50
6.7. Resultado de pruebas con diferente número de rutas	51
6.8. Resultado de pruebas con diferente porcentaje de calificaciones	52
8.1. Diagrama de despliegue del proyecto	59
8.2. Modelo entidad relación del proyecto	60

Lista de tablas

1.1.	Resultados esperados	4
1.2.	Estructura de capítulos	8
4.1.	Ejemplo de una matriz R	21
4.2.	Ejemplo de una matriz R'	22
4.3.	Ejemplo algoritmo Slope One	24
4.4.	Resultado para RMSE y complejidad	25
4.5.	Resultado para el tiempo de ejecución	26
4.6.	Ejemplo de matriz de distancias M	30

Resumen

En este trabajo de grado se diseña, modela e implementa una aplicación para dispositivos móviles Android, mediante el empleo de técnicas de inteligencia artificial y características especiales que presentan esta clase de artefactos, con el fin de utilizarlos como herramientas de soporte útiles en las actividades diarias de los ciclistas.

La aplicación en principio funciona en base a un sistema de recomendación, el cual utiliza los datos compartidos diariamente entre cada persona y su teléfono personal con el fin de generar información relevante y de gran ayuda para la persona en actividades relacionadas con el uso de la bicicleta.

Mediante el uso de un servicio web se mantiene la aplicación en sincronización constante con los datos de todos los usuarios, permitiendo determinar la información más relevante para determinado grupo de personas en base a un estudio de características comunes.

Finalmente, con el desarrollo de la aplicación se busca fomentar el uso de la bicicleta como medio de transporte, deporte y recreación, sacando provecho de sus dispositivos móviles y las ventajas que estos ofrecen en el desarrollo de la vida diaria.

Capítulo 1

Introducción

La bicicleta es uno de los medios de transporte que en las ciudades actuales es poco usual utilizarlo dada las condiciones de vida que se presentan. Sin embargo tiene grandes ventajas el poder usarlo tanto como medio de transporte como siendo una herramienta de trabajo, deporte y recreación. Hoy en día el uso más común de este aparato, es en el deporte, dados los beneficios para mejorar la condición física de las personas.

En el caso del ciclismo, este ha sido uno de los deportes que más ha impactado la historia de Colombia. Desde épocas memorables los escarabajos colombianos han conseguido llevar la bandera tricolor en alto y llenar de orgullo a todo un país [1]. Sin lugar a dudas, lo que más necesitan esta clase de deportistas y en general todas las personas que utilicen este medio, es contar con sugerencias y recomendaciones sobre lo que están haciendo bien, las mejores rutas para algún momento determinado, esto con el fin de motivarlos, plantear nuevas actividades y retos que les permitan seguir utilizando la bicicleta en su vida diaria, teniendo como objetivo principal realizar ejercicio, llevándolos a desarrollar sus capacidades al máximo fuera de su zona de confort.

1.1. Formulación del problema

¿Cómo brindarle a la comunidad ciclista del municipio de Tuluá y los municipios vecinos, información útil para apoyar sus actividades de entrenamiento, transporte, recreación y formación a partir de los datos generados durante el desarrollo de las mismas?

1.2. Descripción del problema

El país está pasando por una etapa de cambio, cada uno de los colombianos estamos buscando integración en la sociedad. Esto se puede lograr dejando de lado conflictos y diferencias para llegar a una paz fundamentada en la igualdad, la educación, el deporte y todos aquellos medios que conlleven hacia este fin [2]. El deporte es un medio muy atractivo para construir paz y lograr el crecimiento de la nación, en especial el ciclismo [3]. Brindando oportunidades para que las personas puedan utilizar este deporte para cumplir sus objetivos personales.

Montar en bicicleta genera mucha información sobre la persona y la actividad que está realizando, como determinar si se trata de un deportista de alto rendimiento o un aficionado que lo hace por recreación, además de información como por ejemplo las rutas de acuerdo al tipo de necesidad, la dificultad y duración de un paseo o entrenamiento. Esta información en la mayoría de los casos se pierde, ya que los datos no son analizados, debido a que muchos ciclistas son aficionados o y no están familiarizados y/o interesados en procesos de formación controlados y rígidos. Además de esto, a pesar de que existen dispositivos como Smartphones y Smartwatches que permiten a través de sus sensores capturar este tipo de datos, estos no son analizados particularmente ciclista por ciclista para generar información relevante y útil que pueda servir de ayuda a los deportistas. Por ejemplo recomendar rutas según su capacidad, distancias de recorrido según la edad, entre muchos otros ítems que se obtienen a través de un análisis de elementos particulares de cada ciclista.

De esta manera, es necesario buscar las herramientas adecuadas para que todos los ciclistas puedan obtener de una manera fácil, rápida y eficaz, recomendaciones, consejos e información útil para apoyar las actividades ciclistas, informarse sobre cada una de las modalidades del ciclismo, utilizar la bicicleta para formarse como deportistas a nivel profesional o simplemente divertirse, crear ambientes de compañerismo y altruismo, fomentar la sana competencia y la motivación de todos los que buscan mejorar su calidad de vida mediante este deporte.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un sistema de recomendación móvil que sirva como un medio para apoyar los procesos de entrenamiento, transporte, recreación, competencia y formación de los ciclistas del municipio de Tuluá y los municipios vecinos, utilizando técnicas de inteligencia artificial.

1.3.2. Objetivos específicos

1. Implementar un módulo de recomendación de rutas para practicar ciclismo, donde estas recomendaciones se muestren al usuario a partir de su información personal y los datos generados en sus procesos de entrenamiento y competencia.
2. Implementar un módulo de eventos que se generaran según la clasificación de usuarios en grupos con información relacionada
3. Desarrollar una sección de retos, en donde se le muestra al usuario una serie de desafíos con actividades a realizar, estas actividades cambiaran en complejidad y duración.

1.4. Resultados esperados

Objetivo	Resultados esperados
Implementar un módulo de recomendación de rutas para practicar ciclismo	• Visualización de las rutas recomendadas por el sistema • Creación de rutas para que sean recomendadas a otros usuarios • Documentación referida al módulo (Historias de usuario, diagramas, arquetipos) • Código fuente del modulo
Implementar un módulo de eventos que se generaran según la clasificación de usuarios en grupos con información relacionada	• Visualizar los eventos que se recomienden a un usuario particular • Documentación referida al módulo (Historias de usuario, diagramas, arquetipos) • Código fuente del modulo
Desarrollar una sección de retos, en donde se le muestra al usuario una serie de retos a cumplir	• Visualizar los retos que tenga un usuario determinado, con su determinado porcentaje para completarlo. • Visualizar el ranking de un usuario que de acuerdo a las retos conseguidos y su seguimiento dentro de la aplicacion. • Documentación referida al módulo (Historias de usuario, diagramas, arquetipos) • Código fuente del modulo

Tabla 1.1: Resultados esperados

Fuente: Elaboración propia

1.5. Antecedentes

Los sistemas de recomendación son parte de un campo extenso y estudiado a profundidad [4], en parte debido al crecimiento exponencial en el uso masivo de medios tecnológicos para realizar una gran cantidad actividades cotidianas. Estas actividades generan datos que parecen triviales, pero que si se analizan a fondo a través de técnicas como el filtrado colaborativo [5], la cual es una de las mas utilizadas para desarrollar este tipo de sistemas, se puede hallar información valiosa, la cual serviría para realizar estudios sobre el comportamiento de un usuario y de esta manera predecir lo que posiblemente desee en un futuro y recomendarle lo que realmente pueda interesarle.

Hoy en día este enfoque es utilizado por muchas compañías para conocer a sus usuarios y obtener una ventaja competitiva, ofreciendole a sus clientes lo que realmente necesitan. Amazon [6], por ejemplo se basa en el historial personal de sus usuarios, para mostrarle los productos que posiblemente estén más interesados en comprar, elevando la probabilidad de que un usuario vuelva a realizar una compra.

Los sistemas de recomendación son herramientas que cada vez se están utilizando más en todos los sectores, debido a que pueden adaptarse casi a cualquier medio para generar predicciones cada vez más acertadas y por consiguiente mejores recomendaciones. A continuación se presentan algunos enfoques y sistemas implementados en este campo.

Shot Recommeder System for NBA Coaches [7]

Este trabajo está enfocado en el desarrollo de un sistema de recomendación, mediante el cual se analizan una gran cantidad de lanzamientos de jugadores de la NBA y de esta manera predecir cuáles son las variables para el mejor lanzamiento.

Se analiza el hecho de que el lanzamiento de un jugador de la NBA tiene una gran cantidad de datos relacionados, como pueden ser: la posición del jugador, la fuerza del disparo, los movimientos del oponente y las habilidades propias del jugador, solo por nombrar algunas. Este hecho, hace que la recolección de información y el análisis de la misma, sea difícil, por lo que utilizan un enfoque nuevo, las máquinas de factorización [8], el cual se utiliza para realizar análisis de datos, modelando todas las relaciones de las variables mediante parámetros factorizados.

Para el diseño del sistema, modelan las relaciones de las variables presentes en los lanzamientos, esto con el fin de realizar una predicción que indique el porcentaje de éxito de un lanzamiento y de esta manera poder recomendar las mejores combinaciones de variables a los jugadores y sus entrenadores.

Movie recommender system using item based collaborative filtering technique [9]

Entre los diferentes tipos de enfoques para desarrollar sistemas de recomendación, la técnica de filtrado colaborativo tiene una muy alta popularidad, debido a que en la mayoría de los casos tiene un porcentaje de éxito bastante alto.

Los sistemas de filtrado colaborativo tradicionales pueden incluso producir buenas recomendaciones, aun para problemas diversos. Para este sistema utilizan el enfoque de filtrado colaborativo basado en elemento. En esta técnica, se examina en primer lugar una matriz con las calificaciones de un usuario activo sobre una serie de películas, después, se identifican relaciones entre estas calificaciones y las realizadas por usuarios que han visto películas similares, por ultimo se utilizan estas relaciones con el fin de calcular las recomendaciones para el usuario.

A survey of recommender system feedback techniques, comparison and evaluation metrics [10]

Con la finalidad de proporcionar las mejores recomendaciones para los usuarios, es necesario construir de manera efectiva el modelo del usuario y sus preferencias. En este proceso de construcción existe la necesidad de obtener información precisa del usuario. Este documento tiene la intención de poner de relieve diversos Sistemas de Recomendación, sus ventajas, desventajas y una visión general del mismo.

Las siguientes son aplicaciones que podrían categorizarse dentro del campo en el que se está trabajando.

Wikiloc

Wikiloc [11] es un sitio web español, donde se puede compartir y descubrir rutas para realizar actividades. La aplicación para dispositivos móviles ayuda a crear rutas, tomar fotos y compartirlas. Estas rutas son públicas para que cualquier persona pueda buscarlas.

Strava

Strava [12] es una aplicación que monitorea las actividades deportivas a través de GPS, permite registrar estadísticas como: distancia, velocidad, altitud, entre otras. Permite tener un ranking de logros según puntuaciones obtenidas.

Se puede compartir la información del perfil con amigos en redes sociales. Para cuentas premium tiene más características relacionadas con el rendimiento del deportista, vídeos de consultoría, descuentos en su tienda, estadísticas más detalladas, entre otras.

Timpik

Timpik [13] es una aplicación para dispositivos móviles que ayuda a organizar eventos deportivos con amigos, se pueden organizar desde partidos de fútbol hasta caminatas. En esta aplicación también se puede:

- Buscar eventos cercanos y organizarlos
- Gestionar mensajes
- Visitar perfiles de otros jugadores

1.6. Justificación

Es frecuente escuchar en el medio deportivo, que el deporte colombiano tiene muchos problemas, que no hay apoyo por parte de las entidades y que se carece de políticas públicas orientadas al deporte. Durante mucho tiempo en nuestro país, el deporte era visto como recreación, los pocos que lograban ser profesionales y triunfar eran casos excepcionales, es tanto así, que la gran mayoría de veces que esto sucedía, eran colombianos en otro país.

Desde hace ya algún tiempo, el gobierno nacional está en el proceso de realizar mejoras en las áreas deportivas de nuestro país. Para esto, desde el año 2009 se viene implementando El Plan Decenal del Deporte, la Recreación, la Educación Física y la Actividad Física, para el Desarrollo Humano, la Convivencia y la Paz, 2009 – 2019 [14] como resultado de la necesidad de realizar la formulación de una política pública, en donde haya una amplia participación de los sectores interesados, aparte de contar con la intervención de entidades y organizaciones que son parte del Sistema Nacional del Deporte.

Gracias a todas estas políticas, Colombia ha tenido un gran fortalecimiento y crecimiento deportivo como se puede notar tras la participación histórica en los pasados juegos olímpicos de Río de Janeiro 2016 [15].

El ministerio de las tecnologías de la información y las comunicaciones a través del sitio web [16], fomenta la creación de ideas y herramientas sostenibles y visionarias que ayuden crear medios para el avance tecnológico y social del país y de todos los colombianos de manera directa o indirecta.

A pesar de que en la actualidad hay redes sociales como Facebook [17] y Twitter [18] que permiten a sus usuarios poder comunicarse en tiempo real para generar comunidades en las cuales se comparten gustos, se encuentra y comparte información, aún hay carencia en este tipo de herramientas debido a que el uso de estas es muy genérico y su finalidad tiene muchas aristas, como pueden ser: entretenimiento, interacción social, recreación, juegos entre otras.

Las políticas deportivas unidas a los proyectos que se llevan a cabo por parte del Ministerio de las TIC, hace que sea necesario buscar maneras de apoyar los procesos de formación de los deportistas, para ofrecerles recomendaciones a través de plataformas digitales. Los dispositivos móviles son una buena alternativa debido a que cada día están más relacionados con nuestras actividades cotidianas, sumado a que cuentan con la portabilidad y los sensores necesarios para obtener los datos en casi cualquier parte que se encuentre el ciclista.

Implementar un sistema de recomendación que puedan utilizar todos los ciclistas del municipio de Tuluá y los municipios vecinos en sus Smartphones, en donde se realicen sugerencias particulares basado en los datos personales de cada ciclista, ayuda en gran medida a mejorar sus entrenamientos, conocer nuevas rutas y de esta manera fortalecer sus condiciones físicas que les permitan subir su nivel a través de competencias, eventos y desafíos que lleven a los ciclistas al máximo de sus capacidades.

El desarrollo de este proyecto es de vital importancia en la etapa final del proceso educativo profesional como estudiante. A través de la carrera se han adquirido una serie de conocimientos analíticos y técnicos que están enfocados en lograr un crecimiento intelectual del estudiante. A pesar de realizar una gran cantidad de proyectos durante el transcurso de la carrera, por lo general el ámbito y el contexto son de tipo académico. El poder realizar un proyecto de esta naturaleza, es vital para el estudiante, ya que permite aplicar sus conocimientos en un proyecto de mayor relevancia.

1.7. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento.

Capítulo	Descripción
Capítulo 2: Alcance de la propuesta	En este capítulo se define el alcance de la propuesta
Capítulo 3: Marco Referencial	Se explican los conceptos necesarios para comprender el problema y su solución, se espera que el lector tenga conocimiento básico en algunos temas relacionados
Capítulo 4: Modelo	Se presenta la propuesta de modelo diseñado para el sistema de recomendación de rutas, la creación de eventos y el sistema de gamificación mediante retos
Capítulo 5: Desarrollo del proyecto	En este capítulo se muestran algunos aspectos que se han tenido en cuenta para el diseño e implementación de la aplicación.
Capítulo 6: Pruebas	Se describen las pruebas que se realizaron durante el proyecto, tanto para la parte de software como para los algoritmos implementados, también se explican los resultados que se obtuvieron.
Capítulo 7: Conclusiones y trabajos futuros	Las conclusiones finales del proyecto y los trabajos futuros que deben realizarse para mejorar la aplicación propuesta en este trabajo de grado.

Tabla 1.2: Estructura de capítulos

Fuente: Elaboración propia

Capítulo 2

Alcance de la propuesta

Se desarrollará un sistema de recomendación para dispositivos móviles, específicamente para dispositivos con sistema operativo Android. Según un estudio realizado sobre estadísticas del sistema operativo Android en el año 2016 [19], se tiene como objetivo el sistema operativo Lollipop 5 con SDK versión 21, ya que es el más utilizado con un porcentaje de 35.5 de todos los dispositivos Android. Se tiene estimado un tiempo de 8 meses para realizar la aplicación.

Se implementará un módulo de recomendación de rutas, en donde los ciclistas tienen la posibilidad de ingresar a la aplicación rutas que frecuentan, de esta manera el sistema puede procesarlas a través de algoritmos de inteligencia artificial y recomendarlas a usuarios potenciales basado en su información personal y los datos generados en sus procesos de entrenamiento y competencia.

Para llevar a cabo la implementación del módulo de eventos, se realizará un proceso de clasificación de ciclistas, esto con el fin de que el sistema pueda desarrollar eventos que tendrán el objetivo de reunir a estos deportistas con características similares y les sirva para compartir información, generar lazos deportivos y crear comunidades para practicar este deporte en el municipio de Tuluá.

Se desarrollará una sección de retos, en donde se tiene como finalidad generar un ranking para que los ciclistas puedan ir subiendo de nivel según su rendimiento; a partir de esto se le propondrán retos como completar 500 kilómetros en una semana. por nombrar un ejemplo. La dificultad de estos retos dependerá del nivel de cada ciclista y le permitirán llegar al límite de su capacidad, además de escalar posiciones en el ranking. Las pruebas para este módulo se desarrollarán en caso de que se cuente con suficiente información cuando la aplicación esté disponible para los usuarios, esto debido a que esta clase de sistemas de gamificación no se pueden probar con datos autogenerados.

Se espera que la aplicación pueda ser probada en diversos sectores urbanos y rurales del municipio de Tuluá. Además es importante especificar que debido a que la aplicación a desarrollar hace parte de un trabajo de grado estudiantil, no se publicará en ninguna tienda de aplicaciones debido a que se pueden presentar inconvenientes legales. No se realizará ningún trabajo de mercadeo, tampoco se solicitará pago por el uso de la aplicación o de alguna funcionalidad.

La aplicación no estará disponible para dispositivos con otros sistemas operativos, debido a que la implementación tardaría mayor tiempo porque se trabaja con lenguajes de programación diferentes, con enfoques y utilidades muy particulares.

Capítulo 3

Marco referencial

3.1. Conceptos básicos

En éste capítulo se abordan los conceptos necesarios para el entendimiento del desarrollo de un sistema de recomendación para dispositivos móviles.

Se recomienda que el lector tenga formación básica en:

- Conceptos de inteligencia artificial:
 - Conocimiento básico sobre sistemas de recomendacion.
 - Conocimiento basico sobre tecnicas de agrupamiento de datos.
- Sistemas de gamificación, como funcionan y cuales son sus objetivos.
- Conocimiento básico sobre desarrollo de aplicaciones móviles.
- Conocer cual es el funcionamiento básico de un telefono inteligente.
- Herramientas para el desarrollo de aplicaciones web y moviles.
- Conocimiento básico sobre el funcionamiento de un servicio web REST.

3.2. Marco Teórico

Los consumidores de hoy en día están inundados de ofertas y productos. Los dispositivos electrónicos proveen contenido variado de todo tipo. Sin embargo dar al consumidor lo que realmente necesita es la clave para enganchar compradores. Los sistemas de recomendación ofrecen sugerencias personalizadas a los usuarios de acuerdo a sus gustos y preferencias [20]. Por una parte están los sistemas que basan sus recomendaciones en el análisis del comportamiento del usuario, por otra parte están los que tienen como base la relación entre el contenido de los productos con las preferencias explicitas o implícitas de los usuarios. Finalmente existe otro tipo de sistemas más específicos para algunas áreas de aplicación según el contexto del problema.

3.2.1. Sistemas de recomendación

Son sistemas diseñado para brindarle a los usuarios una serie de sugerencias personalizadas sobre determinados elementos que puedan ser de su interés, esto lo logra a través de procesamiento de datos, por lo regular aplicando algoritmos de inteligencia artificial [21].

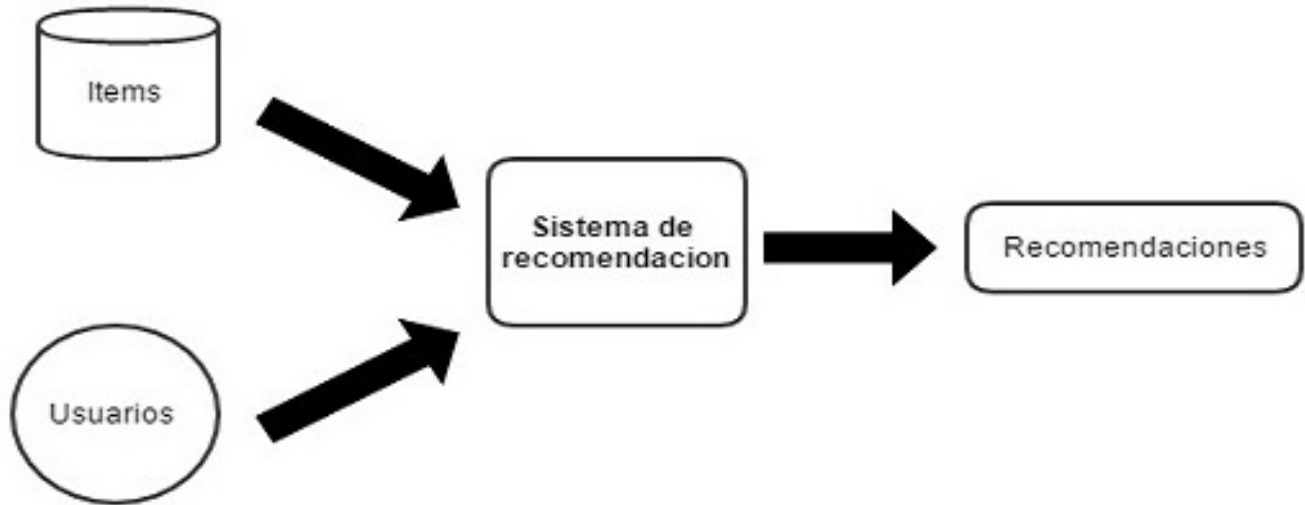


Figura 3.1: Entradas y salidas de un sistema de recomendación

Fuente: Elaboración propia a partir de [21]

En la Figura 3.2 se muestra como es el flujo básico de las entradas y salidas de un sistema de recomendación. El sistema toma los datos de los usuarios y un determinado tipo de elementos, los procesa utilizando algoritmos como los vistos en la Figura 3.2 y por ultimo genera una salida de datos que tienen una alta probabilidad de ser necesaria o útil para el usuario.

3.2.2. Tipos de sistemas de recomendación

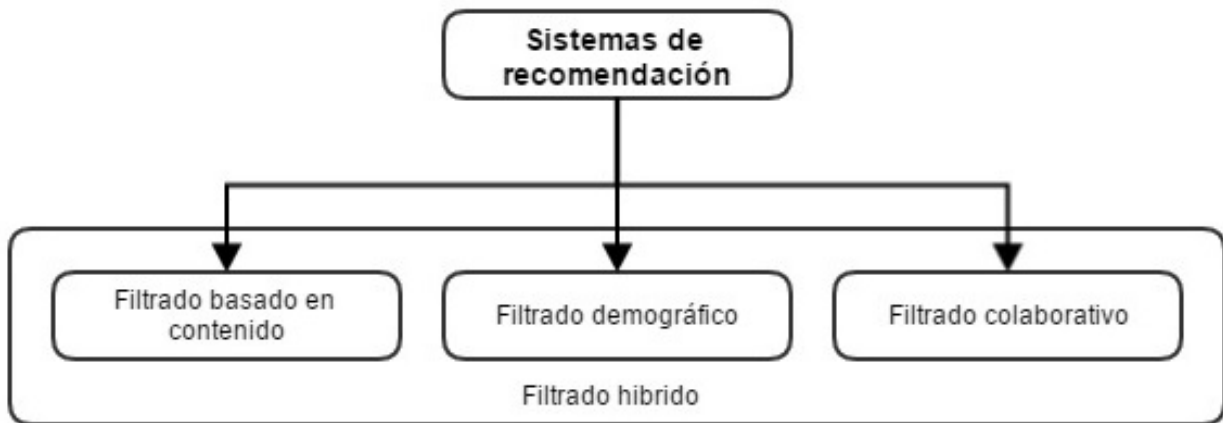


Figura 3.2: Tipos de sistemas de recomendación

Fuente: Elaboración propia a partir de [21, 9]

Técnicas basadas en filtrado colaborativo

El filtrado colaborativo tiene en cuenta un perfil de usuario el cual tiene un historial de comportamiento en el sistema, estas acciones de alguna manera determinan las preferencias del usuario de acuerdo a ciertos elementos. Un ejemplo de estos sistemas puede verse en FilmAffinity [22]. Existen dos métodos principales según el tipo de algoritmo utilizado.

Métodos basados en memoria

Los métodos basados en memoria también son conocidos como algoritmos de filtrado colaborativo basado en vecindad. En estos algoritmos las combinaciones dadas a las calificaciones de usuario o elemento son predecibles a partir de sus vecindades. Esto se hace calculando relaciones entre los elementos del sistema y los usuarios. Hay alternativas para calcular estas vecindades.

■ Filtrado colaborativo basado en usuario

Es un método para hacer predicciones automáticas (filtrado) sobre los intereses de un usuario mediante la recopilación de las preferencias o gustos de información de muchos usuarios (colaboradores).

Ventajas

- Una amplia variedad de algoritmos y técnicas han sido propuestas y evaluadas exitosamente tanto en pruebas como en sistemas del mundo real.
- Aplicar este enfoque, permite evitar la costosa tarea de suministrar y actualizar información detallada de los elementos del sistema, ya que no se necesita, debido a que lo realmente importante es mantener información sobre los usuarios, que son la base de estos sistemas.

Desventajas

- Si el número de personas que han calificado algún elemento es relativamente pequeño comparado con el número de elementos en el sistema, lo más probable es que no se encuentre suficiente similitud entre usuarios.

- Se requiere de la existencia de una gran comunidad de usuarios o de un historial de calificaciones.
- Filtrado colaborativo basado en elemento
Consiste en seleccionar los elementos que un usuario determinado ha calificado y después comprobar que tan similar es cada uno de estos elementos al resto de los elementos del sistema, para al final recomendar los que sean más parecidos.

Ventajas

- Para los elementos la similitud entre ellos es menos variable que la similitud entre usuarios, lo que permite pre-computar estas similitudes y hace el proceso mucho más rápido.

Desventajas

- Ninguna recomendación puede ser generada para un elemento, hasta que alguien no lo haya calificado.

Métodos basados en modelo

En este tipo de métodos, las técnicas de aprendizaje automático y minería de datos son usadas en el contexto de predecir modelos, en tales casos donde el modelo es parametrizado, los parámetros de este modelo son aprendidos dentro del contexto de un sistema de optimización, algunos ejemplos prácticos de uso de métodos basados en memoria incluyen los árboles de decisión, modelos basados en reglas, métodos bayesianos. En esta clase de métodos se crea un modelo de datos previo a las recomendaciones, por lo que se tiene una clara distinción de la etapa de entrenamiento y de prueba.

■ Decision and Regression Trees

Este enfoque realiza una partición jerárquica de los datos con el uso de un conjunto de decisiones en los que cada posible respuesta es un subgrupo de los datos originales, este tipo de preguntas y respuestas va categorizando los datos a medida que se baja en los nodos y termina en una respuesta cuando se llega a una única solución en un nodo hoja.

Una vez que se construye el árbol, es fácil obtener resultados, simplemente se basa en responder una serie de preguntas con respecto a la búsqueda y al final se dará la respuesta.

■ Naive Bayes Collaborative Filtering

En términos simples, un clasificador de Bayes asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable. Por ejemplo, una fruta puede ser considerada como una manzana si es roja, redonda y de alrededor de 7 cm de diámetro. Un clasificador de Bayes considera que cada una de estas características contribuye de manera independiente a la probabilidad de que esta fruta sea una manzana, independientemente de la presencia o ausencia de las otras características.

Esta técnica utiliza la probabilidad para realizar el proceso de recomendar elementos dada la presencia de una o varias características que puedan cumplir los elementos del sistema y así encontrar los de mayor afinidad

■ Modelo de factores latentes

Los modelos de factores latentes comprenden un enfoque alternativo con el objetivo más holístico de descubrir características latentes que explican las calificaciones observadas; se centran en modelos que son inducidos por Singular Value Decomposition (SVD) en la matriz de clasificación de

usuario. Recientemente, los modelos SVD han ganado popularidad, gracias a su atractiva precisión y escalabilidad. Un modelo típico asocia cada usuario y cada elemento con un factor en donde la predicción se realiza a través de la superposición de los factores del usuario y los elementos para encontrar el punto de equilibrio que genera la posible calificación dada por el usuario.

Técnicas basadas en contenido

Este tipo de sistemas de recomendación, tienen como base sugerir en base a elementos preferidos en el pasado. Para esto aprenden las preferencias de un usuario a través de una realimentación, que puede ser implícita, cuando un usuario determina que un elemento es “bueno” o “malo”, o una realimentación explícita, basado en si un usuario eligió un elemento que se le recomendó, o cuánto tiempo paso visualizándolo.

Las preferencias son representadas a través de intereses de un perfil de usuario, que a menudo se muestra como un conjunto de palabras clave que simbolizan cierto tema en específico. Un ejemplo típico de recomendadores basado en contenido es un recomendador de noticias, comparando las palabras clave de una noticia con las palabras clave que aparecen en artículos que el usuario ya ha visto y categorizado de buena manera. [23, 24].

Ventajas

- No requiere que el sistema cuente con un conjunto bastante grande de usuarios.
- Las recomendaciones pueden generarse incluso si solo hay un único usuario
- Variedad de algoritmos y enfoques probados para utilizar mediante esta técnica

Desventajas

- Requiere de guardar y actualizar información de los elementos a recomendar
- Requiere que un usuario tenga como mínimo un conjunto inicial de preferencias
- Requiere mantener y actualizar la información de calificaciones de los usuarios.

Técnicas basadas en conocimiento y demografía

Estos sistemas son particularmente útiles en contextos en donde los elementos no son calificados muy seguido, estos casos pueden ser compra de automóviles, servicios financieros, finca raíz. En estos sistemas los usuarios especifican interactivamente y en tiempo real sus intereses y sus especificaciones para proveer recomendaciones. Este tipo de información es suministrada explícitamente por el usuario, a diferencia de los otros sistemas de recomendación en donde la información para las recomendaciones es inferida por el sistema.

En esta clase de sistemas las sugerencias también se basan características personales de los usuarios como la edad, el sexo, la profesión entre otras. Así un sistema de este tipo realizaría recomendaciones diferentes para un hombre y una mujer.

Técnicas de filtrado híbrido

Esta clase de sistemas mezclan alguno de los tres filtrados mencionados anteriormente para realizar recomendaciones e incluso lo combinan con alguna otra técnica de inteligencia artificial como pueda ser la lógica difusa o la computación evolutiva. Amazon [6] es una de las empresas que utiliza este tipo de recomendaciones para sugerir elementos a sus usuarios.

3.2.3. Agrupamiento de datos

El análisis de grupos o agrupamiento de datos es el proceso de agrupar un conjunto de objetos de tal manera que los miembros del mismo grupo conocido como clúster sean más similares entre sí. Es utilizado en múltiples campos como el aprendizaje automático, el reconocimiento de patrones, el análisis de imágenes, la búsqueda y recuperación de información, la bioinformática, la compresión de datos y la computación gráfica.

El agrupamiento se puede realizar utilizando varios algoritmos que difieren significativamente en su idea de cómo generar el grupo y cómo encontrarlos eficientemente. Las ideas clásicas de grupo incluyen distancias pequeñas entre los miembros del mismo, áreas densas del espacio de datos, intervalos o distribuciones estadísticas particulares. El agrupamiento, por tanto, puede ser formulado como un problema multiobjetivo de optimización.

3.2.4. Modelos de agrupamiento de datos

Agrupamiento jerárquico

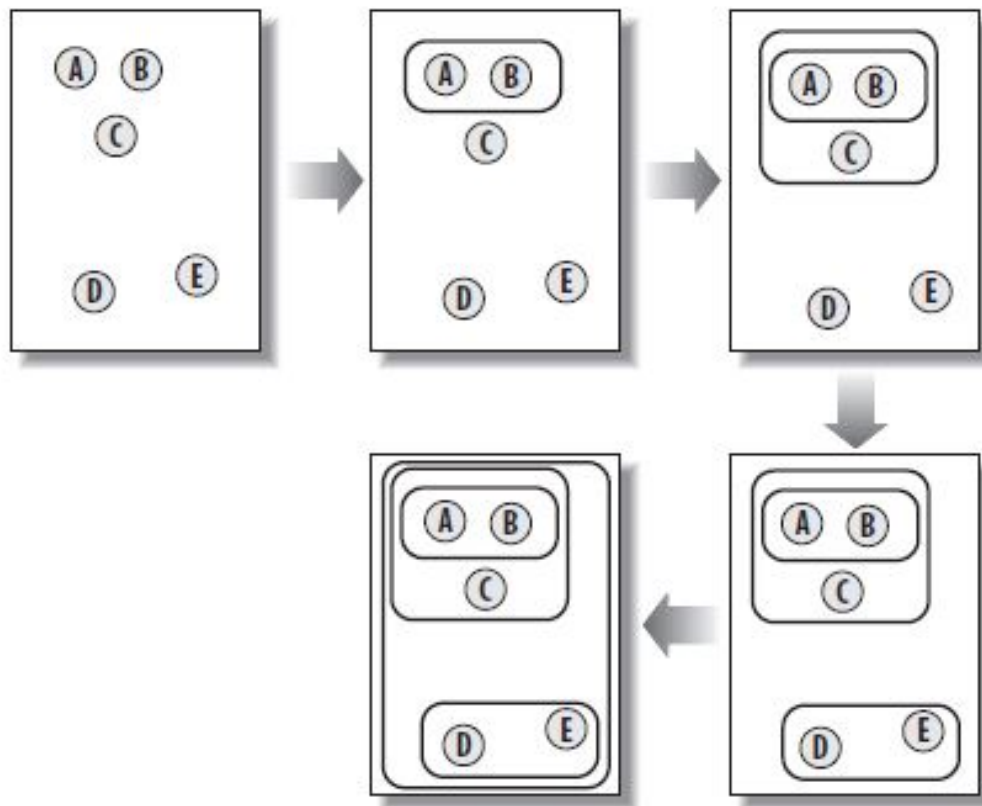


Figura 3.3: Proceso del algoritmo jerárquico

Fuente: Tomado de [25]

El agrupamiento basado en conectividad, también conocido como agrupamiento jerárquico, está basado en la idea principal de que los objetos más cercanos están más relacionados que los que están alejados. Estos algoritmos conectan objetos para formar grupos basados en su distancia. Un grupo puede ser descrito, en gran parte, por la distancia máxima que se necesitó para conectar todas las partes del grupo. A distancias diferentes, se formarán grupos diferentes, los cuales pueden ser representados utilizando un dendrograma, estos algoritmos no solo proporcionan una partición del conjunto de datos, sino en cambio, proporcionan una jerarquía extensa de grupos que se fusionan con cada otro a ciertas distancias [26].

El agrupamiento basado en jerarquía se puede categorizar en aglomerativo (agrupación de lo particular a lo general) y divisivo (división de lo general a lo particular)[26, 27].

Entre las ventajas de la agrupación jerárquica están:

- Flexibilidad incorporada en cuanto al nivel de granularidad
- Facilidad de manejo de cualquier forma de similitud distancia
- En consecuencia, la aplicabilidad a cualquier tipo de atributo

Las desventajas de la agrupación jerárquica están relacionadas con:

- La vaguedad de los criterios de terminación
- El hecho de que la mayoría de los algoritmos jerárquicos no revisitan una vez construido con el fin de mejorarlos

Agrupamiento basado en reubicación de particiones

En este tipo de agrupamiento conocido también como modelos de centroide, los grupos están representados por un vector central, el cual puede no necesariamente ser un miembro del conjunto de datos. Cuando el número de grupos está fijado a k , el algoritmo k -means da una definición formal como un problema de optimización: encontrar los k centros de los grupos y asignar los objetos al centro del grupo más cercano, tal que el cuadrado de las distancias del grupo al centro están minimizadas. Estos algoritmos dividen los datos en varios subconjuntos, debido a que la comprobación de todos los posibles sistemas de subconjuntos es computacionalmente inviable, ciertas heurísticas se utilizan en forma de optimización iterativa para obtener resultados óptimos. Específicamente, esto significa diferentes esquemas de reubicación que iterativamente reasignan puntos entre los k clústeres.

Este problema de optimización es NP-Hard, y por ello el objetivo común es buscar sólo soluciones aproximadas. Un bien conocido método aproximado es el algoritmo de Lloyd, a menudo referido como "k-means". Aun así sólo encuentra un óptimo local, y generalmente se ejecuta varias veces con inicializaciones aleatorias. Variaciones de k -means a menudo incluyen otras optimizaciones como: escoger el mejor resultado de varias corridas, restringir el centroide a miembros del conjunto de datos (k -medoids), escoger medianas (k -medians), escoger los centros iniciales aleatoriamente (K -means++) o permitir una asignación de grupos difusa (Fuzzy C-means).

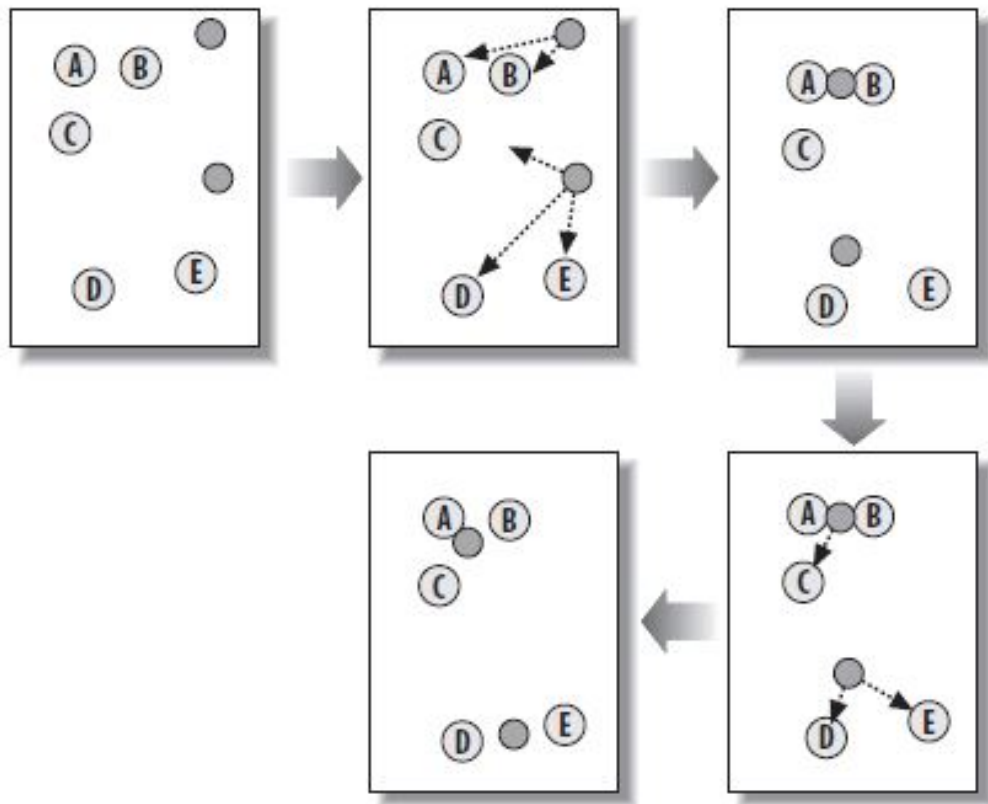


Figura 3.4: Agrupamiento basado en reubicación con K igual a 2

Fuente: Tomado de [25]

Agrupamiento basado en distribuciones

El modelo de agrupamiento más estrechamente relacionado a la estadística es el modelo basado en distribuciones. Los grupos pueden entonces fácilmente ser definidos como los objetos que pertenecen más probablemente a la misma distribución. Una propiedad conveniente de esta aproximación es que esto se parece mucho a la manera en la que los conjuntos de datos artificiales están generados: por muestreos aleatorios de objetos de una distribución.

Mientras la fundación teórica de estos métodos es excelente, adolecen del problema clave conocido como sobreajuste, a no ser que las restricciones estén incluidas en la complejidad del modelo. Un modelo más complejo normalmente será capaz de explicar los datos mejor, el cual hace escoger la complejidad apropiada del modelo inherentemente difícil.

3.2.5. Sistema de gamificación

La ludificación o gamificación es un tipo de aprendizaje que transpone la mecánica de los juegos al ámbito educativo y/o profesional con el fin de conseguir mejores resultados, la gamificación es usada para aumentar conocimientos, mejorar habilidades, o bien recompensar acciones concretas, entre otros muchos objetivos [28].

La ludificación tiene por objeto introducir estructuras creativas e innovadoras provenientes de los juegos para convertir una actividad, a priori aburrida, en otra actividad que motive a la persona a participar en ella. En este sentido, se ha definido que la ludificación pretende persuadir a la persona para convertir una simple tarea en un reto atractivo que merezca la pena continuar [29]

3.2.6. Técnicas de gamificación

Para entender las técnicas involucradas en el concepto es necesario atender a tres conceptos clave: las dinámicas, las mecánicas y los componentes del juego [30, 31].

Mecánicas de juego

Las mecánicas de juego son aquellas reglas que consiguen que la actividad se asimile a un juego o a una actividad lúdica, pues consiguen la participación y el compromiso por parte de los usuarios a través de una sucesión de retos y barreras que han de superar. Existen muchas mecánicas de juego distintas, pero cabe destacar:

- **Recolección:** Se usa la afición de coleccionar de los usuarios y la posibilidad de presumir ante nuestras amistades de estas colecciones. Por ejemplo, en las redes sociales de libros como Anobii los usuarios tienen su propia 'estantería virtual' en la que exponer como trofeos sus libros.
- **Puntos:** Trata de incentivar al usuario mediante un sistema de puntos con el que conseguir algo, como prestigio o premios. Así, algunas cadenas de gasolineras o comercios tienen planes de puntos que prometen descuentos si acumulas cierta cantidad de estos puntos, y consiguen así que acudas a sus establecimientos.
- **Comparativas y clasificaciones:** Someten a los usuarios a un sistema de clasificación que tiene en cuenta su implicación en la actividad. De esta manera se explota el espíritu competitivo de los usuarios.
- **Niveles:** Con este sistema se premia la implicación del usuario en la actividad otorgándole un nivel o descripción con el que distinguirse del resto, y que anima a los usuarios nuevos a igualarles. Así se hace, por ejemplo, en los foros de discusión de Internet en los que en función de la participación del usuario se le asigna un nivel como 'Veterano', 'Novato', etc.

Dinámicas de juego

Son aquellos aspectos y valores que influyen cómo la persona percibe la actividad y que deben ser seleccionados según el propósito que se persiga: la progresión, la narrativa, el compañerismo, etc. Es decir, son los aspectos y valores que se desean añadir a la actividad para que ésta se asimile a una experiencia lúdica o de ocio, pues se refieren a las motivaciones intrínsecas que nos impulsan a seguir jugando. En la ludificación, se utilizan principalmente las siguientes dinámicas de juego:

- **Recompensa:** La recompensa es un incentivo para la realización de una tarea; el jugador se sentirá más atraído hacia el juego.
- **Estatus:** Ser miembro de una comunidad y posicionarse en esta motiva a seguir jugando.
- **Reconocimiento:** Una persona se distingue entre las demás, por ejemplo, por jugar con una buena estrategia. Las personas se sienten comprometidas con actividades que les proporcionan reconocimiento.
- **Expresión o autoexpresión:** El jugador quiere expresar su identidad, su autonomía, su personalidad y su originalidad ante los demás jugadores.
- **Competición:** La competición es la práctica de un juego que tiene como resultado una clasificación de los participantes. La comparación con los demás es una fuente de motivación para muchos jugadores.
- **Juego cooperativo:** Dos o más jugadores no compiten; se esfuerzan por conseguir un mismo objetivo.
- **Altruismo:** Las personas se esfuerzan en ayudar a otras o apoyar causas solidarias sin esperar una recompensa a cambio

Componentes del juego

Son las aplicaciones específicas que se añaden a la actividad. Hay herramientas de ludificación en formato de videojuego muy sofisticadas. En algunas de ellas se introduce a la persona en un ambiente virtual creado por especialistas, donde tendrá que poner a prueba sus capacidades en función de los requerimientos que le piden.

Son sistemas que permiten analizar las reacciones fisiológicas frente a situaciones determinadas y, por otro lado, las reacciones neuronales, con el objetivo de conocer la reacción general del individuo frente a situaciones que pueden intervenir en su ámbito de trabajo, para conocer su capacidad de adaptación, control de la situación, capacidad de liderazgo, etc. Son herramientas que permiten evaluar, pero sobre todo desarrollar competencias, cambiar comportamientos.

Tipos de jugadores

Richard Bartle, autor de *Designing Virtual Worlds*, en su publicación "Teoría sobre los jugadores de Richard Bartle", ha elaborado una clasificación de los diferentes tipos de jugadores que se puede encontrar a la hora de ludificar cualquier actividad o proyecto [29].

Según esta clasificación, existen cuatro tipos de jugadores con diferentes motivaciones e intereses hacia el juego. Por un lado, se encuentran aquellos que se centran en la consecución de los objetivos y en el resultado final del juego, y por el otro, los que se centran más en el valor participativo y lúdico que entraña el juego en sí mismo.

- **El ambicioso:** Su principal motivación es la de ganar y conseguir obtener el primer puesto de la clasificación. La forma para retener a estos participantes en el juego es mediante el uso de listas de clasificación en las que pueden comprobar cómo van superando los distintos niveles y cómo van escalando puestos.

- **El triunfador:** Es un tipo de jugador aventurero, cuya principal motivación es la de continuar descubriendo nuevos escenarios, plataformas o niveles y superar los objetivos marcados en el juego. En este sentido, su motivación es intrínseca y está más relacionada con la satisfacción personal o el bien del grupo.
- **El sociable:** Su motivación es de tipo social por encima de la misma estrategia del juego, compartir con los demás o crear una red de contactos o amigos. Se les retiene en el juego a través de chats o listas de amigos.
- **El explorador:** Es aquel jugador que le gusta descubrir aquello que le es desconocido. Le atraen los retos complejos mediante los cuales pueda superar los diferentes niveles del juego. Su motivación es la auto superación.

Capítulo 4

Modelo

4.1. Modelo de recomendador de rutas

En el sistema de recomendación se tiene un conjunto de preferencias de usuario las cuales se representan mediante una matriz $R = NM$ donde N representan los usuarios y M las rutas, cada celda R_{ij} representa la calificación del usuario i a la ruta j . En condiciones normales la matriz es esparcida dado que la mayoría de los usuarios no han calificado todas la rutas del sistema.

4.1.1. Datos de entrada

- R : Matriz de calicaciones de los usuarios a las rutas del sistema, con valores en el intervalo $(0, 5]$. Para las rutas en las cuales un usuario no ha determinado una calicación, se asigna un carácter diferente.

Rutas Usuarios	Ruta 1	Ruta 2	Ruta 3	Ruta 4	... Ruta M
Usuario 1	1	4.2	-	4	-
Usuario 2	-	4	5	4.3	3.5
Usuario 3	4.5	-	2.5	2	1
...					
Usuario N	3	4	-	5	2.5

Tabla 4.1: Ejemplo de una matriz R
Fuente: Elaboración propia

4.1.2. Datos de salida

- R' : Matriz completa en donde se han determinado los valores aproximados para las rutas que un usuario no ha calificado.

Rutas					
Usuarios	Ruta 1	Ruta 2	Ruta 3	Ruta 4	... Ruta M
Usuario 1	1	4.2	3	4	4
Usuario 2	5	4	5	4.3	3.5
Usuario 3	4.5	5	2.5	2	1
· · · Usuario N	3	4	4.2	5	2.5

Tabla 4.2: Ejemplo de una matriz R'

Fuente: Elaboración propia

4.1.3. Algoritmos planteados

Dado que el primer objetivo define que el sistema de recomendación debe sugerir rutas a los usuarios según las calificaciones conocidas, el filtrado colaborativo es el enfoque más apropiado de acuerdo a las características de los valores presentes, el intervalo de las sugerencias y la naturaleza de aplicación que no ofrece los elementos necesarios para otras técnicas.

Teniendo en cuenta esto, se eligieron dos algoritmos para implementar y poner a prueba dado que son los que tienen mayor desempeño en relación con los otros dentro de este enfoque [24, 32, 33].

Matriz de factorización

La factorización de matrices (MF) es una de las técnicas de filtrado colaborativo basado en modelo. La idea detrás de las técnicas de MF Es muy simple. Se tiene una matriz R de N usuarios y M rutas donde cada celda representa la calificación de un usuario a una ruta, y se desea aproximar R como el producto de dos matrices de la siguiente manera:

$$R \approx PQ \quad (4.1)$$

Donde P es un matriz NK que representa la relación entre un usuario P_i y el factor K y Q es una matriz KM que representa la relación entre una ruta Q_i y el factor K .

K representa las relaciones inherentes entre rutas y usuarios según sus atributos, por ejemplo un factor podría ser la distancia de la ruta en relación con el nivel de desempeño de un usuario. Para efectos del algoritmo esto no tiene mucha importancia dado que precisamente se busca determinar las matrices P y Q que mejor aproximen a la matriz R .

Este metodo es un algoritmo de optimización que realiza el producto vectorial de las matrices P y Q tal que minimizan el error cuadrado regularizado entre este resultado y una matriz R' que contiene un subconjunto de calificaciones conocidas (Conjunto de entrenamiento), la solución se obtiene cuando el algoritmo ha logrado minizar este error a un intervalo de aceptación o se ha llegado a un número máximo de iteraciones. [34].

El algoritmo se define de acuerdo a la ecuación 4.2

$$\min_{q^*, p^*} \sum_{(u,i) \in \mathbf{K}} (r_{ui} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2) \quad (4.2)$$

$$(r_{ui} - q_i^T p_u)^2.$$

(4.3)

La ecuación 4.3 determina el error entre los valores del conjunto de entrenamiento y los valores sugeridos

$$\lambda (\|q_i\|^2 + \|p_u\|^2)$$

(4.4)

En la ecuación 4.4, la constante λ controla los valores extremos a través de una penalización.

Slope One

Slope One es una familia de algoritmos usados para el filtrado colaborativo introducida en [32]. Posiblemente, esta es la forma más simple de filtrado colaborativo basado en elementos. Su simplicidad la hace especialmente sencilla de implementar eficientemente mientras que su exactitud está a la par de algoritmos más complejos y costosos.

Para reducir drásticamente el sobreajuste, mejorar el rendimiento y facilitar la implementación, fue propuesta esta familia de algoritmos. Esencialmente, en vez de usar una regresión lineal de las valoraciones de la forma $f(x) = ax + b$, se usa una forma simple de regresión con un sólo parámetro $f(x) = x + b$.

El parámetro indicado es la diferencia del promedio entre la valoración de dos elementos. Esto ha demostrado ser mucho más exacto que la regresión lineal en algunas instancias, y toma la mitad del espacio de almacenamiento o menos.

Considere la siguiente tabla:

Cliente	Artículo 1	Artículo 2	Artículo 3
Juan	5	3	2
Carlos	3	4	-
Lucia	-	2	5

Tabla 4.3: Ejemplo algoritmo Slope One

Fuente: Elaboración propia

En este caso, el promedio de las diferencias en las valoraciones entre el artículo 2 y el 1 es $\frac{(2+(-1))}{2} = 0,5$, entonces en promedio, el artículo 1 es valorado antes que el 2 por 0.5. De la misma manera, el promedio de las diferencias entre el artículo 3 y el 1 es 3. Por lo tanto, si intentamos predecir la valoración de Lucía para el artículo 1 usando su valoración para el artículo 2, obtenemos $2 + 0,5 = 2,5$. Asimismo, si tratamos de predecir su valoración para el artículo 1 usando la del artículo 3 obtenemos $5 + 3 = 8$.

Si un usuario ha valorado varios elementos, las predicciones son combinadas usando un promedio ponderado donde una buena elección para esta ponderación es el número de usuarios que han valorado dos elementos. En el ejemplo anterior, podemos predecir la siguiente valoración de Lucía para el artículo 1

$$\frac{2*2,5+1*8}{2+1} = 4,3$$

Por lo tanto, teniendo n artículos, para implementar Slope One, todo lo que se necesita es calcular y almacenar una media ponderada, en función de las veces que este valorado el artículo, de las diferencias y el número de valoraciones comunes para cada uno de los n^2 pares de elementos.

4.1.4. Pruebas y resultados experimentales

La efectividad de un algoritmo de recomendación puede ser evaluado a través de métricas de desempeño [33], para los algoritmos planteados se evaluó el error cuadrático medio (*RMSE*). El cual se define de acuerdo a la ecuación 4.5

$$e_{ui} = r_{ui} - \hat{r}_{ui} \quad \text{for } (u, i) \in \mathcal{T}$$

$$\text{SSE} = \sum_{(u,i) \in \mathcal{T}} e_{ui}^2$$

$$\text{RMSE} = \sqrt{\text{SSE}/|\mathcal{T}|}$$

$\mathcal{T}$ = conjunto de validación del algoritmo.

(4.5)

Este valor es una forma de evaluar la diferencia entre un valor estimado y el valor real de la cantidad que se quiere calcular. El *RMSE* mide el promedio del cuadrado del error, siendo el error el valor en la que el estimador difiere de la cantidad a ser estimada.

Para la evaluación se realizaron las pruebas con entradas de tamaño $N * M$, se dividió la entrada en dos partes, 80 % para el conjunto de entrenamiento y 20 % para el conjunto de validación. Las pruebas realizadas evaluaron las siguientes metricas:

- *RMSE*
- Complejidad de implementación y mantenimiento
- Tiempo de ejecución

Para el algoritmo Slope One, se analizaron dos variantes propuestas por [32], dado que son las que tienen mayor aceptación y exactitud. Las tablas 4.4 y 4.5 muestran los resultados de las pruebas realizadas a ambos algoritmos.

Algoritmo	RMSE	Complejidad de mantenimiento(1 a 5)
Matriz de factorización	1.37	5
Slope One ponderado	1.43	1
Slope One doble	1.44	2

Tabla 4.4: Resultado para RMSE y complejidad

Fuente: Elaboración propia

Entrada/ Algoritmo	10X10	20X20	50X50	100X100
Matriz factorización	1.21s	3.72 s	32.2s	712.2 s
Slope One ponderado	0.00023s	0.069s	2.32s	45.2 s
Slope One doble	0.562s	1.87s	18.2s	274 s

Tabla 4.5: Resultado para el tiempo de ejecución

Fuente: Elaboración propia

4.1.5. Justificación del modelo

Después de examinar los resultados obtenidos se eligió el algoritmo de Slope One ponderado para la implementación de la aplicación en desarrollo, dado que tiene una efectividad bastante buena en comparación con la mejor técnica, su tiempo de ejecución supera por mucho a los demás algoritmos y su complejidad para implementar, mantener y escalar es increíblemente baja. Este algoritmo es sencillo pero eficiente, incluso a nivel de las técnicas más avanzadas y complejas.

En la aplicación los usuarios tienen un nivel de acuerdo a su desempeño deportivo, además la mayoría de los dispositivos móviles actuales cuentan con un sensor GPS que permite determinar la ubicación del usuario, según esto, se planteó apoyar el proceso de recomendación de rutas mediante estos atributos.

El diseño es un recomendador que permita sugerir a los usuarios rutas de acuerdo a su nivel de desempeño y su ubicación. La ventaja de realizar este recomendador de apoyo es que obtiene resultados óptimos que complementan la técnica elegida, generando así un sistema de recomendación que se basa en diferentes puntos de vista en relación con el usuario.

Una de las razones de combinar estos resultados, es el hecho de que en algunas situaciones particulares el modelo básico puede tener limitantes, como pueden ser:

- La aplicación se lanza por primera vez a producción.
- La cantidad de usuarios en el sistema es mínima.
- Las rutas no tienen un número suficiente de calificaciones para obtener resultados eficientes
- No se han generado nuevas rutas para ser recomendadas

Finalmente, el sistema de recomendación de rutas para la aplicación es un modelo que combina los resultados de una técnica bien definida de filtrado colaborativo (Slope One) y los resultados de una técnica experimental que utiliza la ubicación y el nivel de desempeño del usuario.

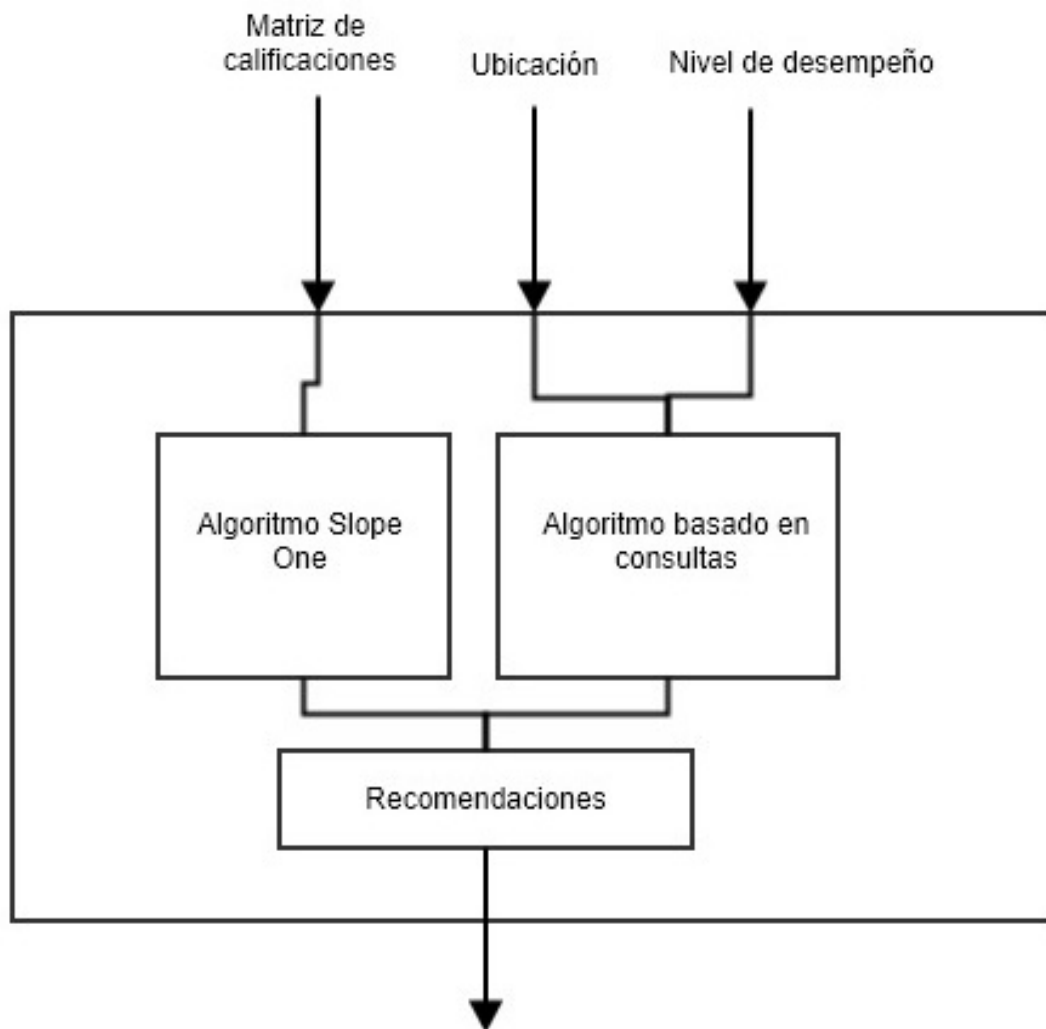


Figura 4.1: Diagrama de flujo del proceso de recomendación para un usuario

Fuente: Elaboración propia

4.2. Modelo de agrupamiento de datos

Según el segundo objetivo que define la realización de una sección de eventos, los cuales deben generarse automáticamente para luego ser sugeridos a los usuarios, se plantea aplicar un proceso de agrupamiento a los usuarios para determinar grupos con características similares basados en los siguientes criterios de afinidad:

- Nivel de desempeño logrado por el usuario como ciclista.
- Días y horarios favoritos para entrenar.
- Rutas preferidas

Después de realizar el proceso de agrupamiento, se obtienen una serie de grupos de usuarios particulares de acuerdo a los criterios y al tiempo en el cual se realice el procedimiento. Estos grupos son la base para la creación de un evento, dado que estos ciclistas serán los que recibirán la invitación al evento.

4.2.1. Algoritmos planteados

Se seleccionaron dos algoritmos para realizar una implementación previa y un análisis experimental. Las técnicas seleccionadas son K-Means y K-Medoids, ambas técnicas de agrupamiento basada en reubicación de particiones, esto dado que el problema requiere formar particiones basándose en atributos de los usuarios, buscando la mayor afinidad entre los elementos de acuerdo a los atributos definidos anteriormente [26].

Algoritmo K-Means

Dado un conjunto de observaciones $(X1, X2, X3 \dots, Xn)$, donde cada observación es un vector real de D atributos, k-means construye una partición de las observaciones en k conjuntos $k \leq n$ a fin de minimizar la suma de los cuadrados dentro de cada grupo $W_{css} : S = s1, s2 \dots sk$.

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

$\boldsymbol{\mu}_i$: Promedio de los puntos de S_i .

(4.6)

Dado un conjunto inicial de k centroides $m1, m2 \dots mk$ el algoritmo continúa alternando entre dos pasos:

- Paso de asignación: Asigna cada observación al grupo con la media más cercana.

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\}$$

(4.7)

Donde cada Xp va exactamente dentro de un $S(i)$, incluso aunque pudiera ir en dos de ellos.

- Paso de actualización: Calcular los nuevos centroides como el promedio de las observaciones en el grupo.

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \quad (4.8)$$

El algoritmo se considera que ha convergido cuando las asignaciones ya no cambian.

Algoritmo K-Medoids

El algoritmo de K-medoids es una variante de K-means que también se conoce como K-means relacional. Similar al algoritmo original, minimiza las distancias entre los puntos de datos y los centroides de los grupos; En contraste con k-means, selecciona los centroides de entre puntos de datos reales y trabaja para cualquier tipo de datos para los cuales podemos definir una distancia. Por lo tanto, el agrupamiento de k-medoids también se aplica a datos que no están incrustados en un espacio vectorial.

Un medoid puede ser definido como el objeto de un grupo cuya disimilaridad media a todos los objetos en el grupo es mínima. Es el punto ubicado más hacia el centro en todo el grupo. El procedimiento del algoritmo es el siguiente:

Seleccionar los k medoids iniciales. $m_1, m_2 \dots m_k$

Iterar mientras el costo disminuya:

- En cada grupo, marcar como medoid el punto que minimiza la suma de distancias dentro del grupo.

$$\mathcal{C}_\kappa = \left\{ \mathbf{x}_i \mid \|\mathbf{x}_i - \mathbf{m}_\kappa\|^2 \leq \|\mathbf{x}_i - \mathbf{m}_\lambda\|^2 \right\} \quad (4.9)$$

Reasignar cada punto al grupo definido por el medoid más cercano determinado en el paso anterior.

$$\mathbf{m}_\kappa = \mathbf{x}_j = \underset{\mathbf{x}_l \in \mathcal{C}_\kappa}{\operatorname{argmin}} \frac{1}{n_\kappa} \sum_{\mathbf{x}_i \in \mathcal{C}_\kappa} \|\mathbf{x}_l - \mathbf{x}_i\|^2 \quad (4.10)$$

4.2.2. Justificación del modelo

Al realizar la implementación de ambos algoritmos se determinó que el algoritmo a utilizar es el K-Medoids esto dado que presenta mejores beneficios que el algoritmo K-Means teniendo en cuenta que:

- Este algoritmo funciona con centroides que pertenecen a los puntos del conjunto de elementos, esta es la principal ventaja dado que nuestro trabajo funciona con puntos (usuarios) con atributos incompletos, es decir las preferencias y los gustos pueden estar o no bien definidos por parte del usuario, lo que para el algoritmo K-means significaría generar ruido ya que los valores no definidos igual contribuyen al promedio del grupo pero con datos erróneos.
- El algoritmo K-Medoids al trabajar unicamente con puntos del conjunto hace que solo sea necesario determinar una métrica de similitud entre usuarios sin necesidad de determinar un vector de atributos. Para una explicación más detallada véase [35].

Datos de entrada

- M : Matriz de distancias entre usuarios que determinan la similitud entre ellos. Una distancia menor indica una grado mayor de similitud y viceversa. Estos valores estan normalizados en el intervalo $[0,1]$

	Usuario 1	Usuario 2	Usuario 3	Usuario 4
Usuario 1	0	0.5	1	0.8
Usuario 2	0.5	0	0.6	0.3
Usuario 3	1	0.6	0	0.4
Usuario 4	0.8	0.3	0.4	0

Tabla 4.6: Ejemplo de matriz de distancias M
Fuente: Elaboración propia

Datos de salida

- C : Lista de vectores, donde cada vector C_i $0 \leq i \leq n$ con $n = \text{número de grupos obtenidos}$, contiene los usuarios pertenecientes a ese grupo para los cuales se va a crear y sugerir un nuevo evento.

$[\{Usuario1, Usuario4\}, \{Usuario2, Usuario3\}]$

Ejemplo de lista de grupos obtenidos C

4.3. Modelo para el sistema de gamificación mediante retos

Para la aplicación desarrollada se modeló un sistema basado en gamificación para que los usuarios tengan incentivos a la hora de interactuar con el dispositivo. El modelo se diseñó de acuerdo a los siguientes elementos:

Componentes del juego

- **Lista de retos:** Se cuenta con una serie de retos para que el usuario pueda cumplir.
- **Monitor de entrenamientos:** Para cumplir con ciertos retos, el usuario debe realizar entrenamientos en su bicicleta, según esto se modela una sección para que el usuario monitoree sus entrenamientos mediante el mapa y la localización del GPS.
- **Crear y calificar:** Uno de los criterios para cumplir algunos retos es crear y calificar cierta cantidad de rutas, para esto se cuenta con el módulo de rutas que permite a un usuario realizar este proceso.

Mecánicas de juego

- **Puntos:** Cada reto tiene un premio, el premio es una cantidad de puntos que varía según la dificultad del reto.
- **Niveles:** A medida que un usuario consigue puntos al cumplir los retos, cuando se llega a cierta cantidad de puntos acumulados se escala en un nivel. Los niveles que un usuario puede obtener son: Novato, aprendiz, campeón, profesional, maestro.

Dinámicas de juego

- **Recompensa:** Cuando los usuarios suban de nivel en la aplicación se le entregaran recompensas, estas son distintivos que caracterizan el desempeño del usuario.

Capítulo 5

Desarrollo del proyecto

En este capítulo se muestran los aspectos más relevantes del proceso de implementación. Se explican qué estrategias se usaron, los lenguajes de programación usados y la estructura del aplicativo.

Un aspecto muy importante tenido en cuenta fue el de la seguridad de los datos y la prevención de fallos.

Para desarrollar el aplicativo fue necesario diseñar e implementar dos estructuras con diferentes técnicas, enfoques y lenguajes de programación, la primera es un servicio web REST y el segundo una aplicación nativa para dispositivos Android.

5.1. Diseño e implementacion del servicio web REST

Para desarrollar la aplicación fue necesario implementar un servicio web REST, este tipo de servicio permite el intercambio y manipulación de datos en las aplicaciones web a través de un protocolo de manejo de recursos que son accedidos desde aplicaciones externas [36]. El código fuente del servicio se encuentra disponible en [37].

5.1.1. Características del modelo REST

- **Protocolo cliente/servidor sin estado:** Cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro:
 - **GET** Recuperan información. Las solicitudes GET son idempotentes, es decir que independientemente de cuántas veces se repitan con los mismos parámetros, los resultados son los mismos.

GET usuarios/1

- **POST** Las solicitudes POST son utilizadas para crear nuevas entidades segun una serie de parámetros pasados a través del cuerpo de la petición.

POST usuarios

```
{
  "id": "1",
  "nombre": "Usuario",
  "direccion": "Universidad del valle"
}
```

- **PUT** Las solicitudes PUT son utilizadas para actualizar entidades identificadas en la URI dependiendo de los parámetros pasados a través del cuerpo de la petición.

PUT usuarios/1

```
{
  "id": "1",
  "nombre": "Usuario actualizado",
  "direccion": "Universidad del valle"
}
```

- **DELETE** Solicitan que se elimine un recurso; sin embargo, el recurso no tiene que quitarse inmediatamente. Podría ser una solicitud asincrónica o de larga duración.

DELETE usuarios/1

- **Manipulación URI:** Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.
- **Interfaz uniforme:** para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- **Sistema de capas:** arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

5.1.2. Entorno de desarrollo

Teniendo en cuenta que el servicio tiene la característica de estar basado en un modelo REST, se planteó la implementación mediante el framework de desarrollo web Django basado en el lenguaje de programación Python. Esta elección se hizo basado en los siguientes criterios:

- Facilidad de implementación de un servicio REST.
- Conocimiento previo sobre el lenguaje de programación.

- Flexibilidad y capacidad de personalización.
- Arquitectura muy comprensible y básica
- Técnicas para el manejo de seguridad y prevención de errores.

Requerimientos

- **Python 3.5:** Lenguaje de programación interpretado para el desarrollo de la aplicación del lado del servidor.
- **Django 3.6:** Framework de desarrollo web de código abierto, escrito en Python, que trabaja mediante un patrón Model-View-Template.
- **Django Rest Framework 1.10:** Librería para la administración de servicios REST de manera fácil, segura y confiable.
- **Numpy 1.10:** Librería para el manejo de operaciones matemáticas especiales.
- **MySQL:** Servidor de administración de base de datos.

5.1.3. Arquitectura

La aplicación se diseña de acuerdo a la siguiente arquitectura:

- **Persistencia:** Está compuesta por la base de datos y la estructura de archivos en el servidor donde se almacena información de forma consistente. Es la parte más importante dado que guarda todo el contenido de la aplicación y de todos los usuarios en sus dispositivos. Para información detallada del modelo de la base de datos véase el anexo en la página 60
- **Modelo:** Son las clases encargadas de administrar la comunicación entre las vistas y la persistencia, estas a su vez definen la estructura de los recursos del servicio y sus atributos.
- **Serializador:** Tiene como función principal definir la estructura de los recursos que serán enviados y recibidos a través de las peticiones en formato JSON.
- **Vista:** Son las encargadas de manejar las peticiones del servidor y re direccionarlas a los modelos de acuerdo al tipo de recurso que se esté solicitando

5.1.4. Implementación del recomendador de rutas

La implementación del recomendador se realizó basado en el modelo planteado, el proceso es el siguiente:

- Se leen todos los usuarios del sistema desde la base de datos y se guardan en una lista.
- Para cada usuario:
 - Se ejecuta el recomendador
 - Se obtiene una lista de tuplas que contienen el identificador de la ruta recomendada y el valor de la recomendación.
 - Por cada recomendación obtenida se guarda en la base de datos y se conecta directamente con el usuario

Este proceso se realiza en el servidor ya que es aquí donde se cuenta con la información de todos los dispositivos y los valores están actualizados. Este proceso se planteó como una tarea periódica que se ejecuta automáticamente los días miércoles y domingo a las 12:00 AM, es de esta manera para permitir que entre días se agreguen calificaciones y nuevos usuarios logrando obtener valores más óptimos a recomendar.

5.1.5. Implementación del modelo de sugerencia de eventos

El modelo se implementa siguiendo las pautas definidas, utilizando para algunas funciones especiales la librería Numpy. El proceso de creación y sugerencia de eventos se realiza de la siguiente manera:

- Se leen todos los usuarios de la aplicación desde la base de datos.
- Se calcula la matriz de distancias entre usuarios.
- Se calcula el número de grupos a crear mediante la siguiente función
- Se obtiene una lista que contiene los grupos obtenidos.
- Para cada grupo obtenido:
 - Se ejecuta el recomendador
 - Se calcula el nivel del grupo promediando los niveles de los usuarios que pertenecen a él.
 - Se genera el nombre para el evento de acuerdo a un diccionario de palabras que se obtiene en base a la dificultad del evento.
 - Se guarda el evento en la base de datos.
 - Se guardan cada uno de los invitados al evento en la base de datos, el cual relaciona el evento, el usuario y la decisión de si asistirá, quizá asista o no asistirá al evento.

Esta ejecución se realiza los días martes y sábado a las 12:00 AM, que permite la creación de nuevos eventos de manera constante, dando la posibilidad de tener alternativas para los usuarios.

5.2. Diseño e implementación de la aplicación Android

Según estudio de una empresa consultora y de investigación de las tecnologías de la información Gartner, en un estudio anual realizado para finales del año 2016 [38] indica que el sistema operativo Android controla el 86 % del mercado global de teléfonos y dispositivos móviles a nivel mundial. Estas cifras sirvieron de base para determinar que para la interfaz del lado del cliente se desarrolló una aplicación nativa para dispositivos con sistema operativo Android.

5.2.1. Entorno de desarrollo

La implementación se realizó en el ambiente de desarrollo integrado Android Studio, el cual es el entorno oficial para el desarrollo de aplicaciones para los dispositivos Android, diseñado por la compañía Google Inc. Entre las ventajas de utilizar este entorno se encuentran:

- Integración de plantillas de código y GitHub para compilar funciones comunes de las apps e importar ejemplos de código
- Manejo de un sistema de compilación basado en Gradle, el cual ayuda a compilar las dependencias de la aplicación.
- Un emulador rápido con varias funciones.
- Gran cantidad de herramientas y frameworks de prueba.
- Instant Run para aplicar cambios mientras la aplicación se ejecuta sin la necesidad de compilar un nuevo APK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.

5.2.2. Estructura del proyecto

Cada proyecto en Android Studio contiene uno o más módulos con archivos de código fuente y archivos de recursos [39]. Entre los tipos de módulos se incluyen los siguientes:

- Módulos de apps para Android
- Módulos de bibliotecas
- Módulos de Google App Engine

Los módulos de la aplicación son los que se depositan los archivos para el desarrollo de la aplicación. Las carpetas se dividen de la siguiente manera:

- **manifests:** contiene el archivo `AndroidManifest.xml`. Este archivo es el encargado de definir los elementos que componen la aplicación, como actividades, servicios, tareas entre otros.
- **java:** contiene los archivos de código fuente de Java.
- **res:** Contiene todos los recursos, como diseños XML, cadenas de IU e imágenes de mapa de bits.

5.2.3. Arquitectura

Es una aplicación nativa para el sistema operativo Android que se diseñó teniendo en cuenta las necesidades de los usuarios. Dada la naturaleza de la aplicación el uso principal se realiza en zonas donde no se tiene conexión a internet y la conexión de datos móviles es solo recomendado para situaciones críticas y especiales.

Según estos criterios la arquitectura de la aplicación se basa en un modelo de funcionamiento offline, es decir todas las tareas realizadas se almacenan de manera local y a través de un sistema de sincronización implementado se actualizan los datos en el servidor cuando el dispositivo se conecta a internet.

Las ventajas de esta arquitectura para la aplicación son:

- Uso de la aplicación en cualquier lugar sin depender de conexión a internet.
- Modelo robusto y eficiente de datos
- Disponibilidad total de la información

Las desventajas de la implementación de esta arquitectura son:

- Mayor esfuerzo para el desarrollo.
- Aumento en la complejidad de la aplicación.
- Necesidad de construcción de un modelo de persistencia local

5.2.4. Módulos de la aplicación

El sistema de desarrollo de manera modular teniendo como base el principio de alta cohesión y bajo acoplamiento, esto dado que se busca que cada módulo realice su tarea eficientemente y lo más independiente del resto, lo que permite que la aplicación sea:

- Escalable a funcionalidades futuras.
- Fácil de depurar y administrar.
- Resistente a fallos. Ítem Código más legible e intuitivo

Se utilizó un modelo de diseño MVP (Model-View-Presenter), el cual es un patrón de desarrollo para aplicaciones móviles que permite escribir un código más legible separando las funcionalidades principales.

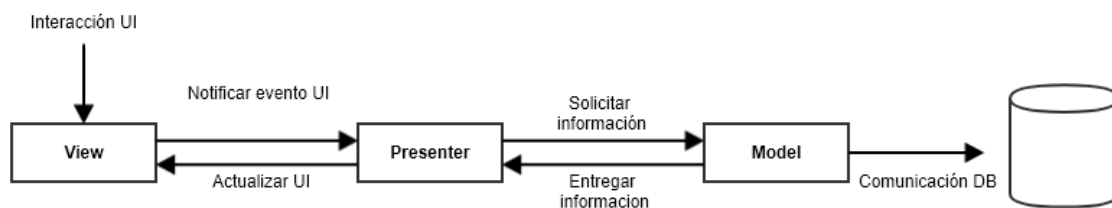


Figura 5.1: Patrón de desarrollo MVP

Fuente: Elaboración propia

- **View:** Representa una abstracción que representa que se puede hacer con la vista, normalmente se asocia a una *interface* para representar la funcionalidad de una vista. La parte importante está en que una Activity o un Fragmento tienen la responsabilidad únicamente de implementar la interface View y conectar las acciones del usuario con el presenter.
- **Presenter:** Es el encargado de coordinar la implementación de la vista y el modelo, actualiza la vista y actúa sobre los eventos de usuario que se envían por la vista. El presenter también recupera los datos del modelo y los prepara para su visualización.
- **Model:** Es el proveedor de los datos que queremos mostrar en la vista, es el que implementa los casos de uso.

El código fuente de la aplicación se encuentra disponible en [40]

Módulo de persistencia

Se encarga de almacenar la información de manera local en una base de datos Sqlite disponible solo para la aplicación y administrada en los archivos del sistema. La base de datos es una copia del servidor. Aquí se define la estructura y la versión.

Módulo de conexión al servidor

Es el módulo que tiene la función de realizar la conexión entre el dispositivo y el servicio REST a través de una interfaz de comunicación. La interfaz utilizada es Retrofit, una librería desarrollada para manipular peticiones de manera fácil y efectiva, mejorando las dificultades que presentan las librerías propias de android SDK [41]. Las principales ventajas son:

- Seguridad en las peticiones, protegiendo la información transmitida.
- Facilidad de uso.
- Resistente a fallos durante la conexión.
- Manejo eficiente de datos consumidos.

Aquí se definen la lista de peticiones REST necesarias para enviar y recibir la información del servidor. Para realizar una comunicación más segura, se implementó una estrategia que tiene como objetivo enviar las peticiones con un encabezado que contiene una clave secreta encriptada para permitir que solo se pueda acceder desde el dispositivo que cuenta con esta y no a través de navegadores web o algún programa para realizar peticiones a una URL determinada. Esta estrategia es la que utilizan la mayoría de sitios web como Netflix cuando definen una API para el acceso a sus datos, mediante la entrega de una clave secreta al cliente.

Módulo de sincronización

La sincronización es uno de los temas sobre desarrollo Android que más trabajo puede exigir, es de vital importancia porque permite que las aplicaciones conserven de forma exitosa los datos de los usuarios.

Teniendo en cuenta que la aplicación funciona de manera offline, era necesario contar con un sistema que sincronizara la información tanto en el dispositivo como en el servidor, este proceso es el más complejo ya que debe cumplir con ciertos criterios que determinan la robustez del módulo:

- **Resistencia a fallos:** Debe funcionar de manera tal que no afecte ningún proceso del sistema en caso de presentar fallos en la comunicación de datos, la conexión a internet u algún otro problema que se presente en el momento de sincronizar.
- **Corrección de errores:** En caso de presentarse fallos durante un proceso de actualización, debe contarse con una técnica que permita recuperar el estado anterior sin perder ninguna información.
- **Trabajo en segundo plano:** La sincronización es una tarea que debe realizarse independientemente de la interacción del usuario con la aplicación, por eso debe funcionar en un proceso diferente, en caso de ser necesario solo puede notificar al usuario que ya se terminó la actualización.
- **Congruencia de la información:** La información debe mantener siempre estable, es decir por ninguna circunstancia deben actualizarse datos con valores incorrectos.
- **Comunicación privada:** El proceso de sincronización es una tarea crítica ya que maneja la información de los usuarios, de esta manera se debe hacer un encapsulamiento de la información a través de proveedores de contenido para no permitir el acceso a la base de datos o el proceso de conexión al servidor por parte de fuentes externas.

Sincronización local

Es el proceso en el cual los datos que existen en el servidor son depositados en el dispositivo móvil, esta tarea realiza una petición que trae todos los registros desde el servidor y los inserta, modifica o elimina según corresponda. Se realiza de manera jerárquica para evitar errores de dependencias a través de claves foráneas. La sincronización se ejecuta automáticamente cada cierto tiempo para mantener los datos al día.

Sincronización remota

Este tipo de sincronización se realiza cada cierto tiempo tomando valores que han sido insertados, modificados o eliminados de manera local para luego a través de peticiones realizar la actualización correspondiente en el servidor y ponerla a disposición del resto de usuarios. Este proceso requiere que los valores locales se marquen con banderas para evitar problemas de congruencia de datos, valores repetidos y realizar operaciones varias veces.

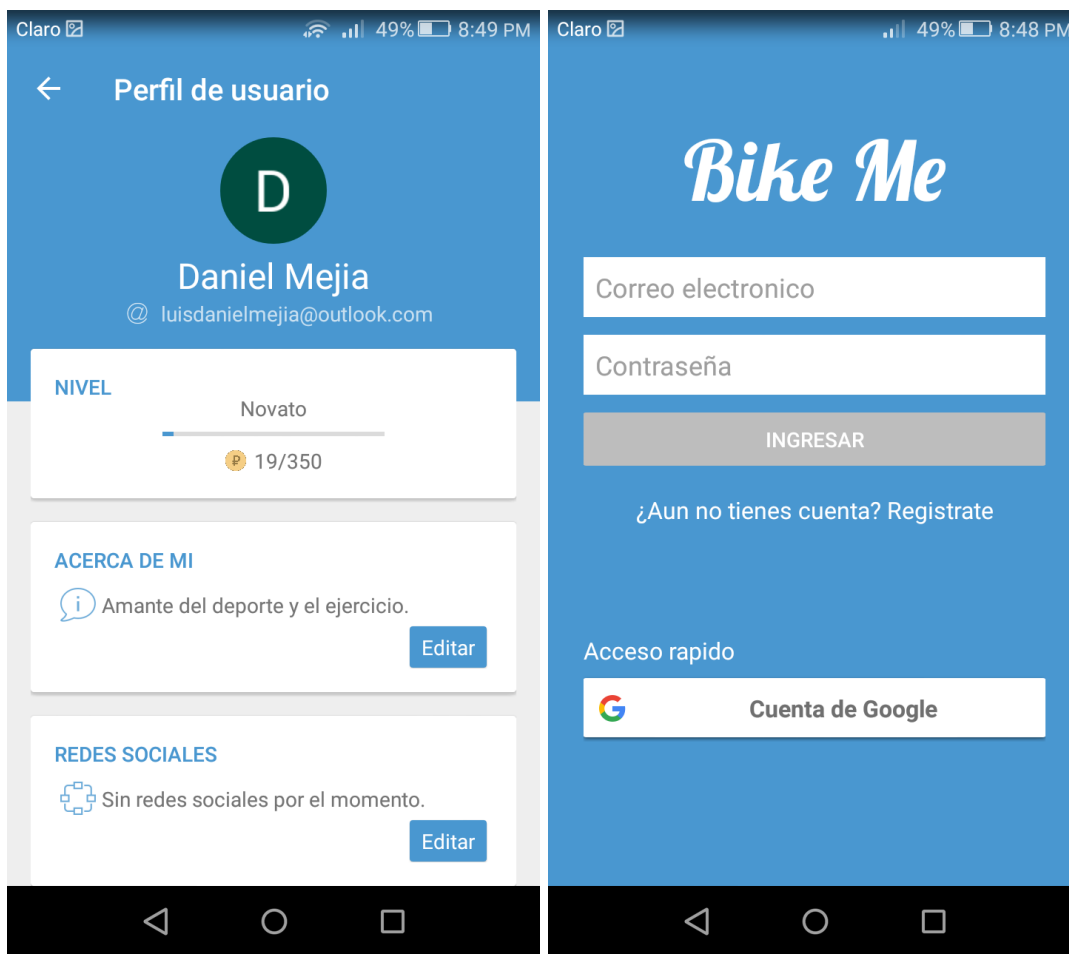
La implementación se realizó a través de un modelo de Sync Adapters que define una serie de elementos necesarios para sincronizar datos entre un servidor externo y dispositivos móviles. Los requerimientos que se implementaron son los siguientes:

- **Sync Adapter:** Es el que define la estructura de la sincronización, allí se escriben los métodos que actualizan la base de datos y acceden al servidor.
- **Service:** La definición del proceso separado para la sincronización.
- **Account Authenticator:** Para realizar la sincronización se requiere autenticar el dispositivo a través de la cuenta registrada, esto para realizar tareas de manera segura.
- **Content Provider:** Se define el proceso de acceso a la base de datos a través de peticiones protegidas de manera encapsulada.

Módulo de usuarios y autenticación

Aquí se dispone de una sección para el registro de usuarios nuevos y la autenticación de usuarios existentes para ingresar a la aplicación. Se cuenta con una implementación de inicio de sesión a través de una cuenta de Google sincronizada en el teléfono.

El registro y autenticación de usuarios se maneja mediante Firebase, una plataforma desarrollada para administrar servicios para aplicaciones, la decisión de elegir esta plataforma es debido a que el proceso de autenticar usuarios en el servidor necesita requerimientos especiales para el control y la seguridad de los datos, teniendo en cuenta que este módulo no hace parte del núcleo principal de los casos de uso de la aplicación, pero es de vital importancia proteger los datos de los usuarios, se decidió elegir este sistema que realiza todo el proceso en un servidor externo y retorna los datos autenticados correctamente. Después de esto lo que único que basta es almacenar los datos personales del usuario en nuestro servidor.



(a) Pantalla de perfil de usuario

(b) Pantalla de inicio de sesión

Figura 5.2: Ejemplos de pantallas del módulo de autenticación y usuarios

Módulo de rutas

Es el modulo que se encarga de controlar todas las operaciones relacionadas con una ruta, aquí se encuentran las siguientes tareas:

- **Crear ruta:** Se permite la creación de una ruta, primero trazándola en el mapa y luego ingresando información descriptiva sobre la misma. Esta tarea es la única para la cual se necesita conexión a internet ya que necesita calcular en tiempo real las carreteras más cercanas para la ruta que el usuario trazo manualmente.
- **Visualizador de rutas:** El usuario puede ver las rutas de la aplicación en base a tres criterios definidos, primero se muestran las rutas sugeridas por el recomendador implementado, segundo las rutas nuevas, es decir que se crearon hace poco tiempo o tienen pocas calificaciones, esto para permitir que las rutas más nuevas sean mostradas a los usuarios y puedan recibir calificaciones para ser sugeridas por el recomendador, por ultimo está la sección de las rutas creadas o calificadas por el usuario.
- **Ver el detalle de una ruta específica:** Se tiene la posibilidad de ver específicamente la información de una ruta, mostrar su recorrido en el mapa y poder calificarla.



Figura 5.3: Ejemplo de una pantalla del módulo de rutas

Módulo de eventos

Se pueden visualizar todos los eventos en una línea de tiempo cronológicamente ordenada en la cual, los eventos sugeridos al usuario se muestran marcados para diferenciarlos del resto, cuando se ingresa a la descripción de un evento particular se puede observar su información, fecha, hora, recorrido en el mapa, además se puede ver los usuarios invitados al evento, se tiene la posibilidad de decidir si se asistirá al evento, quizá asista o definitivamente no asistirá.

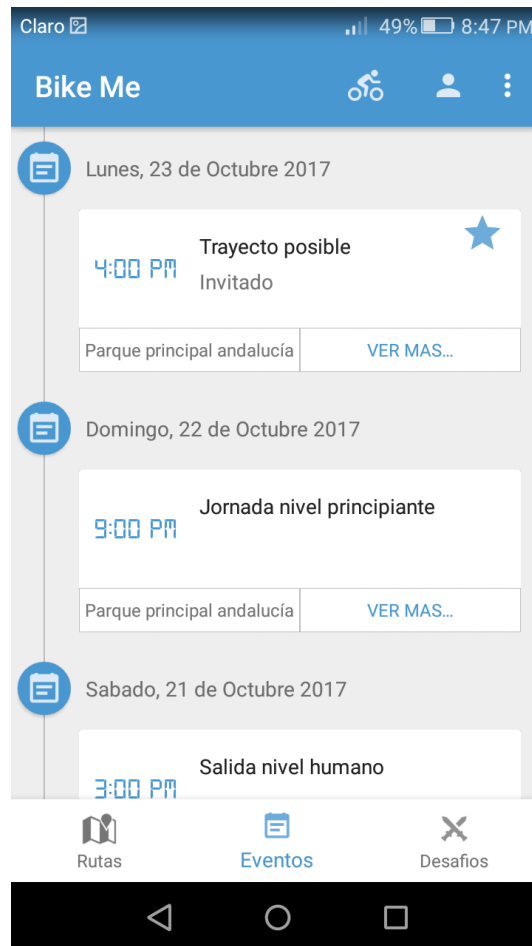


Figura 5.4: Ejemplo de una pantalla del módulo de eventos

Módulo de retos

Se muestran una serie de retos, cada reto define la condición necesaria para cumplirlo, el porcentaje del reto que se ha logrado, los retos se dividen en nueve categorías:

- Entrenamiento
- Rutas creadas
- Rutas calificadas
- Tiempos

- Distancias
- Metas semanales
- Metas mensuales
- Records personales de distancias
- Records personales de tiempo

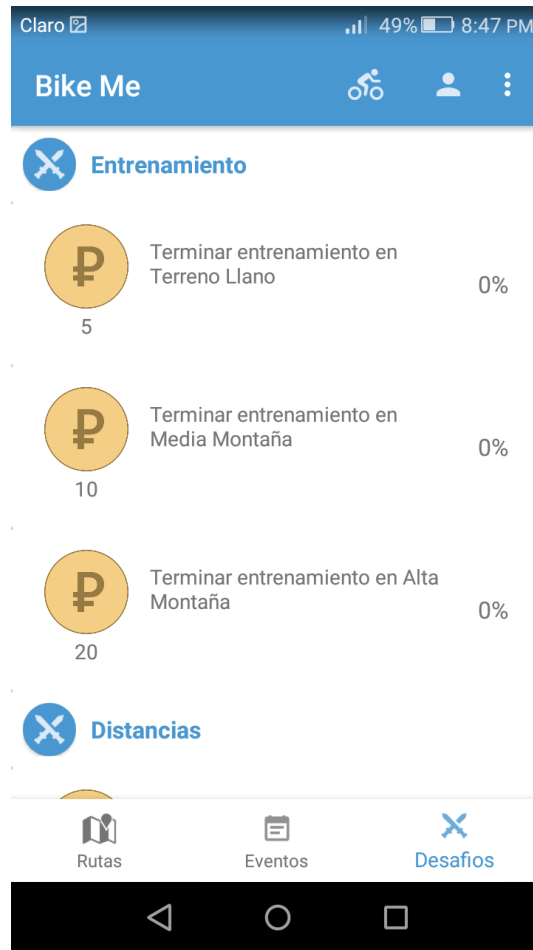


Figura 5.5: Ejemplo de una pantalla del módulo de retos

Módulo de entrenamiento

El usuario puede monitorear un entrenamiento o un recorrido en bicicleta a través del mapa, cuando se inicia, la aplicación automáticamente hace un seguimiento de tu ubicación mostrando el recorrido realizado en el mapa. Luego de terminar el trayecto se puede guardarlo o descartarlo, este muestra las estadísticas del entrenamiento:

- Fecha
- Duración

- Altitud promedio del recorrido
- Distancia total
- Tipo de ruta realizada
- Trayecto en el mapa

Esta tarea funciona en segundo plano por lo que el usuario puede cerrar la aplicación o moverse a otra interfaz y el sistema seguirá monitoreando el recorrido. Cabe resaltar que algunos dispositivos que cuentan con aplicaciones que protegen el uso de la batería eliminan los procesos en segundo plano cuando la aplicación se cierra, por tal motivo se recomienda no cerrarla en caso de estar durante un entrenamiento u otorgarle permisos a la aplicación para no permitir este comportamiento.

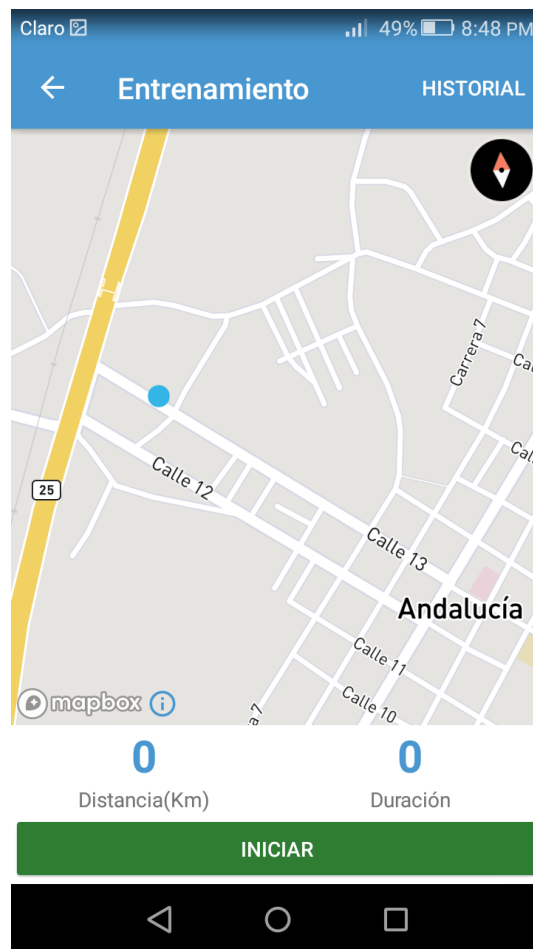


Figura 5.6: Ejemplo de una pantalla del módulo de entrenamiento

Capítulo 6

Pruebas

En este capítulo se describe un proceso de pruebas a la aplicación desarrollada en el proyecto, para realizar un estudio con respecto a qué ventajas presentan ciertas características de la aplicación basado en lo implementado, el motor de recomendación de rutas y el sistema de creación de eventos.

6.1. Pruebas de software

Se realizaron pruebas para probar la aplicación, estas se realizaron para probar los tres elementos principales, rutas, eventos y retos. El objetivo es proporcionar información objetiva e independiente sobre la funcionalidad del sistema.

6.1.1. Pruebas unitarias

Una prueba unitaria nos ayuda a probar una unidad estructural de código. Debido a la diversidad de definiciones, convendremos que una “buena” prueba unitaria tiene las siguientes características:

- **Unitaria:** Prueba solamente pequeñas cantidades de código.
- **Independiente:** No debe depender ni afectar a otras pruebas unitarias.
- **Prueba métodos públicos:** De otra forma la prueba sería frágil a cambios en la implementación y no se podría utilizar en pruebas de regresión
- **Automatizable:** La prueba no debería requerir intervención manual.
- **Repetible y predecible:** no debe incidir el orden y las veces que se repita la prueba, el resultado siempre debe ser el mismo.

Para el proceso de pruebas se utilizan objetos simulados ”mock” que sustituyen a los objetos reales que se necesitan para la unidad a probar. Los elementos que se utilizaron para el proceso de pruebas fueron:

- **JUnit:** Librería utilizada por Android Studio para el proceso de pruebas.
- **Mockito:** Librería para el uso de objetos mock para java.
- **PowerMock:** Librería para hacer mock a objetos propios del sistema.
- **Rx Java:** Librería para administrar el proceso de pruebas a métodos asíncronos.

Las pruebas se diseñaron de tal manera que se validen los elementos básicos de la aplicación, teniendo en cuenta que la aplicación se implementó en base al modelo MVP, las pruebas están enfocadas en verificar estas secciones. Estas se implementaron realizando aserciones, las cuales verifican que las tareas que realizan cada uno de los métodos sea coherente con el valor esperado. Las pruebas se encuentran disponibles en el código fuente de la aplicación en Android en sus respectivos paquetes de pruebas.

Ejemplo

Una de las pruebas realizadas se diseñó para el proceso de rectificar una ruta dibujada en el mapa mediante el API Google Maps, esto para alinearla a puntos que pasen por las calles más cercanas. Este proceso se muestra a continuación.

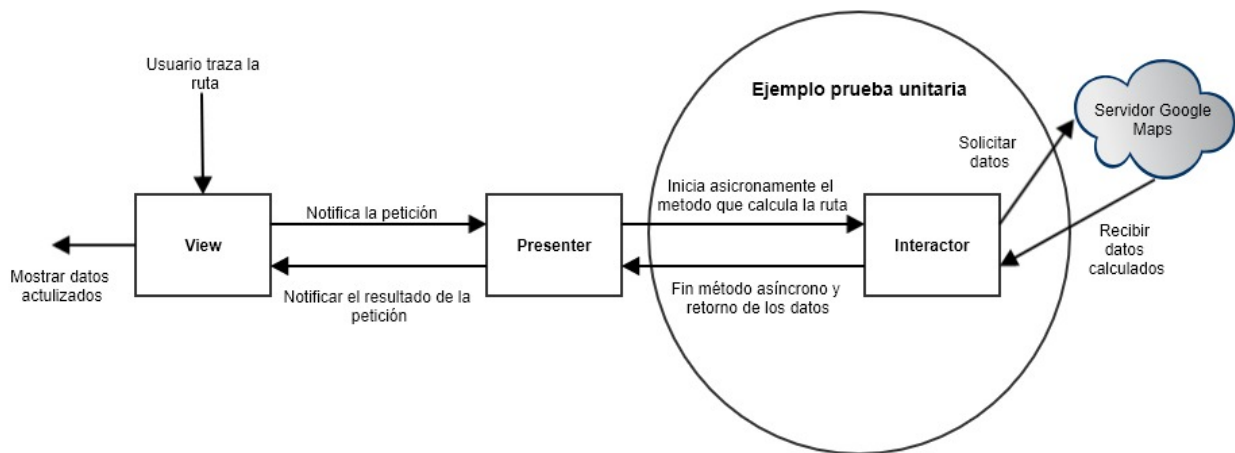


Figura 6.1: Proceso de alineación de ruta al mapa

Fuente: Elaboración propia

Una de las pruebas unitarias fue para el método de calcular ruta que se encuentra en el interactor, allí se construye una petición al servidor de la API, se traen los datos del servidor, se calcula la distancia a partir de los puntos y por último se retorna para el siguiente tratamiento, todo esto se realiza en una tarea asíncrona para no bloquear la interfaz del usuario.

La prueba realiza este proceso por separado, para acceder al servidor se crea un mock que retorne el valor esperado, dado que nos interesa el proceso interno del método mas no lo que se realice en el servidor, aquí se prueba que cada tarea se realizó correctamente hasta que se retorna el valor esperado para el siguiente paso.

6.2. Pruebas generadas a los algoritmos implementados

Para determinar la efectividad de los algoritmos se diseñaron una serie de pruebas con valores generados, estos datos son variables para probar diferentes escenarios a los que se pueden enfrentar. La idea es evaluar cómo se comportan los algoritmos con algunos valores con el fin de verificar si hay escenarios en los que la efectividad del algoritmo disminuya o sea diferente a los esperado.

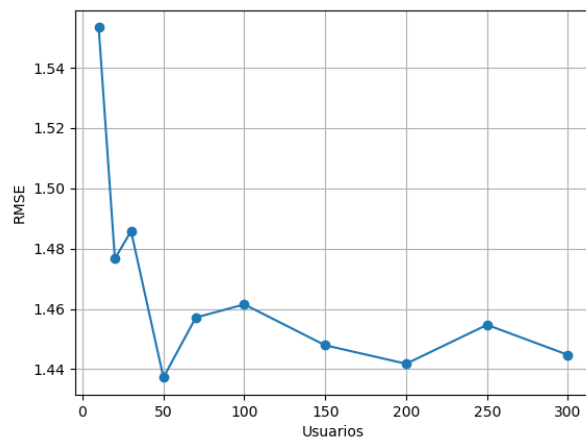
6.2.1. Recomendador de rutas

Para probar la efectividad del recomendador de rutas se evaluó primero la calidad de las recomendaciones mediante el índice RMSE, un indicar que se explicó anteriormente en el capítulo 4. Otro factor importante que se evaluó fue el tiempo de ejecución del algoritmo. Todas las pruebas se realizaron tomando el 20 % de los datos para entrenar el recomendador y el 80 % restante para validar las recomendaciones obtenidas.

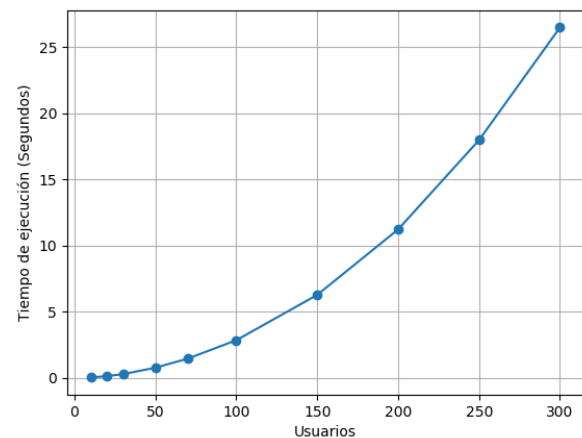
Tipos de prueba

- **Número de usuarios incremental:** Se generaron conjuntos de datos con diferentes cantidades de usuarios para validar el comportamiento del recomendador a medida que ingresan nuevas personas a la aplicación.
- **Número de rutas incremental:** El objetivo era verificar cómo funciona el recomendador cuando se crean nuevas rutas al sistema.
- **Número de rutas y usuarios incremental:** Busca mostrar cómo se comporta el sistema cuando hay tanto nuevas rutas como nuevos usuarios.
- **Diferente cantidad de calificaciones:** La idea es encontrar si la efectividad del algoritmo es diferente cuando hay mayor o menor cantidad de calificaciones disponibles.

Resultados



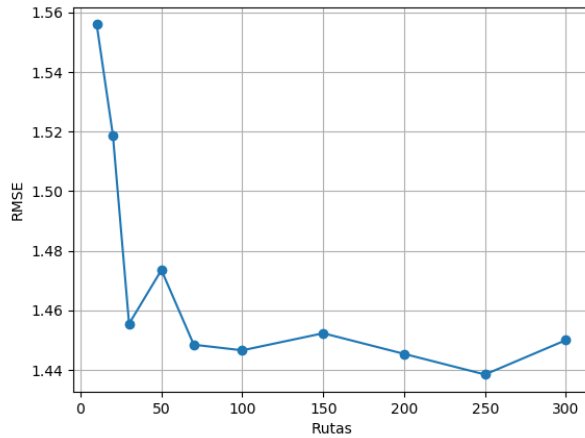
(a) Índice de efectividad RMSE



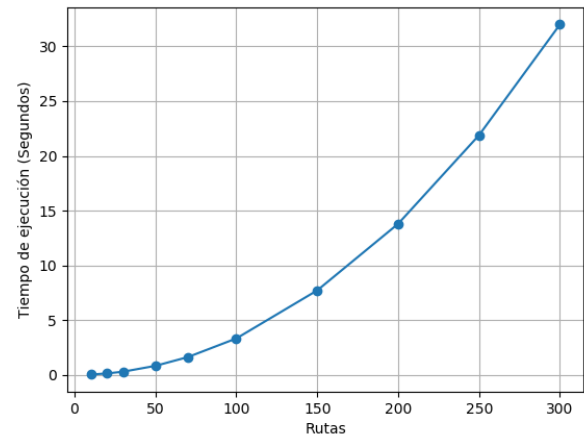
(b) Tiempo de ejecución

Figura 6.2: Resultado de pruebas para usuarios incremental

En la figura 6.2 se muestra que cuando hay pocos usuarios, menos de 50, la efectividad del recomendador se ve afectada un poco, esta diferencia indicaría recomendaciones con unas décimas de error mayor de lo normal, sin embargo después de cierto número de usuarios la efectividad se estabiliza en un intervalo muy pequeño, esto garantiza recomendaciones normales a través del tiempo. Hablando del tiempo de ejecución del algoritmo, este aumenta de manera polinomial a medida que aumenta el número de usuarios, sin embargo la amplitud es grande por lo que aumenta lentamente en intervalo de segundos.



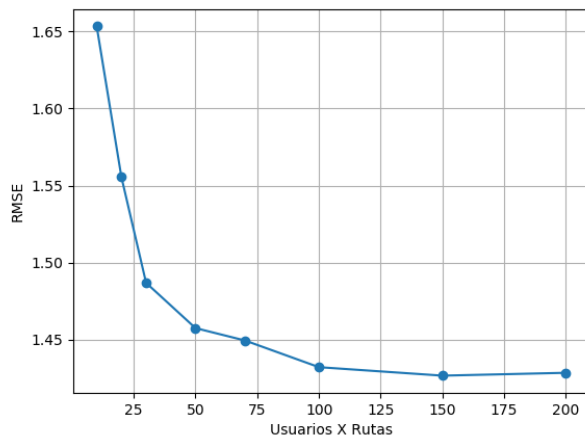
(a) Índice de efectividad RSME



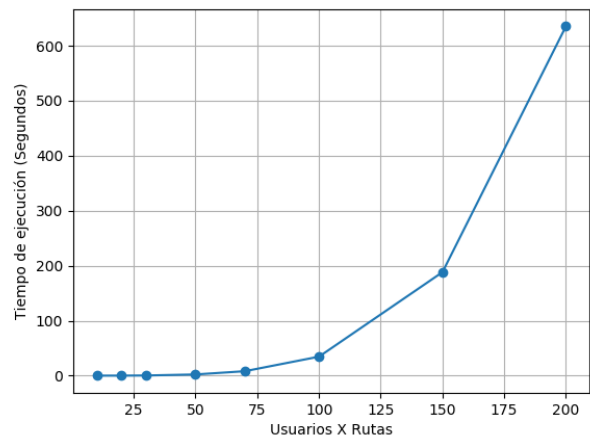
(b) Tiempo de ejecución

Figura 6.3: Resultado de pruebas para rutas incremental

En la figura 6.3 se puede ver que a medida que aumenta el número de rutas en el sistema al igual que con los usuarios, la efectividad del recomendador se estabiliza. De la misma manera sucede con el tiempo de ejecución, ya que se define mediante una función polinomial pero en un intervalo creciente lento.



(a) Índice de efectividad RSME



(b) Tiempo de ejecución

Figura 6.4: Resultado de pruebas para usuarios y rutas incremental

La figura 6.4 muestra que cuando aumentan los usuarios y las rutas conjuntamente, la efectividad del algoritmo aumenta de manera parabólica hasta estabilizarse en un punto en el que es casi asintótica, cabe resaltar que en la gráfica cuando disminuye el RMSE, la efectividad del algoritmo aumenta. En este punto el tiempo de ejecución es diferente, dado que aumenta de manera polinomial pero con una amplitud pequeña, lo que hace que el tiempo crezca rápidamente en intervalo de minutos.

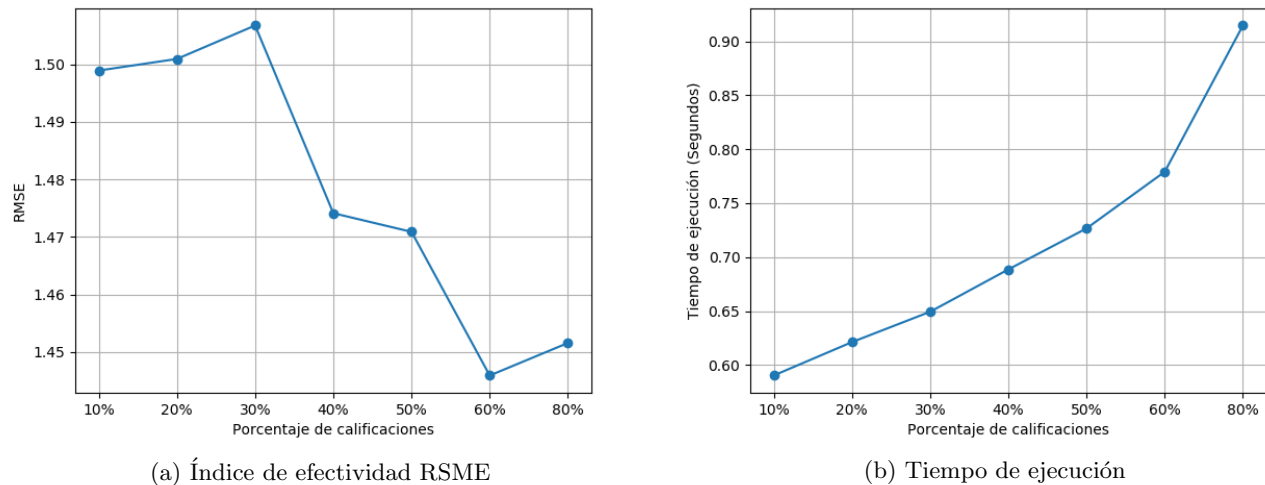


Figura 6.5: Resultado de pruebas para porcentaje de calificaciones

En la figura 6.5 se puede ver que la efectividad del algoritmo empieza a disminuir cuando se tiene al menos el 30 % de calificaciones disponibles, lo que tiene sentido dado que estas son la base del recomendador para encontrar valores nuevos. Por otra parte el tiempo de ejecución del algoritmo aumenta pero de manera muy lenta, analizando que el intervalo es muy pequeño, la ejecución varía en milisegundos por lo que no hay mucha diferencia.

Pruebas con datos en producción

La aplicación estuvo en producción disponible para los usuarios durante 15 días antes de realizar estas pruebas, durante este periodo se logró obtener los siguientes datos:

- 22 Usuarios registrados.
- 18 rutas creadas en el sistema.
- Un promedio de 5 calificaciones por ruta.
- Un promedio de 6 calificaciones realizadas por usuario.
- 4 ejecuciones del recomendador.

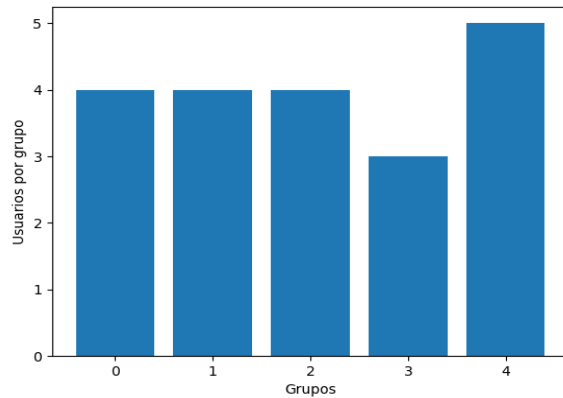
Teniendo en cuenta estos datos se realizó el proceso de pruebas, utilizando el 80 % la matriz de calificaciones de los usuarios como conjunto de entrenamiento del recomendador y el restante 20 % para validar los valores obtenidos. Después de la prueba se obtuvo que el índice de efectividad RMSE fue de 1.47, un índice bueno teniendo en cuenta los resultados de las pruebas al algoritmo de manera automática, sin embargo no es el mejor resultado, dado que la cantidad de información es mínima, a pesar de esto, esta prueba muestra el potencial del algoritmo a medida que aumente el uso de la aplicación.

6.2.2. Creador de eventos

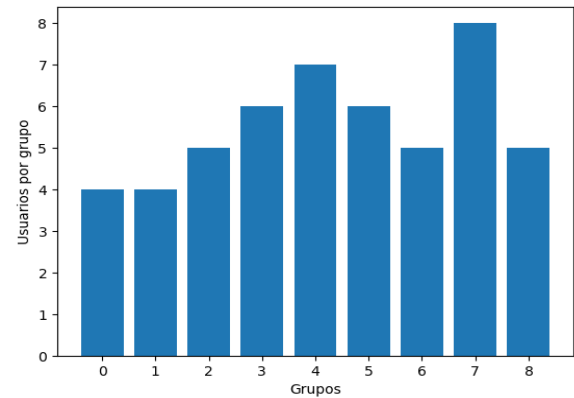
Para crear los eventos se realice un proceso de agrupamiento, para determinar que tan bien formados quedan los grupos, se evalúa la distribución de los usuarios en los K grupos, las pruebas a desarrollar se basan en los siguientes criterios.

- **Número de usuarios incremental:** La idea es validar como se forman los grupos a medida que aumenta el número de usuarios.
- **Número de rutas incremental:** El objetivo evaluar si al incrementar el número de rutas, los grupos se forman de manera diferente.
- **Diferente cantidad de calificaciones:** Que sucede con los grupos si hay menos calificaciones disponibles.

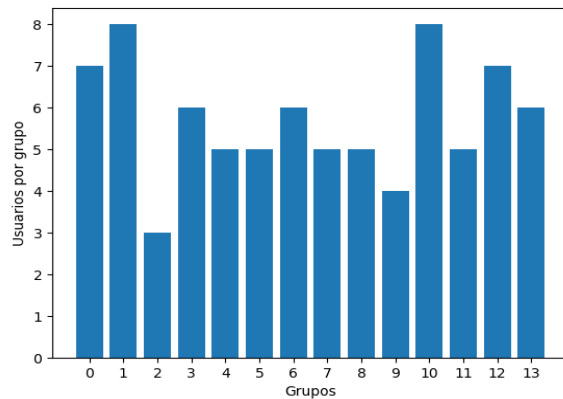
Resultados



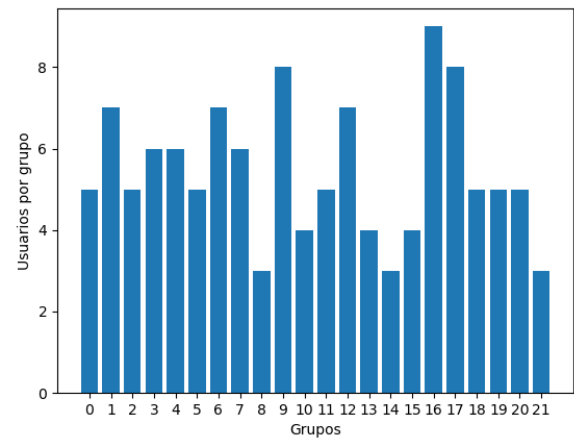
(a) 20 Usuarios



(b) 50 Usuarios



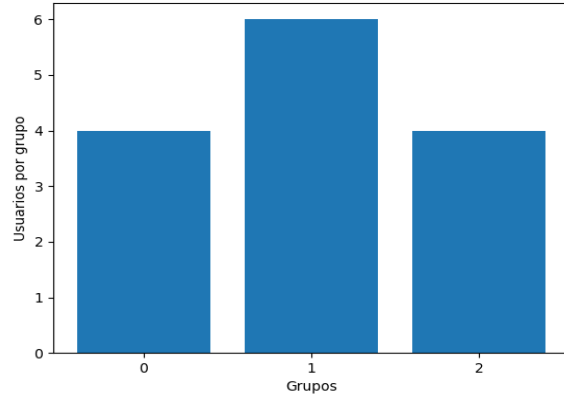
(c) 80 Usuarios



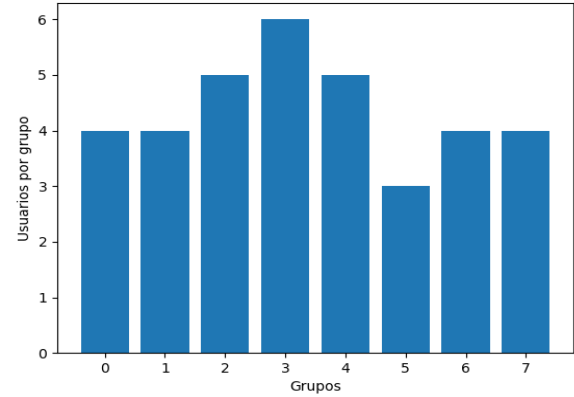
(d) 120 Usuarios

Figura 6.6: Resultado de pruebas con diferente número de usuarios

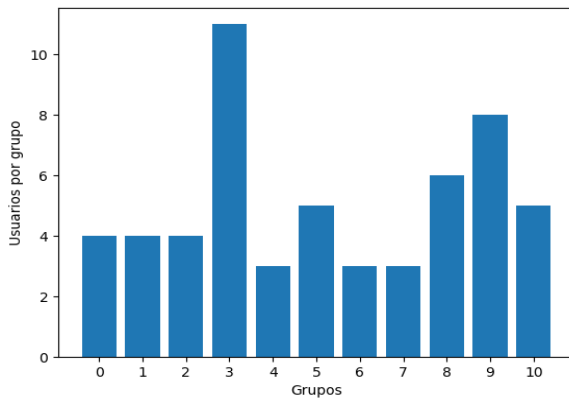
De acuerdo a la figura 6.6, el algoritmo agrupa a los usuarios en una distribución normal, donde los grupos en promedio tienen cantidad equitativa. Un punto muy importante es que sin importar que el conjunto crezca, los grupos se mantienen estables.



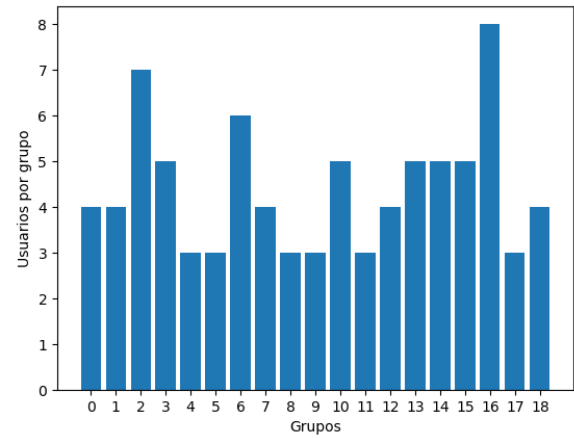
(a) 50 Rutas



(b) 100 Rutas



(c) 150 Rutas



(d) 200 Rutas

Figura 6.7: Resultado de pruebas con diferente número de rutas

La figura 6.7 muestra que los grupos tienen ruido en ciertos escenarios, sin embargo al aumentar la cantidad de rutas los picos tienden a equilibrarse, esto dado que al contar con más rutas hay mayor diversidad de calificaciones que se utilizan para comparar los usuarios y generar los grupos.

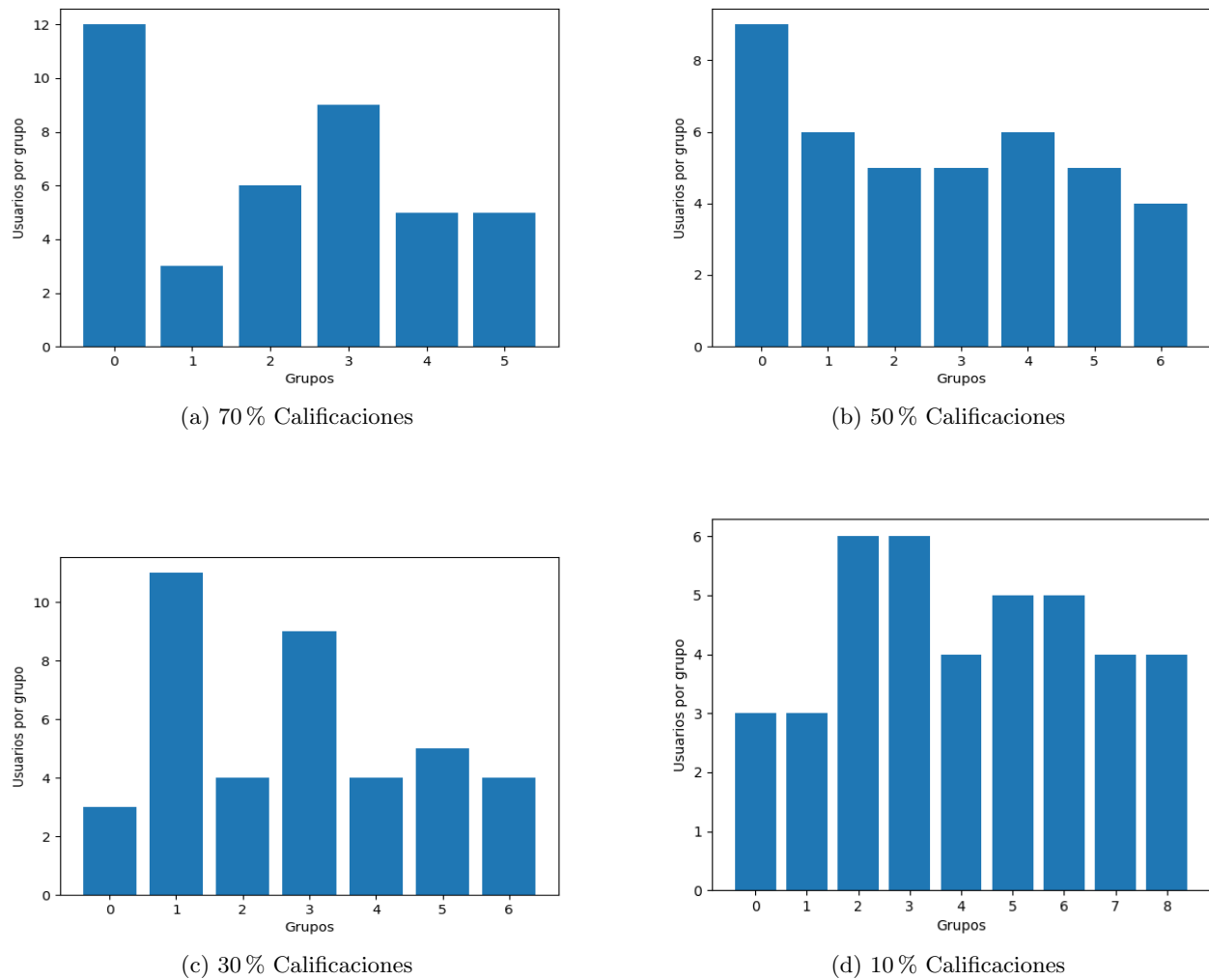


Figura 6.8: Resultado de pruebas con diferente porcentaje de calificaciones

Teniendo en cuenta los resultados de la figura 6.8, se deduce que tener más porcentaje de calificaciones disponibles no necesariamente es un indicativo para obtener mejores grupos, por el contrario, en este caso, los grupos son más equilibrados cuando se cuenta con menor cantidad de calificaciones. Este comportamiento puede deberse a que lo importante es la calidad y diversidad de las calificaciones, es decir contar con valores claves que determinan la mejor combinación de grupos sin importar el resto.

Pruebas con datos en producción

El proceso de creación de eventos durante el periodo que la aplicación estuvo en producción genero los siguientes resultados de los grupos obtenidos:

En la figura se muestra que los grupos formados tienen cantidad diferente de usuarios, esto se debe a que los usuarios al ser nuevos cuentan con el mismo nivel de desempeño y no han interactuado mucho con la aplicación, entonces están al mismo nivel, por otra parte los que están en el grupo más pequeño, se debe a que han calificado una cantidad de rutas considerable que sirve para distinguirlos del resto y son agrupado de acuerdo a sus preferencias.

6.2.3. Sistema de gamificación mediante retos

Dado que se cuenta con un sistema de seguimiento del progreso de un usuario, la forma de probarlo no se realiza con entradas habituales como generalmente se hace con algoritmos tradicionales. Para probar estos sistemas se debe evaluar el seguimiento de los usuarios en tiempo real, algunos de los atributos que se pueden calcular para determinar qué tan interesados y motivados están los usuarios con la aplicación son:

- **Numero de retos alcanzadas:** Evaluar cuantos retos ha logrado terminar un usuario particular.
- **Tiempo en lograr los retos:** El número de retos alcanzados es importante, pero debe contrastarse con el tiempo tardado en conseguirlo, teniendo que cuenta que a menor tiempo, mayor motivación e interacción con la aplicación.
- **Niveles y recompensas logradas:** Este valor ayuda a verificar cual ha sido el trabajo que el usuario le ha dedicado a lograr ciertos objetivos con la finalidad de subir de nivel y lograr recompensas.
- **Tiempo en alcanzar ciertos niveles:** Al igual que con los retos alcanzados, es importante contrastar los niveles obtenidos con el tiempo tardado en lograrlo.
- **Puntos obtenidos:** La cantidad de puntos obtenidos es un buen indicador si el usuario hace las tareas por gusto o solo por alcanzar los niveles y las recompensas.

Teniendo en cuenta que la aplicación ha estado poco tiempo disponible para los usuarios, no se tienen suficientes datos como para generar estadísticas relevantes de su interacción con el sistema de gamificación. El proceso de evaluación se definió como un trabajo futuro, dado que se esperar contar con mayor cantidad de usuarios con el paso del tiempo.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

- Un sistema de autenticación de cuentas como Firebase es una alternativa muy viable cuando se está desarrollando y es necesario un sistema seguro pero que no requiere características especiales o no hace parte del objetivo principal del desarrollo. Este sistema se integra fácilmente con la mayoría de lenguajes de programación y ofrece un modelo seguro con todas las características requeridas en la industria del software.
- En algunas aplicaciones móviles, un sistema de sincronización automático es la clave para hacer que la aplicación este a otro nivel, en la mayoría de situaciones no se cuenta con conexión a internet por lo que es importante permitirle al usuario la interacción en cualquier situación. La sincronización es una tarea que el usuario no la percibe físicamente pero que determina en ciertos casos la satisfacción del mismo.
- El algoritmo Slope One implementado para la recomendación de rutas es eficiente y confiable de acuerdo a los requerimientos que se plantearon y los resultados obtenidos. El tiempo de ejecución no se incrementa demasiado pero en caso de ser necesario se hace muy fácil implementar multi-procesamiento o ejecución por grupos. Sin lugar a dudas es el algoritmo más estable en cuanto a rendimiento y velocidad.
- El modelo de creación de eventos funciona eficientemente teniendo en cuenta que procesa la información de manera correcta, generando grupos de usuarios estables sin importar el crecimiento de los datos, además no necesita de una gran cantidad de entradas para funcionar bien.
- El sistema de gamificación mediante retos es una herramienta de vital importancia en la aplicación, dado que es la base para motivar a los ciclistas a utilizarla y desarrollar sus capacidades a través de un modelo de juego-recompensa.
- Personalmente el desarrollo de este trabajo de grado ha sido el mayor logro como estudiante de Ingeniería de sistemas, porque me permitió conocer en un ámbito más profesional cómo funcionan las tecnologías y herramientas para el desarrollo de software. Además de ayudarme en el crecimiento profesional al adquirir nuevas capacidades y conocer nuevas plataformas acordes con el mundo laboral al que me enfrentare en algún momento.

7.2. Trabajos futuros

De acuerdo al alcance de este trabajo de grado se definieron una serie de objetivos a cumplir, sin embargo hay ciertas funcionalidades y elementos que podrían desarrollarse en un futuro para mejorar el desarrollo e implementación del sistema, a continuación se definen algunos de estos elementos:

- **Extensión de actividades deportivas:** Para el futuro se plantea que la aplicación pueda trabajar con más actividades, es decir, realizar un proceso para que las personas tengan la posibilidad de realizar entrenamientos de caminata, correr, natación, etc. De esta manera extender las alternativas que un usuario pueda realizar.
- **Reconocimiento de actividad:** Los dispositivos modernos cuentan con una serie de sensores que permiten mejor interacción con el usuario, una de las posibilidades es hacer uso de los sensores de movimiento del dispositivo para detectar la actividad que el usuario está realizando y hacer poder recomendarle la mejor alternativa o controlar los entrenamientos automáticamente. La idea básica es permitir nuevos procesos mediante las tecnologías del internet de las cosas.
- **Distribución a través de tiendas de aplicaciones:** La aplicación podría en un futuro estar disponible a través de tiendas de aplicaciones autorizadas para permitir la distribución de manera más fácil y a mayor cantidad de personas.
- **Sincronización directa con redes sociales:** Permitir que los usuarios sincronicen sus redes sociales y darle la posibilidad de invitar amigos, compartir sus logros y objetivos cumplidos.
- **Ampliar radio de acción:** Permitir que los usuarios tengan disponible características como el mapa offline y la región disponible para crear rutas en un radio de acción mayor de acuerdo a su ubicación geográfica, permitiendo aumentar el alcance a nivel departamental y nacional.
- **Pruebas con mayor cantidad de datos:** Realizar un proceso de pruebas y análisis del sistema cuando la aplicación haya estado un tiempo en producción y se tenga mayor cantidad de información.

Capítulo 8

Bibliografía

- [1] M. Rendell, *Reyes de las montañas*. S-Libros, 1994.
- [2] “Fundacion idea para la paz.” www.ideaspaz.org. Consultado 10-Octubre-2016.
- [3] “Agencia presidencial de cooperación internacional de Colombia.” www.apccolombia.gov.co. Consultado 15-Octubre-2016.
- [4] P. Massa¹ and P. Avesani¹, “Trust-aware Collaborative Filterin for recommender systems,”
- [5] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative filtering recommender systems,” *Foundations and Trends® in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011.
- [6] “Amazon online shop.” www.amazon.com.
- [7] R. E. Wright, J. Silva, and I. Kaynar-Kabul, “Shot Recommender System for NBA Coaches,”
- [8] S. Rendle, “Factorization machines,” in *2010 IEEE International Conference on Data Mining*, pp. 995–1000, Dec 2010.
- [9] L. T. Ponnamm, S. D. Punyasamudram, S. N. Nallagulla, and S. Yellamati, “Movie recommender system using item based collaborative filtering technique,” in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–5, Feb 2016.
- [10] M. Lerato, O. A. Esan, A. D. Ebunoluwa, S. Ngwira, and T. Zuva, “A survey of recommender system feedback techniques, comparison and evaluation metrics,” in *Computing, Communication and Security (ICCCS), 2015 International Conference on*, pp. 1–4, Dec 2015.
- [11] “Wikiloc.” es.wikiloc.com.
- [12] “Strava.” www.strava.com.
- [13] “Timpik.” www.timpik.com.
- [14] I. C. del deporte COLDEPORTES, *Plan Decenal del Deporte, la Recreación, la Educación Física y la Actividad Física, para el Desarrollo Humano, la Convivencia y la Paz 2009-2019*. ABC Ltda, 2009.
- [15] “Rio 2016, una experiencia de oro para Colombia.” www.coldeportes.gov.co/index.php?idcategoria=84436.
- [16] “Apps.co.” apps.co.

- [17] “Facebook.” www.facebook.com.
- [18] “Twitter.” www.twitter.com.
- [19] “Paneles de control, Estadísticas sistema operativo Android.” <https://developer.android.com/about/dashboards/index.html?hl=es>. Consultado 9-Diciembre-2016.
- [20] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” vol. 17, pp. 734–749, 07 2005.
- [21] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in *Recommender Systems Handbook* (R. Francesco, R. Lior, B. Shapira, and P. B. Kantor, eds.), pp. 257–297, Springer US, 2010.
- [22] “FilmAffinity.” www.filmaffinity.com.
- [23] “Youtube.” www.youtube.com.
- [24] G. Takács, I. Pilászy, B. Németh, and D. Tikk, “Matrix factorization and neighbor based algorithms for the netflix prize problem,” in *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08*, (New York, NY, USA), pp. 267–274, ACM, 2008.
- [25] T. Segaran, *Programming Collective Intelligence*. O’Reilly, first ed., 2007.
- [26] B. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster analysis*. Wiley, 5th ed., 2011.
- [27] L. Kaufman and P. J. Rousseeuw, *Agglomerative Nesting (Program AGNES)*, pp. 199–252. John Wiley Sons, Inc., 2008.
- [28] C. TIC, “Gamificación educativa canalTIC.com.” canaltic.com.
- [29] R. A. Bartle, “Entry Point.” mud.co.uk.
- [30] H. Sandí and E. Rojas, “La gamificación como participante en el desarrollo del b-learning: Su percepción en la universidad nacional, sede regional brunca,”
- [31] Sebastopol, “Introduction. gamification by design: Implementing game mechanics in web and mobile apps,”
- [32] D. Lemire and A. Maclachlan, “Slope one predictors for online rating-based collaborative filtering,” *CoRR*, vol. abs/cs/0702144, 2007.
- [33] T. H. T. Gladkikh, “Recommender systems comparison: the best performing algorithm.” www.softserveinc.com.
- [34] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, pp. 30–37, Aug. 2009.
- [35] C. Bauckhage, “Numpy / scipy recipes for data science: k-medoids clustering,” 02 2015.
- [36] “Spring Io.” <https://spring.io>.
- [37] “Codigo fuente servicio web REST.” <https://github.com/DanielMejia/BikeMeServer.git>.

- [38] “*Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016.*” <http://www.gartner.com>.
- [39] “*Conociendo Android Studio.*” www.developer.android.com.
- [40] “*Código fuente de la aplicación móvil.*” <https://github.com/DanielMejia/BikeMeApp.git>.
- [41] “*Retrofit Open Source.*” <http://square.github.io/retrofit/>.

Anexos

Diagrama de despliegue

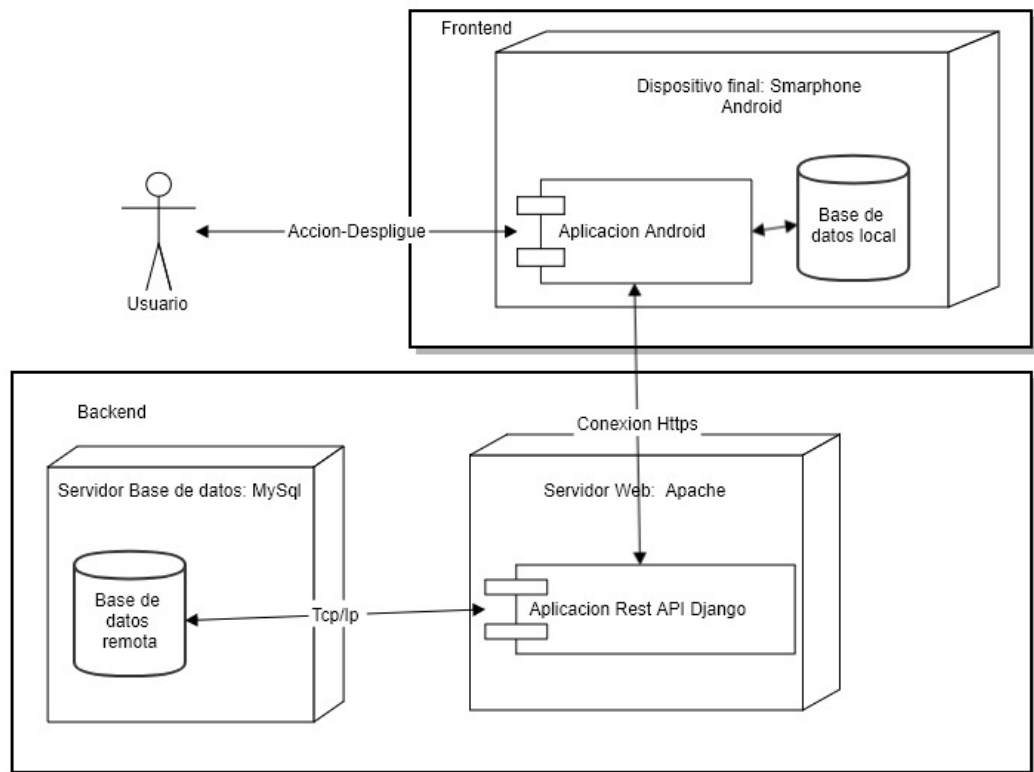


Figura 8.1: Diagrama de despliegue del proyecto

Modelo entidad relación

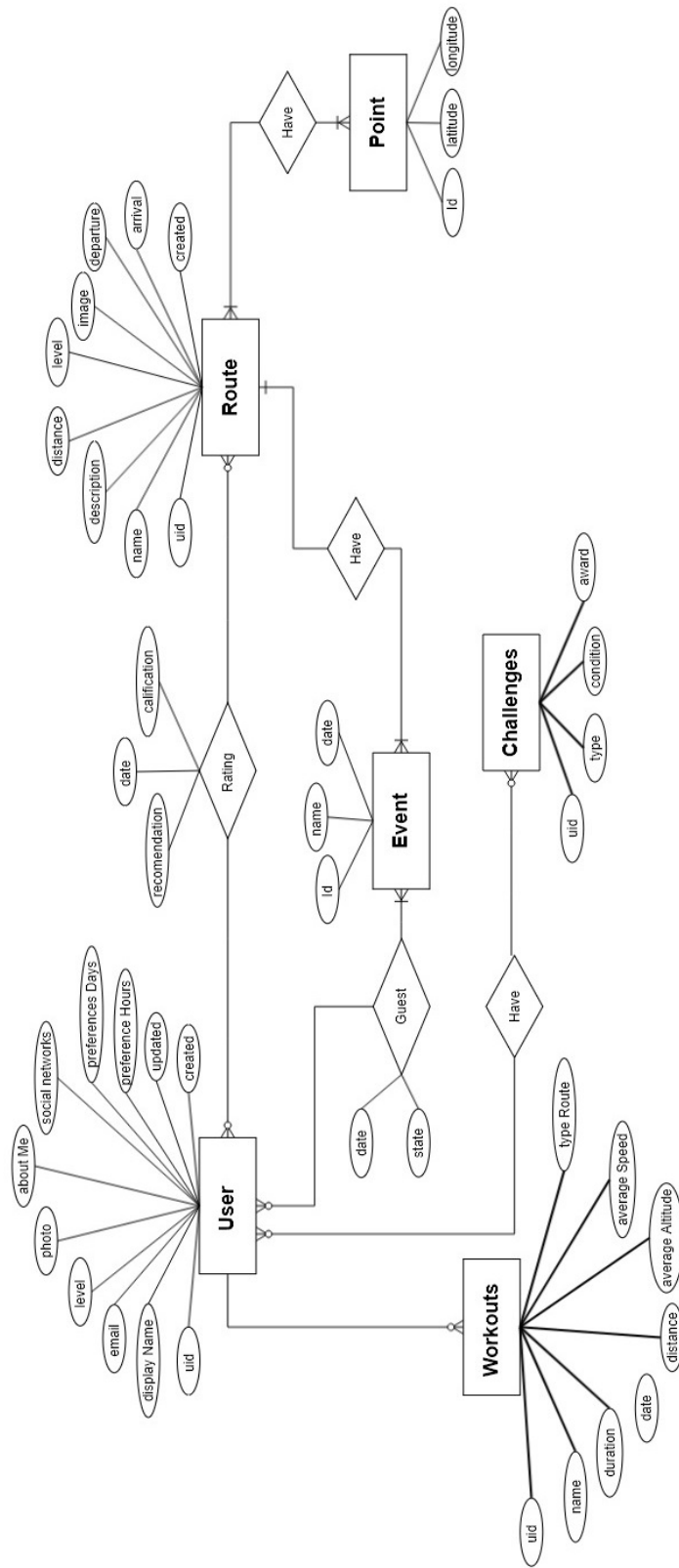


Figura 8.2: Modelo entidad relación del proyecto