

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



“Desarrollo de un sistema de recomendación de libros, haciendo uso de *frameworks* para aplicaciones Web, permitiendo que los usuarios descubran libros de interés”

Trabajo de graduación en modalidad de trabajo profesional presentado por

Marvin Josué Quiñónez Xitumul

para optar al grado académico de Licenciado en Tecnología de Sistemas  
Informáticos

Guatemala

2021



UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



“Desarrollo de un sistema de recomendación de libros, haciendo uso de *frameworks* para aplicaciones Web, permitiendo que los usuarios descubran libros de interés”

Trabajo de graduación en modalidad de trabajo profesional presentado por


Marvin Josué Quiñónez Xitumul

para optar al grado académico de Licenciado en Tecnología de Sistemas  
Informáticos

Guatemala

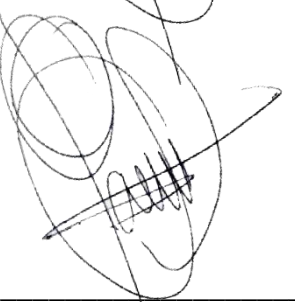
2021

Vo. Bo.:

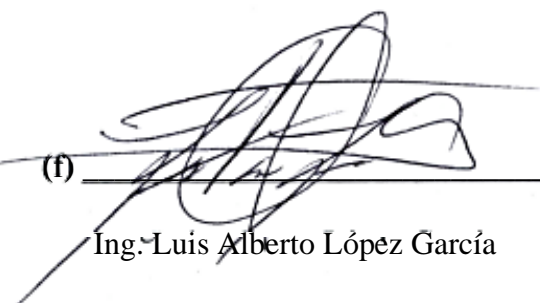
(f) 

Lic. Paulo Saúl García Ramírez

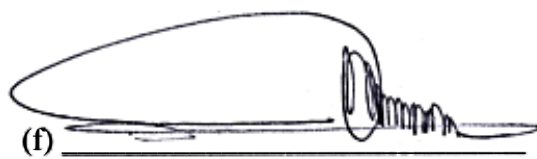
Tribunal Examinador:

(f) 

Lic. Paulo Saúl García Ramírez

(f) 

Ing. Luis Alberto López García

(f) 

MBA. Eddy Omar Arreaga López

Fecha de aprobación: Guatemala, 31 de Mayo de 2021.

## Prefacio

La elaboración de este trabajo de graduación nace del interés personal de apoyar a la comunidad guatemalteca con herramientas digitales que ayuden a fomentar la creación de tecnologías o de empresas tecnológicas en Guatemala.

Desde muy pequeño mostré un interés en la tecnología digital, el cual con el tiempo fue creciendo hasta tal punto de querer estudiar una licenciatura relacionada a este tema. Los conocimientos aprendidos fueron aplicados en esta investigación con la intención de crear una herramienta gratuita que ayude a generar recomendaciones inteligentes de libros en sistemas web, así mismo, mi principal reto fue verificar que las recomendaciones generadas por el sistema fueran funcionales.

Agradezco la ayuda brindada por mi asesor, el Lic. Paulo Saúl García Ramírez, así como la ayuda financiera y emocional brindada por mis padres, hermanas, amigos, y Sabine Voelcker.



Marvin Josué Quiñónez Xitumul

Autor

# Índice

	Página
Prefacio.....	v
Lista de cuadros.....	ix
Lista de figuras.....	x
Lista de ecuaciones.....	xi
Resumen.....	xii
I. Introducción.....	1
II. Objetivos .....	2
A. General .....	2
B. Específicos .....	2
III. Justificación.....	3
IV. Marco teórico.....	5
A. Sistemas de recomendaciones (S.R.) .....	5
1. Técnicas de retroalimentación de información.....	5
2. Enfoques de los sistemas de recomendaciones.....	6
3. Sistemas de recomendaciones en aplicaciones famosas.....	7
4. Precios de implementación.....	11
5. Validación de los sistemas de recomendaciones .....	11
B. Procesado de lenguaje natural.....	12
1. Modelo de Bolsa de Palabras.....	13
2. Term Frequency-Inverse Document Frequency (TF-IDF).....	14
C. Algoritmos de similitud.....	15
1. Similitudes basadas en cosenos .....	15
2. Similitudes basadas en correlación .....	15
3. Distancia de Manhattan .....	16
4. Distancia Canberra .....	16
5. Distancia Euclidiana.....	16
D. K-NN (K nearest network o K vecinos cercanos) .....	17
E. Plataformas de comercialización de libros en línea .....	18
1. Elejandria.com .....	19
2. Casa del libro .....	20

F. Sistema informático .....	20
1. Estructuras .....	21
2. Arquitectura .....	21
3. Ingeniería de software .....	22
V. Identificación del problema.....	25
A. El problema.....	25
B. Mercado objetivo.....	25
VI. Solución propuesta.....	26
A. Arquitectura propuesta .....	26
1. Componentes. ....	27
B. Base de datos propuesta.....	28
C. Algoritmo.....	29
D. Datos a analizar.....	29
1. Portada a información .....	30
2. Resumen .....	34
3. Procesamiento de palabras.....	34
E. Uso del algoritmo .....	34
VII. Desarrollo de la propuesta.....	38
A. Plataforma web .....	38
B. Modelo utilizado .....	38
C. Requerimientos .....	39
1. Requerimientos funcionales .....	40
2. Requerimientos no funcionales.....	40
D. Análisis.....	40
1. Módulo de usuarios.....	41
2. Módulo de administración.....	41
3. Módulo de entretenimiento.....	41
4. Módulo de recolección de datos .....	41
5. Módulo de recomendaciones .....	41
E. Diseño .....	42
1. Alcance.....	42
2. Arquitectura .....	42
3. Proceso de retroalimentación.....	43

4. Motor de recomendaciones.....	44
5. Diagrama de casos de uso.....	48
6. Caso de uso detallado.....	51
7. Base de datos. ....	55
F. Programación.....	56
1. Servidor de recomendaciones .....	56
2. Similitud entre libros.....	56
3. API Rest.....	57
4. Aplicación.....	57
G. Verificación .....	58
1. Pruebas estáticas .....	58
2. Pruebas dinámicas.....	58
H. Aspecto legal .....	62
I. Código fuente .....	63
VIII. Evaluación de las recomendaciones.....	64
A. Evaluación matemática.....	64
B. Prueba de campo (A/B Testing).....	64
IX. Resultados y análisis .....	67
A. Pruebas estáticas .....	67
B. Prueba de casos .....	68
C. Evaluación de las recomendaciones .....	71
1. Evaluación matemática.....	71
2. Prueba de campo .....	73
X. Conclusiones.....	76
XI. Recomendaciones .....	78
XII. Bibliografía.....	79
XIII. Glosario .....	81



## Lista de cuadros

Cuadro 1 Valoraciones de películas .....	6
Cuadro 2 Ejemplo teórico de transformación .....	13
Cuadro 3 Ejemplo práctico de transformación.....	13
Cuadro 4 Descripción del tipo de portada.....	31
Cuadro 5 Descripción del diseño de la portada.....	33
Cuadro 6 Ejemplos de datos transformados.....	35
Cuadro 7 Ejemplo de datos del libro de prueba .....	35
Cuadro 8 Ejemplo con datos de un libro.....	45
Cuadro 9 Caso de uso detallado de autenticación con credenciales.....	51
Cuadro 10 Caso de uso detalla de administración de perfil de usuario .....	52
Cuadro 11 Caso de uso detallado de Dashboard .....	53
Cuadro 12 Caso de uso detallado de páginas principales de interacción .....	53
Cuadro 13 Caso de uso detallado de recolección de datos .....	53
Cuadro 14 Caso de uso detallado de recomendaciones .....	54
Cuadro 15 Rutas de transformación de la API REST .....	57
Cuadro 16 Rutas de generación de recomendaciones de la API REST.....	57
Cuadro 17 Prueba de casos utilizada .....	61
Cuadro 18 Resultado de pruebas estáticas .....	67
Cuadro 19 Resultados de la prueba de casos .....	68
Cuadro 20 Evaluación matemática de las recomendaciones basadas en la portada.....	72
Cuadro 21 Evaluación matemática de las recomendaciones basadas en el resumen .....	73

## Lista de figuras

Figura 1 Captura de pantalla de Netflix.....	8
Figura 2 Captura de pantalla de Amazon.com.....	9
Figura 3 Captura de pantalla de Spotify.....	10
Figura 4 Similitud de cosenos.....	15
Figura 5 Distancia entre dos puntos en un espacio 2D.....	17
Figura 6 Ejemplo de clasificación.....	17
Figura 7 Ejemplo de clasificación más expresiva.....	18
Figura 8 Pseudocódigo del algoritmo KNN.....	18
Figura 9 Captura de pantalla de Elejandría.com.....	19
Figura 10 Captura de pantalla de sección de Elejandría.com.....	19
Figura 11 Captura de pantalla de sección de Elejandria.com.....	20
Figura 12 Captura de pantalla de casadellibro.com.....	20
Figura 13 Cliente-servidor.....	21
Figura 14 Modelo vista controlador.....	23
Figura 15 Arquitectura propuesta.....	26
Figura 16 Base de datos propuesta.....	28
Figura 17 Algoritmo propuesto.....	29
Figura 18 Ejemplo de formulario.....	30
Figura 19 Palabras indeseadas.....	34
Figura 20 Ejemplo de distancias.....	36
Figura 21 Ejemplo del funcionamiento del algoritmo.....	37
Figura 22 Modelo en cascada.....	39
Figura 23 Estructura utilizada.....	43
Figura 24 Diagrama de tipos de recomendaciones.....	44
Figura 25 Datos vectorizados 1.....	45
Figura 26 Datos vectorizados 2.....	46
Figura 27 Diagrama de flujo de procesos de la transformación de datos.....	46
Figura 28 Diagrama de flujo de procesos de solicitud de recomendaciones.....	47
Figura 29 Diagrama de la API-REST.....	48
Figura 30 Diagrama de casos de uso de autenticación.....	49
Figura 31 Diagrama de caso de uso de perfil de usuario.....	49
Figura 32 Diagrama de caso de uso de módulo de entretenimiento.....	50
Figura 33 Diagrama de caso de uso de dashboard.....	50
Figura 34 Estructura de base de datos utilizada.....	55
Figura 35 Estado de una precisión.....	64
Figura 36 Recursos utilizados.....	65
Figura 37 Captura de pantalla de Relibro.....	65
Figura 38 Captura de pantalla de sección recomendados de Relibro.....	66
Figura 39 Prueba de casos, alerta por error.....	69
Figura 40 Alerta por detección de campos obligatorios.....	69
Figura 41 Confirmación de acción mediante contraseña.....	70
Figura 42 Aparición de opciones dependiendo el tipo de usuario.....	70
Figura 43 Datos obtenidos del A/B testing.....	73

Figura 44 Datos de la versión A del A/B Testing .....	74
Figura 45 Datos de la versión B del A/B Testing.....	74
Figura 46 Resultados del A/B Testing.....	75

## Lista de ecuaciones

Ecuación 1 Precisión.....	12
Ecuación 2 Exhaustividad.....	12
Ecuación 3 Term Frequency.....	14
Ecuación 4 Inverse Document Frequency .....	14
Ecuación 5 TF-IDF.....	14
Ecuación 6 Correlación de Pearson .....	15
Ecuación 7 Distancia de Manhattan .....	16
Ecuación 8 Distancia Canberra .....	16
Ecuación 9 Distancia Euclidiana .....	16
Ecuación 10 Ejemplo de distancia euclidiana.....	56
Ecuación 11 Precisión utilizada .....	64

## **Resumen**

Los sitios dedicados a la distribución de libros comúnmente utilizan sistemas de recomendaciones basados en las opiniones de los usuarios o en las calificaciones de sus productos. Estos sistemas basados en el Filtrado Colaborativo tratan al libro como un simple producto y no como un objeto que brinda contenido de interés. Para tener otra alternativa muy distinta a los sistemas de recomendaciones utilizados en estos sitios, se desarrolló un sistema de recomendaciones de libros el cual se basa principalmente en el diseño de la portada y el resumen.

Este sistema utiliza el enfoque Filtrado Basado en Contenido, el cual permite el análisis del contenido de un elemento. En este caso se ha tomado como contenido al diseño de la portada y al resumen. La idea principal es recomendar libros que tengan portadas similares o libros con un tema similar a otro. Para lograr esto se hicieron uso de técnicas de Procesado de Lenguaje Natural en conjunto con un algoritmo de distancia.

Se demuestra por medio de las pruebas realizadas que el sistema desarrollado es capaz de generar recomendaciones de libros en base a datos de posible interés de los usuarios.

# I. Introducción

Los sistemas de recomendaciones son herramientas importantes que ayudan a las aplicaciones web a mostrar información relevante y personalizada a los usuarios. Estos tienen distintas características que ayudan a entender los gustos de los usuarios para mostrarle información similar o de interés. Aplicaciones como Netflix o Spotify las utilizan para mostrar contenido relevante.

El propósito es ayudar a los usuarios a elegir lo que necesitan o les interesa. Cuando hay demasiadas opciones, ayuda y acelera el proceso de toma de decisiones.

Estos sistemas no solo permiten que los usuarios conozcan elementos de interés, sino que también, ayudan a que sigan consumiendo más contenido, generando un aumento de ganancias para las empresas que utilizan estas herramientas en sus aplicaciones.

El área de la literatura no se queda atrás, las páginas que distribuyen contenido literario, como las tiendas de libro en línea, utilizan sistemas que ayudan a recomendar libros de interés para sus usuarios. Estos sistemas generalmente están basados en el producto y calificaciones de este, permitiendo relacionar a usuarios con gustos similares.

En este trabajo se presenta una propuesta diferente, la cual utiliza el diseño de las portadas y resúmenes de los libros, en conjunto con el historial del usuario, para generar recomendaciones de libros personalizadas.

Se explica la estructura de datos que son compatibles con el sistema desarrollado y se definen los algoritmos utilizados para generar las recomendaciones. Se han detallado algunos ejemplos para que el lector tenga una idea más a fondo de cómo funciona el sistema.

Al terminar la lectura, el lector tendrá conocimiento sobre el funcionamiento del sistema propuesto y los resultados obtenidos en el desarrollo de esta. Además, conocerán qué son los sistemas de recomendaciones, cómo funcionan y para qué sirven.

## **II. Objetivos**

### **A. General**

Desarrollar un sistema de recomendación de libros, haciendo uso de frameworks para aplicaciones web, permitiendo que los usuarios descubran libros de interés.

### **B. Específicos**

1. Determinar los requerimientos, analizando el problema, para definir los parámetros a seguir en el desarrollo del sistema.
2. Diseñar un sistema de recomendación de libros, basándose principalmente en las portadas y resúmenes de los libros que el usuario seleccione en el sistema, para generar y mostrar sugerencias de interés.
3. Implementar técnicas de la ingeniería de software, utilizando un patrón de desarrollo, que permitirá la construcción de un software profesional y eficaz.

### III. Justificación

Los sistemas de recomendaciones son sistemas que utilizan algoritmos que analizan los datos de los usuarios, productos, entre otros, para la generación de recomendaciones, muchas de ellas personalizadas.

«Surgen de la necesidad de las personas de determinar qué productos son los que más le pueden interesar para una determinada actividad de entre todas las posibles opciones que hoy en día tenemos a nuestro alcance.» Caro Martínez (2017).

Al entrar a Amazon.com y navegar por su página web se pueden observar secciones de recomendaciones de productos que, en ocasiones, logran cautivar al usuario, influyendo en su toma de decisiones.

Después de ver un video en YouTube, el usuario tiende a visualizar otros, ya que en la aplicación se muestra una lista de recomendaciones. Estas recomendaciones muestran videos de interés para los usuarios, con la intención de que estos encuentren más contenido para consumir, manteniéndolos por más tiempo en la aplicación.

Al utilizar Netflix, se pueden observar una gran variedad de películas que tienen o no relación una con la otra, pero que pueden ser de interés del usuario.

El encontrar en estas plataformas contenido de interés sin buscarlo no es casualidad, en la mayor parte de las veces los sistemas de recomendaciones ayudan a los usuarios a encontrar el contenido que ni siquiera sabían que andaban buscando.

En el área de la literatura también existen los sistemas de recomendaciones, estos están implementados en algunas páginas web dedicadas a la comercialización de libros físicos y digitales. Sin embargo, estos sistemas están basados principalmente en la valoración que tienen los libros o en la popularidad que estos tengan. Estos tipos de sistemas están más enfocados en los productos. Pero un libro no es solo un producto.

Boix García afirma, según en una publicación de Manrique (2017):

«Un libro son muchas cosas: una historia, diferentes textos pequeños que se pueden leer en diferentes formatos. Pueden ser formatos con realidad aumentada, bancos de datos, metadatos. Vamos en esa dirección con los libros. Todo lo que es contenido que se pueda leer en formato digital o impreso para mí es un libro.».

Los libros tratan temas muy diversos. Hay libros que son de ficción y otros que son científicos. Recomendarle a una persona libros que no son de su interés, pero que son los más vendidos, no siempre es una buena idea. Entonces, aquí entran en juego los sistemas de recomendaciones basados en contenidos.

Los sistemas de recomendaciones de contenido son sistemas que, valga la redundancia, están basados en el contenido del producto. Este tipo de sistema se adapta perfectamente a los libros ya que se puede realizar un análisis del contenido de este para mostrar recomendaciones.

Para Adomavicius y Tuzhilin (2005), los sistemas de recomendaciones basados en contenido intentan comprender puntos en común entre los objetos que el usuario ha calificado altamente en el pasado, para generar las recomendaciones.

La implementación de estos sistemas no es sencilla ni complicada. Hay que tener conocimientos en el uso de algoritmos de transformación de datos o tener la idea, y conocimientos de Aprendizaje Automático para la implementación de algoritmos eficaces. Además, los costos de implementación rondan los \$15,000.00, un precio al que muchas personas interesadas no pueden acceder.

Estos sistemas se pueden utilizar para fomentar la lectura, mostrándole a los usuarios libros de interés en base a datos obtenidos de su patrón de lectura. Soy502 (2016) afirma que, según el Consejo de Lectura de Guatemala, de cada cien personas solo una lee por placer. Esto se debe a varios factores en lo que se incluye la baja fomentación de la lectura en Guatemala. Pensando en ayudar a fomentar la lectura en el país se ha decidido desarrollar un sistema de recomendaciones de código libre que ayude a recomendar libros de posible interés. Este sistema puede ser utilizado, modificado o adaptado por personas interesadas en tener recomendaciones inteligentes en aplicaciones web, en tiendas en línea, en sitios de promoción, en sistemas web de bibliotecas de universidades, etc.

La ventaja que se presenta en la propuesta, y que la hace diferente a otros sistemas, es que se basa principalmente en el diseño de la portada del libro y en su resumen.

Se ha decidido usar el resumen del libro porque da una visión general de lo que va a tratar, y se ha utilizado el diseño de la portada porque es la imagen comercial que atrae a la persona interesada.

Según Sánchez *et al.* (2008:62), un libro no es solo un contenido. Afirman que un libro sin una buena portada no alcanzaría los mismos objetivos a uno que si la tenga.

Las personas interesadas podrán utilizar este sistema de cualquier forma y totalmente gratis ya que es de código libre. Además, al implementarlo tendrán un sistema diferente de recomendaciones de libros a los que existen actualmente.



## **IV. Marco teórico**

### **A. Sistemas de recomendaciones (S.R.)**

Con la finalidad de aumentar las ventas, de proporcionar contenido personalizado, de ofrecer productos de interés, entre otros, las empresas han empleado sistemas de recomendaciones que filtran una gran cantidad de datos para mostrar únicamente datos que son de interés para cada usuario.

Un sistema de recomendaciones es una herramienta utilizada para generar y proporcionar sugerencias de artículos que un usuario en específico desea buscar. Estos sistemas utilizan algoritmos que filtran la información en base a los datos con los que se quieren o requieran trabajar. Son utilizados para personalizar lo que se le ofrecerá a cada usuario. Para Herrera-Viedma *et al.* (2004), los sistemas de recomendaciones son herramientas que generan recomendaciones sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por los usuarios.

Para las empresas, un sistema de recomendación es una herramienta eficaz porque genera contenido personalizado a los usuarios, analizando datos que este haya almacenado en los sistemas y ahorrando tiempo, ya que al ser un proceso automático la empresa no tiene que delegar a alguien para que analice a cada usuario. Además, es una herramienta de marketing muy potente.

La información es uno de los elementos más importantes de un sistema de recomendaciones ya que sin datos los algoritmos no pueden funcionar. Los datos que los sistemas de recomendaciones utilizan se pueden obtener de forma explícita o implícita.

Para la creación de estos se utilizan técnicas de Aprendizaje Automático (Machine Learning en inglés) o de Aprendizaje Profundo (Deep Learning en inglés) sobre un conjunto de datos que tengan relación con los productos que el usuario busque. Los datos tienen que estar estructurados de tal forma que los procesos de los algoritmos se ejecuten más rápido.

Al implementar los S.R. se pueden aplicar distintas perspectivas, dependiendo de los objetivos que se quieran alcanzar. Estos se pueden basar en las preferencias individuales de los usuarios, en personas que tengan preferencias similares o en productos similares.

#### **1. Técnicas de retroalimentación de información.**

Obtener la información necesaria que utiliza los sistemas de recomendaciones no es tarea sencilla. Hay que planificar la estructura de la base de datos, determinar qué datos son lo que se van a tratar, saber el tipo de enfoque que tendrá el sistema y lo más importante, saber cómo se obtendrán los datos.

Hay distintas formas de retroalimentación, una de ellas es la retroalimentación explícita. La retroalimentación explícita es la forma de recolectar datos directamente del usuario. En este tipo de técnica, el usuario tiene que rellenar encuestas, realizar valoraciones, responder a preguntas o ingresar comentarios.

Es una forma muy utilizada en las páginas web, sin embargo, afecta la experiencia del usuario. Típicamente, a un usuario no le gusta estar realizando acciones que el crea innecesarias para llegar a su objetivo, el usuario entra a una aplicación a obtener lo que quiere, no entra a ingresar información. Para evitar este problema existe la retroalimentación implícita.

La retroalimentación implícita es la forma de retroalimentación indirecta. Con esta retroalimentación se puede obtener datos sin que el usuario ingrese la información directamente. Por ejemplo, en el momento en que un usuario entra al perfil de un libro, este se guarda en el historial del usuario para ser utilizado posteriormente en el análisis.

## 2. Enfoques de los sistemas de recomendaciones.

Esta sección describe los principales métodos de recomendación ampliamente utilizados en los S.R. clásicos.

Los sistemas de recomendaciones pueden ser clasificados en diferentes enfoques, de acuerdo al tipo de información que se utilizará. Para el diseño de estos sistemas hay que plantear los objetivos y determinar el enfoque que se tendrá.

### a. Filtración colaborativa.

Este tiene un amplio uso. Para Balabanovic y Shoham (1997), los sistemas que usan el filtrado colaborativo no recomiendan productos por ser similares a productos que a los usuarios les gustaban en el pasado, sino que, recomiendan productos que les gustan a otros usuarios similares.

Con este enfoque se logran generar recomendaciones utilizando solo información sobre los perfiles de calificación para diferentes usuarios o elementos.

Supóngase que se tiene la siguiente información de valoraciones de películas:

<b>Usuario 1</b>	<b>Spiderman 3</b>	<b>Gladiator</b>	<b>El hobbit</b>	<b>El señor de los anillos</b>
Luis	7	2	8	7

<b>Usuario 2</b>	<b>Toy Story</b>	<b>Avengers</b>	<b>El diario de Greg</b>	<b>Los cazas fantasmas</b>
Javier	10	5	3	9

<b>Usuario 3</b>	<b>El hobbit</b>
Fernando	7

*Cuadro 1 Valoraciones de películas*

Si el usuario 3 quisiera recomendaciones de películas, el enfoque de filtrado colaborativo permitiría saber cuáles fueron las que visualizó este, para luego mostrarle recomendaciones en base al historial de otros usuarios que comparten películas visualizadas en común.

En este caso el usuario 1 ha visto *El Hobbit*, la misma película que vio el usuario 3, por tal razón se le mostrarían como recomendaciones la película de *El Señor de los Anillos* y *Spiderman 3*, ya que estas fueron las películas que tienen una buena calificación por parte del usuario 1.

#### **b. Filtrado basado en contenido.**

También se le llama sistema de recomendación no colaborativo. Es otro enfoque que se basa en la descripción del elemento y un perfil de las preferencias del usuario. Estos métodos se adaptan mejor a situaciones en las que se conocen los datos sobre un elemento. Los sistemas que utilizan el filtrado basado en contenido utilizan palabras claves para describir los elementos. Como los algoritmos no procesan palabras, estas hay que transformarlas a valores. Para lograr las recomendaciones, el sistema se centra en el historial de la interacción que ha tenido el usuario con el sistema de recomendación.

El sistema de recomendación basado en contenido intenta adivinar las características o el comportamiento del usuario dadas las características del producto.

#### **c. Sistemas de recomendación de criterios múltiples.**

Este enfoque incorpora información de preferencias sobre múltiples factores de un determinado producto.

#### **d. Sistemas de recomendación conscientes de los riesgos.**

Estos sistemas son los que toman en cuenta el riesgo de molestar a los usuarios con acciones o solicitudes en circunstancias donde el usuario no las quiera recibir.

Hay circunstancias en las que el encargado de implementar un sistema de recomendaciones decide utilizar varios enfoques por lo que a esto se le conoce como sistema de recomendación híbrido.

### **3. Sistemas de recomendaciones en aplicaciones famosas.**

Existen varias aplicaciones que utilizan los sistemas de recomendaciones para generar contenido personalizado y de interés para el usuario. Obviamente estos sistemas no tienen solo el objetivo de mostrarle sugerencias al usuario, sino que, también se aplica como estrategia de marketing para mantener a los usuarios en las aplicaciones.

## a. Recomendaciones de Netflix.

Netflix es una plataforma de servicio de streaming que brinda la posibilidad de ver películas y series en formatos que se adaptan a los equipos de los usuarios. Netflix (empresa), sabe que tener un sistema de recomendaciones es una buena forma de mantener a los usuarios consumiendo más contenido y por más tiempo en su aplicación.

Según Muñoz de Frutos (2018), en su artículo para “Business Insider” afirmó que el sistema de recomendación de Netflix es uno de los secretos mejor guardados de la empresa, e internamente está valorado en mil millones de dólares.

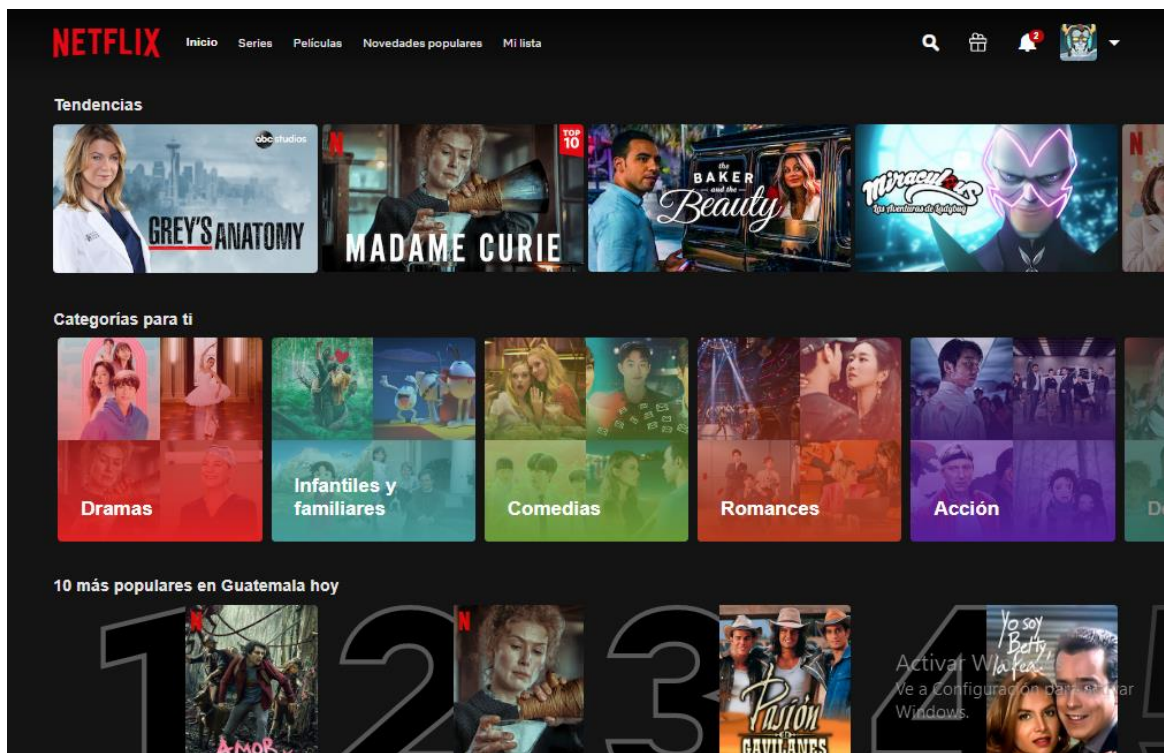


Figura 1 Captura de pantalla de Netflix

No se sabe a ciencia cierta cómo funciona el sistema de recomendaciones de Netflix, pero si se logran identificar algunos detalles. Al momento de ingresar a la aplicación se puede observar que las listas de las películas no siempre son las mismas, éstas van cambiando en base al contenido que se visualiza.

Se ha notado que las portadas de las películas no siempre son las mismas. Estas también cambian en base al contenido visto y otras características. El objetivo es identificar qué portada tiene éxito entre los usuarios.

Para Netflix es mejor que un usuario haga clic en un contenido a tener que rellenar un cuestionario con su sexo o edad, ya que les revela más información. De hecho, Todd Yellin, Vicepresidente de Producto de Netflix, explicó que los datos “sexo” y “edad” no los incluyen en sus algoritmos de recomendación porque no crean que sean útiles.

## b. Recomendaciones de Amazon.

Amazon es una empresa fundada por Jeff Bezos en 1994. En su tienda digital, Amazon.com, se pueden encontrar una gran variedad de productos. En una de las secciones se pueden ver recomendaciones de productos que, sin duda, logran llamar la atención de los compradores.

The screenshot shows the Amazon.com product page for the DJI Mini 2 Fly More Combo. The main product is displayed with its image, title, and price. Below the main product, there are two recommendation sections: "Más artículos para explorar" and "Los clientes que vieron este producto también vieron".

**Más artículos para explorar**

Producto	Precio	Calificación
Base de carga DJI Mavic Mini Parte 19	US\$35.50	4.0
DJI Dron Mavic Air	US\$1,049.00	4.8
DJI Pocket 2 Creator Combo	US\$499.00	4.4
SanDisk MobileMate USB 3.0 lector de tarjetas microSD	US\$12.99	4.5
DJI Mavic Mini Quadcopter portátil 128 GB Bundle	US\$369.00	4.5
SanDisk Extreme Pro Micro SDXC UHS-I US A2 V30 Tarjeta de memoria	US\$53.89	4.8
Tomat Yagi-UDA - Amplificador de señal de antena para DJI Mavic Mini/Mavic 2/PRO/DJI...	US\$9.99	4.5

**Los clientes que vieron este producto también vieron**

Producto	Imagen
DJI Dron Mavic Air	[Image]
DJI Mavic Mini drone	[Image]
DJI Mavic Mini Fly Más	[Image]
DJI Mini 2 Hélices (par)	[Image]
Impermeable Hardshell	[Image]
DJI Mavic 2 Pro Dron	[Image]
DJI Pocket 2 -	[Image]

Figura 2 Captura de pantalla de Amazon.com

Amazon.com también tienen implementado su propio sistema de recomendaciones basado en filtrado colaborativo item a item, como ellos lo llaman. Su sistema asocia cada uno de los productos comprado por un usuario con una lista de productos similares que son obtenidos en base a los elementos que se hayan

comprado en un mismo pedido. También se basa en los ítems que tiene el usuario en su carrito de compra y en la clasificación de los productos según su interés.

Amazon guarda las secuencias de los clics que realizan sus clientes en la aplicación, así como los datos históricos de compras. Al dar clic en alguna sección que tenga relación con el sistema de recomendaciones varios datos son almacenados en su base de datos. Los usuarios podrán pensar que solo se guardó la información del producto, pero posiblemente se esté guardando el tiempo que duró el usuario viendo el perfil del producto. Estos datos son analizados y traducidos a resultados personalizados. Los usuarios de Amazon.com eran aproximadamente 150 millones en el 2012.

### c. Sistema de recomendaciones de Spotify.

Spotify es una plataforma de streaming de música. Cuenta con un gran catálogo de canciones y de usuarios. En la plataforma se pueden observar playlist personalizadas para cada uno de los usuarios. Estos playlist están generados en base a las canciones que fueron escuchadas por el usuario.

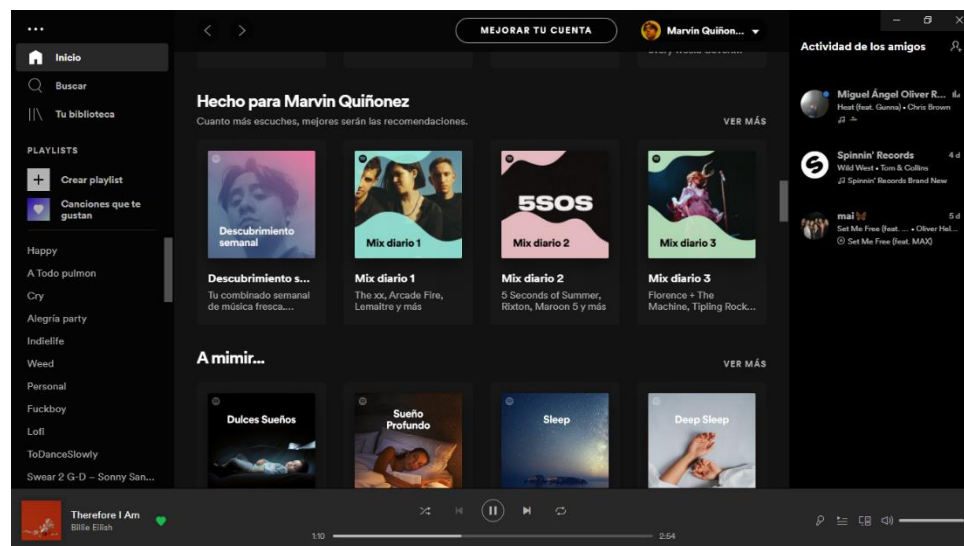


Figura 3 Captura de pantalla de Spotify

Esta plataforma implementa un sistema de recomendaciones que está basado en varios enfoques. Utiliza modelos de filtrado colaborativo con el que analiza el comportamiento de los usuarios de forma individual y global. Implementa Modelos de Procesamiento del Lenguaje Natural y hacen uso de modelos de audio.

Los sistemas de recomendaciones tienen varias características que son ventajosas para un sitio en la web.

- Pueden impulsar el tráfico de la aplicación que la utiliza.
- Entregan contenido relevante haciendo uso de análisis del uso actual del sitio del cliente y de su historial de navegación.
- Los datos se recopilan en tiempo real para que software pueda reaccionar a medida.

- Los compradores se involucran aún más en el sitio cuando se hacen recomendaciones personalizadas de productos.
- Las métricas avanzadas y los informes pueden mostrar definitivamente la eficacia de una campaña.
- Puede aumentar el número de artículos por pedido.

#### **4. Precios de implementación.**

El diseño e implementación de los sistemas de recomendaciones tienen distintos precios. Estos dependen de la magnitud de los proyectos y de las empresas responsables en brindar este servicio.

El costo de desarrollo usual para los motores de recomendaciones haciendo uso de algoritmos de Aprendizaje Automático se divide en varias fases:

- Análisis y estimación - Gratis
- Desarrollo del prototipo - \$5000
- Desarrollo de producto mínimo viable - \$10000
- Implementación y lanzamiento - \$5000

Con estos costos se tendría un total aproximado de \$20000 por proyecto. Los precios varían según los objetivos y tecnologías utilizadas.

#### **5. Validación de los sistemas de recomendaciones.**

Los S.R. tienen que ser evaluados y validados para medir el poder de las predicciones, para determinar la habilidad de predecir los intereses de los usuarios.

Para Shani y Gunawardana (2009:2), existen tres diferentes tipos de experimentos para validar los S.R. los cuales son: Experimentos offline, estudio de usuarios y experimentos online.

Los experimentos offline no tienen interacción con los usuarios, si no que trata de utilizar datasets con datos de usuarios calificando productos. Usando estos datos se puede simular los comportamientos que los usuarios tendrán en el sistema de recomendaciones.

En el estudio de usuarios se reúnen a sujetos y se les indica una serie de pasos para interactuar con el S.R. El investigador tiene que analizar el comportamiento del usuario y el comportamiento del sistema de recomendaciones. El objetivo es determinar si los usuarios obtienen recomendaciones de interés.

La evaluación online consiste en que el diseñador del sistema espera la influencia de la conducta de los usuarios. Con esta evaluación se quiere interactuar con distintos sistemas de recomendaciones para crear uno superior.

Para evaluar de forma matemática las recomendaciones de los S.R. existen varias métricas. Algunas de ellas son:

$$\text{Precisión} = \frac{| \{ \text{documentos relevantes} \} \cap \{ \text{documentos recuperados} \} |}{| \{ \text{documentos recuperados} \} |}$$

*Ecuación 1 Precisión*

La Precisión determina el ratio entre el número de documentos relevantes recuperados entre el número total de documentos recuperados.

$$\text{Exhaustividad} = \frac{| \{ \text{documentos relevantes} \} \cap \{ \text{documentos recuperados} \} |}{| \{ \text{documentos relevantes} \} |}$$

*Ecuación 2 Exhaustividad*

La Exhaustividad es utilizada en menor medida que la Precisión. Esta expresa la proporción de documentos relevantes recuperados entre el total de los documentos que son relevantes en la base de datos.

Aunque las pruebas matemáticas indiquen que el sistema tiene una precisión casi perfecta hay que tomar en cuenta lo que comentan algunos investigadores. Para Yeomans *et al.* (2019:5-18), los humanos tienden a preferir recomendaciones humanas incluso si el sistema de recomendaciones genera recomendaciones más acertadas. Esto se debe a que, según los estudios realizados, las personas están menos dispuestas a aceptar sistemas de recomendaciones que no pueden comprender.

Para ayudar a darle una solución a este problema, los diseñadores de sistemas de recomendaciones tratan que los sistemas no sean solo precisos, sino que también, tratan que sean familiares. La idea es generar confianza en el usuario.

## **B. Procesado de lenguaje natural**

El procesado de lenguaje natural es una técnica utilizada en el área de la computación y de la inteligencia artificial que permite a las computadoras leer, comprender e interpretar el lenguaje humano. Puede ayudar a las computadoras a medir las emociones y determinar qué partes del lenguaje humano son importantes. Para las computadoras, esto es muy difícil debido a la gran cantidad de datos no estructurados, la falta de reglas formales y la falta de un entorno o intenciones reales.

En la informática se utilizan las palabras para buscar información que está relacionada entre sí. Los algoritmos matemáticos no trabajan directamente con las palabras, ya que estos necesitan valores numéricos. Para que un algoritmo empiece a procesar palabras es necesario convertir estas en números. Para ello existen varias formas, en donde algunas son el Modelo de Bolsa de Palabras o TF-IDF.



## 1. Modelo de Bolsa de Palabras.

El Modelo de Bolsa de Palabras (BoW por sus siglas en inglés) no toma en cuenta el orden de las palabras, sino que se centra en la cantidad de veces que aparecen en los documentos.

Para ayudar a entender mejor la transformación se han definido las siguientes palabras:

- Token: Es un conjunto de caracteres que representa una unidad en el análisis de texto.
- Documento: Es el texto a analizar. Puede ser una oración, un tweet, etc.
- Corpus: Conjunto de documentos sobre el que se realiza el análisis.
- Vocabulario: Conjunto de tokens únicos obtenidos de cada documento.

El modelo permite el conteo y la relación de palabras entre los contenidos de los libros. Esta información se transforma en tokens.

Por ejemplo, Se tiene un corpus compuesto por 3 documentos:

- Documento 1: Bienvenida, casa, amigos, familiares.
- Documento 2: juega, casa, pelota, amigos.
- Documento 3: Cocina, hambre, comida, hambre.

Teóricamente el proceso de transformación se miraría así:

	bienvenida	casa	amigos	familiares	juega	pelota	cocina	comida	hambre
Documento 1	1	1	1	1					
Documento 2		1	1		1	1			
Documento 3							1	1	2

*Cuadro 2 Ejemplo teórico de transformación*

En la práctica se miraría así:

	0	1	2	3	4	5	6	7	8
0	1	1	1	1					
1		1	1		1	1			
2							1	1	2

*Cuadro 3 Ejemplo práctico de transformación*

Al tener los datos ya tokenizados se pueden almacenar en matrices para un mejor uso.

## 2. Term Frequency-Inverse Document Frequency (TF-IDF).

Es un algoritmo que es utilizado en los sistemas de recomendación basados en contenido. TF mide la frecuencia con la que aparece un término o frase en un documento determinado y lo compara con el número de documentos de todo el conjunto de documentos que mencionan el término. Para eliminar el problema de las palabras sin relevancia se aplica IDF (Frecuencia inversa de documento).

El IDF disminuye el peso de los términos que se repiten mucho en el total de documentos y le da un valor mayor a las palabras que no se utilizan con frecuencia.

TF-IDF se utiliza para la extracción de características en la recuperación de información y es una subregión del procesamiento del lenguaje natural.

Este método utiliza la frecuencia de un término (TF por sus siglas en inglés). Analiza la frecuencia de un término en un documento y la divide dentro del número total de términos en un documento.

$$Tf(t) = \frac{\text{Frecuencia de aparición del término } t \text{ en el documento}}{\text{Total número de términos en el documento}}$$

*Ecuación 3 Term Frequency*

Este método analiza la Frecuencia Inversa del Documento (IDF por sus siglas en inglés), o sea, el número total de documentos entre el número de documentos que contienen el término evaluado.

$$Idf(t) = \log_{10}\left(\frac{\text{Número total de documentos}}{\text{Número de documentos que contengan el término } t}\right)$$

*Ecuación 4 Inverse Document Frequency*

La fórmula final queda de la siguiente forma:

$$TF - IDF = Tf(t) * Idf(t)$$

*Ecuación 5 TF-IDF*

A diferencia del Modelo de Bolsa de Palabras, que identifica la frecuencia de aparición de un término, el TF-IDF identifica si realmente el término a evaluar es importante. Si un término aparece únicamente en un documento, entonces ese término es importante y lo que se habla en el documento tiene relación con este. En el BoW esto no ocurre ya que este trata de darle importancia a las palabras que más se repiten.

## C. Algoritmos de similitud

Los algoritmos de similitud son algoritmos que determinan la compatibilidad que existen entre dos conjuntos de datos. Permiten medir el espacio que existe entre dos puntos. Se puede utilizar para identificar puntos cercanos a uno en específico. Son algoritmos muy utilizados en el aprendizaje automático.

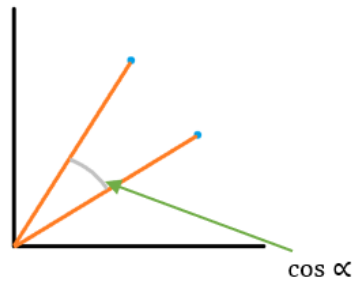
### 1. Similitudes basadas en cosenos.

Es un algoritmo basado en elementos. La similitud de coseno se utiliza para determinar la similitud entre documentos o vectores. Matemáticamente, mide el coseno del ángulo entre dos vectores proyectados en un espacio multidimensional.

Su similitud se calcula mediante:

$$\cos \cos (x_1, x_2) = \frac{\vec{x}_1 \cdot \vec{x}_2}{\|\vec{x}_2\| \|\vec{x}_1\|}$$

*Ecuación 6 Similitud de cosenos*



*Figura 4 Similitud de cosenos*

### 2. Similitudes basadas en correlación.

También existe la forma de buscar la similitud haciendo uso de la correlación. Por ejemplo, el coeficiente de correlación de Pearson es una prueba que se utiliza para medir la relación estadística entre dos variables continuas. Si la relación entre los elementos no es lineal, entonces el coeficiente no está representado adecuadamente.

Esta similitud es calculada mediante:

$$\text{sim}(x_1, x_2) = \frac{\sum_{u \in U} (R_{u,x_1} - \underline{R}_{x_1})(R_{u,x_2} - \underline{R}_{x_2})}{\sqrt{\sum_{u \in U} (R_{u,x_1} - \underline{R}_{x_1})^2} \sqrt{\sum_{u \in U} (R_{u,x_2} - \underline{R}_{x_2})^2}}$$

*Ecuación 6 Correlación de Pearson*

El coeficiente de correlación puede tomar un rango de valores de +1 a -1. Un valor de 0 significa que no existe correlación entre las dos variables. Un valor mayor que 0 indica una asociación positiva.

### 3. Distancia de Manhattan.

Es la distancia entre dos puntos en una cuadrícula la cual se basa en una ruta estrictamente horizontal y/o vertical, en contraposición a la distancia diagonal o en línea recta.

Representada por:

$$d_m(e_i, e') = \sum_{j=1}^N |e_i^j - e'^j|$$

*Ecuación 7 Distancia de Manhattan*

### 4. Distancia Canberra.

Es similar a la distancia de Manhattan, pero es más receptiva a los puntos de datos más cercanos al origen. Está representada por:

$$d(p, q) = \sum_{i=1}^n \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

*Ecuación 8 Distancia Canberra*

### 5. Distancia Euclidiana.

La distancia euclidiana es un número positivo, que indica la distancia entre dos puntos en el espacio euclidiano.

El algoritmo de Distancia Euclidiana es un algoritmo de grafos el cual puede ser utilizado para medir la distancia en línea recta entre dos puntos en un espacio n-dimensional. Haciendo uso de esta distancia se pueden obtener datos que identifiquen elementos que tengan características similares. Se utiliza para medir la similitud entre pares de datos.

La distancia euclidiana es un algoritmo que se utiliza generalmente en la implementación del algoritmo K-NN. Para calcular esta distancia se utiliza:

$$d_E(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

*Ecuación 9 Distancia Euclidiana*

$p, q$  = dos puntos en el espacio euclidiano

$p_i, q_i$  = Vectores euclideanos, empezando desde el origen del espacio

$n$  = limite superior

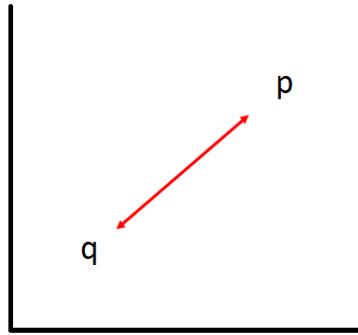


Figura 5 Distancia entre dos puntos en un espacio 2D

## D. K-NN (K nearest network o K vecinos cercanos)

K Vecinos más cercano (KNN) es uno de los algoritmos más simples para problemas de regresión y clasificación en el aprendizaje automático. El algoritmo KNN utiliza datos y clasifica nuevos puntos de datos basándose en medidas de similitud (como funciones de distancia).

Las ventajas de este algoritmo son:

- Facilidad para interpretar la salida
- Tiempo de cálculo
- Poder predictivo

Se puede utilizar K-NN para identificar a qué clase pertenece un elemento.

Para entender mejor el algoritmo se presente el siguiente ejemplo de clasificación en una forma gráfica:

*Se tienen dos clases A (rojo) y B (celeste).*

*Se quiere descubrir a qué clase pertenece el punto de color verde.*

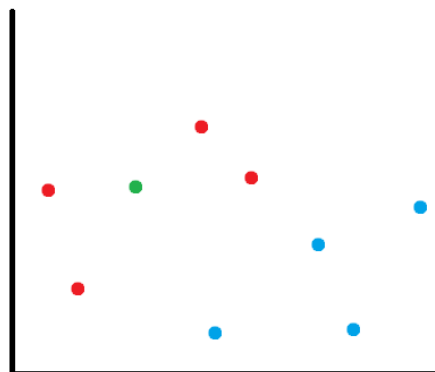


Figura 6 Ejemplo de clasificación

Al realizar un círculo, teniendo en el centro el punto a analizar, se puede observar que el punto verde pertenece a la clase A porque la mayoría de los elementos más cercanos pertenecen a esta clase.

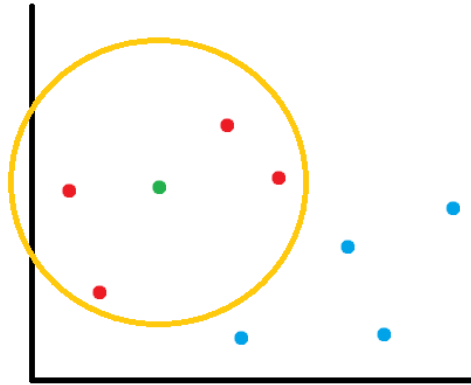


Figura 7 Ejemplo de clasificación más expresiva

El algoritmo K-NN nos ayuda a realizar esta clasificación de forma matemática, y además es aplicable a modelo de datos grandes, aunque hay que tener en cuenta la cantidad de procesamiento que realiza.

Pasos para la implementación:

1. Obtener el dataset de entrenamiento y los datos a evaluar.
2. Calcular la distancia entre los vectores almacenados y el nuevo vector.
3. Seleccionar los k elementos más cercanos.
4. Asignar a x la clase más frecuente

*INICIO*

*Entrada:  $D = \{(x_i, c_i), \dots, (x_i, \dots, C_N)\}$*

*$X = (x_1, \dots, x_n)$  dato a predecir*

*PARA todo objeto ya clasificado  $(x_i, c_i)$*

*calcular  $d_i = d(X_i, X)$*

*Ordenar  $d_i (i=1, \dots, N)$*

*Quedarse con los K casos  $D_X^K$  ya clasificados más cercanos a x*

*Asignar a X la clase más frecuente en  $D_X^K$*

*FIN*

Figura 8 Pseudocódigo del algoritmo KNN

## E. Plataformas de comercialización de libros en línea

La web es un gran conjunto de información a la cual se puede acceder mediante internet. A través de la web se puede acceder a varias páginas o aplicaciones web que intentan comercializar un producto. Algunas de estas se centran en comercializar libros.

Una página o aplicación web de comercialización de libros es una plataforma por la cual se venden, se intercambian, o se pueden descargar o leer libros.

## 1. Elejandria.com.

Elejandria.com es un sitio web dedicado a la comercialización de libros de dominio público. Los libros de dominio público son aquellos en los que los derechos de autor han expirado. Esta página ofrece una gran variedad de libros, basándose en la ley de copyright de España.



Figura 9 Captura de pantalla de Elejandria.com

Cuenta con secciones en las que se les recomienda libros a sus usuarios.

### OTROS LIBROS INTERESANTES



Figura 10 Captura de pantalla de sección de Elejandria.com

Unas de las recomendaciones famosas entre los sitios web con sistemas simples son los basados en los más descargados.



Figura 11 Captura de pantalla de sección de Elejandria.com

## 2. Casa del libro.

Es un sitio web perteneciente a la cadena Casa del Libro en el cual se pueden realizar compras de libros físicos y digitales, además de contar con una sección para libros de dominio público.

Cuenta con varias secciones de recomendaciones. Una de las secciones es la llamada “Novedades destacadas”. En ella se pueden observar productos con calificaciones altas.



Figura 12 Captura de pantalla de casadellibro.com

## F. Sistema informático

Un sistema informático es un conjunto de componentes que recolecta datos, realiza procesos, almacena datos y distribuye datos. Es una combinación de hardware y software que ayudan a resolver problemas, comúnmente de optimización, a través de distintos procesos.

El sistema informático envuelve una gran variedad de tecnologías como computadoras, aplicaciones, base de datos, sistemas de comunicaciones, entre otros, para realizar tareas específicas, interactuar con distintos actores en diferentes contextos.

Para Boell y Cecez-Kecmanovic (2015:4959-4968), Los sistemas de información (SI) involucran una variedad de tecnologías de la información como computadoras, software, bases de datos, sistemas de comunicación, Internet, dispositivos móviles y mucho más, para realizar tareas específicas, interactuar e informar a varios actores en diferentes contextos organizacionales o sociales.



## 1. Estructuras.

Los sistemas informáticos se estructuran por subsistemas: Subsistema físico y Subsistema lógico.

El subsistema físico es el asociado al hardware. Engloba las computadoras, servidores, memorias, discos duros, etc.

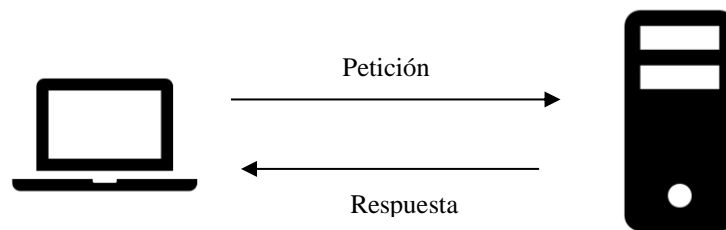
El subsistema lógico es el que está asociado al software y la arquitectura. Engloba las aplicaciones, los sistemas operativos, los gestores de base de datos, etc.

## 2. Arquitectura.

Los sistemas de informáticos se dividen en distintas arquitecturas, dependiendo de los objetivos asignados.

### a. Arquitectura cliente-servidor.

La arquitectura cliente-servidor es una de las arquitecturas más utilizadas en la actualidad. La idea de esta arquitectura es tener un cliente, una aplicación instalada en un ordenador, y un servidor, uno o varios ordenadores que contienen la lógica de los procesos. Los sistemas web utilizan este concepto, ya que para hacer uso de las aplicaciones en la web se necesitaría utilizar un navegador para acceder a este contenido. Con esta arquitectura, un cliente realiza una petición y el servidor le da una respuesta.



*Figura 13 Cliente-servidor*

### b. Arquitectura de 3 capas.

Esta es una arquitectura que segmenta el sistema en tres capas para tener un mejor control y seguridad.

Las tres capas son:

- Presentación
- Negocio
- Datos

La capa de presentación es la encargada de mostrarle al usuario una forma de interactuar con el sistema. La interacción puede ser mediante vistas o formularios.

La capa de negocio es la encargada de la lógica del sistema. Se encarga de ejecutar procesos lógicos que ayuden al funcionamiento del sistema. También, se encarga de obtener y guardar datos en la base de datos. Es el intermediario entre el usuario y otros servicios.

La capa de datos es la encargada de administrar los datos del sistema. El usuario no puede acceder a esta capa para obtener información, si este quiere obtener datos de la base de datos tendrá que hacer la petición a la capa de negocio.

### **c. Arquitectura de N capas.**

Esta arquitectura permite segmentar el sistema en varias partes, aumentando la posibilidad de brindar servicios simultáneamente. También permite la fácil escalabilidad de las partes que componen el sistema.

## **3. Ingeniería de software.**

Se le llama Ingeniería de software al conjunto de técnicas ingenieriles implementadas en el desarrollo de software.

Un software es un programa informático el cual ejecuta distintas instrucciones. Este no se crea, se desarrolla. Un software no se puede, o más bien dicho, no se debe crear, tiene que pasar por distintos procesos que son repetitivos porque siempre tendrá cambios. Todo software debe tener un mantenimiento porque en algún momento puede o existirán cambios. Los cambios pueden ser a nivel de requerimientos, a nivel de error o a nivel de hardware. Si se necesita agregar una opción, arreglar un error o adaptar a un sistema en específico, todos los cambios, por más mínimos que sean, tienen que pasar por el proceso de desarrollo de software.

Para Fernandes y Machado (2016:15), la ingeniería de software es una disciplina de la ingeniería que se centra en todos los aspectos relacionados al desarrollo de software.

El uso de la ingeniería de software permite desarrollar un producto de calidad, siguiendo distintos modelos de pasos que ayudarán a construir un software que cumpla con lo requerido por el interesado.

La ingeniería de software no tiene una lista de pasos ya establecido, estos se determinan dependiendo del tipo de proyecto que se realice. Existen distintos modelos que han sido estudiados e implementados, y que son de mucha ayuda en los proyectos de software.

### **a. Modelo de Cascada.**

El modelo de cascada es un patrón de desarrollo de software secuencial. Este modelo separa el desarrollo de software en distintas fases, de tal forma que se inicia una fase cada vez que se concluya la anterior, esa es la razón por la cual se le llama desarrollo secuencial.

Las fases para el desarrollo de software que propone este modelo varían dependiendo del tipo de proyecto o del ingeniero que lo vaya a implementar. En el transcurso de los años varios autores han definido distintas fases.

Se puede utilizar en el desarrollo de proyectos pequeños o medianos con requerimientos definidos los cuales no pueden cambiar. Se recomienda utilizar en el desarrollo de proyectos donde se necesite un estricto

control, un presupuesto bien definido y un tiempo límite establecido, como por ejemplo los proyectos gubernamentales. Al usar este modelo se debe tener una idea de las tecnologías que se van a utilizar, ya que no puede existir cambios en una fase cuando ya está concluida.

**b. Modelo de validación y verificación (Modelo V).**

Este es otro modelo secuencial, muy similar al modelo de cascada, en el que cada fase tiene una actividad de prueba, implicando un control de calidad excepcional. Este modelo es uno de los más costosos ya que se requiere realizar pruebas por cada una de las fases, aumentando el tiempo de desarrollo. Al igual que el modelo de cascada, este tiene que tener los requisitos definidos al principio y no se pueden cambiar, ya que el cambio durante el desarrollo es muy costoso y difícil de implementar.

Se usa comúnmente en el desarrollo de proyectos donde los fallos no son aceptables, como por ejemplo los softwares de salud, los softwares de guerra, los softwares de aviación.

**c. Modelo en espiral.**

El modelo en espiral es un modelo en el que el análisis de riesgos es esencial. Se le llama modelo en espiral porque las fases se pueden repetir de tal forma que se van analizando los errores y posibles riesgos en cada fase hasta concluir el producto. Los ciclos repetidos en espiral amplían los plazos del proyecto. La ventaja que tiene este modelo es que el cliente tiene una mayor participación en el proceso de exploración y revisión de cada ciclo.

Se utiliza para desarrollar proyectos donde existan requisitos pocos claros o ambiciosos, proyectos grandes, actividades de investigación o en el desarrollo de un nuevo servicio o producto.

**4. Modelo Vista Controlador (MVC).**

El MVC es un patrón de arquitectura de software que se centra en la separación de las vistas, la lógica y los datos. Las capas son independientes entre ellas, permitiendo que los proyectos de grandes dimensiones se trabajen de una manera más eficiente.

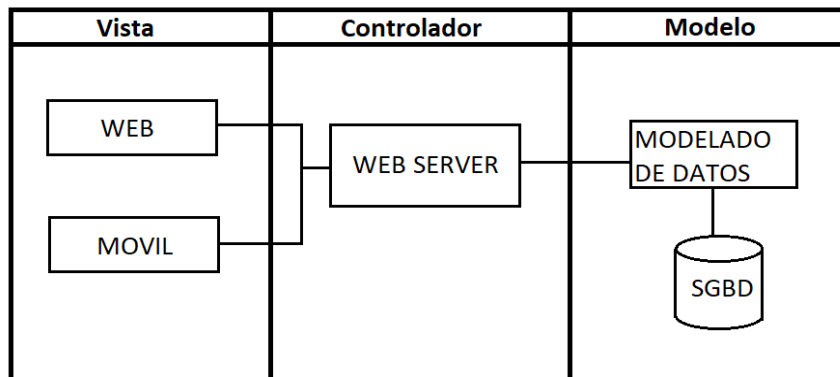


Figura 14 Modelo vista controlador

Este patrón surge de la necesidad de desarrollar software más robusto, donde el mantenimiento sea fácil, al igual que la escalabilidad. La ventaja que tiene es que las distintas partes del código pueden ser reusables. El diseño está dividido de la lógica del sistema, permitiendo un desarrollo más ágil.

En esta arquitectura las vistas son las páginas con las que el usuario interactúa, el controlador es el encargado de tener todas las instrucciones a nivel lógico del software y el modelo es el encargado de administrar los datos.

## **V. Identificación del problema**

### **A. El problema**

El uso de Sistemas de Recomendaciones (SR) ha ayudado a distintos sitios web a filtrar la gran cantidad de información que manejan para mostrarle al usuario solo lo que posiblemente necesita. Estos sistemas ayudan al usuario a encontrar contenido de interés, aumentando su interacción en los sitios que implementan esta tecnología. Los usuarios no son los únicos beneficiados al usar estos sistemas, el sistema de recomendación aumenta la interacción de los usuarios y aumenta las ventas, ya que se están mostrando recomendaciones personalizadas.

El uso del Filtrado Colaborativo, que es un enfoque para la creación de los SR, es muy utilizado por distintas aplicaciones. Ayudan a que un usuario encuentre contenido según los elementos vistos por usuarios con gustos similares al del mencionado de primero. El Filtrado Colaborativo es muy utilizado en los sitios dedicados a la distribución de libros, generan recomendaciones de libros según las calificaciones de otros usuarios. Aquí se ha detectado un problema, las recomendaciones de libros se hacen según gustos similares, no a contenido similar. Los libros son tratados como un producto, no como un elemento que brinda contenido. Para generar recomendaciones basadas en contenido se tendría que utilizar otro enfoque.

El uso del Filtrado Basado en Contenido, que es otro enfoque para la creación de los SR, permite generar recomendaciones en base al contenido de los elementos que al usuario previamente le gustaron, no se implementa demasiado en las recomendaciones de libros. Varios autores han escrito acerca de los beneficios que puede traer el uso de este tipo de filtrado. Algunos de ellos han desarrollado modelos según los párrafos de los libros los cuales han sido detectados como importantes para el usuario, pero estos modelos utilizan muchos recurso computacionales y físicos, aumentando los costos de mantenimiento o de implementación.

### **B. Mercado objetivo**

Personas y entidades públicas o privadas que tengan la necesidad de implementar un sistema web de recomendaciones de libros y que tengan los conocimientos básicos sobre sistemas web.

## VI. Solución propuesta

Las páginas web o aplicaciones web dedicadas a la distribución de libros digitales y físicos tienen sistemas de recomendaciones basados en el filtrado colaborativo, el cual recomiendan elementos según usuarios que tienen gustos similares al de otro, o según a valoraciones.

La propuesta trata en la elaboración de un sistema de recomendaciones de libros que permite la recopilación de datos y el tratamiento de este para mostrar una serie de libros recomendados según el historial del usuario. El sistema se basa en el enfoque Content-based Filtering o Filtrado Basado en Contenido.

Lo innovador en esta propuesta es que se centra en los detalles de la portada de los libros, así como en sus resúmenes, permitiendo que el usuario no solo encuentre libros con portadas similares, sino que, encuentre libros con historias similares.

Este sistema toma en cuenta el resumen y la portada del libro para poder realizar las recomendaciones personalizadas.

El sistema se presenta bajo la licencia MIT, permitiéndole al interesado la libre distribución, modificación, implementación, entre otros, sin tener que realizar algún pago.

### A. Arquitectura propuesta

Al sistema se le ha dado el nombre de Sistema de Recomendaciones de Libros Basado en las Portadas y Resúmenes (SRLBPR). Para este sistema se recomienda la siguiente arquitectura:



*Figura 15 Arquitectura propuesta*

El primer nivel consta de los accesos del cliente con el que el usuario interactuará con el sistema. El segundo nivel está compuesto por dos servidores. El primero, llamado Servidor Web, provee la aplicación web, gran parte de la lógica del sistema. El segundo, llamado Servidor de Recomendaciones, es el que tiene el motor de recomendaciones y procesos de transformación de datos. Los datos transformados se guardan en este servidor. Y el tercer nivel está compuesto por la base de datos. La base de datos almacenará los datos generales del sistema como: perfiles de libros, datos de usuarios e historiales de visualización del perfil del libro.

## **1. Componentes.**

- Cliente web: aplicación web para visualizar recomendaciones de libros. Se puede acceder a ella mediante cualquier navegador. Está conformada por perfiles de los libros y secciones de recomendaciones.
- Cliente administrador: aplicación web para la administración de los perfiles de los libros.
- Servidor de Recomendaciones: encargado de generar las recomendaciones en base al motor de recomendaciones que tiene. También es el encargado de traducir la información a datos procesables por el algoritmo.
- Base de datos: encargada de almacenar los datos de los usuarios y de los libros. También se encarga de proporcionarle la información al Servidor de Recomendaciones.

## B. Base de datos propuesta

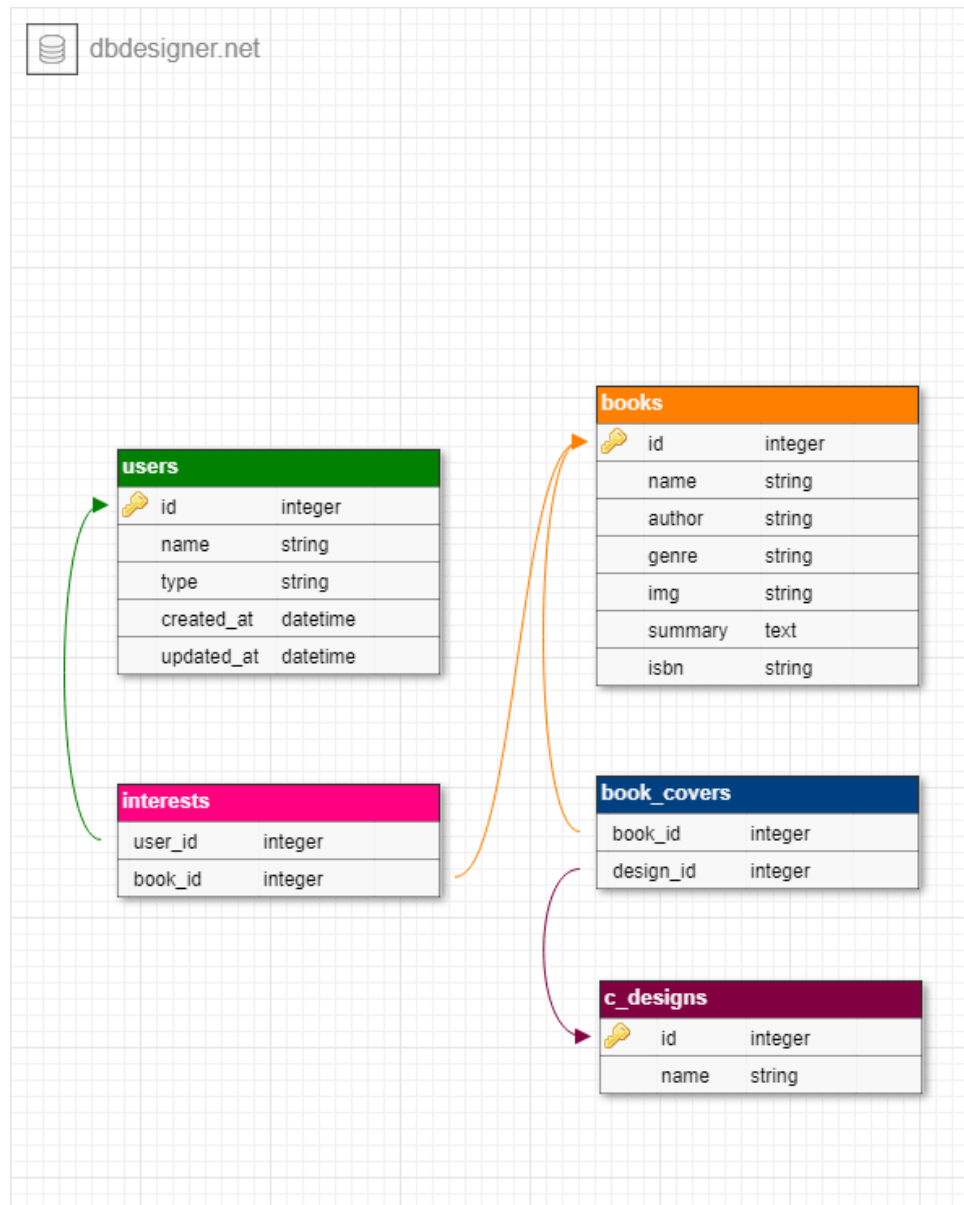


Figura 16 Base de datos propuesta

La tabla “users” almacenaría los usuarios, la tabla “books” los libros, la tabla “c\_designs” almacenaría las características o etiquetas que van a describir a las portadas de los libros, la tabla “book\_covers” almacenaría las características seleccionadas para la portada del libro y la tabla “interests” almacenaría los libros que el usuario guarde.



## C. Algoritmo

Se propone el uso de un algoritmo al que en esta investigación se le ha llamado “Algoritmo de ordenamiento de distancias”, el cual es una modificación del Algoritmo KNN. Este algoritmo obtiene las distancias entre el elemento a evaluar y cada uno de los elementos del set de datos. Al obtener las distancias, estas se ordenan de forma ascendente, quedando los elementos más similares al principio. Los datos donde la distancia tiende a 0 son los más similares al dato evaluado. Este no es un algoritmo de clasificación.

En la implementación de este algoritmo se pueden aplicar distintas distancias, pero la que se ha utilizado para este proyecto ha sido la Distancia Euclidiana.

En la siguiente figura se presenta el pseudocódigo del “Algoritmo de ordenamiento de distancias” (Algoritmo KNN sin la clasificación).

*INICIO*

*Entrada:*

*K= número de datos similares que se quieren encontrar*

*D = {P<sub>1</sub>(x<sub>1</sub>, ..., x<sub>n</sub>), ..., P<sub>n</sub>(x<sub>1</sub>, ..., x<sub>n</sub>)}*

*x = (y<sub>1</sub>, ..., y<sub>n</sub>) dato a predecir*

*PARA todo objeto en D*

*calcular di = d<sub>Euclidiana</sub>(P<sub>i</sub>, x)*

*Ordenar di(i=1, ..., N)*

*Quedarse con los K casos más cercanos a d<sub>Euclidiana</sub>(x, x)*

*FIN*

*Figura 17 Algoritmo propuesto*

## D. Datos a analizar

Los datos son una parte importante de los sistemas de recomendaciones. Estos pueden ser analizados por algoritmos. Tienen que estar estructurados y tienen que aportar valor.

El sistema se propone utilizar el enfoque “Filtrado basado en contenido”, por lo que se requiere que el sistema de recomendaciones trabaje con el contenido del libro. Para este estudio se tomó como contenido a la portada de los libros, sus resúmenes, el título del libro y el género del libro.

Se propone el uso del diseño y significado de la portada y el resumen de los libros para generar las recomendaciones. Con esto se quiere lograr que el usuario vea recomendaciones de libros que tengan diseños de portadas similares o que vea recomendaciones de libros con temas similares.

Se tomó el resumen de los libros ya que es un texto que le brinda al lector la información abreviada del libro, y se tomó en cuenta el diseño de la portada ya que es una herramienta de marketing que ayuda a que los libros tengan una mayor ventaja competitiva. Cada libro tiene un diseño diferente en sus portadas con la cual tratan de expresar la idea principal.

La idea principal es simular las recomendaciones humanas sobre la elección de un libro en base a los elementos descritos anteriormente.

La idea del uso de la portada de un producto en los sistemas de recomendaciones no es nueva. Netflix la utiliza para determinar que portadas son las que han tenido un mayor impacto sobre los usuarios para luego mostrarle recomendaciones en base a ello.

## 1. Portada a información.

Los algoritmos matemáticos no pueden detectar literalmente qué es lo que hay en una portada. Aun no son tan capaces de entender lo que esta parte del libro representa. Para ello se propone el uso de etiquetas o características que describan el diseño y expresión de una portada.

Cuando el administrador del sistema crea un nuevo perfil de un libro, tiene que ingresar datos como nombre, autor, género entre otros. Se sugiere la implementación de un área para la descripción de la portada. Esta área debe de aceptar la selección múltiple ya que una portada no puede ser descrita con una simple palabra.

**Nuevo libro**

**Título**

---

**Diseño de cubierta**

Seleccione cacterísticas ▼

*Figura 18 Ejemplo de formulario*

Para describir a la portada se propone la siguiente estructura de datos.

<b>Tipo de portada</b>	
<b>Nombre</b>	<b>Detalles</b>
simple	La portada cuenta con el nombre del libro, del autor y/o de la editorial.
complejo	La portada cuenta con el nombre del libro, del autor, de la editorial y/o tiene un diseño.

*Cuadro 4 Descripción del tipo de portada*

<b>Diseño de portada</b>	
<b>Nombre</b>	<b>Detalles</b>
minimalista	La portada es de aspecto minimalista.
personas	La portada tiene una fotografía de personas.
persona	La portada tiene una fotografía de una persona.
animales	La portada tiene relación con animales.
animada	La portada tiene un diseño animado.
animada_persona	La portada tiene un diseño animado donde hay una persona.
animada_personas	La portada tiene un diseño animado donde hay varias personas.
historico	Tiene relación con hechos históricos
antiguo	Tiene relación con la antigüedad. Desde el año 1999 hacia atrás.
autoestima	Tiene relación con la autoestima.
abstracto	La portada tiene un diseño abstracto.

epico	La portada tiene un diseño épico. (Batallas, aventuras legendarias, ficticias, fabulosos, personajes imaginarios )
sombras	La portada muestra figuras de un color sólido y fondo de un color, diferente al de las figuras.
letras	La portada muestra palabras con letras grandes.
rasgado	La portada muestra una imagen rasgada.
hoja_doblada	La portada muestra una imagen de una hoja doblada.
romantico	La portada tiene relación con el amor entre personas.
cara	La portada muestra principalmente la cara de una o varias personas.
collaje	La portada muestra la unión de varias imágenes.
objeto	La portada muestra a un o varios objetos.
simbolo	La portada muestra a uno o varios símbolos.
paisajes	La portada muestra un paisaje.
flores	La portada contiene flores.
tenebroso	La portada tiene un aspecto tenebroso (oscuro)
ciencia	La portada está relacionada a la ciencia
familia	La portada está relacionada a la familia.
guerra	La portada está relacionada a la guerra.
suspense	La portada tiene un aspecto de suspense
terror	La portada está relacionada con el terror.
espacio	La portada tiene relación con el espacio (universo).

alegria	La portada representa alegría.
tristeza	La portada representa tristeza.
pintada	La portada tiene una imagen con características de ser una pintura
medieval	La portada tiene relación con la era medieval
soldado	La portada muestra soldados.
deporte	La portada tiene relación con el deporte.
comida	La portada muestra comida.
tecnologia	La portada muestra tecnología.
arte	La portada muestra arte.
negocios	La portada muestra relación con los negocios.
empoderamiento	La portada tiene relación con el empoderamiento.
ciencia_ficcion	La portada tiene relación con la ciencia ficción.
tenebroso	La portada tiene personajes tenebrosos.
asesinato	La portada tiene relación con un o varios asesinatos.
blanco_negro	La portada tiene un aspecto blanco y negro.
misterio	La portada tiene un aspecto blanco y negro
sepia	La portada tiene un color sepia.

*Cuadro 5 Descripción del diseño de la portada*

Todos estos datos se encontrarían en la tabla `c_designs` de la base de datos. Se le puede agregar más palabras a esta sección. Mientras más palabras se usen al describir a una portada, mejor será la predicción del algoritmo.

## 2. Resumen.

Los resúmenes de los libros se pueden utilizar literalmente, sin embargo, estos tienen preposiciones, lo que ocasiona que algunas palabras se repitan, no porque sean importantes, sino porque son necesarias para que las personas entiendan el contexto de una oración. Al utilizar técnicas de procesamiento de palabras las preposiciones podrían generar valores significativos.

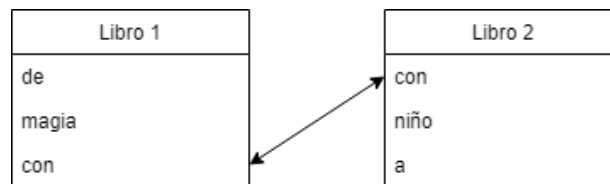


Figura 19 Palabras indeseadas

En el diagrama anterior podemos ver un ejemplo de cómo una preposición puede aumentar la relación entre dos libros. Para evitar esto se implementó una lista de palabras “vacías” a eliminar.

## 3. Procesamiento de palabras.

El Algoritmo de Ordenamiento de Distancias funciona con valores, no entiende las palabras, por ello se propone el uso de técnicas de procesamiento de palabras. Para transformar las etiquetas que describen las portadas de los libros se sugiere el Modelo de Bolsa de Palabras (BoW por sus siglas en inglés). Esta técnica permite contar las veces que una palabra aparece en cada uno de los documentos. Para el procesamiento de palabras del resumen se sugiere el uso del modelo Term Frequency-Inverse Document Frequency, el cual valora las palabras no por las veces que se repite, sino que, por la importancia que esta tiene. Estos valores comúnmente se trabajan en matrices.

Se propone formar dos documentos por cada uno de los libros, de tal forma que se tenga al final dos sets de datos.

El primer documento tendría los datos título, diseño de portada y género, mientras que el segundo documento tendría al resumen, sin las preposiciones.

Los datos procesados se generarían y se almacenarían en el Servidor de Recomendaciones cada vez que el administrador del sistema ingrese, edite o elimine algún libro. Se guardaría un dataset con los datos procesados del primer documento (título+diseño de portada+género) de cada uno de los libros y se guardaría un dataset con los datos procesado del segundo documento (resumen) de cada uno de los libros.

## E. Uso del algoritmo

Para el uso del algoritmo tienen que existir los dos dataset almacenados en el Servidor de Recomendaciones.

1. El Servidor Web obtiene el id de uno de los perfiles de los libros vistos o guardados por el usuario.
2. El Servidor Web solicita las recomendaciones al Servidor de Recomendaciones, enviándole el id seleccionado y el tipo de recomendación que quiera hacer (por portada o por resumen).
3. El Servidor de Recomendaciones obtiene el dataset perteneciente al tipo de recomendación solicitada y obtiene los datos procesados del libro con el id recibido (datatest).
4. El Servidor de Recomendaciones ejecuta el algoritmo enviándole a este el dataset, el datatest y el valor de K. El algoritmo retornaría los K elementos similares al datatest con su respectivo id.
5. El Servidor de Recomendaciones envía una cadena json con los k primeros id.
6. El Servidor Web consulta los libros con esos id's en la base de datos y los muestra en el área del cliente.

A continuación, se presenta un ejemplo del uso del algoritmo en el sistema de recomendaciones.

Supóngase que se tienen los siguientes *datos transformado* de los libros (dataset):

<b>Nombre del libro</b>	<b>Datos del libro</b>
Libro1	[2,7,5,8]
Libro2	[4,9,4,1]
Libro3	[3,8,9,7]
Libro4	[4,2,9,8]

*Cuadro 6 Ejemplos de datos transformados*

Se quiere predecir los 3 (K=3) libros más similares a uno de los últimos libros visto por el usuario.

Supóngase que el último libro visto o descargado por el usuario es el libro 3:

LibroTest	[3,8,9,7]
-----------	-----------

*Cuadro 7 Ejemplo de datos del libro de prueba*

Entonces se aplica la Distancia Euclidiana entre el LibroTest y cada uno de los libros del dataset.

Las distancias obtenidas son:

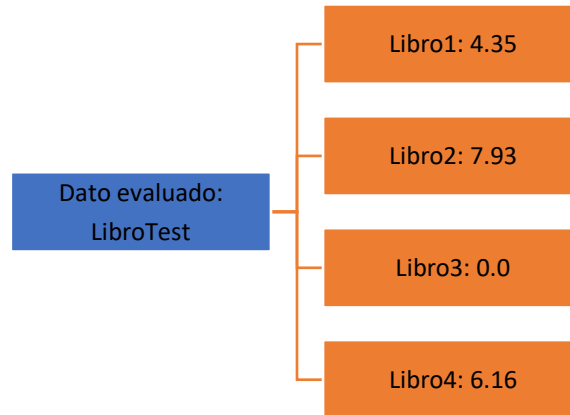


Figura 20 Ejemplo de distancias

Al ordenar de forma ascendente las distancias, se obtiene:

1. Libro3: 0.0
2. Libro1: 4.35
3. Libro4: 6.16
4. Libro2: 7.93

Nuestro resultado serían los  $K=3$  primeros datos: {Libro3, Libro1, Libro4}

Se puede observar que el libro más similar que existe es el mismo libro evaluado, porque obviamente comparten los mismos datos.

Luego se pueden observar los datos que le siguen. Mientras la distancia de un libro tienda a 0, más similar es al libro evaluado.

Se sugiere dejar el elemento donde la distancia es 0 (el mismo libro evaluado) ya que en un sistema de recomendaciones no se tiene que basar netamente en lo matemático, tiene que intentar pensar como un humano. Las recomendaciones pueden ser las mejores, pero si el usuario no tiene confianza con ellas no servirán de nada. Al momento de dejar el mismo elemento evaluado en los resultados, que es un libro que el usuario ya leyó, el usuario podría tener una mayor confianza, ya que en los resultados hay un libro que ya haya visto.

Se plantea el uso de este algoritmo basado en distancias ya que se podría aprovechar el error que pueda existir entre ellos. En este proyecto se les llama error a las distancias mayores a 0.

Por ejemplo, si se tienen los siguientes libros con sus distancias respecto al libro 5 y queremos obtener  $K=5$ , el algoritmo podría retornar 1 libro con un error igual a 0, que es el mismo libro evaluado, 3 libros con el error cercano a 0, que son libros similares, y 1 libro con un error muy alejado a 0, el cual posiblemente solo comparte una característica con el libro evaluado. Al mostrarle estas recomendaciones al usuario podrá



ver el libro cuyo perfil visitó o leyó, libros que son similares y un libro con características muy diferentes pero que podría ser de interés.

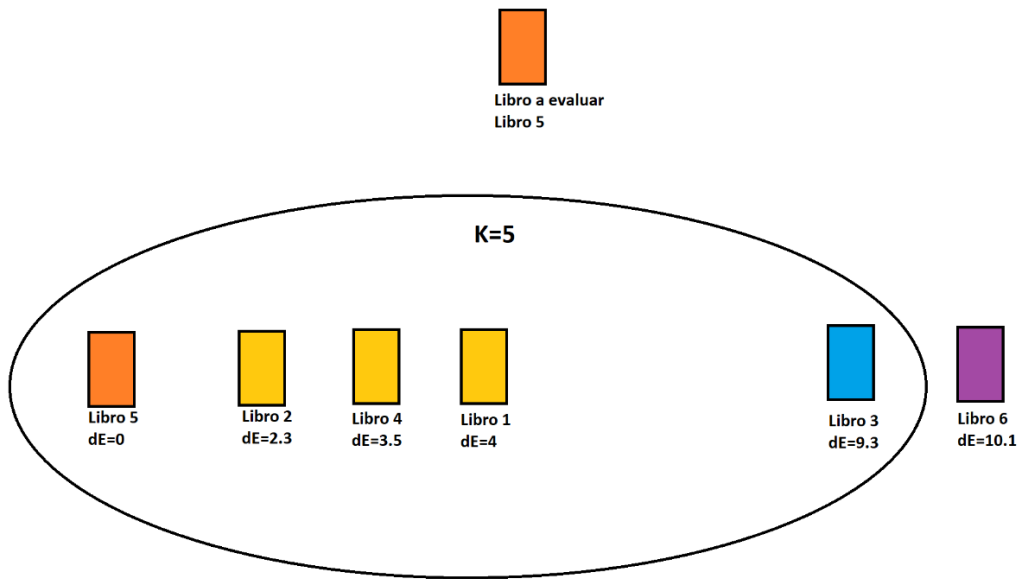


Figura 21 Ejemplo del funcionamiento del algoritmo

## **VII. Desarrollo de la propuesta**

### **A. Plataforma web**

La plataforma es una aplicación web de recomendación de libros, a la cual los usuarios acceden mediante el cliente, en donde se pueden ver perfiles de los libros y secciones de recomendaciones donde se le muestran al usuario una serie de libros que podrían ser de su interés.

Tiene un área de administración donde el administrador de la plataforma puede acceder para administrar los perfiles de los libros.

Con el desarrollo de esta plataforma se busca crear un sistema de código libre el cual tenga un sistema de recomendaciones centrado en las características de la portada de los libros, así como en sus resúmenes.

En la plataforma se puede:

- Descubrir libros mediante recomendaciones
- Crear los perfiles de los libros (solo administrador)
- Editar los perfiles de los libros (solo administrador)
- Eliminar los perfiles de los libros (solo administrador)

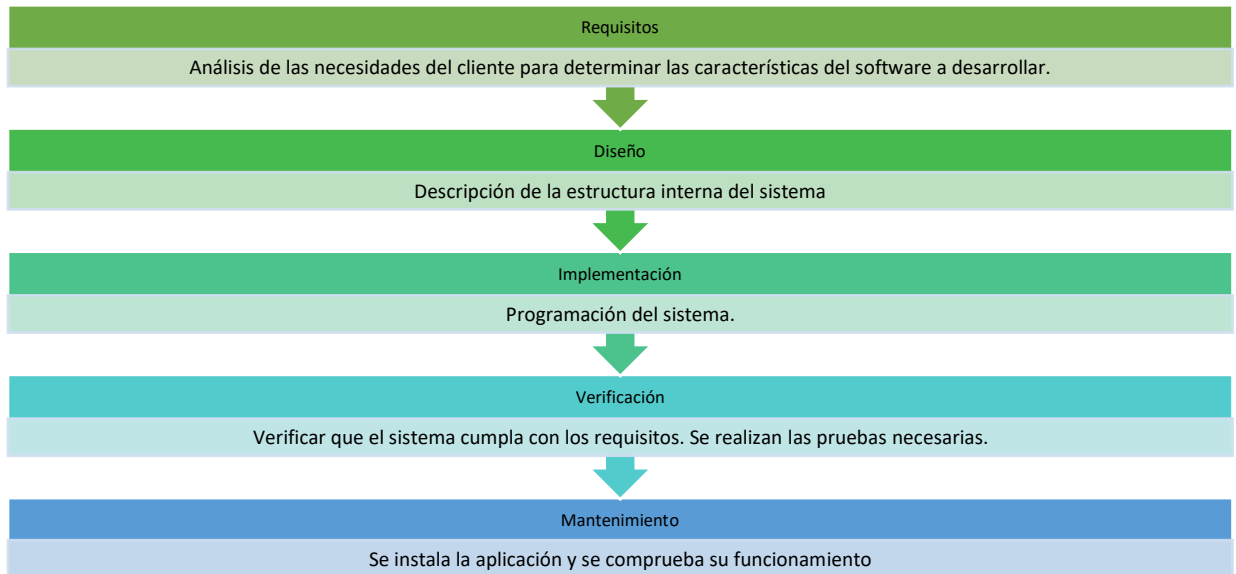
Para el desarrollo de esta propuesta se siguieron algunos lineamientos de la ingeniería de software.

La ingeniería de software es una rama de las ciencias de la computación que estudia el desarrollo de software basándose en métodos y técnicas de la ingeniería. Según Ian Sommerville (2005), la ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

### **B. Modelo utilizado**

Se ha decidido trabajar este proyecto haciendo uso de una adaptación del modelo en cascada propuesto por Winston W. Royce. A este modelo también se le llama secuencial.

El desarrollo en cascada es un modelo de ciclo de vida del software en el que cada etapa se ejecuta una tras otra. El inicio de cada etapa debe esperar a que la etapa anterior finalice.



*Figura 22 Modelo en cascada*

Dentro del método de cascada se cubrieron las siguientes etapas:

- Requisitos: Análisis, especificación de los requisitos y planificación.
- Diseño: Diseño y especificación del sistema.
- Implementación: Programación del sistema.
- Verificación: Pruebas del sistema. Se verifica que todos los componentes del sistema funcionen correctamente y que cumplan con los requisitos.

El desarrollo del sistema se centró en la web, por lo que se utilizaron herramientas necesarias para construir el mismo. Se utilizó la estructura Modelo Vista Controlador (MVC) para desarrollarla.

## C. Requerimientos

La recopilación y análisis de requisitos es muy importante en la ingeniería de software ya que establece el objetivo de este, las acciones que tiene que realizar, las opciones que debe tener, entre otras. El sistema final deberá cumplir en un cien por ciento lo establecido en los requisitos.

En esta etapa se encuentra en forma detallada la información requerida para comprender el sistema desarrollado, a la vez se pueden definir estrategias que se aplicarán para cumplir con los requisitos.

Para Pressman (2010), los requerimientos llevarán a la comprensión de cuál será el efecto que tendrá el software en el negocio, qué es lo que quiere el cliente y cómo interactuarán los usuarios finales con el software.

El propósito de este proyecto de desarrollo es la creación de un sistema web de recomendaciones de libros, haciendo uso de técnicas de la ingeniería de software, con la finalidad de tener una alternativa a los sistemas de recomendaciones actuales, que comúnmente están basados en el Filtrado Colaborativo.

Este desarrollo exige que se realicen las acciones necesarias para que las recomendaciones se muestren de manera automatizada, sin que interfiera en las acciones de los usuarios.

## **1. Requerimientos funcionales.**

### **a. Generales.**

El usuario ingresará al sistema web con su correo electrónico y contraseña. El usuario navegará por la aplicación web y seleccionará los libros que podrían gustarle. La aplicación web le deberá mostrar recomendaciones las cuales serán generadas de forma automatizada y personalizada. El sistema deberá de contar con un área de administración en el que se administren los libros.

### **b. Recolección automatizada de la información.**

El sistema tendrá que recolectar los libros que los usuarios marcarán como “Guardar”.

### **c. Uso de los datos.**

El sistema tiene que usar datos recolectados para mostrar una serie de libros que puedan ser de interés para el usuario.

### **d. Tipo de recomendaciones.**

Se requiere que las recomendaciones sean personalizadas, basadas en el diseño de la portada de los libros, así como en el resumen de este.

### **e. Experiencia del usuario.**

Se necesita un diseño en el que los usuarios utilicen de manera sencilla la aplicación web. El usuario no tiene que darse cuenta de las acciones que están ocurriendo en el momento.

La retroalimentación explícita que se implementa permite que el usuario no tenga que estar realizando acciones extras para poder ver las recomendaciones. Las recomendaciones aparecerían y se actualizarían automáticamente.

## **2. Requerimientos no funcionales**

### **a. Requerimientos legales.**

Se requiere que el software web desarrollado esté bajo la licencia MIT.

### **b. Encriptación.**

Los datos sensibles deben estar encriptados.

## **D. Análisis**

Al analizar los requisitos, se establecieron módulos para segmentar los procesos del sistema web.

## **1. Módulo de usuarios.**

El módulo de usuario permite crear usuarios nuevos. También permite ingresar y verificar las credenciales para que el usuario pueda acceder a las áreas a las que tenga acceso. Los datos de las credenciales son:

- Correo
- Contraseña

## **2. Módulo de administración.**

El módulo de administración permite la administración de los perfiles de los dos tipos de usuarios:

- Usuario administrador
- Usuario

Permite:

- El cambio de correo y contraseña.
- La solicitud de datos almacenados del usuario que ha iniciado sesión.
- La eliminación de la cuenta y de todos los datos almacenados del usuario que ha iniciado sesión.
- La administración de las sesiones activas del usuario.

El usuario *administrador* tiene acceso a un tablero (dashboard) de administrador de los perfiles del libro en el cual se puede: agregar, editar y eliminar los perfiles de los libros.

## **3. Módulo de entretenimiento.**

Este módulo permite mostrarle al usuario las páginas de interacción donde se pueden ver los libros que se han ingresado al sistema.

- Permite mostrar recomendaciones
- Permite mostrar perfiles de libros.

## **4. Módulo de recolección de datos.**

Permite la recolección de datos de manera explícita. Capta y detecta los perfiles de los libros que el usuario haya guardado.

## **5. Módulo de recomendaciones.**

Permite obtener el id de los perfiles de los libros que el usuario haya visitado. Estos id's son enviados a un motor de recomendaciones que procesa, mediante técnicas de aprendizaje automático, los datos para obtener recomendaciones de libros. Permite procesar las palabras a valores numéricos.

## **E. Diseño**

En esta sección se presenta la descripción detallada del Sistema de Recomendaciones de Libros Basado en las Portadas y Resúmenes (SRLBPR). Se explica el diseño de la arquitectura creada, así como los elementos a tomar en cuenta y el por qué se tomaron en cuenta.

### **1. Alcance.**

El diseño y desarrollo del Sistema de Recomendaciones se limita a la creación del motor de recomendaciones, el proceso de retroalimentación explícita, la estructura de la base de datos y la aplicación web. El sistema está basado en la web por lo que también se le llama sistema web.

### **2. Arquitectura.**

Las páginas web o aplicaciones web que comercializan libros, ya sean digitales o físicos, tienen secciones las cuales le brindan al usuario una serie de sugerencias respecto a libros nuevos, en los cuales puedan estar interesados. Estas sugerencias comúnmente se basan en mostrar los libros más descargados, los más vistos, los más comentados, los más recientes, los más valorados, entre otros detalles. Sin embargo, al aplicar estos tipos de sugerencias están tratando a los libros como un producto, olvidando los detalles del contenido.

Con la finalidad de tener una solución alternativa y gratuita a los sistemas de recomendaciones de libros comunes, se ha desarrollado un sistema diferente, el cual toma en cuenta los detalles de diseño y expresión de la portada de los libros, así como los resúmenes de este.

Esta arquitectura ha sido creada tomando como base el enfoque Content-based Filtering o Filtrado Basado en Contenido. Este enfoque permite analizar el contenido de un elemento. En este caso el elemento es el libro y se ha definido como contenido a los detalles del diseño de la portada y el resumen.

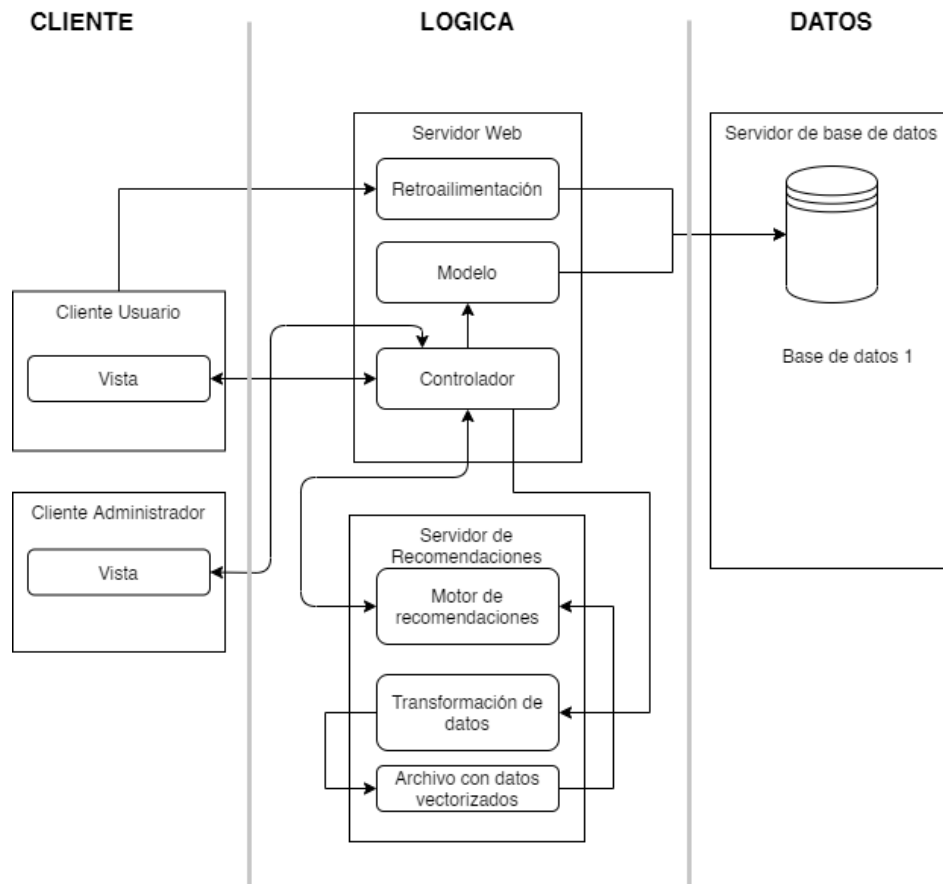


Figura 23 Estructura utilizada

El diagrama anterior muestra la arquitectura del SRLBPR, el cual está dividido en diferentes partes. El cliente web hace referencia a la aplicación web que utiliza el usuario para acceder al sistema y en el cual podrá visualizar los perfiles de los libros y las recomendaciones. Como la aplicación incluye un proceso llamado “Proceso de retroalimentación”, podrá capturar información relacionada a los gustos del usuario y almacenarla en la base datos.

El motor de las recomendaciones está separado del servidor web. Está instalado en su propio servidor al que se le llama “Servidor de Recomendaciones”. El Servidor de recomendaciones también convierte los datos, de los campos necesarios, en información procesable para los algoritmos.

La aplicación web es la encargada de mostrarle al usuario las recomendaciones generadas.

### 3. Proceso de retroalimentación.

El sistema necesita información para funcionar. Si el sistema no cuenta con datos para realizar relaciones sobre los posibles intereses de los usuarios no se podrán mostrar las recomendaciones personalizadas. Para obtener estos datos se ha implementado un proceso de retroalimentación.

El proceso de retroalimentación es el encargado de obtener la información del usuario a través del cliente usuario. Como fue detallado en el Marco Teórico, existen dos técnicas: retroalimentación explícita y retroalimentación implícita.

Para el desarrollo del sistema se ha tomado en cuenta la retroalimentación explícita, ya que el usuario tendrá que darle clic al botón “Guardar” que está presente en cada perfil del libro para guardar libros de interés.

#### 4. Motor de recomendaciones.

El motor de recomendaciones es el encargado de generar una serie de recomendaciones en base al id recibido. El motor toma el enfoque del Filtrado Basado en Contenido para generar las recomendaciones. Hace uso del Algoritmo de ordenamiento de distancias, el cual es una modificación del Algoritmo KNN, para obtener un listado de libros similares al que se esté evaluando. El motor se encuentra en el Servidor de Recomendaciones, el cual es gestionada por una API REST.

El controlador del Servidor Web decide aleatoriamente cuál de los dos tipos de recomendaciones va a mostrar.

- Recomendaciones según la portada, título y género  
En esta acción se generan 5 (valor de K) recomendaciones.
- Recomendaciones según el resumen  
En esta acción se generan 5 (valor de K) recomendaciones.

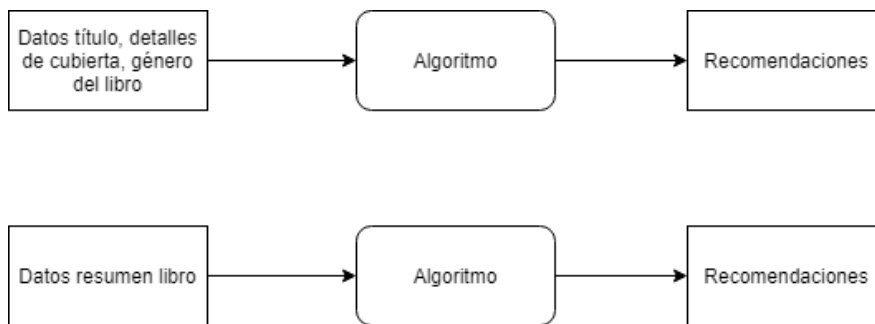


Figura 24 Diagrama de tipos de recomendaciones

##### a. Transformación de datos.

La información a tratar se concatenó para formar dos documentos por cada uno de los libros:

Documento1 = Título + Diseño de portada + Género

Documento2 = Resumen

Por ejemplo, si se tiene la siguiente información de un libro:



<b>Título</b>	<b>Autor</b>	<b>Género</b>	<b>Diseño de portada</b>	<b>Resumen del libro</b>
Mujercitas	Louisa May Alcott	Ficción, Novela, Comedia	personas, antiguo, familia, alegría	Jo es una chica que disfruta de escribir historias. Ella y sus hermanas pasan por momentos muy felices y tristes.

*Cuadro 8 Ejemplo con datos de un libro*

al concatenar los datos se tendría los siguientes documentos:

Documento1 = “Mujercitas personas, antiguo, familia, alegría Ficción, Novela, Comedia”

Al aplicarle el proceso de transformación, basado en el Modelo de la Bolsa de Palabras al Documento1, se tendría un vector similar a:

(0, 5)	1
(0, 7)	1
(0, 1)	1
(0, 3)	1
(0, 0)	1
(0, 4)	1
(0, 6)	1
(0, 2)	1

*Figura 25 Datos vectorizados 1*

Documento2 = “Jo es una chica que disfruta de escribir historias. Ella y sus hermanas pasan por momentos muy felices y tristes.”

Y al transformar el Documento2 se tendría un vector similar a:

(0, 9)	1
(0, 4)	1
(0, 17)	1
(0, 0)	1
(0, 14)	1
(0, 2)	1
(0, 1)	1
(0, 5)	1
(0, 8)	1
(0, 3)	1
(0, 15)	1
(0, 7)	1
(0, 12)	1
(0, 13)	1
(0, 10)	1
(0, 11)	1
(0, 6)	1
(0, 16)	1

Figura 26 Datos vectorizados 2

Este mismo proceso le ocurre a la información de cada libro. Los datos tokenizados que están relacionados a la portada, género y título del libro son almacenados en un archivo llamado vector-cover, y los datos tokenizados que están relacionados al resumen se almacenan en un archivo llamado vector-summary. De este modo se tendrían dos dataset que se usan como datos de entrenamiento (datatrain).

**b. Diagramas.**

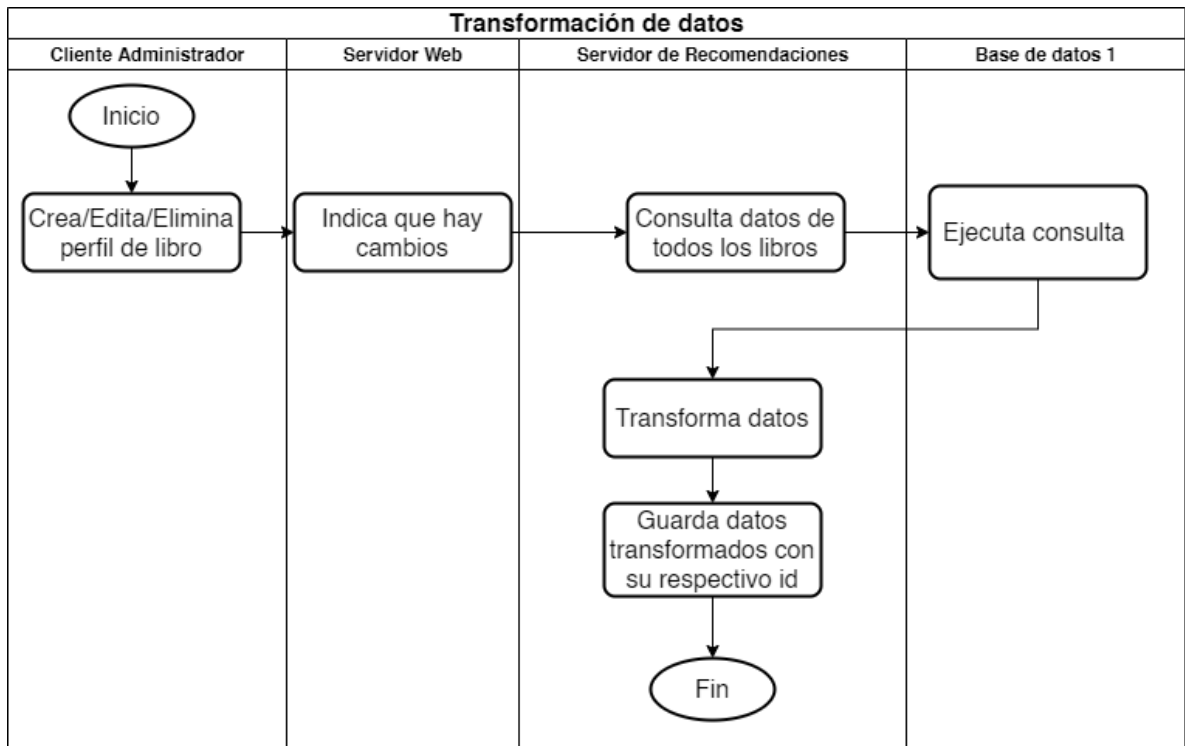


Figura 27 Diagrama de flujo de procesos de la transformación de datos

La transformación se realiza en el momento en el que se crea, edita o elimina un perfil de algún libro, y se guarda en el Servidor de Recomendaciones para ser utilizada posteriormente.

Al momento de realizar el proceso de generar recomendaciones ya no se necesita aplicar la transformación porque se tienen los datos transformados en el Servidor de Recomendaciones, optimizando así el tiempo de cómputo.

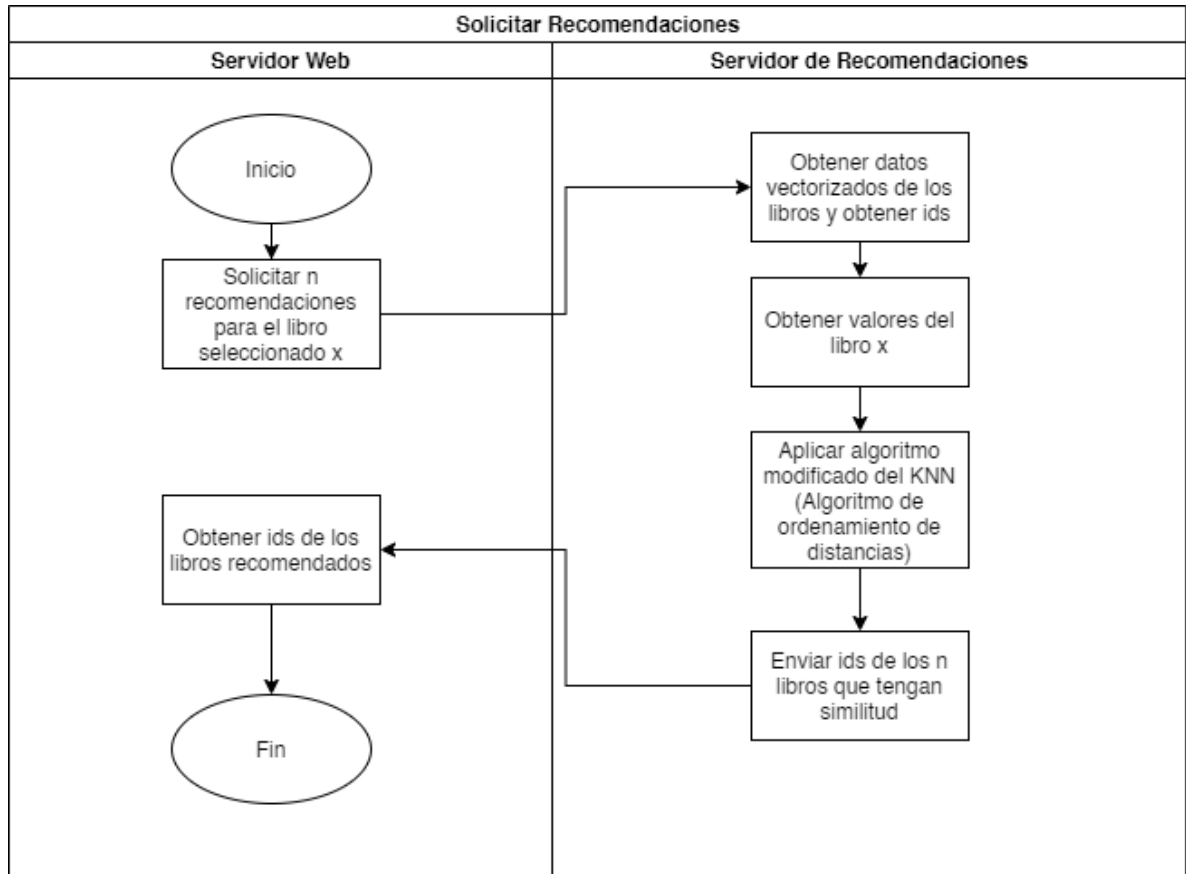


Figura 28 Diagrama de flujo de procesos de solicitud de recomendaciones

Las recomendaciones están basadas en el interés del usuario, por lo que cada vez que el usuario guarde un libro, este se almacena como interés del usuario.

El diagrama anterior muestra el proceso general de las recomendaciones, donde se observa la falta del proceso de transformación de datos porque ya se ha realizado.

## API REST

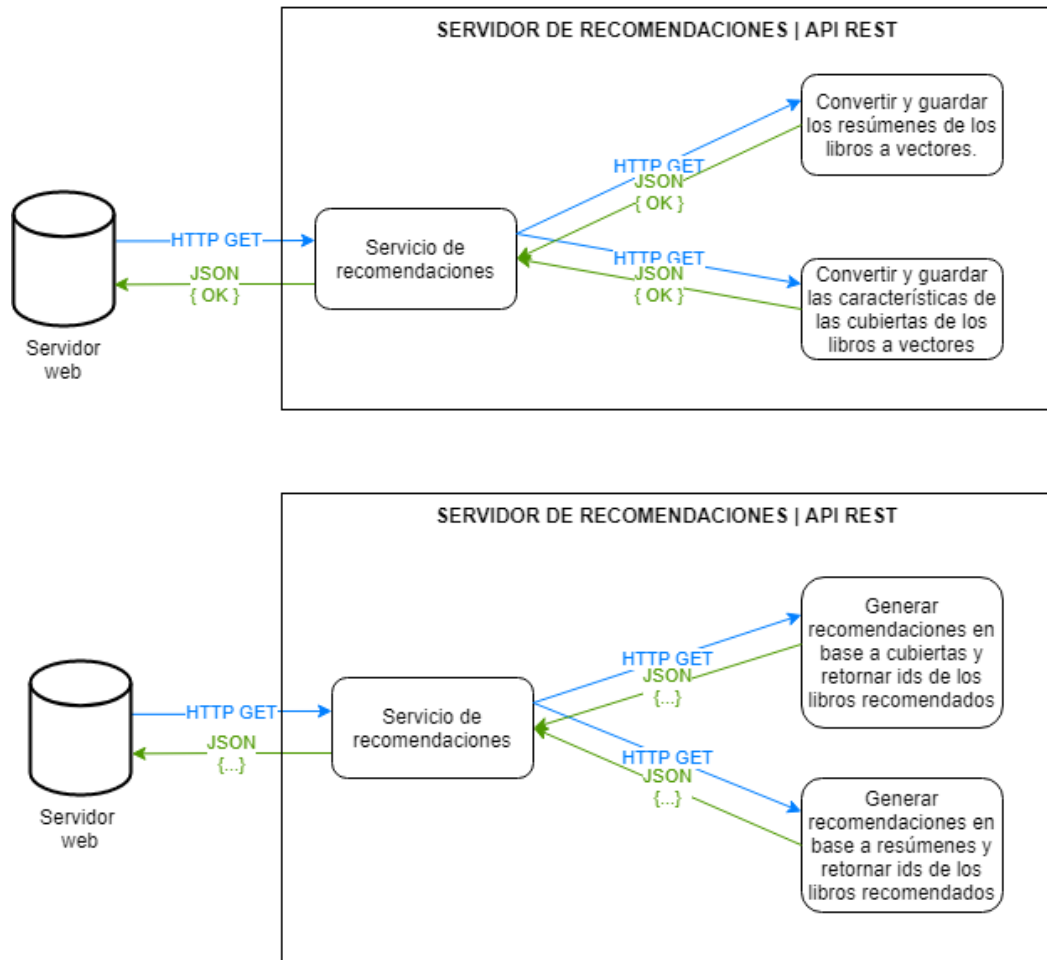


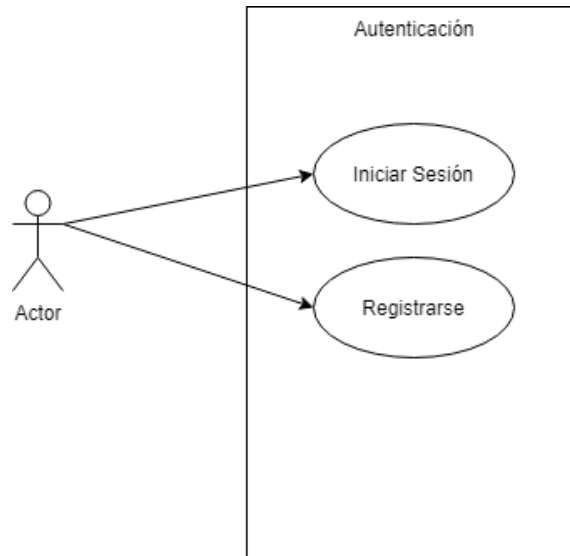
Figura 29 Diagrama de la API-REST

El sistema de recomendaciones es gestionado por una API REST la cual se estableció que sería desarrollada en Python.

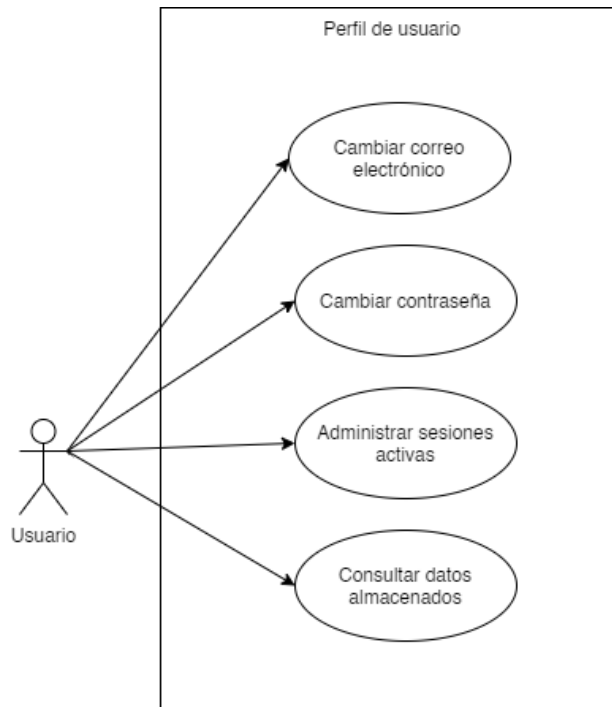
## 5. Diagrama de casos de uso.

El diagrama de casos de uso permite describir el comportamiento general entre el sistemas y usuarios u otros sistemas. Mediante estos diagramas resulta fácil entender cómo el usuario podría interactuar con el sistema y cómo el sistema responderá. A continuación, se presentan los diagramas de los módulos de la aplicación.

**a. Módulo de usuarios**

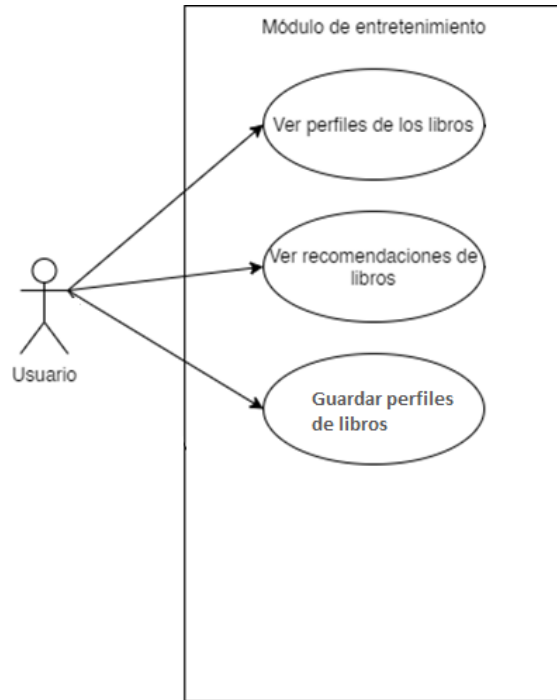


*Figura 30 Diagrama de casos de uso de autenticación*



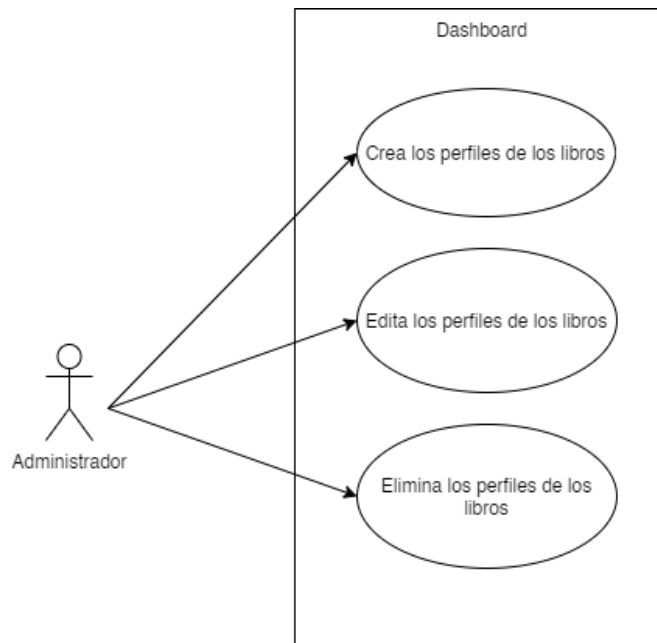
*Figura 31 Diagrama de caso de uso de perfil de usuario*

**b. Módulo de entretenimiento.**



*Figura 32 Diagrama de caso de uso de módulo de entretenimiento*

**c. Módulo de administración**



*Figura 33 Diagrama de caso de uso de dashboard*

## 6. Caso de uso detallado.

En esta sección se presentan las especificaciones detalladas de casos de uso. Esto es muy diferente al diagrama de casos de uso ya que se presenta con mayor detalle la interacción entre el sistema y el usuario.

### a. Usuarios.

#### 1) Autenticación con credenciales.

<b>Nombre</b>	Autenticación de usuarios
<b>Descripción</b>	Permite la validación del usuario para entrar a los distintos módulos.
<b>Actor(es)</b>	Usuario, Administrador, aplicación, base de datos
<b>Lógica del proceso</b>	<ol style="list-style-type: none"><li>1. El sistema muestra una página de inicio de sesión en la cual el usuario debe de ingresar sus credenciales (email, contraseña).</li><li>2. El sistema valida las credenciales del usuario.</li><li>3. El sistema permite el ingreso a los módulos permitidos en base a los roles del usuario.</li></ol>

*Cuadro 9 Caso de uso detallado de autenticación con credenciales*

### b. Módulo de Administración.

#### 1) Administración el perfil del usuario.

<b>Nombre</b>	Perfil del usuario
<b>Descripción</b>	Permite la modificación de datos del usuario.
<b>Actor(es)</b>	Usuario, administrador, aplicación, base de datos
<b>Lógica del proceso</b>	<ol style="list-style-type: none"><li>1. El usuario entra a su perfil.</li><li>2. El usuario realiza cambios en los datos de información personal.</li><li>3. Si el usuario se sitúa en la sección <b>Cambio de correo electrónico:</b><ol style="list-style-type: none"><li>3.1 La aplicación le muestra un formulario para cambiar el correo electrónico.</li></ol></li><li>4. Si el usuario se sitúa en la sección <b>cambio de contraseña:</b></li></ol>

	<p>4.1 La aplicación le muestra un formulario para cambiar la contraseña. En uno de los campos se le solicita la contraseña actual.</p> <p>4.2 Si la aplicación valida la contraseña actual:</p> <p>4.2.1 Se sustituye y almacena la nueva contraseña.</p> <p>4. Si el usuario se sitúa en la sección <b>Administrar sus sesiones:</b></p> <p>4.1 El usuario puede administrar sus sesiones activas.</p> <p>5. Si el usuario se sitúa en la sección <b>Eliminar cuentas:</b></p> <p>5.1 El usuario puede borrar su cuenta y todos sus datos almacenados.</p>
--	--

*Cuadro 10 Caso de uso detalla de administración de perfil de usuario*

## 2) Administración del “Dashboard”

<b>Nombre</b>	Dashboard
<b>Descripción</b>	Permite agregar, modificar o eliminar perfiles de los libros.
<b>Actor(es)</b>	Administrador, aplicación, base de datos
<b>Lógica del proceso</b>	<p>1. El administrador ingresa a la página del “Dashboard”.</p> <p>2. El administrador puede ver todos los perfiles de los libros almacenados</p> <p>3. Si selecciona la opción <b>Nuevo:</b></p> <p>3.1 La aplicación mostrará un formulario para la creación de un perfil de un libro.</p> <p>3.2 Se ingresan los datos.</p> <p>3.3 Si los datos ingresados son válidos, se almacenan.</p> <p>4. Si selecciona la opción <b>Eliminar</b> de un libro:</p> <p>4.1 La aplicación mostrará una alerta de confirmación.</p> <p>4.2 Si la alerta se confirma</p> <p>4.2.1 La aplicación elimina el perfil del libro seleccionado.</p>



	<p>5. Si selecciona la opción <b>Editar</b>:</p> <p>5.1 La aplicación mostrará un formulario para la edición de la información del libro:</p> <p>5.2 Si los datos ingresados son correctos:</p> <p>5.2.1 Se almacenan los datos.</p>
--	--

*Cuadro 11 Caso de uso detallado de Dashboard*

## c. Módulo de entretenimiento

### 1) Páginas principales de interacción

<b>Nombre</b>	Presentación de las vistas
<b>Descripción</b>	Permite la visualización de las páginas de interacción de los usuarios
<b>Actor(es)</b>	Usuario, aplicación, base de datos
<b>Lógica del proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a las páginas permitidas.</li> <li>2. La aplicación obtiene la información necesaria de la base de datos y se la muestra a los usuarios mediante las vistas diseñadas.</li> <li>3. El usuario interactúa con las páginas a las cuales tiene acceso.</li> <li>4. El usuario guarda los libros de interés.</li> </ol>

*Cuadro 12 Caso de uso detallado de páginas principales de interacción*

### 2) Recolección de datos

<b>Nombre</b>	Almacenamiento de datos de perfiles vistos
<b>Descripción</b>	Permite la recopilación de datos en base al tipo de usuario y de forma implícita.
<b>Actor(es)</b>	Usuario, aplicación, base de datos
<b>Lógica del proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario da clic en un libro para visualizar el perfil de este.</li> <li>2. La aplicación detecta el momento en el que el usuario da clic el botón guardar.</li> <li>3. Capta el id del libro visualizado y guarda en la base de datos qué usuario realizó la visualización, a qué hora y en qué fecha se realizó.</li> </ol>

*Cuadro 13 Caso de uso detallado de recolección de datos*

### 3) Recomendaciones

<b>Nombre</b>	Mostración de las recomendaciones al usuario
<b>Descripción</b>	Permite la visualización de parte del usuario de las recomendaciones.
<b>Actor(es)</b>	Usuario, aplicación, base de datos, servidor de recomendaciones
<b>Lógica del proceso</b>	<ol style="list-style-type: none"><li>1. El usuario ingresa a las áreas con las secciones de recomendaciones.</li><li>2. La aplicación detecta uno de los libros guardado por el usuario.</li><li>3. La aplicación manda una solicitud de recomendaciones al servidor de recomendaciones con el id del libro.</li><li>3. El motor de recomendaciones lo analiza y envía de vuelta una lista de ids de libros que le podrían gustar al usuario.</li><li>4. La aplicación obtiene de la base de datos la información de los libros que fueron generados por el servidor de recomendaciones.</li><li>5. La aplicación le muestra al usuario las recomendaciones</li></ol>

*Cuadro 14 Caso de uso detallado de recomendaciones*

Para el módulo de las recomendaciones se usó la estructura del SRLBPR planteada.

## 7. Base de datos.

Se estableció el uso de una base de datos relacional utilizando el gestor MariaDB.



Figura 34 Estructura de base de datos utilizada

## F. Programación

### 1. Servidor de Recomendaciones.

Se desarrolló una API REST en el lenguaje de programación Python 3.8.6 y se hizo uso del framework Flask 1.1.2.

Flask es un framework para el desarrollo de aplicaciones y servicios web haciendo uso de Python. Se utilizó este framework ya que el motor de recomendaciones necesita algoritmos y procesos que pueden trabajarse de forma sencilla en Python.

Se usaron las siguientes librerías matemáticas, de análisis de datos y de aprendizaje automático:

- Pandas 1.1.4
- Numpy 1.20.2
- Scikit-learn 0.23.2

### 2. Similitud entre libros.

Para la similitud entre libros se ha medido la distancia euclidiana entre los datos del libro de prueba y los datos de cada uno de los libros del dataset. La distancia euclidiana es una medida de distancia entre un par de muestras  $p$  y  $q$  en un espacio  $n$ -dimensional.

La distancia euclidiana entre dos puntos del espacio euclídeo  $n$ -dimensional se define por la *ecuación 5*.

Ejemplo del uso de las distancias:

Se tienen dos puntos:  $P = (1,0,0)$  y  $Q = (1,0,1)$  donde  $P$  es el punto del primer libro y  $Q$  es el punto del segundo libro.

La Distancia Euclidiana de estos puntos es:

$$d_E(P, Q) = \sqrt{(1-1)^2 + (0-0)^2 + (1-0)^2} = 1$$

*Ecuación 10 Ejemplo de distancia euclidiana*

En este caso  $d_E$  es la distancia que hay entre ambos libros. Mientras las distancias de los libros tiendan más a 0 más similares son.

Esta similitud es ordenada por el algoritmo KNN modificado, que en esta investigación se le ha dado el nombre de Algoritmo de Ordenamiento de Distancias.

### 3. API Rest.

Esta API utiliza cuatro rutas las cuales están divididas en dos grupos, el grupo de transformación y el grupo de generación de recomendaciones.

Las rutas utilizadas fueron las siguientes:

Rutas de transformación			
Método	Ruta	Datos de envío	Respuesta
GET	/transform-cover	apikey (clave privada)	[ OK ]
GET	/transform-summary	apikey (clave privada)	[OK]

*Cuadro 15 Rutas de transformación de la API REST*

Rutas de generación de recomendaciones			
Método	Ruta	Datos de envío	Respuesta
GET	/recommendation-cover	apikey (clave privada), data (id del libro a evaluar), k (cantidad de recomendaciones a obtener)	JSON {...} con ids de los k libros recomendados
GET	/recommendation-summary	apikey (clave privada), data (id del libro a evaluar), k (cantidad de recomendaciones a obtener)	JSON {...} con ids de los k libros recomendados

*Cuadro 16 Rutas de generación de recomendaciones de la API REST*

### 4. Aplicación.

Ya teniendo listo el Servidor de Recomendaciones se trabajó la aplicación del Servidor Web haciendo uso del framework de desarrollo de aplicaciones web “Laravel” en su versión 8 y PHP 7.4.

Laravel es un framework para desarrollar aplicaciones y servicios web con PHP. Es de código abierto y permite la implementación de distintas librerías que ayudan al programador a ahorrar tiempo en el desarrollo de sus proyectos. Está enfocada en la arquitectura de software MVC.

Laravel permite comunicarse con la base de datos por medio de Eloquent y QueryBuilder, permitiendo usar un mismo código para distintos gestores de base de datos, por lo que no es necesario escribir consultas en SQL, aunque sí se tiene que usar la misma lógica. Además, Eloquent usa el controlador PDO para evitar inyecciones SQL, aumentando la seguridad de las aplicaciones hechas en Laravel.

Se utilizaron transacciones de Laravel, permitiendo cortar de forma automática una inserción, edición o eliminación si ocurre algún tipo de error.

Para la protección de las contraseñas de los usuarios se ha utilizado el sistema de autenticación que utiliza este framework, permitiendo que estas sean encriptadas. Además, Laravel implementa un token CSRF por cada sesión de usuario ayudando a prevenir las falsificaciones de solicitudes entre sitios.

Para el almacenamiento de los datos de los usuarios y de los libros se utilizó el gestor de base de datos MariaDB.

MariaDB es un gestor de base de datos el cual está basado en MySQL, uno de los sistemas de administración de bases de datos relacionales más populares.

## G. Verificación

Una de las fases más importantes en el desarrollo del software es la verificación. En esta fase se verifica que el sistema cumpla con los requerimientos establecidos al principio del desarrollo. Los requerimientos tienen que estar completados en un cien por ciento.

Asegura la calidad del producto y satisface las necesidades del cliente. Además, se espera que brinde una mejor optimización comercial (reduciendo los costos de mantenimiento), confiabilidad y una excelente experiencia de usuario.

En esta fase se realizan también las pruebas de procesos. Los procesos del software tienen que funcionar como se espera.

### 1. Pruebas estáticas.

Se realizaron pruebas estáticas en las que se revisa la documentación referente al desarrollo del software. En estas se siguen los flujos de la aplicación para determinar si el funcionamiento es correcto.

### 2. Pruebas dinámicas.

Estas pruebas prueban las funciones de la aplicación o módulo construido desde la perspectiva del usuario final (con diferentes roles) para verificar si el software puede realizar el trabajo especificado. Se realizaron este tipo de pruebas para identificar errores en la ejecución de la aplicación. Se ejecutó siguiendo una serie de pasos ya establecidos.

#### a. Pruebas de casos.

Una prueba de casos es un conjunto de objetivos, condiciones y pasos que se tienen que ejecutar sobre el software para determinar si cumple con los requisitos establecidos. Se han realizado casos de prueba del sistema para validar que este funcione como se espera.

Nombre Identificador	Descripción	Precondición	Entradas	Pasos	Resultados esperados	Prioridad
CP_auth1	Crear usuario		Datos del usuario	<ol style="list-style-type: none"> <li>Clic en el botón de Registro</li> <li>Ingresar datos</li> </ol>	Inicia la sesión del usuario y retorna al usuario a la	Alta

				3. Dar clic en crear	pantalla principal	
<b>CP_auth2</b>	Crear usuario con campos vacíos		Sin datos	<ol style="list-style-type: none"> <li>1. Clic en el botón de registro.</li> <li>2. Dejar los campos vacíos</li> <li>3. Dar clic en crear</li> </ol>	Retorna un error de validación de campos.	Media
<b>CP_auth3</b>	Iniciar sesión	El usuario debe de estar registrado.	Credenciales "correo y contraseña".	<ol style="list-style-type: none"> <li>1. Clic en el botón de inicio de sesión.</li> <li>2. Ingresar credenciales.</li> <li>3. Dar clic en iniciar.</li> </ol>	Retorna al usuario a la pantalla principal.	Alta
<b>CP_auth3</b>	Iniciar sesión con campos vacíos		Sin datos	<ol style="list-style-type: none"> <li>1. Clic en el botón de inicio de sesión.</li> <li>2. Dejar los campos vacíos.</li> <li>3. Dar clic en iniciar.</li> </ol>	Retorna un error de validación de campos.	Media
<b>CP_auth4</b>	Detectar cuenta de administrador	Debe de existir la cuenta de administración.	Credenciales	<ol style="list-style-type: none"> <li>1. Clic en el botón de inicio de sesión.</li> <li>2. Ingresar credenciales de</li> </ol>	Redirige al administrador al dashboard de administración.	Alta

				administrador. 3. Dar clic en iniciar.		
<b>CP_dash1</b>	El Dashboard de administración únicamente es visible para el administrador.	Un usuario debió haber iniciado sesión.		1. Escribir "localhost:8000/dashboard".	Redirección a la pantalla principal del usuario.	Alta
<b>CP_dash2</b>	Creación de perfil de los libros	El administrador tiene que estar logueado.	Datos del libro.	1. Ingresar a la sección de "agregar nuevo libro". 2. Ingresar los datos. 3. Clic en crear.	Redirección a dashboard con un mensaje de éxito.	Alta
<b>CP_dash3</b>	Creación de perfil de los libros con campos vacíos.	El administrador tiene que estar logueado.	Sin datos.	1. Dar clic en el libro a editar. 2. Borrar todos los campos permitidos. 3. Dar clic en editar.	Retornar mensaje de error de validación de campos.	Media
<b>CP_dash4</b>	Eliminación del perfil de los libros	El administrador tiene que estar logueado.		1. Dar clic en el libro a eliminar. 2. Dar clic en el botón de eliminar.	La página se actualiza y el libro ya no aparece en	Alto



				3. Aceptar el mensaje de confirmación	las búsquedas.	
<b>CP_home1</b>	Página en la que el usuario interactúa para encontrar libros.			1. Entrar a "localhost:8000"	La página muestra los libros que se pueden descargar.	Medio
<b>CP_recom1</b>	La página muestra recomendaciones a usuario logueados.	El usuario tiene que estar logueado.		1. Entrar a "localhost:8000/profile/1"	La página muestra libros en base a las recomendaciones.	Alto
<b>CP_authp</b>	Los usuarios pueden administrar sus perfiles.	El usuario tiene que estar logueado		1. Clic en el nombre de usuario ubicado en el NavBar. 2. Probar opciones de administración de perfil.	La página permite hacer los cambios correspondientes.	Alto

Cuadro 17 Prueba de casos utilizada

## H. Aspecto legal

El uso de este software se rige bajo la licencia MIT.

Licencia original:

Copyright 2021 Marvin Josué Quiñónez Xitumul

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Traducción no oficial en español:

Copyright 2021 Marvin Josué Quiñónez Xitumul

<<Por la presente se concede permiso, libre de cargos, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados (el "Software"), a utilizar el Software sin restricción, incluyendo sin limitación los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias del Software, y a permitir a las personas a las que se les proporcione el Software a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "COMO ESTA", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO PERO NO LIMITADO A GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR E INCUMPLIMIENTO. EN NINGÚN CASO LOS AUTORES O PROPIETARIOS DE LOS DERECHOS DE AUTOR SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS

U OTRAS RESPONSABILIDADES, YA SEA EN UNA ACCIÓN DE CONTRATO, AGRAVIO O CUALQUIER OTRO MOTIVO, DERIVADAS DE, FUERA DE O EN CONEXIÓN CON EL SOFTWARE O SU USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE.>>

Wikipedia.

## **I. Código fuente**

El código fuente del proyecto se encuentra alojado en el siguiente repositorio de GitHub:

<https://github.com/marvinhere/bookrecommendation>

## VIII. Evaluación de las recomendaciones

### A. Evaluación matemática

Para la evaluación matemática del Sistema de Recomendaciones se ha realizado una prueba la cual trató en seleccionar el cincuenta por ciento de los libros de un grupo total de cuarenta y dos, para obtener cinco recomendaciones por cada uno de ellos. A estos datos se le aplicó la métrica de Precisión para determinar la precisión de las recomendaciones respecto al libro seleccionado. La fórmula utilizada quedó de la siguiente manera:

$$\text{Precisión} = \frac{|\text{Recomendaciones funcionales}|}{\text{Recomendaciones obtenidas}} = \frac{|\text{Recomendaciones funcionales}|}{5}$$

*Ecuación 11 Precisión utilizada*

Donde las recomendaciones funcionales son las recomendaciones aceptadas como buenas por el evaluador, en este caso por el autor de este trabajo.

Las recomendaciones aceptadas se basan en la similitud observable de cada libro en base a las etiquetas utilizadas para describir a las portadas de los libros o el contenido escrito del libro. En estas recomendaciones se incluyen el mismo libro evaluado por razones que se explican en el capítulo IV. MARCO TEÓRICO.

Para determinar el estado de una precisión se utilizaron los siguientes datos:

*Precisión  $\Rightarrow$  0.5 la respuesta es "positiva"*

*Precisión  $<$  0.5 la respuesta es "negativa"*

*Figura 35 Estado de una precisión*

### B. Prueba de campo (A/B Testing)

Para identificar el impacto del sistema de recomendaciones en el usuario se utilizó la metodología A/B Testing. Esta es una metodología para la investigación de experiencias de usuarios la cual consiste en mostrar de forma aleatoria dos versiones de un mismo elemento, A y B, para saber cuál funciona mejor.

Para Kohavi y Longbotham (2017:3-4), la configuración experimental más simple es evaluar un factor con dos niveles, versión A (control) y versión B (tratamiento). Normalmente el control es la versión básica y el tratamiento es el cambio que se está probando.

Para esta prueba se lanzó en el municipio de Panajachel, Sololá, Guatemala la aplicación web Relibro, la cual hace uso del sistema desarrollado, a través del link <http://relibro.xiflat.com>, en la cual usuarios

exploraron la plataforma mientras el sistema recogía información que ayudó a identificar la interacción de los usuarios con el sistema de recomendaciones.

Nota: La credencial de administración predeterminada del sistema es: correo: admin@admin.com y contraseña: GradGua2016.

En el área “Recomendados” se les mostraba a los usuarios dos tipos de recomendaciones al azar. El primer tipo (Versión A) mostraba cinco libros, también al azar, cada vez que abrían el perfil de un libro. El segundo tipo (Versión B) mostraba cinco recomendaciones generadas por el motor de recomendaciones. Cuando el usuario interactuaba con los libros del área Recomendados el sistema guardaba el tipo de recomendación que sedujo al usuario.

Este sistema se montó en recursos en la nube pertenecientes a Microsoft Azure haciendo uso de los siguientes servicios:

SERVICE NAME	SERVICE RESOURCE
Virtual Machines	D2 v3/D2s v3
Storage	S10 Disks
Virtual Network	Static Public IP
Bandwidth	Data Transfer Out
Bandwidth	Data Transfer Out - NAM or EU To Any
Bandwidth	Data Transfer In
Bandwidth	Data Transfer Out - Free
Storage	Disk Operations - Free
Virtual Network	Dynamic Public IP - Free

Figura 36 Recursos utilizados

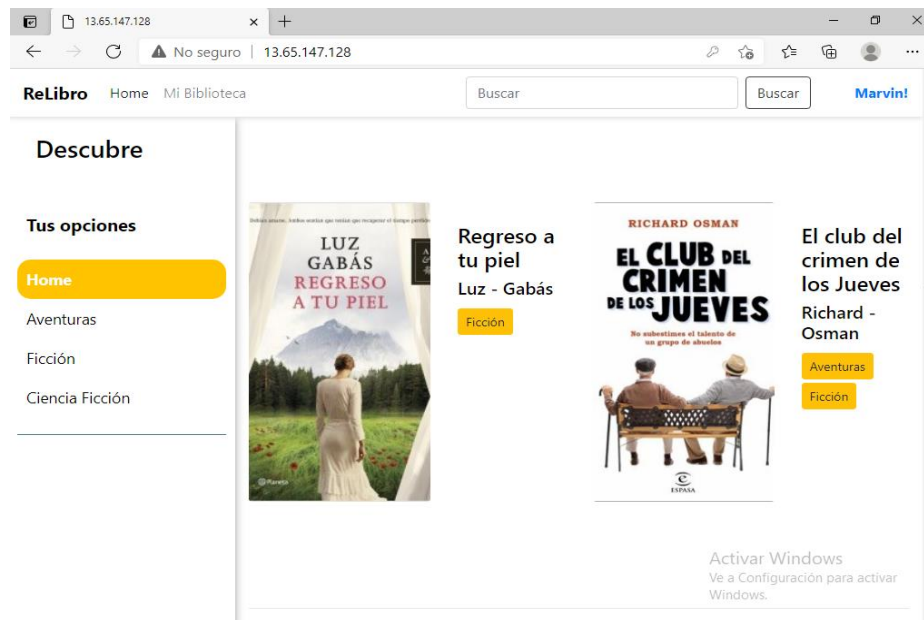


Figura 37 Captura de pantalla de ReLibro

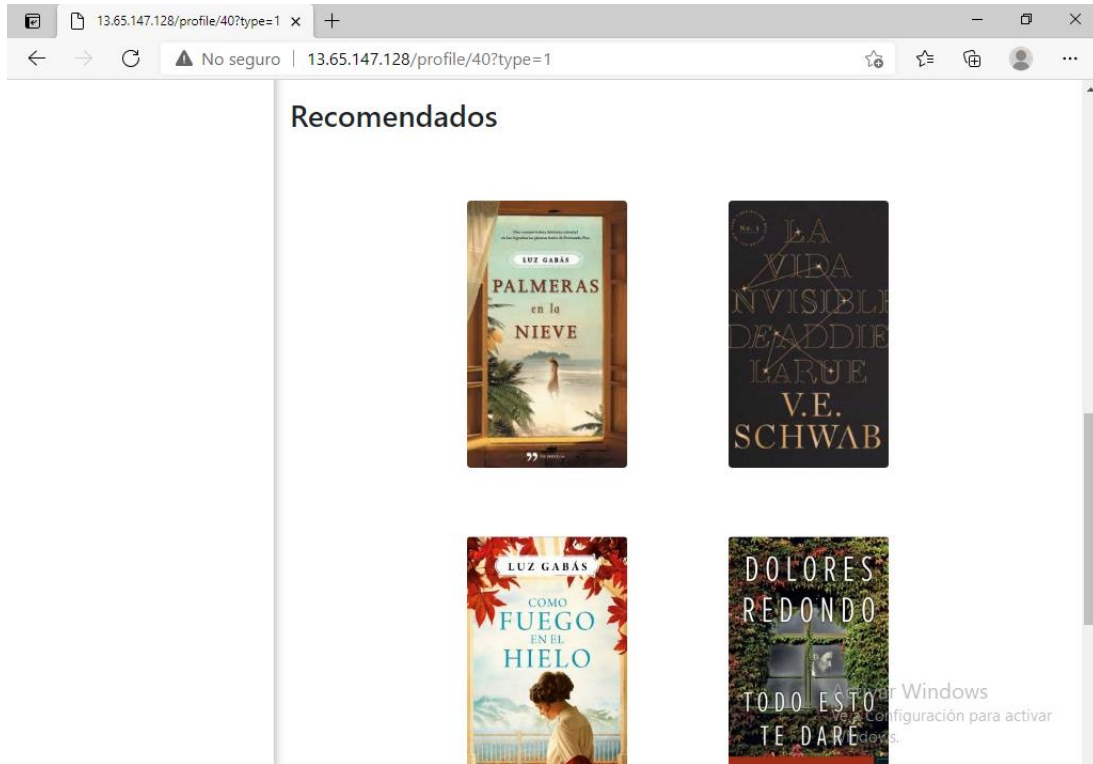


Figura 38 Captura de pantalla de sección recomendados de Relibro

## IX. Resultados y análisis

En este capítulo se presenta los resultados obtenidos durante el proceso de análisis de resultados de las pruebas realizadas sobre el Sistema de Recomendaciones de Libros Basado en las Portadas y Resúmenes (SRLBPR).

En la fase de pruebas se realizaron pruebas estáticas y pruebas dinámicas para comprobar la funcionalidad y calidad del software. Estas permitieron determinar el cumplimiento de los requisitos.

### A. Pruebas estáticas

En las pruebas estáticas se realizaron análisis estáticos de código para analizar y verificar la lógica de este. Se realizó con el objetivo de determinar que la lógica de los códigos cumpla con lo establecido en la sección de diseño. Se establecieron formularios en el que se valida si la acción coincide con lo diseñado.

Los siguientes cuadros muestran los resultados obtenidos de las pruebas.

Nombre general del proceso	Cumple con lo solicitado en el diseño (Sí/No)	Número de revisiones	Observaciones
Proceso de autenticación	Sí	2	
Proceso de creación de perfiles de libros	Sí	3	
Proceso de edición de perfiles de libros	Sí	2	
Proceso de eliminación de perfiles de libros	Sí	1	
Proceso de retroalimentación explícita	Sí	2	
Proceso de generación de recomendaciones	Sí	10	

*Cuadro 18 Resultado de pruebas estáticas*

## B. Prueba de casos

La prueba de casos ayudó a determinar el funcionamiento de cada área del programa a nivel usuario. Se hicieron pruebas en las que se utilizaba el programa como si de un usuario se tratase.

<b>Nombre identificador</b>	<b>Descripción</b>	<b>Estado</b>
CP_auth1	Crear usuario	Funcionando
CP_auth2	Crear usuario con campos vacíos	Funcionando
CP_auth3	Iniciar Sesión	Funcionando
CP_auth3	Iniciar Sesión con campos vacíos	Funcionando
CP_auth4	Detectar cuenta de administrador	Funcionando
CP_dash1	El Dashboard de administración únicamente es visible para el administrador.	Funcionando
CP_dash2	Creación de perfil de los libros	Funcionando
CP_dash3	Creación de perfil de los libros con campos vacíos.	Funcionando
CP_dash4	Eliminación de perfil de los libros	Funcionando
CP_home1	Página en la que el usuario interactúa para encontrar libros.	Funcionando
CP_recom1	La página muestra recomendaciones a usuario logueados.	Funcionando
CP_authp	Los usuarios pueden administrar sus perfiles.	Funcionando

*Cuadro 19 Resultados de la prueba de casos*



La prueba de casos logró identificar el comportamiento del sistema en cuanto al ingreso de datos incorrectos.

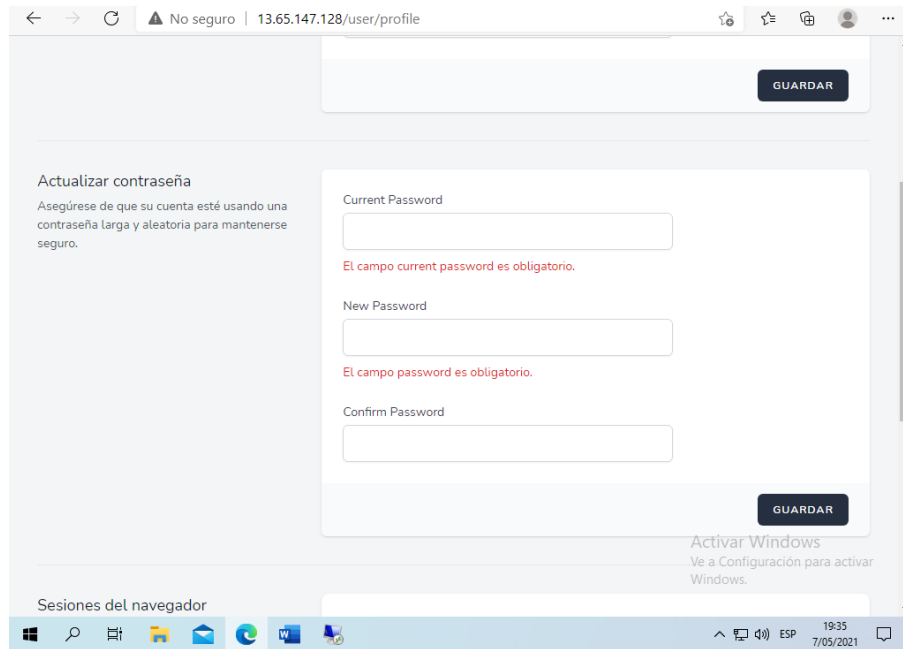


Figura 39 Prueba de casos, alerta por error

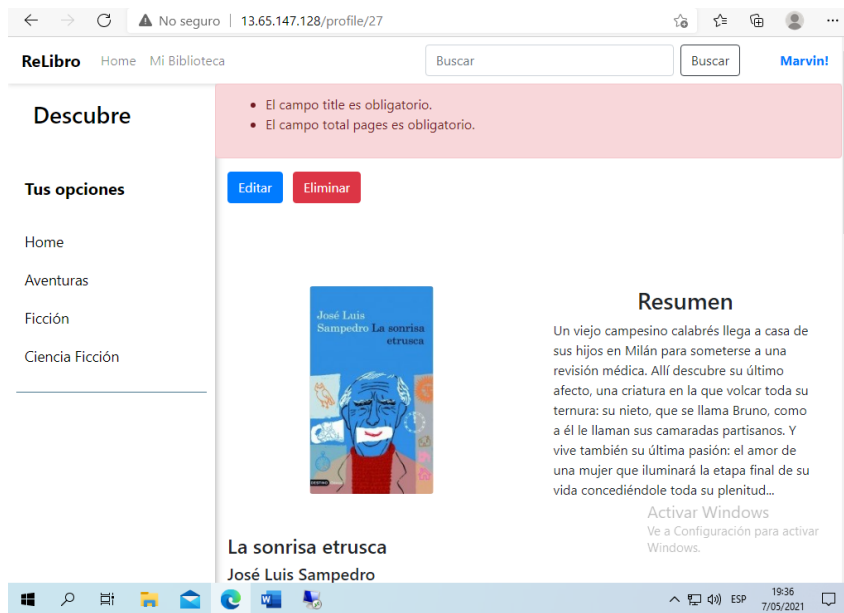


Figura 40 Alerta por detección de campos obligatorios

También identifiqué el comportamiento del sistema en cuanto a la seguridad de los accesos de este. En un caso el sistema pedía la confirmación por contraseña para validar la acción del usuario. En otros el sistema únicamente mostraba las opciones indicadas para cada tipo de usuario.

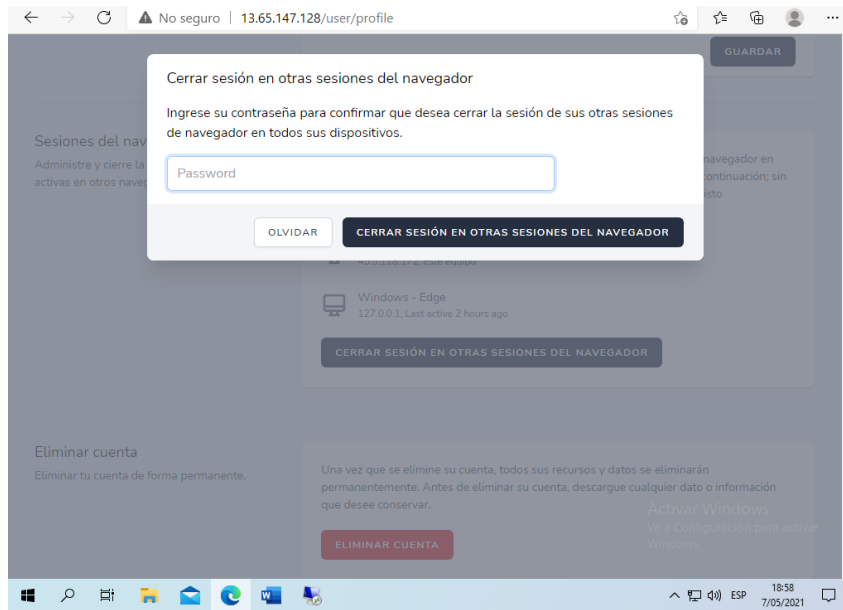


Figura 41 Confirmación de acción mediante contraseña

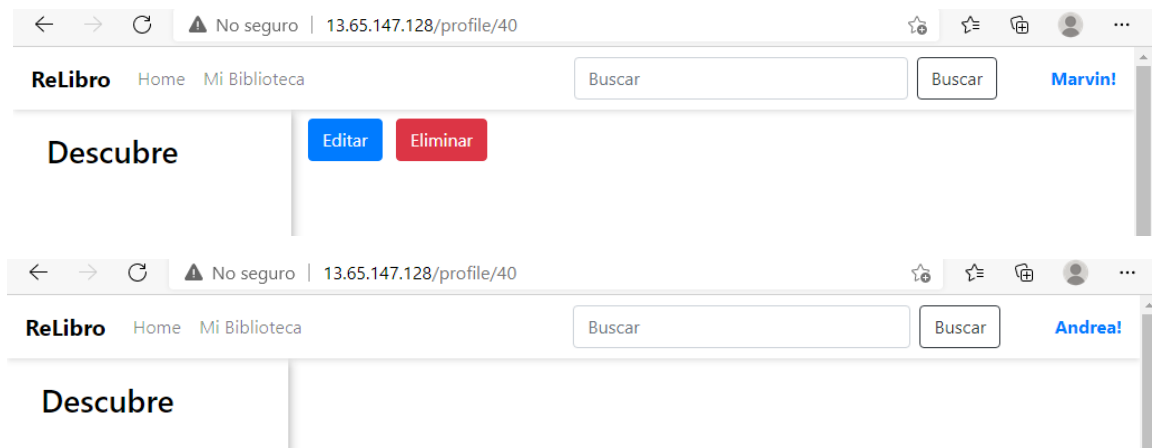


Figura 42 Aparición de opciones dependiendo el tipo de usuario

## C. Evaluación de las recomendaciones

### 1. Evaluación matemática.

Según la portada del libro					
Nombre del libro	Autor	Recomendaciones obtenidas	Recomendaciones funcionales	Precisión	Respuesta
<b>El libro de Gianni Rodari</b>	Gianni Rodari	5	5	1	Positiva
<b>La voz dormida</b>	Dulce Chacón	5	3	0.6	Positiva
<b>El nombre del viento</b>	Patrick Rothfuss	5	4	0.8	Positiva
<b>Misery</b>	Stephen King	5	3	0.6	Positiva
<b>Monte Sinaí</b>	José Luis Sampedro	5	4	0.8	Positiva
<b>La suerte del enano</b>	Cesa Pérez Gallida	5	2	0.4	Negativa
<b>Spin Nulo</b>	Juan Vicente	5	0	0	Negativa
<b>Mientras Escribo</b>	Stepehn King	5	4	0.8	Positiva
<b>Adiós al frío</b>	Elvira Sastre	5	5	1	Positiva
<b>Dime quién soy</b>	Julia Navarro	5	0	0	Negativa
<b>Diario de una mujer muerta y otros cuentos</b>	Dulce Chacón	5	2	0.4	Negativa
<b>Sapiens: De animales a Dioses</b>	Yuval Noah Harari	5	3	0.6	Positiva
<b>El caballo desnudo</b>	José Luis Sampedro	5	4	0.8	Positiva
<b>Décimo tercer viaje: Las siete pruebas del reino de las fantasías</b>	Geronimo Stilton	5	5	1	Positiva
<b>Regreso a tu piel</b>	Luz Gabás	5	1	0.2	Negativa
<b>Cicatriz</b>	Juan Gómez Jurado	5	4	0.8	Positiva
<b>V de Vendetta</b>	Alan Moore	5	3	0.6	Positiva
<b>El club del crimen de los Jueves</b>	Richard Osman	5	2	0.4	Negativa

Nombre del libro	Autor	Recomendaciones obtenidas	Recomendaciones funcionales	Precisión	Respuesta
<b>Todo lo que somos tú y tú y yo</b>	Martha Cillán	5	2	0.4	Negativa
<b>Palmeras en la nieve</b>	Luz Gabás	5	3	0.6	Positiva
<b>Congreso en Estocolmo</b>	José Luis Sampedro	5	4	0.8	Positiva
Promedio de precisión: 0.6					
Porcentaje de resultados positivos 66.66%					

*Cuadro 20 Evaluación matemática de las recomendaciones basadas en la portada*

<b>Según el resumen del libro</b>					
Nombre del libro	Autor	Recomendaciones obtenidas	Recomendaciones funcionales	Precisión	Respuesta
<b>Regreso a tu piel</b>	Luz Gabás	5	4	0.8	Positiva
<b>La milla verde</b>	Stephen King	5	5	1	Positiva
<b>Sherlock Holmes: El signo de los cuatro</b>	Sir Arthur Conan Doyle	5	5	1	Positiva
<b>Vestido de novia</b>	Pierra Lemaitre	5	5	1	Positiva
<b>Palmeras en la nieve</b>	Luz Gabás	5	4	0.8	Positiva
<b>La sonrisa etrusca</b>	José Luis Sampedro	5	3	0.6	Positiva
<b>La genealogía de la moral</b>	Friedrich Nietzsche	5	2	0.4	Negativa
<b>El discurso del método</b>	René Descartes	5	1	0.2	Negativa
<b>Gerra Mundial Z</b>	Max Brooks	5	2	0.4	Negativa
<b>Décimo tercer viaje: Las siete pruebas del reino de las fantasías</b>	Geronimo Stilton	5	3	0.6	Positiva
<b>Dune</b>	Frank Herbert	5	2	0.4	Negativa
<b>Las 150 vidas de Horacio Echevarrieta</b>	Javier Amezaga	5	3	0.6	Positiva
<b>Mientras Escribo</b>	Stephen King	5	2	0.4	Negativa
<b>Mortadelo y Filemón</b>	Francisco Ibáñez	5	1	0.2	Negativa
<b>Spin Nulo</b>	Juan Vicente	5	2	0.4	Negativa
<b>El guardián invicible</b>	Dolores Redondo	5	5	1	Positiva

Nombre del libro	Autor	Recomendaciones obtenidas	Recomendaciones funcionales	Precisión	Respuesta
Legado en los huesos	Dolores Redondo	5	4	0.8	Positiva
La voz dormida	Dulce Chacón	5	4	0.8	Positiva
El club del crimen de los Jueves	Richard Osman	5	5	1	Positiva
La leyenda de la peregrina	Carmen Posadas	5	3	0.6	Positiva
Adiós al frío	Elvira Sastre	5	4	0.8	Positiva
Promedio de precisión: 0.6571					
Porcentaje de resultados positivos: 67%					

*Cuadro 21 Evaluación matemática de las recomendaciones basadas en el resumen*

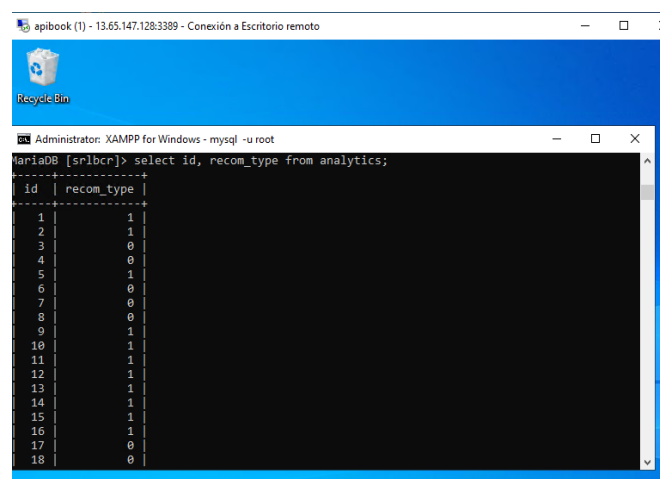
### a. Análisis.

La evaluación matemática a través de la métrica de Precisión determinó que el Sistema de Recomendaciones SRLBPR tiene un promedio de Precisión de 0.6 para las recomendaciones basadas en la portada y un promedio de Precisión de 0.6571 para las recomendaciones basadas en el resumen.

Para esta prueba se utilizó un dataset de 42 libros de los cuales 21 libros fueron seleccionados al azar para obtener la Precisión. En el transcurso de la evaluación se observó que la deficiencia de Precisión del 100% de las recomendaciones basadas en el resumen que dieron resultado negativo se debió a la cantidad baja de diversidad temática de los libros.

Se observó que al agregar más libros de temática variada y al agregar más características de las portadas la precisión suele aumentar generando mejores recomendaciones. Entre más datos existan las recomendaciones irán mejorando.

## 2. Prueba de campo.



*Figura 43 Datos obtenidos del A/B testing*

Se tuvieron en total 134 interacciones con los libros del área Recomendados.

Para la Versión A se utilizó el identificador 0 el cual indicaba que el usuario había dado clic en un libro que fue generado aleatoriamente. Los usuarios interactuaron 39 veces con estos libros.

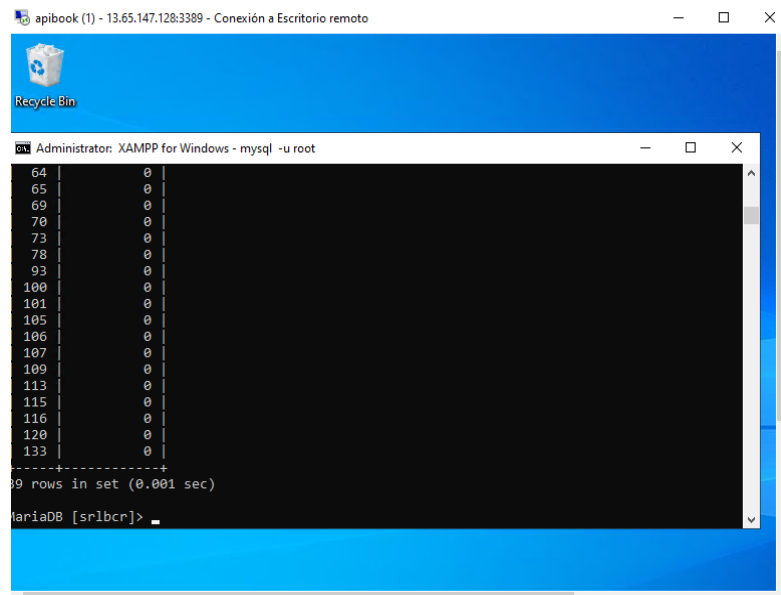


Figura 44 Datos de la versión A del A/B Testing

Para la Versión B se utilizó el identificador 1 el cual indicaba que el usuario había dado clic en un libro que fue generado por el sistema de recomendaciones desarrollado. Los usuarios interactuaron 95 veces con estos libros.

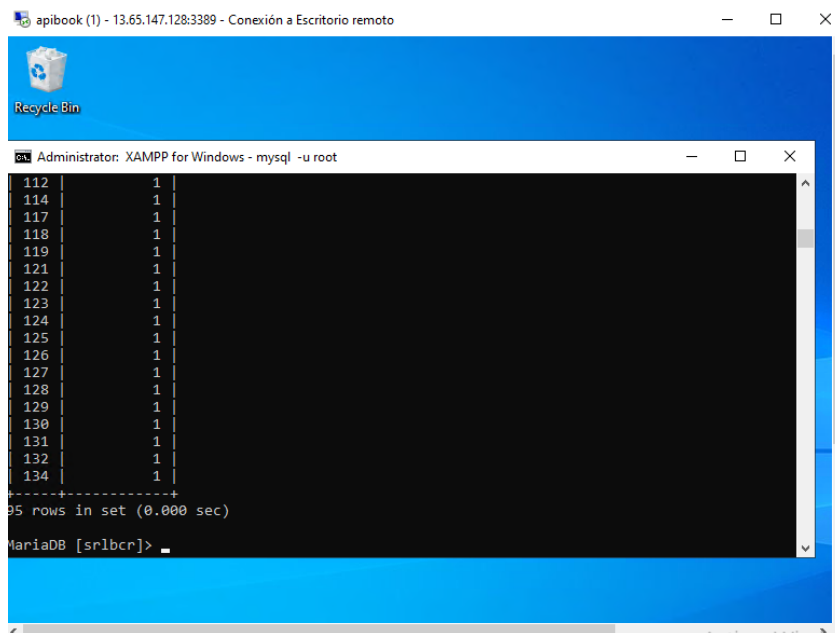


Figura 45 Datos de la versión B del A/B Testing



*Figura 46 Resultados del A/B Testing*

Al comparar el impacto que tuvieron las dos versiones se puede observar que los usuarios tuvieron una atracción mayor hacia las recomendaciones generadas por el sistema desarrollado.

## X. Conclusiones

Se ha logrado desarrollar un sistema de recomendación de libros, mediante el uso de frameworks para aplicaciones web, permitiendo que los usuarios descubran libros de interés. Los frameworks utilizados disminuyeron el tiempo de desarrollo considerablemente.

El establecimiento de requerimientos ayudó a identificar el tipo de sistema que se quería desarrollar. Las pruebas del sistema arrojaron resultados positivos, demostrando que los requerimientos planteados se cumplieron de forma correcta.

El desarrollo de un sistema web que ofrezca recomendaciones de libros, basándose principalmente en las características del diseño de la portada y el resumen ha demostrado ser eficiente para la generación de recomendaciones de libros. Las precisiones obtenidas y las observaciones confirman que este sistema cumple con su objetivo.

A nivel de usuario se pudo identificar por medio de la prueba A/B Testing que los usuarios se sintieron atraídos en un 71% hacia los libros de las recomendaciones generadas por el sistema desarrollado, mientras que las recomendadas generadas aleatoriamente tuvieron un impacto bajo (29%), confirmando el buen funcionamiento del sistema desarrollado en los usuarios.

Al identificar a una portada de un libro mediante palabras, que expresan los diseños de las portadas y lo que intentan expresar, ayudan a que el algoritmo utilizado pueda relacionar características que existen en otros libros, teniendo como resultado una serie de libros similares. La estructura de datos planteada permite que el sistema obtenga los resultados obtenidos en esta prueba.

El sistema de recomendaciones no genera recomendaciones en base al género de los libros por lo que puede existir un problema al momento de tener como usuarios a niños. Con este tipo de recomendaciones los niños podrían encontrar contenido no apto para su edad.

Los algoritmos de aprendizaje automático ayudan a automatizar procesos a partir de predicciones. El uso modificado del algoritmo K-NN (llamado en este trabajo como algoritmo de ordenamiento de distancias) para el análisis del contenido (portada y resumen) del libro, ayudó, de manera sencilla, a que el sistema estudie el historial del usuario para generar recomendaciones.

La retroalimentación explícita resultó ser una técnica ventajosa de la recopilación de datos. Permite obtener información brindada directamente por el usuario. La correcta implementación de este tipo de retroalimentación mejora la experiencia de usuario mientras alimenta con información las bases de datos.



El uso de técnicas de la ingeniería de software, en base al modelo “cascada”, ha permitido desarrollar un sistema web profesional. Permitió realizar un desarrollo lineal y ordenado que ayudó a determinar los detalles de la construcción del software y de las pruebas que se realizaron.

Las evaluaciones del software son una fase importante, ya que se valida el cumplimiento de los requerimientos especificados en el inicio del desarrollo, la adaptabilidad del diseño al sistema, la lógica de los algoritmos, el funcionamiento del software y la aceptabilidad de los modelos utilizados. La validación del modelo utilizado es una parte necesaria de la fase de pruebas, ya que verifica que el modelo usado es correcto para el conjunto de datos con el que se está trabajando.

La información no solo es un conjunto de datos que almacenar, sino que también, son activos que se estudian para generar información nueva. El uso de los datos para la generación de contenido personalizado seguirá en aumento en los próximos años, y las empresas, páginas, aplicaciones que no los utilicen tendrán una desventaja considerable respecto a los que si lo hagan.

## **XI. Recomendaciones**

El uso de palabras determinadas en este trabajo para la representación del diseño de la portada permite que el sistema de recomendaciones funcione y obtenga los resultados que se esperaron en este proyecto. Sin embargo, se recomienda una profundización sobre este tema para lograr detectar las palabras que mejor se adapten para la descripción de las portadas. El uso de un sistema, similar al de reCAPTCHA (propiedad de Google), para que varios usuarios de internet apoyen en la identificación de las palabras esenciales, podría beneficiar y mejorar el sistema de recomendaciones.

Se recomienda la realización de más casos de pruebas que ayuden a optimizar el algoritmo del sistema de recomendaciones. Mientras más estudio haya sobre el diseño implementado o la estructura de los datos utilizados, habrá mejores resultados.

La implementación de futuros algoritmos podría lograr que los resultados del sistema de recomendaciones de libros estudien mejor al usuario, lo que logrará generar un efecto más personalizado. El motor de recomendaciones se podría modificar para que agrupe los resultados por géneros.

El código de este sistema, y de sus diseños, están bajo la licencia MIT, la cual le permite a cualquier persona utilizar este software sin restricción, mientras se sigan las condiciones de uso. Se recomienda la participación y apoyo de parte de la comunidad de desarrolladores de software libre para seguir contribuyendo al desarrollo de este proyecto.

## XII. Bibliografía

- <<The MIT Licence>>. Open Source Initiative. Obtenido de Open Source Initiative:  
<https://opensource.org/licenses/MIT>
- <<Licencia MIT>>. Wikipedia. Obtenido de Wikipedia.com: [https://es.wikipedia.org/wiki/Licencia\\_MIT](https://es.wikipedia.org/wiki/Licencia_MIT)
- Adomavicius, G. y Tuzhilin, A. 2005. <<*Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions*>>. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6. págs 734-749.
- Boell, K. S. y Cecez-Kecmanovic, D. 2015. <<*What is an Information System?*>> 48th Hawaii International Conference on System Sciences, págs. 4959-4968.
- Manrique Sabogal, Winston. 2017. <<*¿Qué es un libro en el siglo XXI?*>> WMagazín.com. Obtenido de WMagazín.com: <http://wmagazin.com/relatos/quince-escritores-y-editores-latinoamericanos-participan-en-el-diccionario-que-es-un-libro-en-el-siglo-xxi/#aaamarif%20a9-boix-garc%20ada-arlene-carballo-y-manuela-espinal-solano>
- Caro Martínez, M. 2017. <<*Sistemas de recomendación basados en técnicas de predicción de enlaces para jueces en línea*>>E-Prints Complutense. Obtenido de E-Prints Complutense:  
<https://eprints.ucm.es/43975/1/Sistemas%20de%20Recomendaci%C3%B3n%20basados%20en%20t%C3%A9cnicas%20de%20predicci%C3%B3n%20de%20enlaces%20para%20jueces%20en%20l%C3%ADnea%20-%20Marta%20Caro%20Mart%C3%ADnez.pdf>
- Cuesta, E., Pérez, M.; Rietveldt de Arteaga, F. y Gutierrez, M. 2009. <<*Investigación tecnológica, una experiencia compartida*>>. Revista de Artes y Humanidades UNICA, 10 No. 3, 99.
- Fernandes, J. M. y Machado, R. J. 2016. <<*Requirements in Engineering Projects*>>. Springer International Publishing. Pág. 15.
- Herrera-Viedma, E.; Porcel, C. y Hidalgo, L. 2004. <<*Sistemas de recomendaciones: herramientas para el filtrado de información en Internet*>>. núm. 2, 2004. Obtenido de <http://www.hipertext.net>
- Kohavi, R. y Longbotham, R. 2017. <<*Online Controlled Experiments and AB Testing*>>. Págs. 3-4. Obtenido de ResearchGate:  
[https://www.researchgate.net/publication/316116834\\_Online\\_Controlled\\_Experiments\\_and\\_AB\\_Testing/link/59b7583b458515c212b3cd46/download](https://www.researchgate.net/publication/316116834_Online_Controlled_Experiments_and_AB_Testing/link/59b7583b458515c212b3cd46/download)
- Muñoz de Frutos, A. 2008. <<*Así es como funciona el sistema de recomendaciones de Netflix*>>. Business Insider [En español]. Obtenido de Business Insider: <https://www.businessinsider.es/como-funciona-sistema-recomendaciones-netflix-292001>
- Pressman, R. S. 2010. <<*Ingeniería del Software. Un enfoque práctico*>> 7ª ed. México, D. F. McGRAW-HILL INTERAMERICANA EDITORES, S. A. DE C.V.
- Sánchez Vigil, J. M.; Marcos Recio, J. C. y Villegas Tovar, R. 2008. <<*Las cubiertas de los libros como mecanismo de marketing editorial*>>. pág. 62. Ibersid.com. Obtenido de Ibersid.com:  
<https://ibersid.eu/ojs/index.php/ibersid/article/view/2204/1965>

Shani, G. y Gunawardana, A. 2009. <<*Evaluating Recommendation Systems*>>. Pág. 2. Obtenido de www.microsoft.com: <https://www.microsoft.com/en-us/research/publication/evaluating-recommender-systems/?from=http%3A%2F%2Fresearch.microsoft.com%2Fpubs%2F115396%2FevaluationmetFeva.tr.pdf>

Sommerville, I. 2005. <<*Ingeniería del Software*>> 7ª ed. Madrid: Pearson Educación, S. A.

Yeomans, M.; Shah, A. K.; Mullainathan, S. y Kleinberg, J. 2019. <<*Making Sense of Recommendations*>>. Págs. 5 -18. Obtenido de Scholars at HARVARD: <https://scholar.harvard.edu/files/sendhil/files/recommenders55.pdf>

## XIII. Glosario

**Algoritmo:** Conjunto de operaciones ordenadas que tienen un inicio y un fin.

**Arquitectura de sistema:** Estructura y herramientas físicas o digitales del sistema.

**Arquitectura de software:** Forma de construir, diseñar o separar un software.

**Cliente:** Parte del software con el que interactúa directamente el usuario. Persona a la que se le brinda un servicio.

**Código libre o código abierto:** Se le llama al código fuente de un software que es accesible, editado y distribuido por cualquier persona.

**Dataset:** Set de datos. Conjunto de datos.

**Datatest:** Dato o datos de prueba.

**Deep Learning o Aprendizaje Profundo:** Es una metodología de análisis de datos que se utiliza en la rama de la inteligencia artificial la cual utiliza redes neuronales artificiales que ayudan a tomar decisiones.

**Framework web:** Conjunto de herramientas estructuradas que ayudan al desarrollo de aplicaciones web.

**Gestor de base de datos:** Herramienta que gestiona los datos.

**Historial:** Antecedentes.

**HTTP request:** Modelo utilizado para enviar solicitudes y recibir respuestas.

**Id:** Identificador.

**Item:** Elemento de un grupo.

**Machine Learning o Aprendizaje Automático:** Es una metodología de análisis de datos que se utiliza en la rama de la inteligencia artificial para obtener respuestas en base a datos o a patrones de datos.

**Offline:** Fuera de línea.

**Online:** En línea.

**Portada:** Imagen que identifica a un elemento, en este caso a un libro.

**Rutas:** URL por el cual se accede para mandar u obtener información.

**Servidor:** Es un equipo lógico o físico que tiene como función almacenar y ejecutar archivos, programas u otros datos.

**Streaming:** Transmisión en continuo de datos de audio y video.

**Transacción:** En base de datos es un conjunto de órdenes las cuales pueden ser controladas para tomar acciones en caso de errores.