Pontificia Universidad
JAVERIANA
Cali

“DISEMINACIÓN SELECTIVA DE LA INFORMACIÓN USANDO CIENCIA DE DATOS: RECOMENDACIÓN DE LIBROS Y LECTURAS EN LAS BIBLIOTECAS COMFAMA”

Edwin José Bedoya Henao

Código 8986487

Proyecto Aplicado para optar al título de

Magister en Ciencia de Datos

Director(a)

GLORIA INÉS ÁLVAREZ VARGAS

Codirector(a)

DIEGO LUIS LINARES OSPINA

FACULTAD DE INGENIERÍA Y CIENCIAS

MAESTRÍA EN CIENCIA DE DATOS

SANTIAGO DE CALI, OCTUBRE 30 DE 2024

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTA DE GRÁFICAS

LISTA DE TABLAS

1 INTRODUCCIÓN

En la era de la información, las bibliotecas han evolucionado de ser simples espacios de préstamo de libros a convertirse en verdaderos centros de conocimiento, innovación y comunidad. La digitalización y el acceso a vastos volúmenes de datos han transformado la manera en que los usuarios interactúan con los contenidos bibliográficos. Este contexto plantea un desafío crítico para las bibliotecas públicas, como las de Comfama: ¿cómo ofrecer recomendaciones de lecturas relevantes y personalizadas que respondan a los intereses y necesidades de sus usuarios?

El objetivo principal de este proyecto fue desarrollar un sistema de recomendación de libros y lecturas basado en técnicas avanzadas de ciencia de datos. A través de modelos como Word2Vec y algoritmos de agrupamiento como K-Means, se analizaron patrones de lectura, preferencias individuales y comportamientos históricos de los usuarios, permitiendo la generación de recomendaciones personalizadas. Los resultados obtenidos demuestran que el sistema logra identificar de manera precisa intereses específicos, aumentando la satisfacción de los usuarios y fomentando un mayor uso de los recursos bibliotecarios.

El trabajo está estructurado en varias etapas, cada una contribuyendo al desarrollo y evaluación del sistema. Primero, se plantea el problema y se formulan preguntas clave que guían la investigación. Segundo, se definen los objetivos generales y específicos del proyecto, con un enfoque en la implementación y validación de modelos de aprendizaje automático. Tercero presenta el marco teórico y antecedentes relevantes, analizando enfoques previos en sistemas de recomendación y su aplicabilidad en bibliotecas.

Como cuarto se detalla la preparación y limpieza de datos, describiendo cómo se consolidó una base robusta de información a partir de historiales de préstamos y metadatos bibliográficos. Quinto se aborda el modelado del sistema, implementando técnicas de clustering y vectorización de textos. Sexto se evalúa el desempeño del modelo a través de métricas como el Silhouette Score y el Davies-Bouldin Index. Finalmente. Séptimo se describe la implementación de un sistema de recomendación visualmente interactivo y personalizado, seguido por un análisis detallado de los resultados, conclusiones y trabajos futuros

Entre los resultados destacados, se observa una aceptación promedio del 62% en las recomendaciones ofrecidas, con usuarios como el 1 y el 6 mostrando una alta afinidad hacia los títulos sugeridos (100% y 90% de aceptación, respectivamente). El sistema demostró ser especialmente eficaz en captar intereses específicos, como el gusto por novelas de aventuras (Usuario 1) y títulos infantiles (Usuarios 3 y 5), además de identificar géneros recurrentes como la autoayuda y la psicología (Usuario 2). La segmentación en 30,000 clústeres optimizó la relevancia de las sugerencias, superando la limitada diferenciación observada en pruebas iniciales con menos clústeres.

Más allá de su relevancia tecnológica, este proyecto tiene un impacto cultural significativo. En un entorno donde la lectura compite con otros medios digitales, un sistema de recomendación eficiente puede convertirse en un recurso clave para fomentar el hábito lector, facilitar el descubrimiento de nuevas obras y géneros, y reforzar el rol de las bibliotecas como agentes en la difusión del conocimiento.

En conclusión, el sistema de recomendación desarrollado para las bibliotecas de Comfama no solo aborda un desafío técnico, sino que también transforma la relación entre los usuarios y la lectura. Este enfoque mejora la personalización de los servicios bibliotecarios, fortalece el hábito lector y consolida el rol de las bibliotecas como agentes clave en la democratización del acceso al conocimiento y la cultura.

2 DEFINICIÓN DEL PROBLEMA

2.1 PLANTEAMIENTO DEL PROBLEMA

En el marco de la sociedad actual, caracterizada por un constante flujo de información y la diversificación de los medios de entretenimiento y comunicación, la promoción de la lectura y la oferta de recomendaciones personalizadas de libros se erigen como elementos importantes para fomentar el hábito de la lectura y satisfacer las expectativas de los usuarios. El propósito superior de Comfama es "consolidar y expandir la clase media trabajadora antioqueña para que sea consciente, libre, productiva y feliz" [1]; en este sentido, la persona a la que va dirigida el portafolio de servicios de la Caja es curiosa, busca alternativas constantemente, afina la mirada ante cualquier oportunidad, desea aprender conocimientos y habilidades nuevas y reflexiona sobre sus hábitos.

Además, "las bibliotecas Comfama son lugares abiertos para todos los públicos y, en su interés por ser espacios vivos que representen la magia única y particular de cada uno de los territorios que habitan, tienen usuarios, afiliados o no, que van desde la primera infancia hasta la vejez, que desean encontrarse con otros para compartir pasiones, refugiarse en la compañía de los libros y la conversación, asombrarse por una idea o experiencia novedosa, resolver sus inquietudes sobre el mundo, aprender a través del juego y descubrir diversas maneras de leerse a sí mismos y a todo lo que les rodea" [1].

Comfama, como institución comprometida con el bienestar y la difusión cultural, se encuentra ante el desafío de promover la cultura a través de sus bibliotecas, utilizando los libros y la lectura como medios fundamentales. Por eso es necesario un sistema de recomendación de libros y lecturas para ofrecer sugerencias más precisas y adaptadas a las preferencias individuales de sus usuarios.

Uno de los desafíos fundamentales radica en la complejidad inherente a las preferencias de lectura, un ámbito que puede ser abordado eficazmente mediante técnicas avanzadas de ciencia de datos. La diversidad de gustos, intereses y perfiles de usuarios dentro de la comunidad de Comfama demanda un enfoque sofisticado que pueda abordar la amplia gama de géneros, autores y temas de interés. Además, el no contar con un sistema de recomendación requiere una mejora significativa para superar las limitaciones de

adaptabilidad y respuesta en tiempo real, asegurando así una experiencia de usuario más enriquecedora. La integración de algoritmos de aprendizaje automático y técnicas de procesamiento de datos se convierte en un componente esencial para superar estas limitaciones. La presencia de una "larga cola" de libros menos populares, pero potencialmente valiosos, plantean desafíos algorítmicos que deben abordarse para garantizar la equidad en las recomendaciones y la consideración de la diversidad literaria.

2.2 **FORMULACIÓN DEL PROBLEMA**

¿Cómo aplicar técnicas avanzadas de ciencia de datos y aprendizaje automático para desarrollar un sistema de recomendación de libros y lecturas altamente preciso y personalizado, capaz de satisfacer las variadas preferencias de los usuarios de las bibliotecas de Comfama?

El proyecto de recomendación de libros y lecturas para las bibliotecas de Comfama se enfrenta a desafíos fundamentales que requieren una cuidadosa formulación del problema. Para abordar eficazmente estos desafíos, es esencial plantear preguntas clave que guiarán el desarrollo y la implementación del sistema de recomendación:

¿Qué tipos de datos específicos se requieren para la construcción del sistema de recomendación? ¿Cuáles son las fuentes principales de información para obtener historiales de préstamos, interacciones de usuarios con libros y preferencias declaradas? ¿Cómo se priorizan y seleccionan los textos y metadatos bibliográficos para el procesamiento y análisis? ¿Qué metodologías y algoritmos de aprendizaje automático se consideran más adecuados para la construcción del sistema de recomendación? ¿Cómo se diseña la interfaz para que sea fácil de usar y permita una interacción efectiva con las recomendaciones? ¿Cuáles son los criterios y métricas utilizados para validar la eficacia y la precisión de los modelos de aprendizaje automático?

3 OBJETIVOS DEL PROYECTO

3.1 OBJETIVO GENERAL

Desarrollar un sistema de recomendación de libros y lecturas para el servicio de préstamo de materiales bibliográficos en el Sistema de Bibliotecas de Comfama usando técnicas de aprendizaje automático.

3.2 OBJETIVOS ESPECÍFICOS

- Identificar y recolectar datos relevantes, incluyendo historiales de préstamos, interacciones de usuarios con libros y preferencias declaradas, priorizando el procesamiento de textos y metadatos bibliográficos.
- Diseñar e implementar un sistema de recomendación utilizando métodos y algoritmos de aprendizaje automático.
- Validar los modelos de aprendizaje automático utilizados en el sistema de recomendación.
- Desarrollar una interfaz intuitiva que facilite la interacción de los usuarios con el sistema de recomendación.

4 MARCO TEÓRICO Y ANTECEDENTES

4.1 MARCO TEÓRICO

Los sistemas de recomendación que a continuación serán explorados, desde los basados en contenido hasta los colaborativos y demográficos, ofrecen valiosas estrategias para comprender cómo optimizar la entrega de sugerencias acordes con las preferencias individuales de los usuarios, teniendo presente que debemos de contar con bases para poder evaluar estos sistemas de recomendación. Paralelamente, el concepto de perfiles de usuario se convierte en la base esencial, ya que, la recopilación y comprensión detallada de las características y comportamientos de los usuarios permitirá la creación de perfiles que sirvan como base sólida para el sistema de recomendación, facilitando así la generación de sugerencias personalizadas y adaptadas a las necesidades y gustos de cada individuo que frecuente las Bibliotecas de Comfama.

En este espacio se definen los términos que fueron considerados para el presente proyecto, esto con el fin de que todas las personas comprendan cada uno de los elementos y términos usados para entender las ideas que aquí se plantean.

3.1.1. Sistemas de recomendación:

Un sistema de recomendación es una herramienta que emplea criterios y evaluaciones basadas en los datos de los usuarios con el fin de predecir sugerencias valiosas o útiles para ellos. Estos sistemas evalúan datos proporcionados directa o indirectamente por los usuarios, analizando su historial para convertir esa información en recomendaciones útiles [2].

Hoy en día, los sistemas de recomendación son altamente eficientes, ya que pueden relacionar elementos de nuestros patrones de consumo, como compras previas, preferencias de contenido e incluso libros y lecturas más consultadas, para generar recomendaciones personalizadas [2].

En este proyecto la recomendación está enfocada en la búsqueda de libros y lecturas según los perfiles

de los usuarios que se encuentran registrados en las bases de datos de Comfama (Data lake y Software Alma). Esto con el objetivo de obtener listados de libros y lecturas que puedan ser útiles para los usuarios y suplan sus necesidades de información y lectura.

Para poder construir un sistema de recomendación, existen diversidad de técnicas y estrategias, ente ellas están:

Sistemas de recomendación basados en contenido: son ampliamente empleados en la actualidad [3], y se encuentran integrados, aunque como parte de sistemas más complejos, en plataformas en línea muy reconocidas y usadas como Youtube, Amazon, Facebook, Netflix, entre otras. Su objetivo principal radica en sugerir elementos del sistema a los usuarios según sus perfiles. Dichos perfiles reflejan, en cierta medida, las preferencias o intereses del usuario, definidos a partir de los elementos que el usuario mismo ha marcado como relevantes.

Estos modelos usan características de los ítems para generar recomendaciones. Cada ítem tiene una serie de atributos que se comparan con los intereses del usuario para hacer sugerencias. Técnicamente, el modelo más común en esta estrategia es el TF-IDF (Term Frequency-Inverse Document Frequency), que calcula la relevancia de términos para predecir preferencias. Las recomendaciones se basan en la similitud entre el perfil del usuario y el contenido de los ítems, calculada mediante métricas como el cosine similarity.

- **Funcionamiento técnico:** En estos sistemas, el perfil de un usuario es modelado mediante un vector ponderado, basado en las características de los ítems que ha preferido en el pasado. La similaridad entre el vector de un ítem nuevo y el vector del usuario determina si el ítem será recomendado.
- **Modelo matemático:** El cálculo de similitud, usualmente, se lleva a cabo mediante la distancia coseno o la distancia euclidiana, ambas métricas miden qué tan similares son dos vectores en un espacio multidimensional.
- **Métricas de desempeño:** Las métricas de evaluación más comunes incluyen precisión, recall y F1-score, que se calculan observando cuántas de las recomendaciones son realmente relevantes para el usuario.

Sistemas de recomendación basados en conocimiento: aprovechan un área específica de información para generar recomendaciones al estimar las necesidades del usuario y vincularlas con posibles soluciones [4]. Este enfoque es valioso cuando se trata de artículos poco frecuentes, como bienes raíces, vehículos, solicitudes de turismo, servicios financieros o productos de lujo costosos. En estos casos, la escasez de calificaciones disponibles dificulta el proceso de recomendación, ya que estos artículos se adquieren con poca frecuencia y ofrecen diversas opciones detalladas, lo que complica obtener suficientes calificaciones para una combinación específica del artículo.

Utilizan reglas explícitas para vincular la información del usuario con posibles soluciones. Los modelos de rule-based reasoning (RBR) suelen ser utilizados aquí, basando sus recomendaciones en un conjunto de reglas previamente definidas.

- **Funcionamiento técnico:** Se construyen reglas que permiten asociar características del usuario con ítems específicos. Por ejemplo, un usuario que vive en una ciudad con clima cálido podría recibir recomendaciones de libros sobre jardinería tropical.
- **Modelo matemático:** El enfoque RBR se basa en lógica condicional, con reglas "si-entonces" que determinan el proceso de recomendación.
- **Métricas de desempeño:** Al igual que otros sistemas de recomendación, el recall es esencial aquí para medir cuántas recomendaciones son relevantes respecto a todas las opciones posibles.

Sistemas de recomendación basados en filtrado: emplean la colaboración de las calificaciones otorgadas por múltiples usuarios para generar recomendaciones. El mayor desafío al diseñar estos métodos radica en la dispersión de las matrices subyacentes de calificaciones [4], consideremos una aplicación de libros digitales donde los usuarios dan calificaciones para expresar su preferencia por libros específicos. Dado que la mayoría de los usuarios solo ven una pequeña parte del amplio conjunto de libros disponibles, la mayoría de las calificaciones quedan sin especificar. Y es aquí donde está la idea de este modelo, el cual radica en validar los gustos similares de los usuarios. Esta similitud se puede usar para hacer inferencias sobre valores incompletamente especificados. La mayoría de los modelos para el filtrado colaborativo se enfocan en aprovechar las correlaciones entre ítems o usuarios para el proceso de predicción [4].

Este enfoque utiliza interacciones de múltiples usuarios para generar recomendaciones. El filtrado colaborativo puede implementarse mediante dos técnicas principales:

- **Filtrado colaborativo basado en usuarios:** Usa la similitud entre usuarios para recomendar ítems.
- **Filtrado colaborativo basado en ítems:** Calcula la similitud entre ítems en función de las calificaciones otorgadas por diferentes usuarios.
- **Funcionamiento técnico:** La similitud se calcula utilizando correlación de Pearson o distancia coseno entre vectores de usuarios o ítems.
- **Modelo matemático:** El modelo matemático detrás del filtrado colaborativo es generalmente un sistema matricial, en el que se calculan las similitudes entre vectores de calificaciones.
- **Métricas de desempeño: Mean Squared Error (MSE) y Root Mean Squared Error (RMSE)** son comunes para evaluar la precisión de las predicciones.

Sistemas de recomendación demográfica: se utiliza la información de características del usuario para desarrollar clasificadores capaces de asociar datos demográficos específicos con calificaciones o preferencias de compra. Este tipo de sistema se fundamenta en aspectos demográficos del usuario, como el idioma, las recomendaciones de productos según la ubicación del usuario, así como también en consideraciones relacionadas con el género y la edad, sugiriendo restaurantes cercanos, entre otros [4]. Por ejemplo, supongamos que tenemos toda la información demográfica de los usuarios que visitan las distintas bibliotecas que pertenecen al Sistema de Biblioteca de Comfama, allí podemos identificar según su edad, localidad, sexo y factores como la economía del sector que habita qué libros y lecturas se adecuan a sus gustos y cuál es la biblioteca más cercana para que los preste.

Aquí se utilizan datos demográficos del usuario para clasificar y predecir sus preferencias. Los modelos de clasificación como los árboles de decisión, SVM (Support Vector Machines) o Naive Bayes son ejemplos típicos.

- **Funcionamiento técnico:** A partir de los datos demográficos como la edad, ubicación y género, el sistema clasifica a los usuarios en categorías y asigna preferencias en función de patrones observados.

- **Modelo matemático:** El modelo más común es un clasificador, que predice una etiqueta para el usuario basada en su conjunto de características.
- **Métricas de desempeño: Accuracy, precision, recall, y ROC-AUC** son usadas para medir la capacidad del sistema para clasificar correctamente las preferencias.

Sistemas de recomendación híbridos y basados en conjuntos: los sistemas de recomendación explotan distintas entradas y se desempeñan bien en diferentes escenarios. Los de filtrado colaborativo usan calificaciones de los usuarios, los basados en contenido dependen de descripciones y calificaciones propias, mientras que los basados en conocimiento emplean interacciones en contextos de bases de datos. Los sistemas demográficos usan perfiles del usuario. [4] Cada sistema tiene sus propias entradas, fortalezas y debilidades. Algunos, como los basados en conocimiento, son útiles en entornos con pocos datos; otros, como los colaborativos, funcionan mejor con más información. Cuando hay múltiples entradas, se pueden usar distintos sistemas y mezclarlos para obtener mejores resultados. Los híbridos combinan elementos de varios sistemas, similar al análisis de conjuntos, mejorando la efectividad general [4].

3.1.2. Evaluación de los sistemas de recomendación

El análisis de los sistemas de recomendación se basa en enfoques y herramientas similares a los utilizados en la clasificación y la regresión. En esos modelos, se anticipa un valor o resultado faltante a partir de los datos disponibles. De manera análoga, en los sistemas de recomendación, se prevé cualquier dato faltante de una matriz considerando la información existente. Esta relación permite adaptar modelos de clasificación para evaluar los sistemas de recomendación, aunque se necesitan ajustes específicos.

La evaluación de estos sistemas abarca diversas técnicas, como la predicción de calificaciones y la clasificación. La predicción de calificaciones guarda similitudes con la clasificación y la regresión, mientras que la clasificación se vincula más a la evaluación de la efectividad en búsquedas e información. Los métodos para evaluar los sistemas de recomendación se detallan minuciosamente en el respectivo capítulo del texto.

Los sistemas de recomendación son útiles cuando se tiene pensado desarrollar proyectos enfocados en las preferencias de las personas respecto a un producto y servicio, como en este caso, que se desea crear un sistema que recomiende libros y lecturas a las personas.

3.1.3. Machine Learning

Es una de las ramas de la Inteligencia Artificial, ya que, su propósito es crear modelos que aprenden automáticamente [8]. Para entender el concepto de aprendizaje en este contexto se habla de generalizar el conocimiento a partir de conjuntos de experiencias. Esto basado en que los sistemas pueden ir adquiriendo conocimiento y aprendizaje a través de los datos que tenga disponibles, identificando patrones sin la intervención de personas.

Dependiendo de los datos que se tengan disponibles para el análisis, se tienen diferentes tipos de aprendizaje, los más comunes son el supervisado y el no supervisado, el primero se basa en la construcción de modelos a partir de datos ya categorizados, permitiendo al algoritmo ajustarse para predecir resultados coincidentes con los conocidos [8].

Como segundo el aprendizaje no supervisado busca descubrir patrones y estructuras en datos no clasificados, empleando técnicas como la clusterización, la reducción de dimensiones y las reglas de asociación para clasificar nuevos datos según las clasificaciones previas [8].

La integración del Machine Learning en el presente proyecto representa un vínculo directo entre la ciencia de datos avanzada y la satisfacción del usuario. Este enfoque tecnológico permite el análisis profundo de datos, como historiales de préstamos y preferencias declaradas, para modelar perfiles de usuario precisos. Al aplicar algoritmos de aprendizaje automático, se busca comprender los patrones individuales de interacción con la información, con el propósito de generar recomendaciones altamente personalizadas.

3.1.4. Word2Vec

Word2Vec es un modelo predictivo basado en aprendizaje profundo desarrollado por Google en 2013, diseñado para generar representaciones densas y continuas de palabras en un espacio vectorial. Estas representaciones, también conocidas como embeddings, capturan relaciones semánticas y contextuales entre

las palabras, facilitando una comprensión más profunda de los datos textuales. A diferencia de los modelos tradicionales como Bag of Words, Word2Vec genera embeddings en un espacio de menor dimensionalidad, lo que permite un análisis eficiente incluso con grandes volúmenes de datos. Este modelo es entrenado de manera no supervisada, procesando grandes corpus textuales para construir un vocabulario y generar vectores densos que representan cada palabra [14].

Word2Vec incluye dos arquitecturas principales: Continuous Bag of Words (CBOW) y Skip-Gram. El modelo CBOW predice una palabra objetivo basada en su contexto (palabras circundantes), mientras que Skip-Gram realiza la tarea opuesta, utilizando una palabra objetivo para predecir su contexto. Ambas arquitecturas son entrenadas como modelos de clasificación, donde el objetivo es minimizar el error de predicción durante el entrenamiento. Estas capacidades hacen de Word2Vec una herramienta versátil en tareas como análisis semántico, clasificación de textos y recomendaciones personalizadas, destacándose como un enfoque clave en el procesamiento de lenguaje natural y la minería de texto [14].

3.1.5. K-means

El algoritmo de agrupamiento k-means es un enfoque centrado en centroides que busca dividir un conjunto de datos en grupos o clústeres con varianza homogénea. Su propósito fundamental es minimizar la inercia, o suma de cuadrados dentro del clúster, una medida que refleja la dispersión de los datos en relación con sus centroides. Este modelo se distingue por su simplicidad y facilidad de implementación, lo que lo convierte en uno de los algoritmos de agrupamiento más utilizados, especialmente en contextos donde se manejan grandes volúmenes de datos.

El funcionamiento del k-means se puede desglosar en tres pasos esenciales: primero, se inicializan los centroides; segundo, se asignan los puntos de datos a los centroides más cercanos; y tercero, se recalibran los centroides en función de las nuevas asignaciones. Sin embargo, uno de los principales desafíos de este algoritmo es la necesidad de determinar el número de clústeres (k) de antemano. Este requerimiento puede limitar su aplicabilidad en situaciones en las que no se conoce el número óptimo de clústeres desde un principio. Además, aunque el algoritmo converge, no siempre asegura un resultado óptimo global, ya que puede quedar atrapado en un mínimo local. Para superar este inconveniente, se recomienda el uso del método k-means++, que elige centroides iniciales que se encuentran a una mayor distancia entre sí [13].

La aplicación del k-means es habitual en diversas áreas, como el análisis de datos, la segmentación de clientes y la clasificación de textos. Por ejemplo, en el análisis cinematográfico, este algoritmo se puede utilizar para agrupar películas basándose en características extraídas de sus descripciones. Al asignar etiquetas de clúster a cada título, los investigadores tienen la oportunidad de examinar la distribución de películas en cada clúster y extraer características significativas que revelen las similitudes dentro de cada grupo. Este tipo de análisis no solo enriquece nuestra comprensión de los patrones presentes en los datos, sino que también facilita la toma de decisiones informadas en distintos contextos.

3.1.6. Perfil de usuario

El usuario es el protagonista de los sistemas de información, servicios y de toda la "trama informática, es el principio y el fin del ciclo de transferencia de la información" [5]. El perfil del usuario se refiere al conocimiento que se tiene sobre la información que resulta interesante para ese usuario en particular. Se pueden identificar dos tipos de perfiles: los simples, que consisten en un conjunto de condiciones extraídas de documentos considerados relevantes para el usuario, permitiendo expandir la búsqueda inicial para encontrar más documentos relacionados o similares. [6] Por otro lado, los perfiles extendidos se describen como un conjunto de cuatro variables distintas. Estas variables abarcan aspectos demográficos, como la edad o nivel educativo del usuario; la identificación, entre otros.

El perfil del usuario es una representación organizada y coherente de las características del usuario, buscando precisión en su descripción. Originalmente vinculado a la psicología, se refiere al conjunto de medidas que definen a una persona o grupo en una misma unidad de medición. En el ámbito bibliotecológico, se asocia con el servicio de Diseminación Selectiva de Información (DSI), concebido por H.P. Luhn en la década de 1950, destinado a proporcionar información relevante a individuos según sus necesidades laborales o de interés [7].

El DSI requiere la definición de un perfil que represente los intereses del usuario, facilitando la identificación de documentos relevantes [6]. Este perfil inicialmente se refería solo a necesidades de información e incluía datos generales para la identificación. Con el tiempo, ha evolucionado, agregando características como tipo de documento, autor, fecha de publicación, entre otros, para hacer los perfiles más especializados.

El concepto de perfil de usuario surge como una extensión del DSI, definiéndose como el conjunto de

características distintivas que identifican al usuario. Similar a su uso en psicología, donde ayuda en diagnósticos, en bibliotecología describe al usuario y sirve de base para planificar procedimientos [6].

Los datos que conforman un perfil de usuario incluyen intereses disciplinarios, nivel educativo, función principal, recursos de información utilizados o requeridos, métodos de búsqueda, comportamiento en la búsqueda de información y manejo del lenguaje. Mayormente, estos perfiles se basan en supuestos de necesidades de información percibidas por los bibliotecarios, sin ser confirmados o refutados.

3.1.7. Procesamiento de textos

El procesamiento de textos es un campo interdisciplinario que se ocupa de desarrollar y aplicar técnicas computacionales para comprender, analizar y manipular el lenguaje humano en forma de texto. Se basa en una comprensión profunda de la lingüística y la estructura del lenguaje, desde la sintaxis y la semántica hasta la pragmática y la adquisición del lenguaje. Utilizando herramientas y algoritmos de procesamiento de lenguaje natural (NLP), el procesamiento de textos abarca una amplia gama de tareas, que van desde la tokenización y el análisis morfológico hasta la traducción automática y la generación de texto. Al aprovechar técnicas como el aprendizaje automático y el procesamiento del lenguaje profundo, el procesamiento de textos tiene aplicaciones prácticas en campos como la búsqueda de información, la extracción de conocimiento y la interacción humano-computadora [12].

Por ello, el procesamiento de textos permite a las computadoras entender y procesar el lenguaje humano de manera automatizada, lo que abre la puerta a una amplia variedad de aplicaciones en el ámbito del análisis de datos, la inteligencia artificial y la comunicación digital. Mediante el uso de herramientas y técnicas avanzadas, los profesionales del procesamiento de textos pueden abordar desafíos complejos relacionados con el manejo y la interpretación de grandes volúmenes de texto, lo que facilita la extracción de información útil y la toma de decisiones basada en datos en una amplia gama de contextos y aplicaciones.

4.2 ANTECEDENTES

Se realizó la búsqueda de trabajos referentes en sistemas de recomendación y predicción. Entre los trabajos consultados se encontró:

- Modelo inteligente de recomendación de campañas, basado en perfilamiento de hábitos de consumo [9]: Este trabajo se sumerge en el desarrollo de un modelo inteligente para recomendar campañas basado en el análisis de los hábitos de consumo. Explora la teoría y la práctica detrás del perfilamiento de consumidores, buscando comprender cómo los patrones de comportamiento influyen en las estrategias promocionales. La investigación comienza estableciendo los fundamentos teóricos, identificando segmentos de mercado a partir de datos demográficos y psicográficos. Esta base teórica se traduce en un componente estático del modelo, permitiendo clasificar a los usuarios y comprender sus preferencias. El análisis se adentra en el diseño de un componente dinámico que responde a eventos del mundo físico y virtual. Este componente reacciona ante transacciones financieras, ubicación geográfica y actividad en redes sociales, adaptándose en tiempo real a las nuevas intencionalidades de consumo. Un hito importante es la creación del motor de reglas, el núcleo del modelo. Este motor deduce campañas promocionales a partir de la información estática y dinámica del cliente, generando recomendaciones personalizadas y precisas. Si bien este modelo se enfoca en hábitos de consumo, difiere en su contexto y enfoque. La propuesta para las bibliotecas de Comfama se centra en los hábitos de lectura y preferencias literarias, lo que implica una adaptación específica de las técnicas de recomendación para libros y lecturas en lugar de productos o campañas promocionales.

- College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm [10]: Este trabajo desarrolla un sistema de recomendación personalizado para bibliotecas universitarias, utilizando un enfoque híbrido que combina filtrado colaborativo y basado en contenido. El modelo integra datos de perfiles demográficos, patrones de comportamiento como historiales de préstamo y calificaciones otorgadas a los recursos bibliográficos. Este enfoque captura tanto las preferencias explícitas como implícitas de los estudiantes. La investigación aborda las limitaciones de los métodos

tradicionales, como la dependencia exclusiva de datos históricos. Se implementa un algoritmo híbrido que usa aprendizaje automático para mejorar la precisión de las recomendaciones, adaptándose a cambios en las preferencias de los usuarios. Además, emplea técnicas avanzadas de preprocesamiento para manejar grandes volúmenes de datos y resolver problemas como el sesgo hacia ciertos recursos.

Se destaca la segmentación de usuarios y la integración de métricas para evaluar el desempeño del sistema, como la precisión, cobertura y satisfacción del usuario. Los resultados muestran que el sistema híbrido supera a los métodos tradicionales, proporcionando recomendaciones más relevantes, especialmente en el entorno académico. Aunque centrado en bibliotecas universitarias, el enfoque es aplicable a las Bibliotecas Comfama, adaptándose a las necesidades de lectura y promoción cultural.

- Diseño e implementación de un sistema de recomendación para facilitar la elección de programas académicos en educación superior [11]: este proyecto comenzó con la definición de variables relevantes, para abarcar tanto las preferencias y habilidades de los estudiantes como los atributos de los programas. Se diseñó y desarrolló una prueba vocacional específico para capturar de manera precisa las preferencias e intereses de los estudiantes, proporcionando una información valiosa que fue integrada a los datos obtenidos de pruebas estandarizadas como las pruebas ICFES, permitiendo complementar el perfil académico de los estudiantes. Además, se establecieron fases estructuradas para el procesamiento de datos, desde su extracción hasta la presentación, asegurando una gestión eficiente y precisa de la información recolectada. Se desarrollaron reglas de cálculo meticulosamente definidas, considerando metodologías ponderadas que incorporaran tanto las preferencias como los puntajes y características de los programas académicos. Se implementaron módulos centrales como el de explicación y administración, fortaleciendo la confianza del estudiante en las recomendaciones y permitiendo una gestión integral de los datos utilizados en el proceso. Finalmente, se diseñó una interfaz gráfica intuitiva y funcional dividida en secciones específicas para diferentes roles, facilitando la interacción y la navegación dentro del sistema. Estos hitos representan aspectos esenciales y críticos en la resolución del desafío de proporcionar recomendaciones académicas precisas y efectivas en el contexto de la educación superior. Este antecedente se enfoca en la selección de programas académicos en educación superior. A diferencia de este proyecto, que se centra en la selección personalizada de libros y lecturas, el contexto y los datos utilizados son completamente distintos. La

propuesta bibliotecas de Comfama se basa en patrones de préstamo, interacciones de usuarios con libros y preferencias declaradas para generar recomendaciones bibliográficas.

Los antecedentes recopilados en relación con sistemas de recomendación y predicción ofrecen una panorámica valiosa para el presente proyecto. El modelo inteligente de recomendación de campañas destaca la importancia del análisis de hábitos de consumo y la comprensión de perfiles de usuarios, aspectos cruciales para comprender las preferencias individuales. Por otro lado, sistema de recomendación personalizado para bibliotecas universitarias, resalta el uso de algoritmos híbridos para personalizar las recomendaciones en bibliotecas académicas. Este enfoque combina técnicas basadas en contenido y filtrado colaborativo, logrando resultados más precisos al integrar información demográfica y de comportamiento de los usuarios. Además, el diseño de un sistema de recomendación para la elección de programas académicos enfatiza la importancia de seleccionar variables relevantes y realizar un procesamiento riguroso de datos, elementos esenciales que se alinean con los desafíos de implementar un sistema adaptado a las preferencias de los usuarios de las Bibliotecas de Comfama. Estos antecedentes brindan bases claras para el desarrollo de una solución robusta y efectiva.

5 PREPARACIÓN DE DATOS

El primer paso para el desarrollo del sistema de recomendación fue la identificación y recolección de datos. Esta fase se enfocó en obtener información de diversas fuentes disponibles en las bibliotecas Comfama, las cuales fueron esenciales para construir una base sólida para el análisis de las preferencias de los usuarios. Los datos recogidos incluyen historiales de préstamos, interacciones con libros y preferencias declaradas por los usuarios a través de encuestas y formularios, permitiendo una mejor comprensión de sus comportamientos y gustos.

5.1 Estrategia de recolección de datos

Para garantizar que los datos recopilados fueran adecuados y representativos, se utilizó una estrategia estructurada de recolección de datos. Esta estrategia incluyó la colaboración con el personal de las bibliotecas para obtener información precisa sobre los historiales de préstamo y las interacciones con el sistema de bibliotecas. También se trabajó con el área de tecnología para obtener acceso al lago de datos de Comfama, donde se pudo obtener más información sobre las transacciones de los usuarios en el ILS (Integrated library system) Alma, que es el gestor bibliográfico que tiene el Sistema de Bibliotecas de Comfama.

Otro aspecto fundamental en la recolección de los datos fue que se tuvo acceso en tiempo real al ILS y al lago de datos de Comfama. Allí se construyó una data que permitió conocer los títulos que prestan los usuarios, el número de préstamos, las materias o temáticas y los autores, además, de validar sus edades, municipios de residencia y género. Esta base de datos se consolidó desde el 01/09/2018 al 31/08/2024 con las variables ya mencionadas (Tabla1), para un total de 1.457.562 de registros.

Tabla 1. Variables de la base de datos.

Variable	Descripción
Primary Identifier	Código de identificación de los usuarios en el ILS.
Full Name	Nombre completo del usuario.
Gender	Género del usuario.
Birth Date	Fecha de nacimiento del usuario.
City	Ciudad o municipio donde reside el usuario.
ISBN	Número de identificación único del libro (International Standard Book Number).
Title	Título del libro.
Author	Autor del libro.
Subjects	Temas, materias o tópicos relacionadas con el libro.
Loans (In House + Not In House)	Número total de préstamos y consultas

5.2 **Preprocesamiento de datos y limpieza**

El preprocesamiento de los datos es uno de los pasos más críticos en la construcción de un sistema de recomendación, ya que asegura que los datos sean consistentes, limpios y utilizables para los algoritmos de recomendación. En el proyecto, el conjunto de datos se sometió a varias etapas de limpieza y transformación, con el objetivo de garantizar la calidad de los datos y prepararlos adecuadamente para su análisis.

Etapas:

Eliminación de valores faltantes (NAN) y registros erróneos: En primer lugar, se identificaron y eliminaron registros con valores NAN.

Tabla 2. Variables con valores faltantes.

Variable	Cantidad de valores faltantes
Primary Identifier	0
Full Name	0
Gender	168472
Birth Date	0
City	20914
ISBN	57170
Title	1
Author	149974
Subjects	22067
Loans (In House + Not In House)	0

Luego de identificar los valores NAN se procedió con su eliminación utilizando "dropna()", quedando un total de 1.114.880 registros.

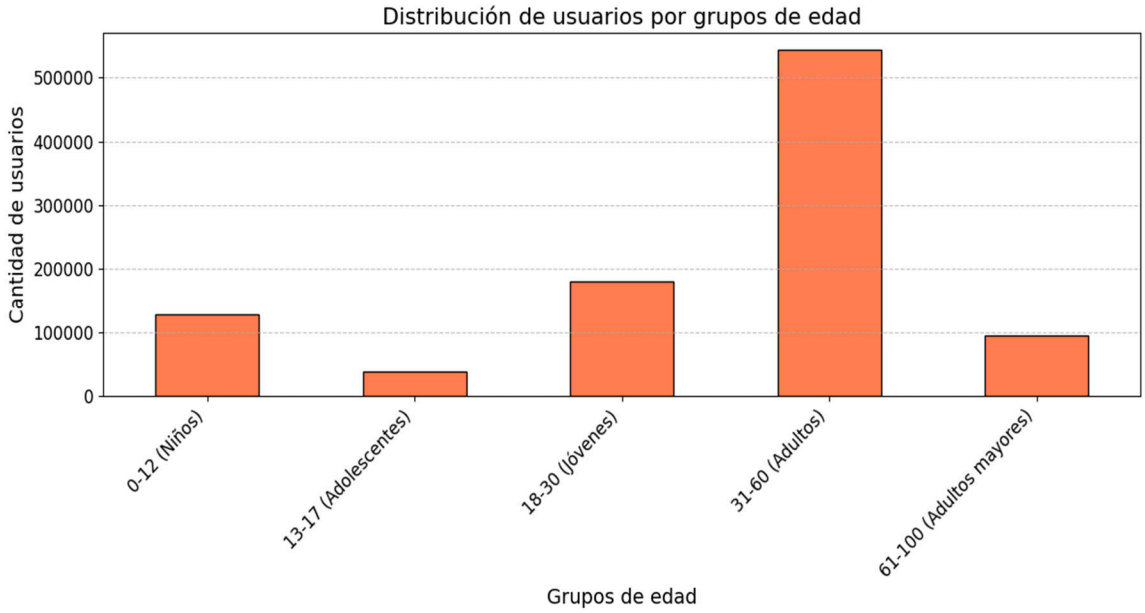
Análisis de la base de datos:

Codificación de variables:

Con base en el análisis realizado, se decidió codificar las siguientes variables para facilitar la posterior aplicación del algoritmo de clustering K-means. Las variables seleccionadas son: “Birth Date”, “Gender” y “City”.

En el caso de la variable ***“Birth Date”*** se realizó el proceso para calcular la edad de cada usuario según la fecha de nacimiento y estas edades quedaron alojadas en una nueva variable nombrada **“Age”.**

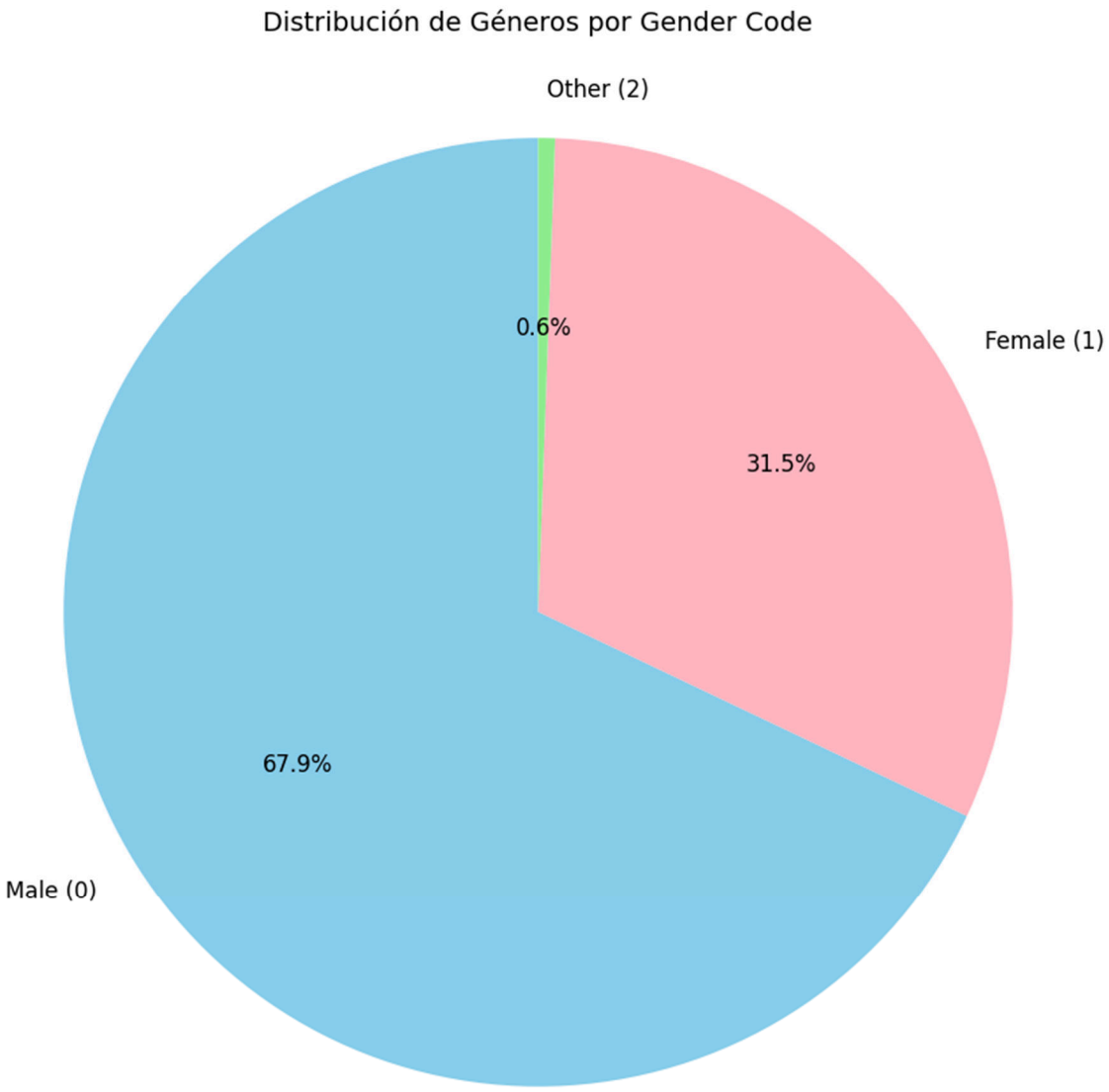
Gráfica 1. Distribución de usuarios por grupos de edad.



Según la gráfica 1, se evidencia que la mayoría de los usuarios se encuentra entre los 31 a 60 años, siguiendo los jóvenes de 18 a 30 años. Esto llama mucho la atención, ya que, en los anteriores análisis se evidencia que los títulos más prestados son para el público infantil, esto puede darse porque los adultos prestan libros para sus hijos, sobrinos, nietos o familiares.

Para la variable de ***“Gender”*** se utilizaron los códigos 'Female': 1, 'Male': 0 y 'Other': 2 y quedaron en una nueva variable nombrada **“Gender code”.**

Gráfica 2. Distribución por el código del género.

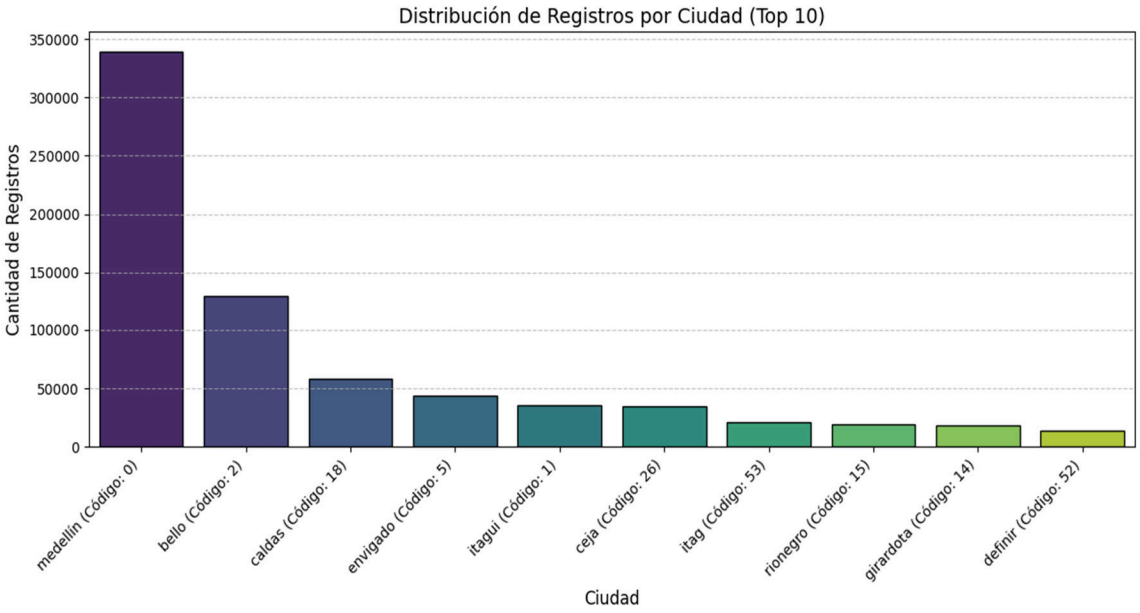


En la

2 podemos ver cómo cambia la distribución y los hombres ya son mayoría con un 67.9% en comparación con las mujeres que son 31.5%. Esto se presenta porque al organizar las edades, se evidenció que la variable “Birth Date” quedo con valores faltantes. Se procedió a eliminar estos NAN y la base de datos quedó con 984.954 registros.

Ya para la variable **“City”** se utilizaron códigos de 0 en adelante, y estos quedaron alojados en la variable nombrada **“City code”**.

Gráfica 3.Ciudades codificadas.



En la gráfica 3 se pueden ver el top de las 10 ciudades donde viven los usuarios que acceden al servicio de préstamo en las bibliotecas de Comfama. Acá es de esperarse que Medellín sea la ciudad en la que más usuarios hay, ya que es donde se concentran el mayor número de presencias de bibliotecas, porque Comfama tienen bibliotecas en los barrios de Pedregal, Aranjuez, Manrique, El Poblado, El perpetuo Socorro, Barrio Colombia, entre otros. Ya en las otras ciudades tiene poca presencia.

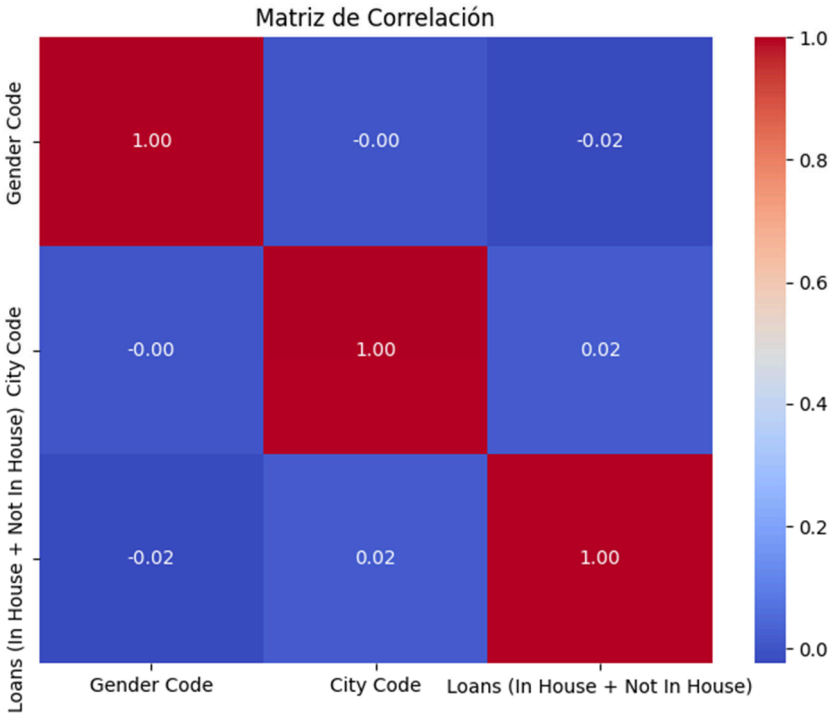
Datos duplicados:

Para asegurar que cada usuario aparezca una única vez en el conjunto de datos, se agruparon en cada celda de la variable **“Title”**, todos los títulos que se ha leído cada usuario, así como las variables de **“Subjects”, “Author”, “Loans (In House + Not In House)”**. Las entradas duplicadas basadas en los identificadores clave: Primary Identifier, Full Name, Gender Code, Age, y City Code.

Correlación de las variables:

Se validó la correlación de las variables codificadas.

Gráfica 4. Matriz de correlaciones.



En la gráfica 4 se muestra la matriz de correlación, que evalúa las relaciones entre las variables Gender Code, City Code y Loans (In House + Not In House), en el contexto bibliotecario mostrando el grado de asociación lineal entre ellas. La correlación de **Gender Code** y **City Code** es prácticamente nula (-0.000159), lo que indica que el género y la ciudad de los usuarios no están relacionados linealmente en este contexto. Para **Gender Code y Loans (In House + Not In House)**, la correlación es muy débil y negativa (-0.024470). Esto sugiere que no existe una relación significativa entre el género y el número total de préstamos y consultas realizadas. En **City Code y Loans (In House + Not In House)**, la correlación es muy débil y positiva (0.017626). Esto indica que la ciudad de residencia tiene un impacto prácticamente inexistente en el número total de préstamos.

Para ampliar la información de la base de datos analizada ver Anexo 1.

5.3 Estructuración final de los datos

Una vez que los datos fueron limpiados y preprocesados, se organizaron en un DataFrame de Pandas, lo cual facilitó su manipulación y uso en los modelos de recomendación. El DataFrame integra tanto las características demográficas de los usuarios como su historial de interacciones con libros, proporcionando una base sólida para la aplicación de algoritmos de recomendación basados en similitudes y agrupación. El DataFrame quedó con un total de 107.744 registros y 107 variables.

Estructura y características principales del DataFrame final:

Las siguientes variables ayudan a entender las preferencias y comportamientos de los usuarios, permitiendo personalizar las recomendaciones y agrupar a los usuarios según sus similitudes:

Tabla 3. DataFrame final.

Variable	Descripción
Primary Identifier	Código por el cual se relaciona el documento de identidad de los usuarios.
Full Name	Nombre completo de la cuenta de los usuarios.
Age	Edad del usuario, calculada a partir de la fecha de nacimiento.
City Code	Código numérico que representa la ciudad o municipio de residencia.

Gender Code	Género del usuario, codificado en un formato numérico (0 para masculino, 1 para femenino, 2 para otro).
Loans (In House + Not In House)	Número total de consultas y préstamos realizados por el usuario.
Title	Lista de títulos de los libros prestados por el usuario.
Subjects	Lista de temas o materias relacionadas con los libros que el usuario ha prestado o consultado.

La tabla 3 contiene información detallada sobre 107.744 usuarios únicos, a cada uno de los cuales se les han asignado variables demográficas como edad, género y localización. Estas características permiten realizar segmentaciones y personalizar las recomendaciones de manera más efectiva.

El historial de préstamos, integrado en el mismo dataframe, captura las preferencias y comportamientos pasados de los usuarios en la biblioteca. Este historial incluye 107.744 interacciones, registrando no solo los usuarios y los libros que han tomado en préstamo, sino también atributos clave como los títulos de los libros, las materias asociadas, así como la frecuencia con la que cada usuario interactúa con los diferentes materiales. En la transformación de los datos, se consolida un único registro por usuario, que incluye los Subjects y los títulos de los libros asociados a su historial de préstamos.

Tabla 4. Ejemplo del registro único por usuario.

Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Subjects	Author	Loans (In House + Not In House)
C01C1001500943	TABORDA POSSO SEBASTIAN ANDRES	0	23	0	['La edición digital de la imagen fotográfica', '1968: el nacimiento de un mundo nuevo']	['Fotografías; Fotografía digital; Procesamiento de imagen; Sistema de imágenes; Fotografía - Técnicas digitales; Fotografía digital - Manuales', 'Revoluciones sociales--Historia--Francia--1968; Movimientos	['Meehan, Les', 'González Férriz, Ramón 1977-,']	2

						estudiantiles --Francia-- 1968; Historia mundial-- Europa']		

En la tabla 4 se puede observar un ejemplo de cómo quedó el registro único por cada usuario, allí se agruparon los títulos, Subjects, número de préstamos y autores en cada registro de usuario.

5.4 **Vectorización de textos.**

Para garantizar la calidad y consistencia de los datos textuales, se realizó un proceso de normalización que incluyó la identificación de las materias dividiéndolas en oraciones mediante expresiones regulares adaptadas a su estructura y, posteriormente, la tokenización de dichas oraciones. Este proceso abarcó la conversión de todo el texto a minúsculas, la eliminación de caracteres especiales conservando únicamente letras y guiones relevantes, y la lematización para reducir las palabras a su forma base, optimizando así el análisis semántico. Además, se eliminaron palabras vacías comunes en español que no aportaban valor significativo al contexto.

Para la vectorización de los textos se utilizó Word2Vec es una técnica de modelado de lenguaje que captura relaciones semánticas en grandes corpus de texto. Al utilizar esta técnica, es posible mapear libros con descripciones o títulos similares a un espacio vectorial común, lo que mejora las recomendaciones colaborativas.

Esta técnica tiene como objetivo convertir las características textuales de los libros, como títulos, géneros y descripciones, en vectores numéricos para poder calcular las similitudes entre libros. Esto permite recomendar libros que sean semánticamente similares a los que un usuario ha leído o mostrado interés.

Para la optimización se realizó lo siguiente:

Dimensionalidad del vector: se ajustó a 100 dimensiones, lo cual es un valor comúnmente utilizado para encontrar un equilibrio entre la precisión de las representaciones y la eficiencia computacional. Ver ilustración 1.

Tamaño de la ventana: Se utilizó un tamaño de ventana de 5 palabras, lo que permitió capturar las relaciones semánticas dentro de un contexto razonable de cada libro.

Frecuencia mínima: Para evitar el ruido, se configuró una frecuencia mínima de 5 apariciones de palabras, lo que ayuda a eliminar términos irrelevantes y mejorar la calidad de las representaciones.

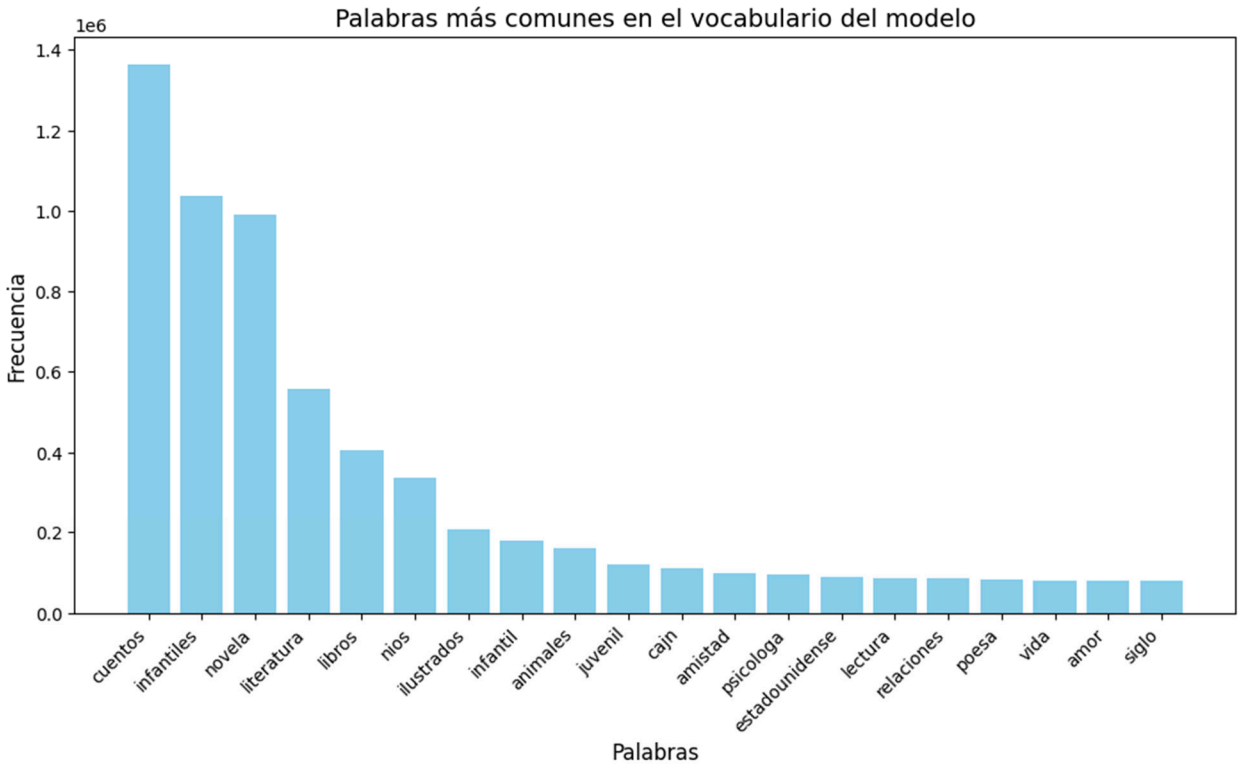
0	1	2	...	90	91	92	93	94	95	96	97	98	99
0.280615	-0.052416	0.307992	...	-0.032930	0.135688	0.498295	-0.426712	0.287736	0.009747	0.321784	0.313229	-0.150987	0.099512
0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-0.177818	-0.199111	-0.268580	...	0.051444	0.100448	0.224243	0.284885	-0.113078	0.294446	-0.079034	-0.526583	0.079784	0.309629
-0.263781	-0.385908	-0.466116	...	0.189849	-0.032371	0.396008	0.444555	-0.374318	0.328343	-0.177303	-0.833552	0.254091	0.415318
0.199063	-0.292346	-0.435913	...	-0.755373	0.079594	-0.833571	0.448213	-0.088917	0.452563	-1.398970	-0.745161	-0.688367	1.291091

Ilustración 1. Representación de los Subjects en vectores.

En la ilustración 1, se muestra una matriz numérica que representa vectores generados mediante la técnica Word2Vec. Las columnas están etiquetadas de 0 a 99, lo que indica que cada vector tiene 100 dimensiones o características. Las filas corresponden a diferentes elementos del conjunto de datos, y cada fila contiene valores numéricos que oscilan entre rangos positivos y negativos, los cuales representan relaciones semánticas en el espacio vectorial.

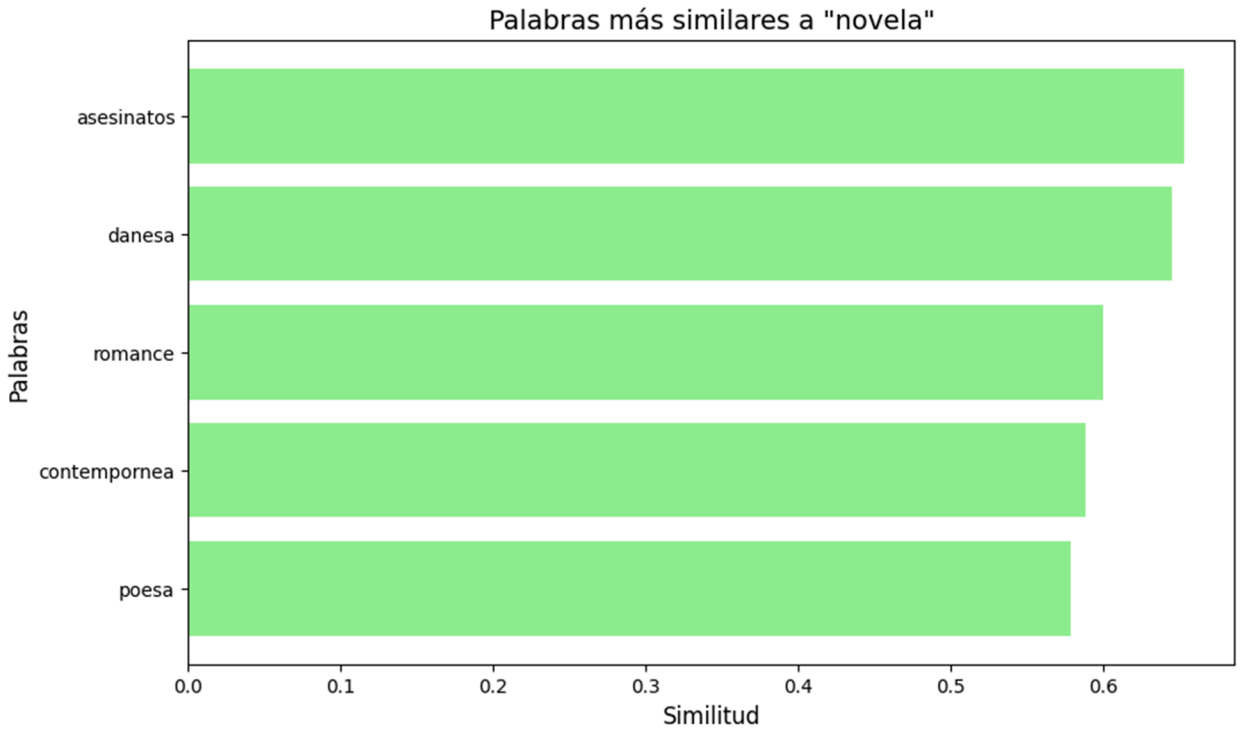
Para validar los resultados de Word2Vec, se evaluó la calidad de los vectores generados mediante la cohesión semántica y su capacidad para capturar relaciones entre elementos similares. Se analizaron las recomendaciones producidas, comprobando su coherencia con las preferencias de los usuarios, y se realizaron pruebas de consistencia para asegurar la estabilidad de los resultados frente a cambios en los datos de entrada.

Gráfica 5. Vocabulario del modelo Word2Vec



Una de las primeras validaciones realizadas al Word2Vec fue el análisis de las palabras más comunes en su vocabulario, ver gráfica 5. Este paso es esencial para validar que el modelo ha capturado términos relevantes del dominio en cuestión, en este caso, las materias y temas bibliográficos. Al revisar las primeras 20 palabras del vocabulario, se pudo observar una fuerte presencia de términos relacionados con literatura y géneros específicos, lo que demuestra que el modelo ha aprendido correctamente las representaciones semánticas a partir del corpus entrenado. Este análisis garantiza que las palabras clave necesarias para las recomendaciones están adecuadamente representadas en el espacio vectorial del modelo.

Gráfica 6. Palabras similares.



Word2Vec también fue validado mediante pruebas de similitud y analogías para verificar la calidad de sus representaciones. Por ejemplo, al consultar las palabras más similares a "novela", el modelo identificó términos estrechamente relacionados, como géneros literarios y términos asociados a la narrativa. Esto confirma que el modelo captura correctamente las relaciones semánticas entre las palabras. Además, se realizaron pruebas de analogías, ver gráfica 6, como "libro es a autor como pintura es a", para analizar la capacidad del modelo de comprender relaciones conceptuales. En este caso, el modelo sugirió palabras relevantes, lo que refuerza su utilidad para representar de manera efectiva las relaciones entre conceptos literarios y bibliográficos.

Gráfica 7. Analogía.

En este capítulo 4, se describió detalladamente el proceso de recolección, limpieza, estructuración y transformación de los datos provenientes de las bibliotecas Comfama. Estas acciones permitieron consolidar un conjunto de datos robusto y funcional, integrado por variables demográficas, históricas y textuales que reflejan los patrones de comportamiento y preferencias de los usuarios.

Asimismo, se destaca la importancia de la codificación de variables y la vectorización de textos como pasos fundamentales para optimizar el análisis semántico y la implementación de

algoritmos de recomendación. El resultado es un DataFrame, limpio, estructurado y preparado para abordar las necesidades del sistema de recomendación. Ver tabla 5

Tabla 5.DataFrame final para el modelado.

Variable	Descripción
Primary Identifier	Código por el cual se relaciona el documento de identidad de los usuarios.
Full Name	Nombre completo de la cuenta de los usuarios.
Age	Edad del usuario, calculada a partir de la fecha de nacimiento.
City Code	Código numérico que representa la ciudad o municipio de residencia.
Gender Code	Género del usuario, codificado en un formato numérico (0 para masculino, 1 para femenino, 2 para otro).
Loans (In House + Not In House)	Número total de consultas y préstamos realizados por el usuario.
Title	Lista de títulos de los libros prestados por el usuario.
Vectores	Representación de los Subjects en vectores numéricos.

La tabla 5 describe un conjunto de datos relacionados con los usuarios de un sistema de préstamos bibliotecarios. La variable **Primary Identifier** representa un código único vinculado

al documento de identidad de cada usuario. **Full Name** contiene el nombre completo del usuario asociado a la cuenta. **Age** indica la edad, calculada a partir de la fecha de nacimiento. **City Code** es un número que identifica la ciudad o municipio de residencia del usuario. **Gender Code** clasifica el género utilizando valores numéricos: 0 para masculino, 1 para femenino y 2 para otro. **Loans (In House + Not In House)** refleja el total de consultas y préstamos realizados. **Title** presenta una lista de los libros prestados, y **Vectores** contiene representaciones numéricas de los temas (Subjects) asociados a los títulos, permitiendo análisis más complejos.

6 MODELADO

En este capítulo, se abordará el modelado desde un enfoque colaborativo, centrado en identificar patrones y agrupar a los usuarios según sus interacciones y preferencias de lectura. Este enfoque aprovecha la información compartida entre usuarios con comportamientos similares, permitiendo generar recomendaciones personalizadas y relevantes. A través de técnicas como el análisis de agrupamiento (clustering) y la vectorización de datos textuales, el sistema buscará mejorar la experiencia de los usuarios y optimizar el acceso a los recursos bibliotecarios.

En la ilustración 2 se pueden observar las etapas de del sistema de recomendación colaborativo

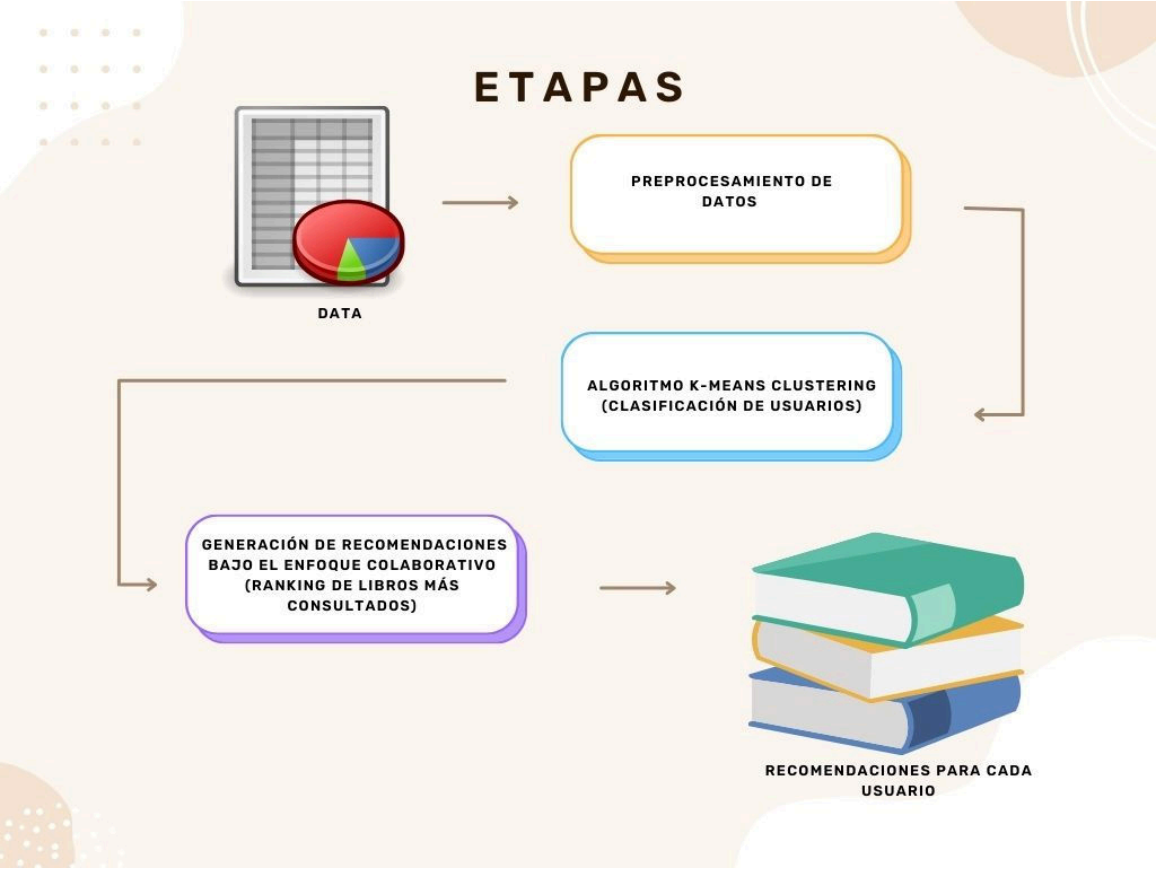


Ilustración 2. Etapas modelado y recomendación.

Al inicio del proceso de segmentación de usuarios, se optó por utilizar el algoritmo DBSCAN debido a su

capacidad para identificar grupos de datos sin necesidad de definir un número fijo de clústeres. Este enfoque resultaba atractivo, ya que DBSCAN es particularmente útil en contextos donde los datos presentan distribuciones densas y con ruido, como en el caso de las interacciones de los usuarios con las bibliotecas.

Sin embargo, durante la implementación, se evidenció un problema crítico relacionado con el rendimiento computacional. A medida que el algoritmo procesaba los datos, el tiempo de ejecución se volvía excesivo. En pruebas preliminares, la ejecución del modelo se extendía por más de seis horas sin completar el procesamiento, lo que ocasionaba que la máquina dejara de responder y la ejecución se detuviera abruptamente. Como resultado, era necesario reiniciar el proceso desde cero, lo que generaba una gran pérdida de tiempo y recursos.

Ante esta limitación, se tomó la decisión de cambiar a K-Means, un algoritmo más eficiente en términos de tiempo de ejecución y escalabilidad. Aunque K-Means requiere definir un número fijo de clústeres, su capacidad para agrupar grandes volúmenes de datos de manera rápida y efectiva lo convirtió en la opción más viable para este proyecto. Gracias a este cambio, fue posible optimizar el proceso de segmentación y generar recomendaciones personalizadas sin comprometer el rendimiento del sistema.

6.1 **Clusterización con K-Means**

Se seleccionó el algoritmo K-Means Clustering, el objetivo fue clasificar a los usuarios en grupos o clústeres según sus comportamientos de lectura y hábitos de préstamo, lo que permite detectar patrones comunes en las preferencias literarias de los usuarios.

El algoritmo K-Means es ampliamente utilizado en problemas de clasificación no supervisada. Su capacidad para segmentar usuarios en grupos homogéneos facilita la personalización de las recomendaciones, al sugerir libros que han sido apreciados por otros usuarios dentro de un mismo clúster.

En la implementación del algoritmo K-Means, se utilizaron variables clave para reflejar las características demográficas y comportamentales de los usuarios. Estas incluyen **Age**, **Gender Code**, **City Code**, **Loans (In House + Not In House)**, **Title**, **Subjects** y **Vectores**. Estas variables fueron seleccionadas estratégicamente para capturar aspectos esenciales como la edad, el género, la ubicación geográfica, el historial de préstamos, y las preferencias literarias, permitiendo una segmentación efectiva de los usuarios en clústeres significativos.

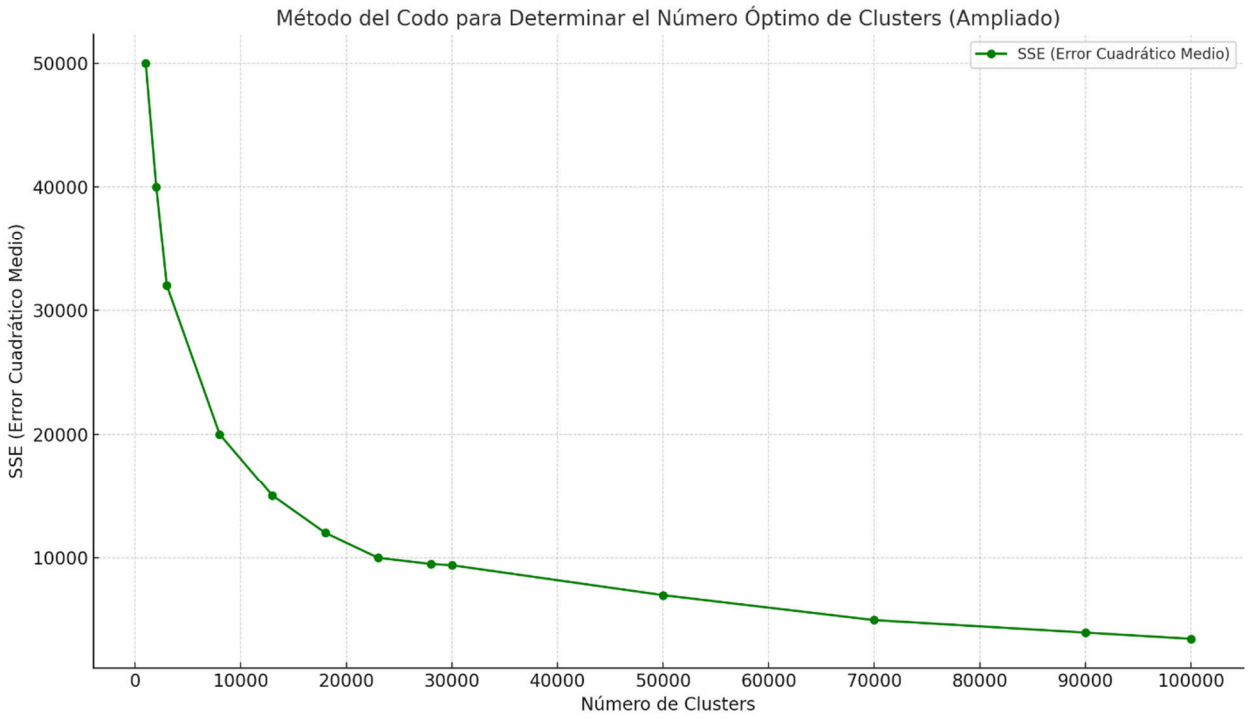
Para optimizar el modelo K-Means, se utilizó el método del codo, que permite determinar el número óptimo de clústeres al calcular el Error Cuadrático Medio (SSE) para distintos valores de K. Este análisis identificó un punto de inflexión en el que se equilibraba la precisión del modelo y su

complejidad, sugiriendo una cantidad inicial de clústeres adecuada. Sin embargo, al aplicar esta configuración en un dataset de 107,744 registros, los resultados no fueron satisfactorios. Los clústeres generados no lograron capturar de manera efectiva los patrones de comportamiento de los usuarios, y las recomendaciones presentaban títulos repetidos asignados a diferentes clústeres, evidenciando la necesidad de aumentar el número de clústeres para reflejar con mayor precisión la diversidad de perfiles y preferencias. Ver más información en **Anexo 2.**

Inicialmente, se intentó segmentar los datos con valores de K entre 2 y 10, pero estos generaban grupos demasiado grandes y heterogéneos que no diferenciaban efectivamente los patrones de comportamiento. Por ejemplo, con 2 clústeres, cada uno contenía un promedio de 50,000 registros, lo que era insuficiente para representar las variaciones en los datos.

Ante esta limitación y Según los resultados obtenidos y que se observan en las **ilustraciones 6 y 7**, se adoptó una estrategia de aumento progresivo del número de clústeres. Se comenzó con 1,000 clústeres, incrementando progresivamente en 1,000, y posteriormente en 5,000, hasta llegar a 30,000 clústeres, el cual fue el número optimo a usar según se ve en la **gráfica 8**. Este enfoque permitió reducir significativamente el tamaño promedio de los clústeres, logrando una segmentación más granular.

Gráfica 8. Método codo.



Después de aplicar el método del codo, se implementó el algoritmo K-Means con 30,000 clústeres, segmentando un conjunto de datos de 107,744 registros. Los resultados reflejaron una notable variación en los tamaños de los clústeres: algunos, como el clúster 3 y el clúster 608, agruparon 215 y 206 usuarios respectivamente, mientras que otros, como los clústeres 14066 y 27450, contuvieron únicamente un usuario. Esta diversidad en los tamaños de los clústeres es indicativa de la capacidad del modelo para capturar tanto grupos más generales como casos únicos o atípicos. La presencia de clústeres con un solo usuario sugiere que ciertos patrones o comportamientos no comparten similitudes suficientes con otros grupos.

Tabla 6. Muestra de clusters y cantidad de usuarios por cluster.

Clusters	Cantidad
3	215

608	206
49	153
144978	144
4603	135
14066	1
21450	1

Para abordar los clústeres que contenían solo un usuario, se implementó una estrategia basada en la similitud semántica utilizando los vectores generados previamente con Word2Vec. En lugar de depender exclusivamente de las interacciones dentro del clúster (que no era posible por la ausencia de otros usuarios), se calculó la similitud coseno entre el vector del usuario único y los vectores de todos los demás usuarios del dataset. Este cálculo permitió identificar a los usuarios más cercanos en términos de comportamiento y preferencias.

Una vez que la base de datos fue clusterizada, se tiene una última data que se utilizó para generar las recomendaciones. En la tabla 5 ya fueron descritas las variables que quedaron para la data, solo se agrega la variable **Cluster** que es la distribución de los usuarios en un grupo según sus similitudes y preferencias en cuanto a la información demográfica y su historial de préstamos, después de aplicar el algoritmo K-Means.

6.2 **Generación de recomendaciones.**

Después de contar con una base limpia, con la aplicación del algoritmo K-Means para segmentar a los usuarios en clúster, con su respectiva evaluación, se procedió a la generación de recomendaciones personalizadas. Este proceso se diseñó para aprovechar las similitudes dentro de cada clúster y garantizar que las recomendaciones fueran relevantes y adaptadas a los intereses de los usuarios.

En el proceso de generación de recomendaciones, se implementaron diversas estrategias para asegurar que las sugerencias fueran personalizadas y alineadas con las preferencias de los usuarios agrupados en clústeres. Estas estrategias se detallan a continuación:

Para cada clúster generado, se analizó cuidadosamente las preferencias de los usuarios agrupados. Este análisis incluyó la identificación de los títulos más leídos y las temáticas predominantes dentro de cada grupo, permitiendo establecer un perfil colectivo del clúster. Este perfil sirvió como base para comprender mejor los intereses compartidos de los usuarios.

Dentro de cada clúster, se calculó el top de libros más leídos mediante un conteo de interacciones (consultas y préstamos) agrupadas por el título de los libros. Posteriormente, los títulos se ordenaron según la cantidad total de préstamos, identificando así los libros más populares y representativos dentro de cada grupo. Este ranking permitió generar recomendaciones que reflejan los intereses predominantes en el clúster.

Las recomendaciones fueron personalizadas para cada usuario basándose en el top de títulos de su clúster. Este enfoque asegura que las sugerencias reflejen patrones de comportamiento y gustos compartidos con otros usuarios del mismo grupo. Además, se filtraron los títulos ya leídos por el usuario, evitando así recomendaciones repetitivas y promoviendo el descubrimiento de nuevos contenidos que enriquecieran la experiencia del lector.

Finalmente, las recomendaciones se enriquecieron con información adicional, como el autor y los temas principales de cada título. Este detalle ayudó a contextualizar las sugerencias y a proporcionar una experiencia más informada y satisfactoria para los usuarios.

7 EVALUACIÓN DEL MODELO

7.1 Evaluación de la calidad del agrupamiento

Una vez ajustado el modelo de K-Means y asignado un clúster a cada usuario del dataset, es fundamental evaluar la calidad de las agrupaciones generadas. Este análisis asegura que los clústeres reflejen patrones significativos en los datos y respalden recomendaciones efectivas. Para este

proyecto, se utilizaron tres métricas ampliamente reconocidas para evaluar algoritmos de clustering: **Silhouette Score, Davies-Bouldin Index y Calinski-Harabasz Index**. Estas métricas ofrecen perspectivas complementarias sobre la coherencia y separación de los clústeres.

Tabla 7. Resultados de evaluación de segmentación de Clusters.

Métrica	Resultado
Silhouette Score:	0.2045
Davies-Bouldin Index	0.8869
Calinski-Harabasz Index	378.315

En la tabla 7 podemos observar que El **Silhouette Score** que mide la coherencia interna de los clústeres, comparando las distancias entre los puntos del mismo clúster y los de otros clústeres. Con valores que van de -1 a 1, un puntaje cercano a 1 indica una buena cohesión dentro del clúster y una separación clara entre los demás. En este caso, el puntaje promedio fue de 0.2045, lo que se evidencia que está más cercano a 1. Esto se puede traducir en que se tiene una leve separación de los clusters.

El **Davies-Bouldin Index** evalúa la calidad de los clústeres considerando la distancia entre los centros de los clústeres y su dispersión interna. Un valor bajo indica que los clústeres están bien separados y son compactos. En este caso, el índice fue de 0.8869, lo que sugiere una separación moderada entre los clústeres, aunque aún puede haber espacio para una mejor distinción.

Por último**, Calinski-Harabasz** Index mide la relación entre la dispersión entre los clústeres y la dispersión interna de los clústeres. Un valor más alto indica una mayor separación y cohesión entre los clústeres. En este caso, el índice fue de 37.8315, lo que indica una separación decente entre los clústeres, con una estructura de agrupamiento relativamente clara.

8 SISTEMA DE RECOMENDACIÓN

El proceso de generación de recomendaciones personalizadas se fundamentó en el análisis de patrones y comportamientos de los usuarios, agrupados en clústeres mediante el algoritmo K-Means. Dentro de cada clúster, se identificaron las preferencias predominantes, tales como los géneros más leídos, autores recurrentes y temáticas populares. Posteriormente, se calcularon las interacciones más significativas entre los usuarios y los libros, priorizando títulos con altos índices de préstamos y consultas. Este enfoque permitió construir un sistema de recomendaciones que ofrece lecturas relevantes y alineadas con los intereses individuales de los usuarios, promoviendo un mayor aprovechamiento de la colección bibliotecaria.

Además, para evitar la repetición de recomendaciones, se implementó un filtro que excluyó los libros previamente leídos por cada usuario. Este filtro, en combinación con la segmentación por clústeres, facilitó el descubrimiento de nuevos títulos y temáticas que enriquecen la experiencia de los lectores. A su vez, el sistema utilizó técnicas avanzadas de análisis semántico, como Word2Vec, para garantizar que los títulos sugeridos compartieran similitudes contextuales y conceptuales con las preferencias de los usuarios. Este enfoque integral mejoró no solo la pertinencia de las recomendaciones, sino también su capacidad para captar la diversidad de intereses presentes en la comunidad de usuarios de las bibliotecas Comfama.

8.1 Visualización de recomendaciones.

Tras la generación de recomendaciones personalizadas, se implementó una interfaz amigable para facilitar a los usuarios el acceso y visualización de las sugerencias generadas por el sistema de recomendación, la cual se puede ver en la ilustración 3. Esta interfaz fue desarrollada con un enfoque centrado en la experiencia del usuario, garantizando que fuera intuitiva, funcional y estéticamente agradable.

La interfaz fue instalada localmente, permitiendo su uso dentro de las bibliotecas Comfama. A través de esta herramienta, los usuarios pueden ingresar su identificador único (Primary Identifier) para acceder a sus recomendaciones personalizadas. En la ilustración 4 se observa un ejemplo de la

recomendación personalizada, la cual incluye detalles de cada libro recomendado, como el título, el autor y las temáticas principales, lo que ayuda a los usuarios a comprender por qué se les sugirió cada título. Este nivel de personalización busca enriquecer la experiencia del usuario y fomentar la exploración de nuevos contenidos.

Ilustración 3. Interfaz de recomendación.

Ilustración 4. Ejemplo de recomendaciones.

9 EVALUACIÓN DEL SISTEMA DE RECOMENDACIÓN

Para la evaluación del sistema de recomendación se generaron varias recomendaciones a usuarios de prueba. Para ello se verificaron los números de documento, se generó la recomendación y vía WhatsApp fueron enviadas con las siguientes preguntas:

- ¿De los títulos en el listado le gustaría leer?
- ¿Le interesan las materias allí recomendadas?
- ¿Le interesan los autores que aparecen en el listado?

Los usuarios de prueba respondieron de forma general a cada una de las preguntas. Es importante resaltar que acá se muestran las tablas con los resultados de recomendación de cada usuario y sus respectivas respuestas a las preguntas.

9.1 **Recomendaciones Generadas.**

Se generaron listas personalizadas para usuarios específicos utilizando datos de prueba del período 2018-2024. A continuación, se describen los resultados por usuario:

Usuario 1:

Las recomendaciones generadas (Tabla 8) incluyeron varios títulos de Julio Verne, alineándose directamente con las preferencias del usuario, quien expresó su interés en novelas de aventuras y viajes. Este caso refleja una alta efectividad del sistema al identificar patrones claros de interés dentro del clúster correspondiente y ofrecer recomendaciones acordes.

Tabla 8. Títulos recomendados usuario 1.

#	Libro	Conteo	Author	Subjects
0	La isla misteriosa	4	Verne, Julio, 1828-1905.	Novela francesa; Marineros--Novela; Novela de ...
1	Proyecto Hail Mary	3	Weir, Andy, 1972-,.	Novela estadounidense; Novela de ciencia ficci...
2	Viaje al centro de la tierra	3	Verne, Julio, 1828-1905.	Novela juvenil francesa; Viajes en la literatu...
3	El extraño caso del Dr. Jekyll y Mr. Hyde	3	Stevenson, Robert Louis, 1850-1894.	Novela escocesa--Siglo XIX.; Novela psicológic...
4	De la tierra a la luna	3	Verne, Julio, 1828-1905.	Viajes interplanetarios; Novela francesa; Aven...
5	Los hijos del capitán Grant	3	Verne, Julio, 1828-1905.	Novela francesa; Novela de aventuras; Viajes p...
6	Cinco semanas en globo	2	Verne, Julio, 1828-1905.	Novela francesa; Aventuras--Novela; Viajes--No...
7	El cementerio de los libros olvidados. El pris...	2	Ruiz Zafón, Carlos 1964-2020	Novela española; Rehenes--Novela; Historias de...
8	Las minas del rey Salomón	2	Haggard, H. Rider., 1856-1925 Henry Rider Haggard	Novela inglesa--Siglo XIX; Minas de diamantes-...
9	Trilogía de Marte. Marte rojo (v1)	2	Robinson, Kim Stanley 1952-,	Novela estadounidense; Futuro en la literatura...
10	Magnus Chase y los dioses de Asgard. El barco ...	2	Riordan, Rick, 1964-, (Richard Russell Riordan)	Literatura norteamericana; Novela estadouniden...

11	Canción de hielo y fuego. Juego de tronos (v1)	2	Martin, George R. R., 1948-, (George Raymond R...	Sagas; Novela gráfica; Aventuras; Fantasía; Comic
12	Cuentos esenciales	2	Poe, Edgar Allan, 1809-1849.	Poe, Edgar Allan, 1809-1849--Colecciones; Cuen...
13	El valle del miedo	2	Doyle, Sir Arthur Conan. 1859-1930	Novela infantil inglesa; Detectives en la lite...
14	Relato de un náufrago	2	García Márquez, Gabriel, 1927-2014.	Novela colombiana--Siglo XX.; Naufragios--Nove...
15	La metamorfosis	2	Kafka, Franz, 1883-1924.	Novela checa--Siglo XX.; Novela psicológica.; ...
16	Túneles (v1)	2	Gordon, Roderick, 1960-,.	Novela inglesa; Secretos--Novela; Personas des...
17	El hombre de negro	2	Stevenson, Robert Louis, 1850-1894.	Cuentos escoceses; Condición humana--Cuentos; ...
18	El hombre que sabía demasiado	2	Keith Chesterton, Gilbert. 1874-1936	Novela inglesa
19	El mundo perdido	2	Conan Doyle, Arthur Ignatius 1859 - 1930	Novela gráfica; Literatura juvenil inglesa; Ti...

Usuaria 2:

En las recomendaciones generadas (Anexo 3, Tabla 10), la usuaria destacó su interés por los títulos 0, 2, 6, 12, 14, 15 y 18, además de mencionar su afinidad hacia libros relacionados con psicología. Este caso evidencia la capacidad del sistema para ofrecer sugerencias en múltiples géneros, incluyendo novelas psicológicas, permitiendo que los usuarios seleccionen opciones según sus intereses específicos.

Usuaria 3:

Las recomendaciones (Anexo 3, Tabla 11) fueron evaluadas positivamente, ya que la usuaria indicó que los títulos sugeridos son adecuados para sus hijos. Este caso demuestra que el sistema puede identificar libros que encajen con necesidades familiares, ampliando el impacto del sistema más allá del usuario directo.

Usuaria 4:

En las recomendaciones (Anexo 3, Tabla 12), la usuaria mostró interés específico en títulos como Momo, Odisea, Historia secreta de la música, y autores relacionados con Frankenstein y Black music. Asimismo, valoró las materias asociadas a las recomendaciones, evidenciando que el análisis semántico basado en Word2Vec permitió captar intereses particulares de manera efectiva.

Usuaria 5:

Las recomendaciones generadas (Anexo 3, Tabla 13) resultaron de interés para esta usuaria, quien es promotora de lectura para niños. Aunque mencionó que ya ha leído varios de los títulos sugeridos, estos no fueron prestados en la biblioteca, lo que subraya la relevancia de las recomendaciones basadas en patrones de lectura observados.

Usuario 6:

Las recomendaciones (Anexo 3, Tabla 14) fueron altamente relevantes para este usuario, quien indicó interés en todos los títulos sugeridos, con énfasis en Hijos de las estrellas y Mujeres que corren con lobos. Este caso refuerza la precisión del sistema al capturar intereses específicos en géneros y materias.

Tabla 9. Análisis de resultados de respuestas de los usuarios.

Usuario	Porcentaje de Aceptación	Detalles
Usuario 1	100%	100% de aceptación de los títulos relacionados con Julio Verne.
Usuaria 2	35%	35% de aceptación, destacando su interés en títulos relacionados con psicología.
Usuaria 3	50%	50% de aceptación, principalmente en títulos para sus hijos.
Usuaria 4	40%	40% de aceptación, con énfasis en títulos específicos y materias relacionadas.
Usuaria 5	60%	60% de aceptación, enfocada en libros para niños, aunque ya había leído algunos sugeridos.
Usuario 6	90%	90% de aceptación, mostrando afinidad hacia todos los títulos recomendados y énfasis en dos específicos.

9.2 **Análisis general de las recomendaciones**

El sistema de recomendación implementado demostró ser efectivo en mejorar la experiencia de los usuarios de las bibliotecas de Comfama, al proporcionar recomendaciones personalizadas que se alinean con sus intereses y preferencias literarias.

Un análisis detallado de los resultados reveló varios hallazgos clave:

El sistema permitió detectar hábitos recurrentes, como la preferencia por libros infantiles en usuarios adultos, lo que sugiere que muchos de ellos acceden a libros para niños a través de sus cuentas personales. También se identificaron tendencias en géneros específicos, como novelas clásicas y autoayuda, que son populares entre usuarios de 31 a 60 años.

Las recomendaciones ayudaron a diversificar el uso de la colección al promover libros menos solicitados pero relevantes para los intereses de los usuarios. Esto es particularmente valioso para equilibrar la carga sobre los títulos más demandados y mejorar la rotación de la colección.

La segmentación basada en clústeres permitió una personalización más precisa, agrupando a usuarios con intereses similares. Esto mejoró la relevancia de las recomendaciones y contribuyó a una mayor satisfacción del usuario.

10 CONCLUSIONES

El desarrollo e implementación de un sistema de recomendación de libros para las bibliotecas de Comfama representa un avance significativo en la manera en que se gestionan los servicios bibliotecarios. A través de la utilización de algoritmos de agrupamiento y técnicas de análisis de datos, este sistema no solo mejora la eficiencia en la entrega de recomendaciones personalizadas, sino que también enriquece la experiencia de los usuarios. A continuación, se presentan las conclusiones más relevantes de este trabajo:

Se logró desarrollar un sistema de recomendación, combinando técnicas de agrupamiento con análisis de similitudes basadas en un sistema colaborativo. Este enfoque permitió ofrecer recomendaciones personalizadas a los usuarios, incrementando la pertinencia de las sugerencias literarias y optimizando el uso del catálogo bibliotecario.

La aplicación de algoritmos como K-Means ha demostrado ser eficaz para agrupar libros en clústeres que reflejan características y temas similares. Esta segmentación precisa facilita la búsqueda de lecturas relevantes para los usuarios, aumentando la probabilidad de préstamos exitosos y maximizando el impacto del sistema en las bibliotecas.

La calidad de los datos fue muy importante para el desempeño del sistema de recomendación. La recolección de información precisa, como historiales de préstamos y preferencias de lectura, permitió al sistema adaptarse mejor a las necesidades cambiantes de los usuarios. Esto subraya la importancia de implementar un sistema robusto de gestión de datos, que garantice la actualización y el análisis constante de la información.

Otro aspecto importante es el impacto en la experiencia del usuario. Un sistema de recomendación bien implementado enriquece la interacción del usuario con la biblioteca, permitiéndole descubrir libros que, de otro modo, podrían haber pasado desapercibidos. Esto fomenta un entorno de lectura más activo y participativo, promoviendo el hábito de la lectura y el uso de los recursos bibliotecarios.

Finalmente, aunque la creación de un sistema de recomendación presenta diversos retos, como la selección

de algoritmos apropiados y la integración de múltiples fuentes de datos, las oportunidades que ofrece son significativas. Un sistema eficaz no solo puede optimizar el servicio bibliotecario, sino también contribuir a la democratización del acceso a la cultura y al conocimiento, reafirmando el papel fundamental de las bibliotecas en la sociedad contemporánea.

10.1 Trabajos futuros.

A medida que la tecnología evoluciona y las expectativas de los usuarios se diversifican, el sistema de recomendación propuesto en este proyecto puede beneficiarse de diversas mejoras y expansiones. Uno de los principales enfoques futuros radica en la optimización del algoritmo de recomendación. En este sentido, es posible implementar modelos más avanzados, como sistemas híbridos que combinen enfoques basados en contenido con técnicas colaborativas. De igual forma, el uso de redes neuronales o técnicas de aprendizaje profundo podría perfeccionar la capacidad del sistema para ofrecer recomendaciones aún más personalizadas y ajustadas a las necesidades de los usuarios.

Asimismo, la incorporación de nuevas fuentes de datos se presenta como una oportunidad para mejorar la precisión del sistema. En el futuro, se podría considerar la integración de datos adicionales, como el tiempo dedicado a la lectura de ciertos materiales, las búsquedas dentro del catálogo o las interacciones en redes sociales y plataformas de lectura. Además, incluir metadatos más detallados sobre los libros, como reseñas de usuarios o recomendaciones de expertos, permitiría generar sugerencias más enriquecidas y ajustadas a los intereses específicos de los usuarios.

Para asegurar la mejora continua del sistema, resulta relevante en trabajos futuros incluir un mecanismo de retroalimentación que permita a los usuarios valorar las recomendaciones y el material prestado. Este tipo de feedback sería valioso para ajustar los parámetros del sistema de manera más precisa, respondiendo a las experiencias y expectativas de los usuarios. Asimismo, la implementación de sistemas de explicabilidad, que proporcionen al usuario una comprensión clara de por qué se le ha recomendado cierto material, contribuiría a generar confianza y aumentar la interacción con el sistema.

11 REFERENCIAS BIBLIOGRÁFICAS

[1] Comfama, “Las Bibliotecas Comfama, un lugar para el encuentro, la conversación y la cultura”, Comfama - Caja de Compensación Familiar de Antioquia, Medellín, 2019.

[2] Grapheverywhere, “Sistema de recomendación, qué son, tipo y ejemplos”. Acceso: nov. 15, 2023 [En línea]. Disponible: https://www.grapheverywhere.com/sistemas-de-recomendacion-que-son-tipos-y-ejemplos/

[3] J. C. González, PI161-3-RunayaySoft - RunayaySoft: sistema de recomendación de actividades de enriquecimiento en entornos educativos. Acceso: Nov. 16, 2023 [En línea]. Disponible: http://hdl.handle.net/10554/19633.

[4] Ch. C. Aggarwal, Recommender Systems the Textbook, New York: Springer, 2016, pp. 7-20.

[5] H. S. Patricia, “Perfil del usuario de la información”, Revista de Investigación Bibliotecológica, [s.f.], pp. 16-22. Acceso: nov. 16, 2023. [En línea]. Disponible: http://www.ejournal.unam.mx/ibi/vol07-15/IBI000701502.pdf

[6] E. J. Víctor, M. B. María y V. María. “Perfil De Usuario Y Lógica Difusa: Modelo De Representación De Perfil De Usuario”, Trilogía 27, no. 37 (Jul 2015): 148–58. Acceso: nov. 17, 2023. [En línea]. Disponible: https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=asn&AN=116683404&lang=es&site=eds-live&scope=site.

[7] M. Alexander. Scientific communications and informatics. Arling-ton, Virginia: Information Resources, 1984. p. 234.

[8] M. Salvador, Machine Learning aplicado al traiding, Tesis de grado, Facultad de Ciencias Económicas y Empresariales, Universidad Pontificia Comillas, Madrid, 2019. Acceso: nov. 20, 2023. [En línea]. Disponible:

https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/27863/TFG%20Salvador%20Maceira%2c%20Macarena.pdf?sequence=1&isAllowed=y

[9] W. H. Suarez Modelo inteligente de recomendación de campañas, basado en perfilamiento de hábitos de consumo PI14104. [online]. Disponible: http://hdl.handle.net/10554/18924.

[10] X. Zhang y Y. Liu, "College library personalized recommendation system based on hybrid recommendation algorithm", CIRP, vol. 83, pp. 490-494, 2019. Acceso: Ene. 8, 2025. [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/S2212827119307401?via%3Dihub

[11] J. Daniel, J. Carlos, V. Darío, M. Julio. "Diseño e implementación de un sistema recomendador para apoyar la selección de programas académicos en educación superior" Ingeniería e innovación, vol. 2, no. 2, Jun 2013. Acceso: nov. 19, 2023. [En línea]. Disponible: https://revistas.unicordoba.edu.co/index.php/rii/article/view/766

[12] D. Sarkar. Natural Language Processing Basics en Text Analytics with Python. Bangalore, Karnataka, India: Apress, 2019, cap. 1, pp. 1-68.

[13] D. Sarkar, Text Analytics with Python, 2nd ed. Apress, 2019, pp. 501-508.

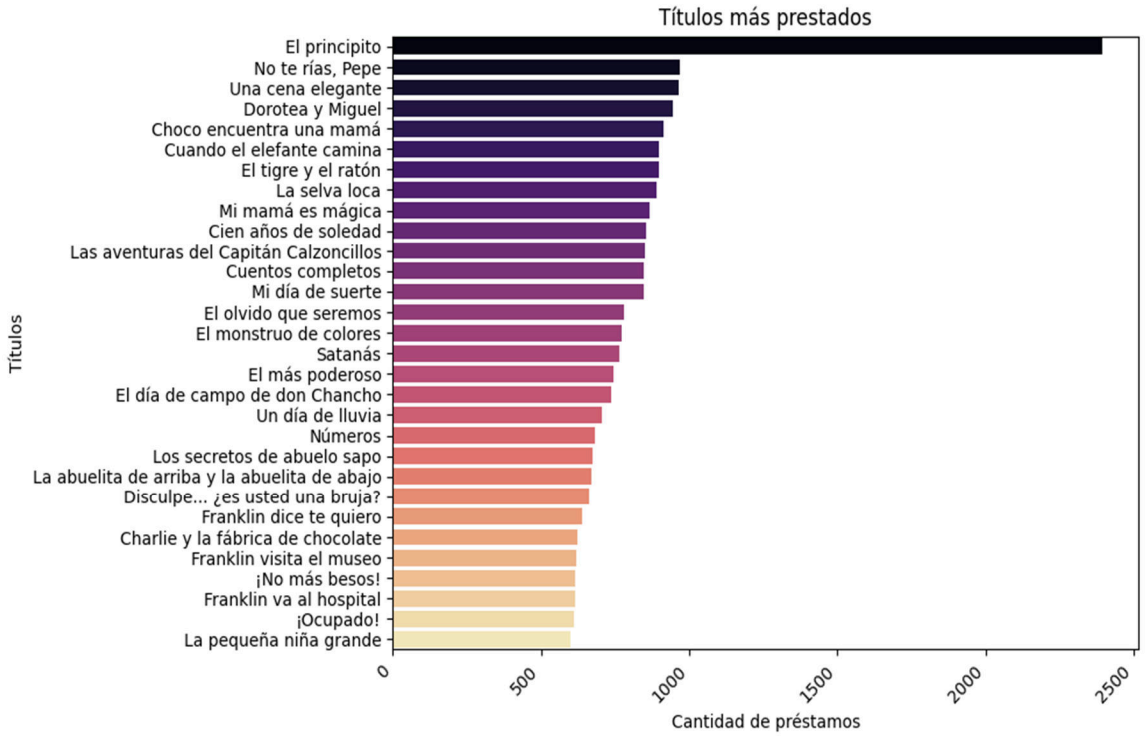
[14] D. Sarkar, Text Analytics with Python, 2nd ed. Apress, 2019, pp. 234-235.

12 ANEXOS

12.1 ANEXO 1 Análisis de la base de datos

Con estos datos se continuó con un análisis de la base de datos. Allí se verificaron los títulos más prestados:

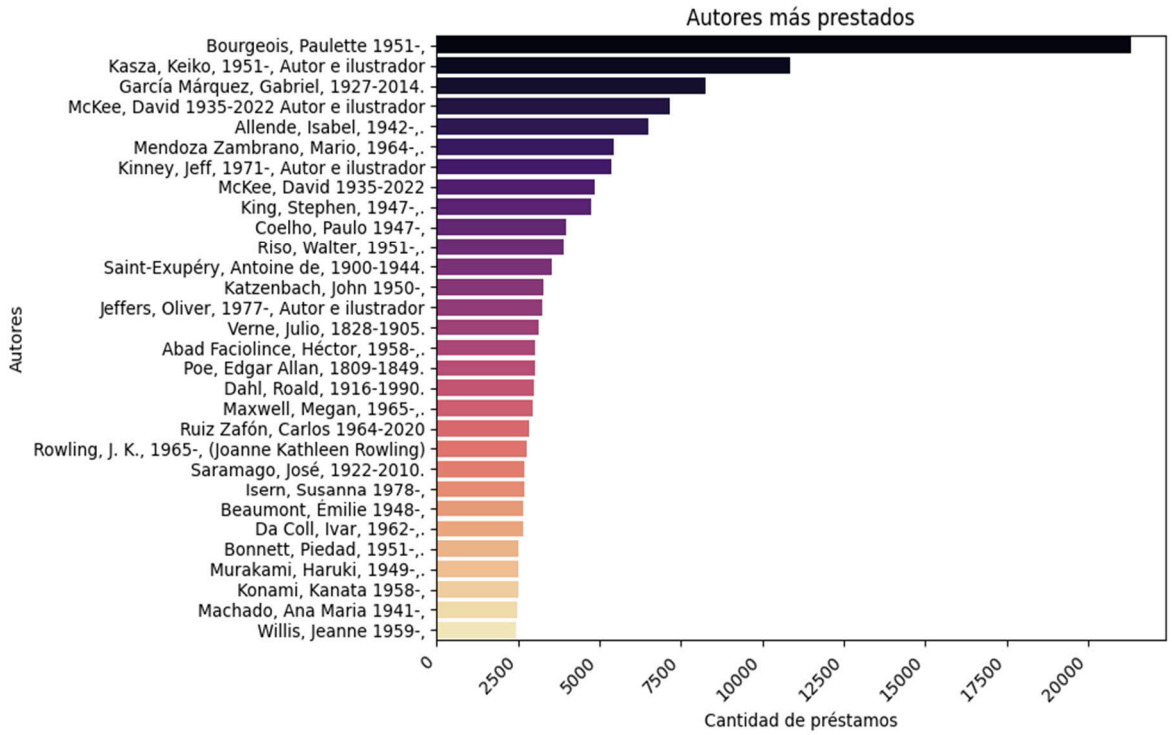
Gráfica 9. Títulos más prestados



Se puede observar que el título más prestado en las bibliotecas de Comfama es “el principito” con 2395 préstamos, seguido por “No te rías Pepe” con 969. Allí por medio de la variable “Loans (In House + Not In House)” se puede validar que estos libros fueron prestados y también consultados en las salas de las bibliotecas por los usuarios. También es posible observar que en la mayoría los títulos más prestados son de temáticas infantiles.

Con este primer acercamiento a la base surge la pregunta por los autores más consultados y prestados:

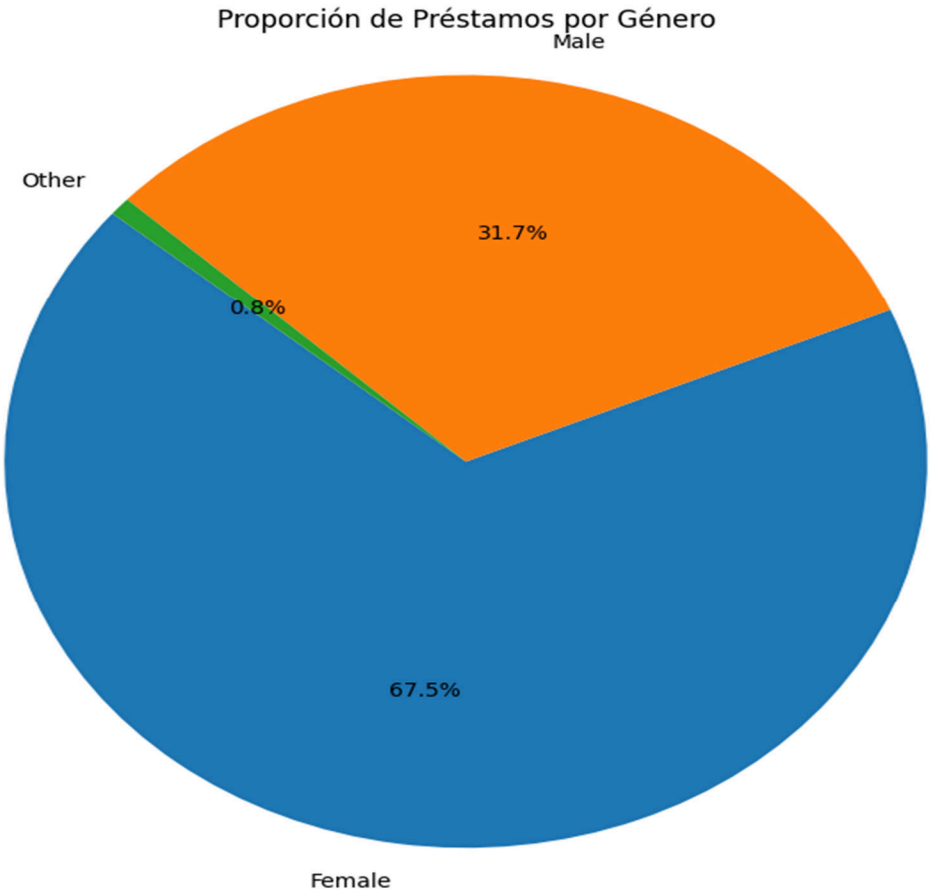
Gráfica 10. Autores más prestados.



Validando la gráfica 10 se puede decir que el autor más prestado en las bibliotecas Comfama desde el mes de septiembre del 2018 y hasta el momento es “Paulette Bourgeois” con 21305 préstamos y consultas, ella es escritora de libros infantiles, también la sigue “Keiko Kasza” con 10.831 préstamos y consultas, también escritora de libros para niños. Acá en comparación con los títulos más prestados llama la atención que se encuentren autores de libros juveniles y para adultos, como “Gabriel García Marquez”, “Isabel Allende”, “Stephen King”, “Pulo Coelho”, “Walter Riso”, entre otros. Es evidencia que, si bien es cierto que la mayoría de los títulos más prestados son para el público infantil, se puede validar que también hay préstamos de libros para públicos juveniles y adultos.

Al revisar estas dos variables de títulos y autores, es importante validar los usuarios, por ello se analiza por medio del género cómo están distribuidos:

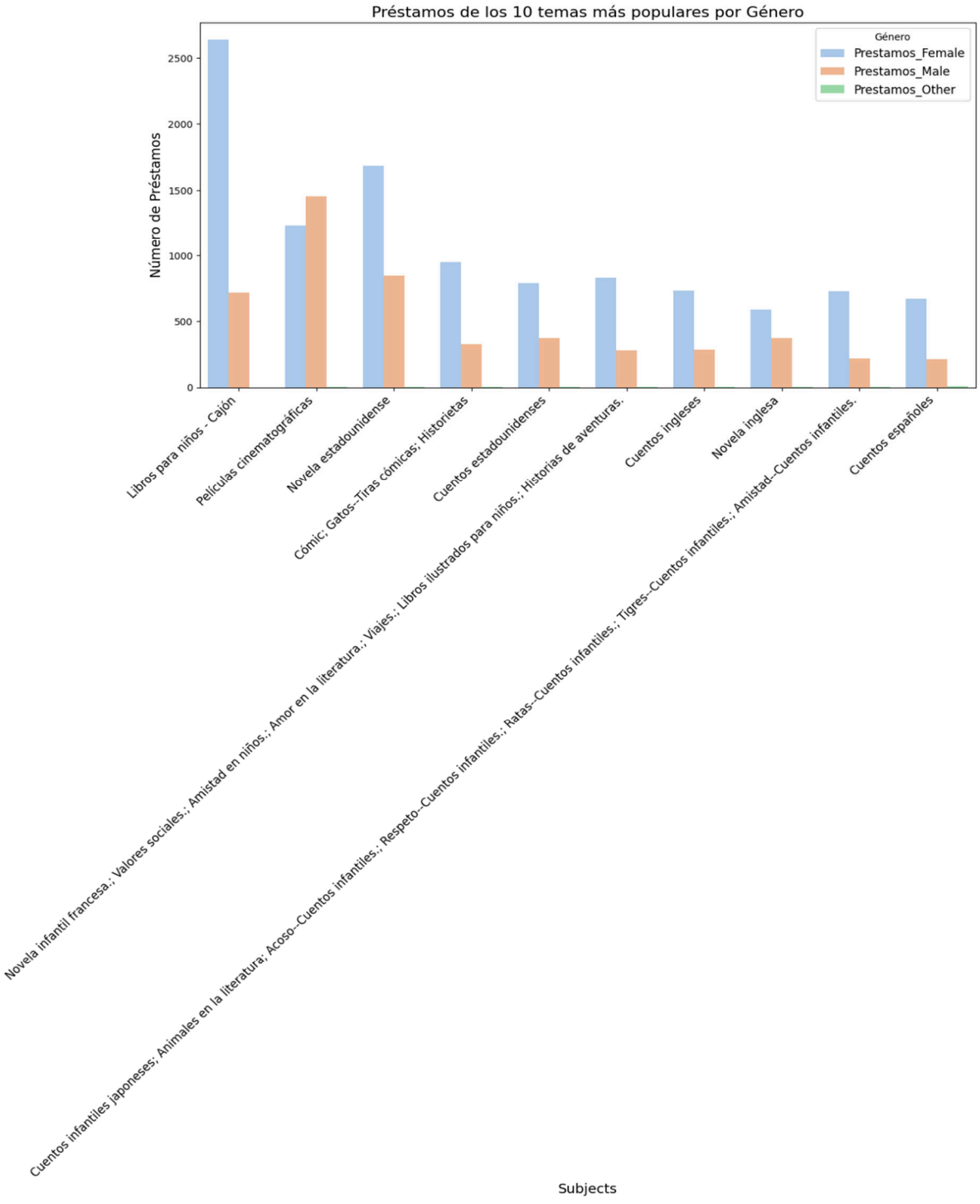
Gráfica 11. Distribución de usuarios por género.



Según la gráfica 11 el 67.5% de los usuarios de identifican como mujeres, el 31.7% como hombres y el 0.8% no se identifican ni mujer, ni hombre. Lo anterior evidencia que la mayoría de los usuarios que acceden al servicio de préstamo de libros en las bibliotecas son mujeres.

Continuando con el análisis es preciso validar los temas que más consultan o prestan los usuarios, ver gráfica 12:

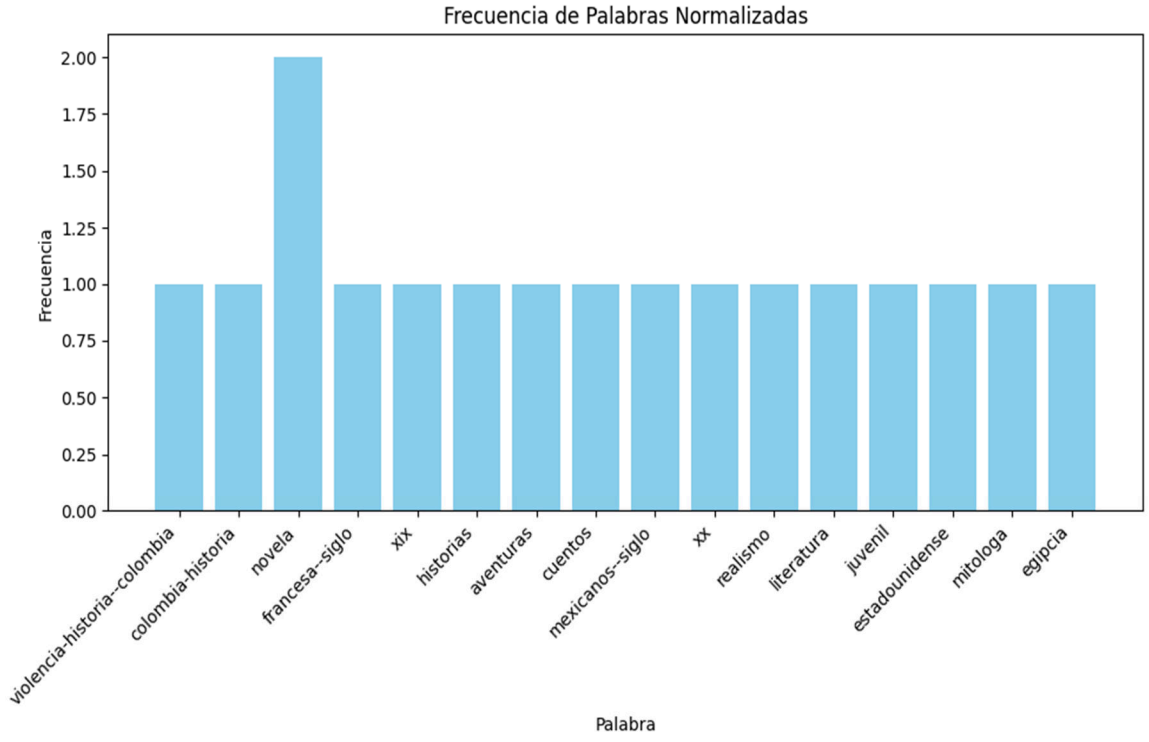
Gráfica 12. Temas más consultados o prestados por los usuarios.



Según la gráfica 12 se observa por medio de las barras que el tema más consultado sigue siendo los "libros infantiles de cajón", que son los que se usan para la promoción de la lectura en la primera infancia y es lógico que lo presten o consulten más las mujeres, ya que, según la información que comparten los mediadores y gestores en las bibliotecas, la madres en la mayoría llegan con niños en brazos o muy pequeños. Otro contraste es el tema de las "películas cinematográficas", que son más pretadas por los hombres, llama la atención que en un mundo donde hay tantas plataformas streaming, se sigan prestando o consultando películas. Siguiendo con esta gráfica se puede ver que tanto mujeres como hombres consultan o prestan novelas estadounidenses, es importante validar en las bibliotecas qué tanta colección de este tema tienen para seguir fortaleciendolo y también mirar por qué no salen otros tipos de novelas.

En este orden de ideas es preciso analizar cómo está distribuida la colección por medio de los subjects (materias)

Gráfica 13. Distribución de las materias.



Gráfica 14. Nube de palabras.

De acuerdo con las gráficas 12 y 13, la novela se destaca como uno de los temas más consultados y prestados, así como uno de los que cuenta con una mayor cantidad de ejemplares y títulos en las colecciones de las bibliotecas de Comfama. Este predominio es comprensible, ya que el gestor bibliográfico y el catálogo incluyen novelas dirigidas a públicos diversos, como niños, jóvenes y adultos, además de abarcar obras provenientes de diferentes países.

Procesamiento de los textos de la variable “subjects”

Para procesar los textos de la variable “Subjects”, se empleó un enfoque sistemático y detallado, ya que este fue uno de los pasos más demandantes del flujo de trabajo. La variable "Subjects" contenía una amplia variedad de términos, algunos de los cuales debían ser normalizados para asegurar una representación consistente en los modelos de recomendación.

El proceso comenzó con la importación de las librerías necesarias para el procesamiento del lenguaje natural. Se utilizó nltk, que ofrece herramientas fundamentales para tareas de tokenización, eliminación de palabras vacías (stopwords) y lematización. De igual forma, se emplearon expresiones regulares (re) para realizar la limpieza y estructuración del texto. La lematización se llevó a cabo mediante el uso de WordNetLemmatizer, que reduce las palabras a su forma base, asegurando que variaciones como “correr” y “corría” se consideraran como la misma palabra.

El primer paso fue la descarga de los recursos necesarios, como las listas de stopwords y el lematizador de NLTK. Luego, se aplicó una limpieza básica al texto, convirtiéndolo a minúsculas y eliminando caracteres no alfabéticos (a excepción de los guiones, que se mantuvieron). Para la tokenización, se usaron expresiones regulares que permitieron dividir el texto en palabras individuales, asegurando que solo se mantuvieran los términos relevantes para el análisis.

Posteriormente, cada término fue lematizado, lo que significa que se redujo a su raíz o forma base, eliminando las conjugaciones y variaciones. Este paso ayudó a reducir la complejidad del texto y a mejorar la calidad de los datos para su posterior análisis. Además, se eliminaron las stopwords, esas palabras que, aunque son comunes en el lenguaje, no aportan valor semántico para los modelos de recomendación.

12.2 ANEXO 2 Resultados de recomendaciones con 5 clusters

A continuación, se muestran los resultados de las recomendaciones dividiendo los datos en 5 clusters

	Libro	Conteo	Author	Subjects
0	Satanás	469	Mendoza Zambrano, Mario, 1964-,.	Novela colombiana.; Asesinos--Bogotá (Colombia...
1	Cuentos completos	402	Fuentes, Carlos 1928-2012	Cuentos mexicanos; Realidad en la literatura; ...
2	1984	368	NaN	Novela gráfica.; Literatura--Adaptaciones.; Gu...
3	El cementerio de los libros olvidados. La somb...	331	Ruiz Zafón, Carlos 1964-2020	Novela histórica; Amor en la literatura; Novel...
4	El principito	314	Saint-Exupéry, Antoine de, 1900-1944, Autor e ...	Novela infantil francesa.; Valores sociales.; ...
5	El psicoanalista	296	Katzenbach, John, 1950-,.	Novela estadounidense.; Novela de suspenso.; P...
6	El retrato de Dorian Gray	275	Wilde, Oscar, 1854-1900.	Novela inglesa; Belleza personal--Novela.; Juv...
7	El cementerio de los libros olvidados. El jueg...	263	Ruiz Zafón, Carlos 1964-2020	Novela española; Amor en la literatura--Novela...

Ilustración 5. Sugerencias del usuario de prueba 1 con 5 clusters.

	Libro	Conteo	Author	Subjects
0	Satanás	469	Mendoza Zambrano, Mario, 1964-,.	Novela colombiana.; Asesinos--Bogotá (Colombia...
1	Cuentos completos	402	Fuentes, Carlos 1928-2012	Cuentos mexicanos; Realidad en la literatura; ...
2	1984	368	NaN	Novela gráfica.; Literatura--Adaptaciones.; Gu...
3	El cementerio de los libros olvidados. La somb...	331	Ruiz Zafón, Carlos 1964-2020	Novela histórica; Amor en la literatura; Novel...
4	El principito	314	Saint-Exupéry, Antoine de, 1900-1944, Autor e ...	Novela infantil francesa.; Valores sociales.; ...
5	El psicoanalista	296	Katzenbach, John, 1950-,.	Novela estadounidense.; Novela de suspenso.; P...
6	El retrato de Dorian Gray	275	Wilde, Oscar, 1854-1900.	Novela inglesa; Belleza personal--Novela.; Juv...
7	El cementerio de los libros olvidados. El jueg...	263	Ruiz Zafón, Carlos 1964-2020	Novela española; Amor en la literatura--Novela...
8	Crimen y castigo	261	Dostoyevski, Fiódor Mijáilovich, 1821-1881.	Novela rusa--Siglo XIX.; Novela psicológica.; ...
9	Fahrenheit 451	250	Bradbury, Ray 1920-2012	Novela estadounidense--Siglo XX.; Censura--nov...

Ilustración 6. Sugerencias del usuario de prueba 2 con 5 clusters.

Si observamos los resultados de las recomendaciones en las ilustraciones 6 y 7. Se puede validar que para estos usuarios que no están en el mismo cluster, se hacen las mismas recomendaciones, debido a esto se descarta utilizar la cantidad de 5 clusters.

12.3 ANEXO 3 Resultados de recomendaciones por usuario

Tabla 10. Títulos recomendados usuario 2.

#	Libro	Conteo	Author	Subjects
0	Pensar bien, sentirse bien : nada justifica el...	4	Riso, Walter, 1951-,.	Psicología cognitiva; Terapia cognitiva.; Auto...
1	El juego : 77 tácticas para alcanzar tu paz in...	2	Corzo, Silvia. 1973-,	Autorrealización (Psicología); Paz interior; E...
2	Las razones del amor : el sentido de nuestras ...	2	Frankfurt, Harry G., 1929-2023.	Amor--Aspectos psicológicos; Conducta afectiva...
3	Manual para no morir de amor : diez principios...	2	Riso, Walter, 1951-,.	Amor--Aspectos psicológicos; Afecto; Amor (Psi...
4	Te amo... pero soy feliz sin ti : cómo vivir ...	2	Jaramillo, Jaime 1956-,	Conducta dependiente; Relaciones humanas
5	Maravillosamente imperfecto, escandalosamente ...	2	Riso, Walter, 1951-,.	Superación personal; Autoayuda; Crecimiento pe...

6	El duelo : crecer en la pérdida	2	Nevado, Manuel	Muerte; Pérdida (Psicología); Duelo (Muerte); ...
7	Trilogía de la nube blanca. En el país de la n...	1	Sarah Lark (Seudónimo de Christiane Gohl)	Novela alemana; Literatura alemana; Amor en la...
8	La esposa entregada una novedosa estrategia pa...	1	Doyle, Laura. 1967-,	Matrimonio; Relaciones de pareja; Relaciones f...
9	Melany : historia de una anoréxica	1	Harris, Dorothy Joan. 1931-,	Novela juvenil canadiense; Trastornos del apet...
10	¡Habla!	1	Halse Anderson, Laurie. 1961-,	Novela juvenil estadounidense; Marginados soci...
11	Resultados extraordinarios técnicas y estrateg...	1	Stamateas, Bernardo 1965-,	Autoayuda; Exito; Mejoramiento continuo; Soluc...
12	Suicidio involuntario	1	Burgell, Jaume. 1966-,	Novela infantil española; Muerte en la literat...
13	120 preguntas y respuestas para ser mejores pe...	1	Alegría, Cecilia. 1958-,	Crecimiento personal; Relaciones familiares; R...
14	Tanatología: la inteligencia emocional y el p...	1	Castro González, María del Carmen.	Muerte--Aspectos psicológicos; Técnicas de aut...
15	Cómo afrontar la muerte de un ser querido	1	Markham, Ursula	Muerte; Duelo; Consuelo; Perdida (Psicología);...
16	Una vida genial: sana tu mente, fortalece tu ...	1	Lugavere, Max, 1982-,	Cerebro.; Nutrición.; Enfermedades.; Mente y c...

17	La llave: el secreto perdido para atraer todo...	1	Vitale, Joe, 1953-	Superación personal; Exito; Conducta (Etica); ...
18	La psicóloga	1	Flood, Helene, 1982-	Novela noruega; Personas desaparecidas--Novela...
19	La fidelidad es mucho más que amor: cómo prev...	1	Riso, Walter, 1951-,.	Adulterio; Relaciones de pareja; Infidelidad--...

Tabla 11. Títulos recomendados usuario 3.

#	Libro	Conteo	Author	Subjects
0	¡No más besos!	5	Chichester Clark, Emma 1955-	Cuentos infantiles ingleses; Besos--Cuentos in...
1	El más poderoso	4	Kasza, Keiko, 1951-, Autor e ilustrador	Cuentos infantiles; Literatura infantil japone...
2	¿Por qué lloramos?	3	Pintadera, Fran, 1982-,.	Cuentos infantiles españoles; Emociones--Cuent...
3	Nano y sus amigos	3	Da Coll, Ivar, 1962-,.	Cuentos infantiles colombianos; Amistad en la ...
4	Mi mamá	3	Browne, Anthony, 1946-, (Anthony Edward Tudor ...	Cuentos infantiles ingleses; Madre e hijo--Cue...
5	Una cena elegante	3	Kasza, Keiko, 1951-, Autor e ilustrador	Animales en la literatura; Cuentos infantiles;...

6	¿Cómo era yo cuando era un bebé?	3	Willis, Jeanne 1959-,	Cuentos infantiles ingleses; Niños--Cuentos in...
7	Mi mamá es mágica	3	Norac, Carl, 1960 -,.	Cuentos infantiles franceses.; Madres e hijas-...
8	¡Cuidado! ¡Palabra terrible!	3	Schreiber-Wicke, Edith 1943-,	Cuentos infantiles; Literatura infantil aleman...
9	Cuentos pintados	3	Pombo, Rafael, 1833-1912.	Cuentos infantiles colombianos; Cuentos popula...
10	El tigre y el ratón	3	Kasza, Keiko, 1951-, Autor Ilustrador	Cuentos infantiles japoneses; Animales en la l...
11	El perro que quiso ser lobo Keiko Kasza; Tradu...	3	Kasza, Keiko, 1951-, Autora e ilustradora	Cuentos infantiles; Literatura infantil japone...
12	Infinito : los ciclos mágicos del universo	3	Romero Mariño, Soledad	Evolución.; Historia natural; Cosmología.; Cos...
13	El monstruo de colores	3	Llenas, Anna, 1977-,.	Cuentos infantiles españoles; Monstruos--Cuent...
14	Mi día de suerte	2	Kasza, Keiko, 1951-, Autor e ilustrador	Cuentos infantiles japoneses; Inteligencia--Cu...
15	Franklin juega al fútbol	2	Bourgeois, Paulette 1951-,	Cuentos infantiles canadienses; Tortugas--Cuen...

16	Sofía y las lechuguitas	2	Green, Ilya, 1976-, Autor e ilustrador	Cuentos infantiles franceses; Verdad y mentita...
17	Franklin es un mandón	2	Bourgeois, Paulette 1951-,	Cuentos infantiles canadienses; Tortugas--Cuen...
18	La extraña mamá	2	Baek, Heena, 1971-, Autor e ilustrador	Cuentos infantiles coreanos; Madres e hijos--C...
19	Un mar de tristeza	2	Ludica, Anna	Cuentos infantiles; Peces--Cuentos infantiles;..

Tabla 12. Títulos recomendados usuario 4.

#	Libro	Conteo	Author	Subjects
0	Más fuerte que la adversidad : cómo afrontar l...	2	Riso, Walter, 1951-,.	Personalidad.; Conducta humana.; Ansiedad; Est...
1	Momo	2	Ende, Michael 1929-1995	Novela juvenil Alemana; Novela Fantástica; Ami...
2	Maravillosamente imperfecto, escandalosamente ...	2	Riso, Walter, 1951-,.	Superación personal; Autoayuda; Crecimiento pe...
3	Historia de la sexualidad. La voluntad de sabe...	2	Foucault, Michel 1926-1984	Psicología de la sexualidad; Conducta sexual; ...

4	Odisea	2	Homero.	Homero--Crítica e interpretación.; Literatura ...
5	Hazte dueño de ti : una guía para vivir con pr...	2	Martinez, Efrén.	Autorrealización (Psicología); Amor.; Conducta...
6	Mujeres conscientes : diez movimientos para la...	2	Echegoyen, Agustina. 1988-,	Mujeres - Conducta de vida; Psicología de la m...
7	Dioses sois	2	Miranda, Mercedesi	Jesucristo - Enseñanzas; Jesucristo--Vida cris...
8	Hombres sin mujeres	1	Murakami, Haruki, 1949-,.	Cuentos japoneses; Mujeres en la literatura; A...
9	Ilustración de moda : dibujo plano	1	Lleonart, Aitana	Diseño de vestidos; Moda--Dibujo; Corte y Conf...
10	Historia secreta de la música	1	Marín, Alejandro.	Música--Historia; Canciones--Historia; Música-...
11	Diciembre, otra vez	1	Cruz, Santiago, 1976-	Cruz, Santiago,1976--Autobiografía; Cruz, Sant...
12	Frankenstein	1	NaN	Shelley, Mary,--1797-1851.; Novela juvenil ing...
13	Black music : free jazz y conciencia negra 195...	1	Jones, Leroi	Música de Jazz--1957-1967--Estados unidos; Mús...

14	Veronika decide morir	1	Coelho, Paulo 1947-,	Literatura brasileña; Vida--Aspectos psicológi...
15	Ilustración de moda	1	Morris, Bethan	Diseño de modas; Moda--Dibujo; Dibujo de modas...
16	Punto a punto	1	Machado, Ana Maria, 1941-,.	Cuentos infantiles brasileños--Siglo XXI.; Muj...
17	La condición humana	1	Arendt, Hannah, 1906-1975.	Psicología social; interacción social; Filosof...
18	Guía completa de bordado : 500 combinaciones d...	1	Bothell, Valerie, 1962-	Bordado; Labores de aguja; Punto de cruz; Patc...
19	Y si me gustara morir	1	Cardoso Martins, Rui, 1967-,.	Novela portuguesa--Siglo XXI.; Suicidio--Novel...

Tabla 13. Títulos recomendados usuario 5.

#	Libro	Conteo	Author	Subjects
0	El infinito en un junco : la invención de los ...	10	Vallejo, Irene, 1979-,.	Ensayo español; Libros - Historia; Libros - Hi...

1	Guillermo Jorge Manuel José	6	Fox, Mem, 1946-,.	Cuentos infantiles australianos.; Emociones in...
2	Listo para cualquier cosa	5	Kasza, Keiko, 1951-, Autor e ilustrador	Cuentos infantiles japoneses; Patos--Cuentos i...
3	Haiku	5	NaN	Poesía japonesa; Naturaleza en la poesía; Emoc...
4	De vuelta a casa	5	Jeffers, Oliver, 1977-, Autor e ilustrador	Cuentos infantiles australianos; Literatura in...
5	Los dinosaurios	5	Díaz, Ana María	Cuentos infantiles colombianos.; Dinosaurios--...
6	Un día de lluvia	5	Rueda, Claudia	Cuentos infantiles colombianos; Gatos--Cuentos...
7	Cultivar un jardín en miniatura : terrarios, j...	5	Farrell, Holly	Jardines - Diseño; Horticultura; Terrarios; Ja...
8	Números	5	Donaldson, Julia 1948-,	Números--Educación primaria.; Aprender a conta...
9	Lo que construiremos : planes para nuestro fut...	5	Jeffers, Oliver, 1977-,.	Cuentos infantiles australianos; Amor--Cuentos...
10	Lo que no tiene nombre	5	Bonnett, Piedad, 1951-,.	Segura Bonnett, Daniel, 1983-2011--Suicidio; N...

11	El árbol rojo	4	Rosen, Michael, 1946-,.	Cuentos infantiles australianos; Condición hum...
12	Montañas : cumbres, montes y valles de la tierra	4	Cassany, Mia 1986-,	Ecología montañosa; Flora andina; Flora alpina...
13	¿Qué puedo esperar? el libro de las preguntas	4	Teckentrup, Britta, 1969-,.	Preguntas y respuestas; Curiosidades y maravil...
14	Perdido y encontrado	4	Jeffers, Oliver, 1977-, Autor e ilustrador	Cuentos infantiles australianos; Pingüinos--Cu...
15	Cómo atrapar una estrella	4	Jeffers, Oliver, 1977-, Autor e ilustrador	Cuentos infantiles australianos; Cuentos fantá...
16	Lobito aprende a ser malo	4	Whybrow, Ian 1941-,	Cuentos infantiles ingleses; Diversidad--Cuent...
17	Malena Ballena	4	Cali, David 1972-,	Cuentos infantiles italianos; Autoestima--Cuen...
18	Hansel y Gretel	4	NaN	Grimm, Jacob--1785-1863.; Grimm, Wilhelm--1786...
19	El bosque que más quiero	4	Berloso Clara`, Laia. 1991-, Autor e ilustrador	Cuentos infantiles españoles; Niños y medio am...

Tabla 14. Títulos recomendados usuario 6.

#	Libro	Conteo	Author	Subjects

0	El olvido que seremos	2	Abad Faciolince, Héctor, 1958-,.	Abad Gómez, Héctor, 1921-1987--Biografías; Nov...
1	¡Que viva la música!	1	Caicedo, Andrés, 1951-1977.	Novela colombiana--Siglo XXI.; Música y litera...
2	Bocababa	1	Valle`s, Tina, 1976-	Cuentos infantiles; Niños y mascotas--Cuentos ...
3	El último encuentro	1	Márai, Sándor 1900-1989	Novela húngara; Amistad--Novela; Hombres y muj...
4	Las brujas	1	Dahl, Roald, 1916-1990.	Novela infantil inglesa; Abuelos--novela; Nove...
5	Naturaleza	1	Emerson, Ralph Waldo, 1803-1882.	Ensayos estadounidenses; Naturaleza--Ensayos; ...
6	Cartas a un buscador de sí mismo	1	Thoreau, Henry David, 1817-1862.	Thoreau, Henry David,--1817-1862--Corresponden...
7	La teoría feminista del margen al centro	1	Bell hooks (Seudónimo de Gloria Jean Watkins) ...	Feminismo; Feministas; Derechos de la mujer; T...
8	Pedagogía de la esperanza : un reencuentro con...	1	Freire, Paulo 1921-1997	Freire, Paulo, 1921-1997; Educación y sociedad...

9	Mientras el cielo esté vacío	1	Vélez Saldarriaga, Marta Cecilia, 1954-2019.	Novela colombiana; Empatía--Novela; Mujeres en...
10	El sol que nunca vimos	1	Restrepo Cuartas, Jaime 1944-,	Novela colombiana; Literatura realista; Violen...
11	¿Quién soy yo para juzgar?	1	Francisco, Papa, 1936- (Jorge Mario Bergoglio)	Ética cristiana; Iglesia católica--Problemas s...
12	Los niños de Asperger : el exterminador nazi d...	1	Sheffer, Edith	Asperger, Hans, 1906-1980; Asesinos - Guerra m...
13	Mujeres que corren con los lobos	1	Estés, Clarissa Pinkola, 1945-,.	Psicología de la mujer; Sexualidad femenina; A...
14	El país de los niños perdidos	1	Martín Garzo, Gustavo, 1948-, autor personal	Novela juvenil española--Siglo XXI.; Novela fa...
15	Diferentes diferencias	1	Penide, Silvia	Personalidad; Diferencias individulaes; Respet...
16	Somos muchos	1	Ordóñez, Nicolás, 1996- Autor e ilustrador	Novela gráfica; Vida cotidiana--Novela gráfica...
17	Hijos de las estrellas	1	Ruíz, María Teresa, 1946-	Astronomía; Ciencias del espacio; Espacio y ti...

18	Celos : una historia cultural	1	Sissa, Giulia 1954-,	Celos; Celos--Historia; Relaciones de pareja; ...
19	Los años terribles	1	Yolanda Reyes Villamizar	Novela juvenil colombiana; Vida cotidiana--Nov...

12.4 ANEXO 4. Código fuente

https://colab.research.google.com/drive/15UczNoYnzGS1-AXqeHmSZy3w4Lr1J_wR?usp=sharing

ANEXO 4. CÓDIGO FUENTE Y RESULTADOS

Sistema de recomendación Bibliotecas Comfama

```
# Sistema de recomendación Bibliotecas Comfama

import numpy as np
import pandas as pd
import sklearn as sk
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from gensim.models import Word2Vec
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Montar Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Cargar los datos
data = pd.read_csv("/content/drive/MyDrive/datas/data_Comfama_20241.csv")
data.head()
```

	Primary Identifier	Full Name	Gender	Birth Date	City	ISBN	Title	Author	Subjects	Loans (In House + Not In House)
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	01/30/1998	MEDELLÍN	9789584291431; 9584291432	Crea & divaga : vida y reflexiones de Jeff Bezos	Bezos, Jeff. 1964-,	Bezos, Jeff.--1964---Relatos personales; Amazo...	1
1	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	01/30/1998	MEDELLÍN	9789589007938; 9589007937	Sanación con cristales : las claves para inici...	Cuellar B., Andrea	Cristales--Uso terapéutico; Terapias de sanaci...	1
2	1035973109	RODRIGUEZ LAVERDE	Female	10/21/2005	Itagui	9789584287724; 9584287729	Colombia una historia mínima : una mirada	Melo, Jorge Orlando	Violencia-Historia--Colombia;	1

```
# Número de registros y atributos
shape = data.shape
print("Número de registros y atributos:", shape)
```

```
Número de registros y atributos: (1457562, 10)
```

```
nan_values = data.isna().sum()
print("Valores NaN:\n", nan_values)
```

```
Valores NaN:
 Primary Identifier          0
Full Name                    0
Gender                  168472
Birth Date                   0
City                     20914
ISBN                     57170
Title                        1
```

```
Author                          149974
Subjects                         22067
Loans (In House + Not In House)      0
dtype: int64
```

```
data = data.dropna()
```

```
nan_values = data.isna().sum()
print("Valores NaN después de eliminar filas:\n", nan_values)
```

```
Valores NaN después de eliminar filas:
 Primary Identifier                0
Full Name                         0
Gender                            0
Birth Date                        0
City                              0
ISBN                              0
Title                             0
Author                            0
Subjects                          0
Loans (In House + Not In House)   0
dtype: int64
```

```
# Número de registros y atributos
shape = data.shape
print("Número de registros y atributos:", shape)
```

```
Número de registros y atributos: (1114880, 10)
```

```
# Tipo de los atributos
print("Tipos de los atributos:", data.dtypes)
```

```
Tipos de los atributos: Primary Identifier                object
Full Name                         object
Gender                            object
Birth Date                        object
City                              object
ISBN                              object
Title                             object
Author                            object
Subjects                          object
Loans (In House + Not In House)    int64
dtype: object
```

Análisis estadístico

```
# 1. Estadísticas Básicas de Préstamos
loans_by_title_author = data.groupby(['Title', 'Author']).size().reset_index(name='Total_Loans')
loans_by_title_author = loans_by_title_author.sort_values(by='Total_Loans', ascending=False)
```

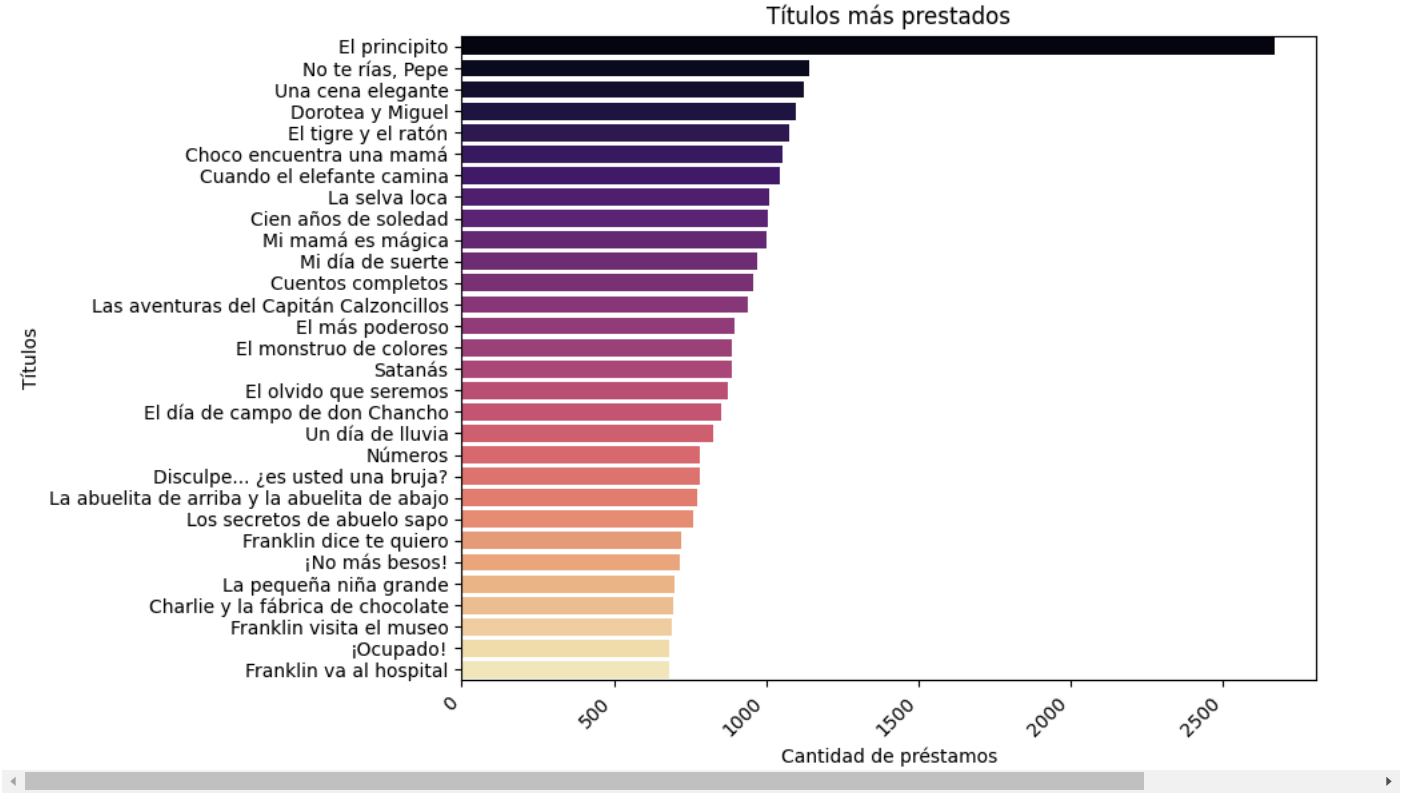
```
titulo_prestamos = data.groupby('Title')['Loans (In House + Not In House)'].sum()

# Ordenar los títulos por la cantidad de préstamos en casa en orden descendente
titulo_prestamos = titulo_prestamos.sort_values(ascending=False)

# Mostrar los títulos más prestados
print("Títulos más prestados:")
print(titulo_prestamos.head(30))
```

```
Títulos más prestados:
Title
El principito                                 2671
No te rías, Pepe                              1141
Una cena elegante                             1123
Dorotea y Miguel                              1096
El tigre y el ratón                           1074
Choco encuentra una mamá                      1055
Cuando el elefante camina                     1046
La selva loca                                 1012
Cien años de soledad                          1004
Mi mamá es mágica                             1001
Mi día de suerte                               970
```

```
Cuentos completos                          958
Las aventuras del Capitán Calzoncillos     942
El más poderoso                            896
El monstruo de colores                     889
Satanás                                    886
El olvido que seremos                      874
El día de campo de don Chancho             853
Un día de lluvia                           826
Números                                    784
Disculpe... ¿es usted una bruja?           781
La abuelita de arriba y la abuelita de abajo    774
Los secretos de abuelo sapo                760
Franklin dice te quiero                    720
¡No más besos!                             715
La pequeña niña grande                     698
Charlie y la fábrica de chocolate          695
Franklin visita el museo                   690
¡Ocupado!                                  682
Franklin va al hospital                    680
Name: Loans (In House + Not In House), dtype: int64
```

```
# Visualización: Gráfico de barras
import matplotlib.pyplot as plt
import seaborn as sns


top_titulos = titulo_prestamos.head(30)


plt.figure(figsize=(10, 6))
sns.barplot(x=top_titulos.values, y=top_titulos.index, palette="magma")
plt.xlabel('Cantidad de préstamos')
plt.ylabel('Títulos')
plt.title('Títulos más prestados')
plt.xticks(rotation=45, ha='right')  # Rotar los nombres de los títulos para una mejor legibilidad
plt.tight_layout()  # Ajustar el diseño para evitar que los nombres de los títulos se superpongan
plt.show()
```

```
<ipython-input-11-be86c2ce75c6>:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend

  sns.barplot(x=top_titulos.values, y=top_titulos.index, palette="magma")
```



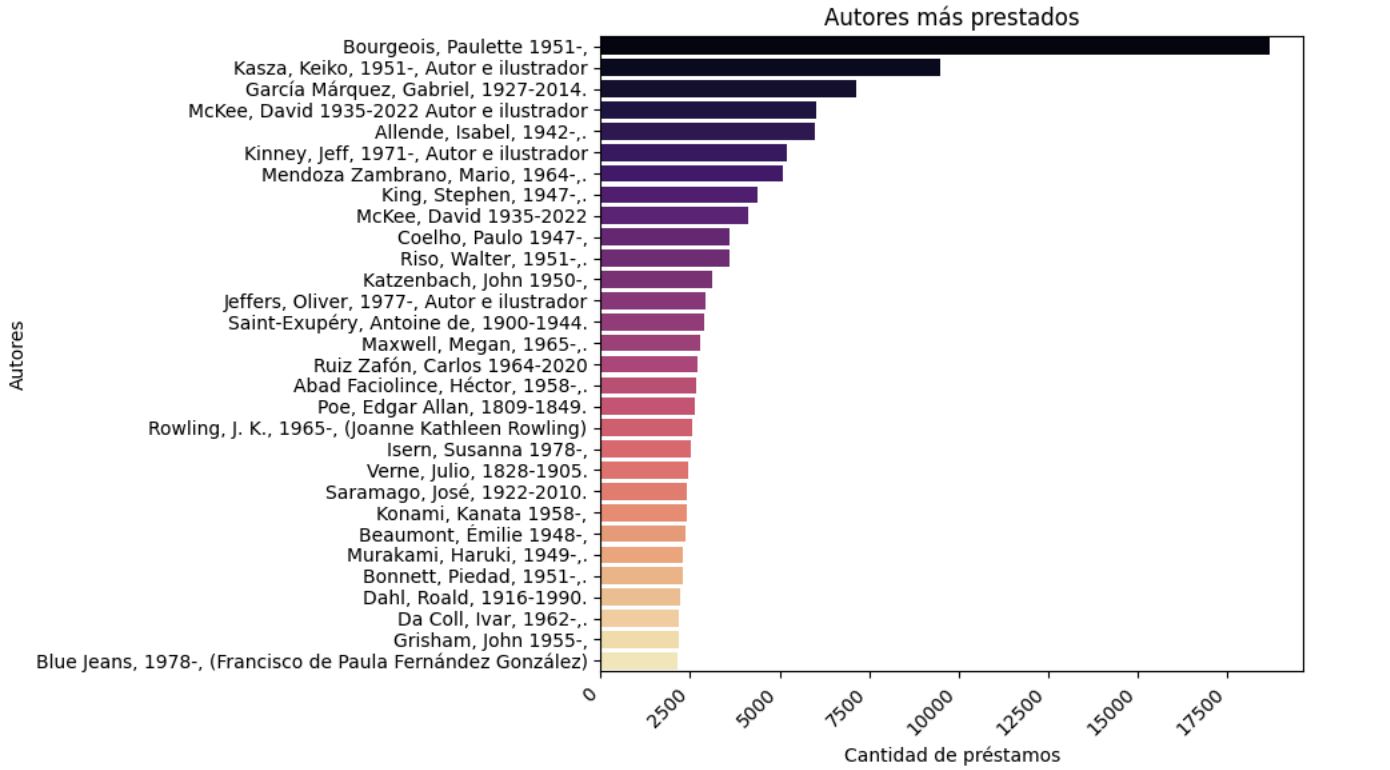

```
autor_prestamos = data.groupby('Author')['Loans (In House + Not In House)'].sum()

# Ordenar los títulos por la cantidad de préstamos en casa en orden descendente
```

```
autor_prestamos = autor_prestamos.sort_values(ascending=False)

# Mostrar los títulos más prestados
print("Títulos más prestados:")
print(autor_prestamos.head(30))
```

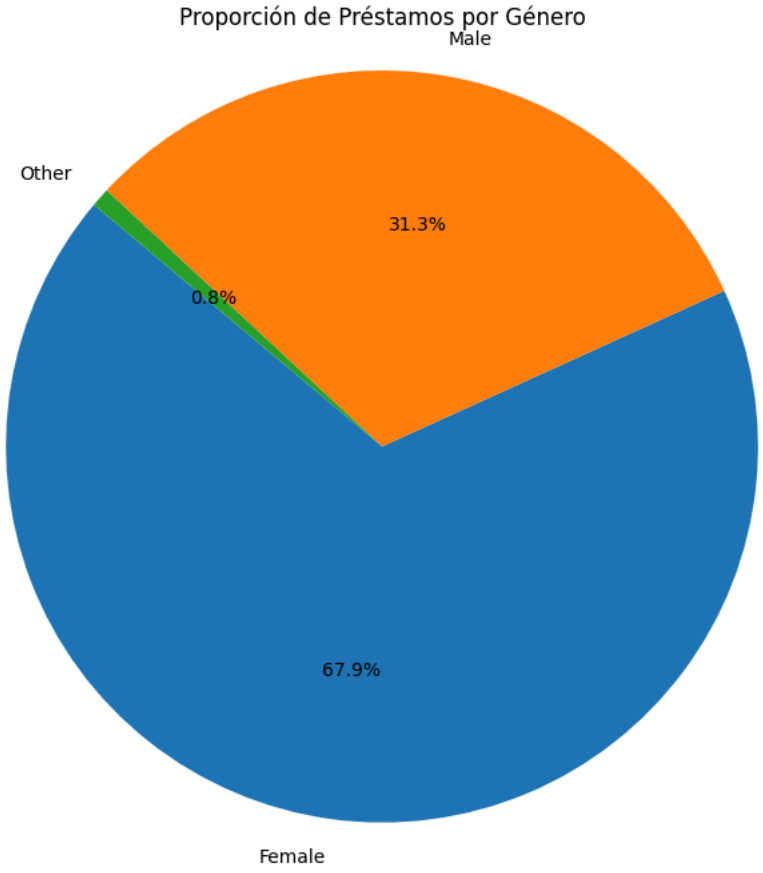
```
Títulos más prestados:
Author
Bourgeois, Paulette 1951-,                                  18675
Kasza, Keiko, 1951-, Autor e ilustrador                      9466
García Márquez, Gabriel, 1927-2014.                          7144
McKee, David 1935-2022 Autor e ilustrador                    6012
Allende, Isabel, 1942-,.                                     5993
Kinney, Jeff, 1971-, Autor e ilustrador                      5219
Mendoza Zambrano, Mario, 1964-,.                             5076
King, Stephen, 1947-,.                                       4391
McKee, David 1935-2022                                       4114
Coelho, Paulo 1947-,                                         3601
Riso, Walter, 1951-,.                                        3598
Katzenbach, John 1950-,                                      3099
Jeffers, Oliver, 1977-, Autor e ilustrador                   2935
Saint-Exupéry, Antoine de, 1900-1944.                        2878
Maxwell, Megan, 1965-,.                                      2788
Ruiz Zafón, Carlos 1964-2020                                 2710
Abad Faciolince, Héctor, 1958-,.                             2675
Poe, Edgar Allan, 1809-1849.                                 2647
Rowling, J. K., 1965-, (Joanne Kathleen Rowling)             2564
Isern, Susanna 1978-,                                        2514
Verne, Julio, 1828-1905.                                     2436
Saramago, José, 1922-2010.                                   2427
Konami, Kanata 1958-,                                        2426
Beaumont, Émilie 1948-,                                      2366
Murakami, Haruki, 1949-,.                                    2291
Bonnett, Piedad, 1951-,.                                     2286
Dahl, Roald, 1916-1990.                                      2234
Da Coll, Ivar, 1962-,.                                       2200
Grisham, John 1955-,                                         2168
Blue Jeans, 1978-, (Francisco de Paula Fernández González)   2144
Name: Loans (In House + Not In House), dtype: int64
```

```
# Visualización: Gráfico de barras
import matplotlib.pyplot as plt
import seaborn as sns


top_autores = autor_prestamos.head(30)


plt.figure(figsize=(10, 6))
sns.barplot(x=top_autores.values, y=top_autores.index, palette="magma")
plt.xlabel('Cantidad de préstamos')
plt.ylabel('Autores')
plt.title('Autores más prestados')
plt.xticks(rotation=45, ha='right')  # Rotar los nombres de los títulos para una mejor legibilidad
plt.tight_layout()  # Ajustar el diseño para evitar que los nombres de los títulos se superpongan
plt.show()
```

```
<ipython-input-13-b0084d2930dd>:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend

  sns.barplot(x=top_autores.values, y=top_autores.index, palette="magma")
```




```
# 2. Préstamos por Género
loans_by_gender = data.groupby('Gender').size().reset_index(name='Total_Loans')


plt.figure(figsize=(8, 8))
plt.pie(loans_by_gender['Total_Loans'], labels=loans_by_gender['Gender'], autopct='%1.1f%%', startangle=140)
plt.title('Proporción de Préstamos por Género')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

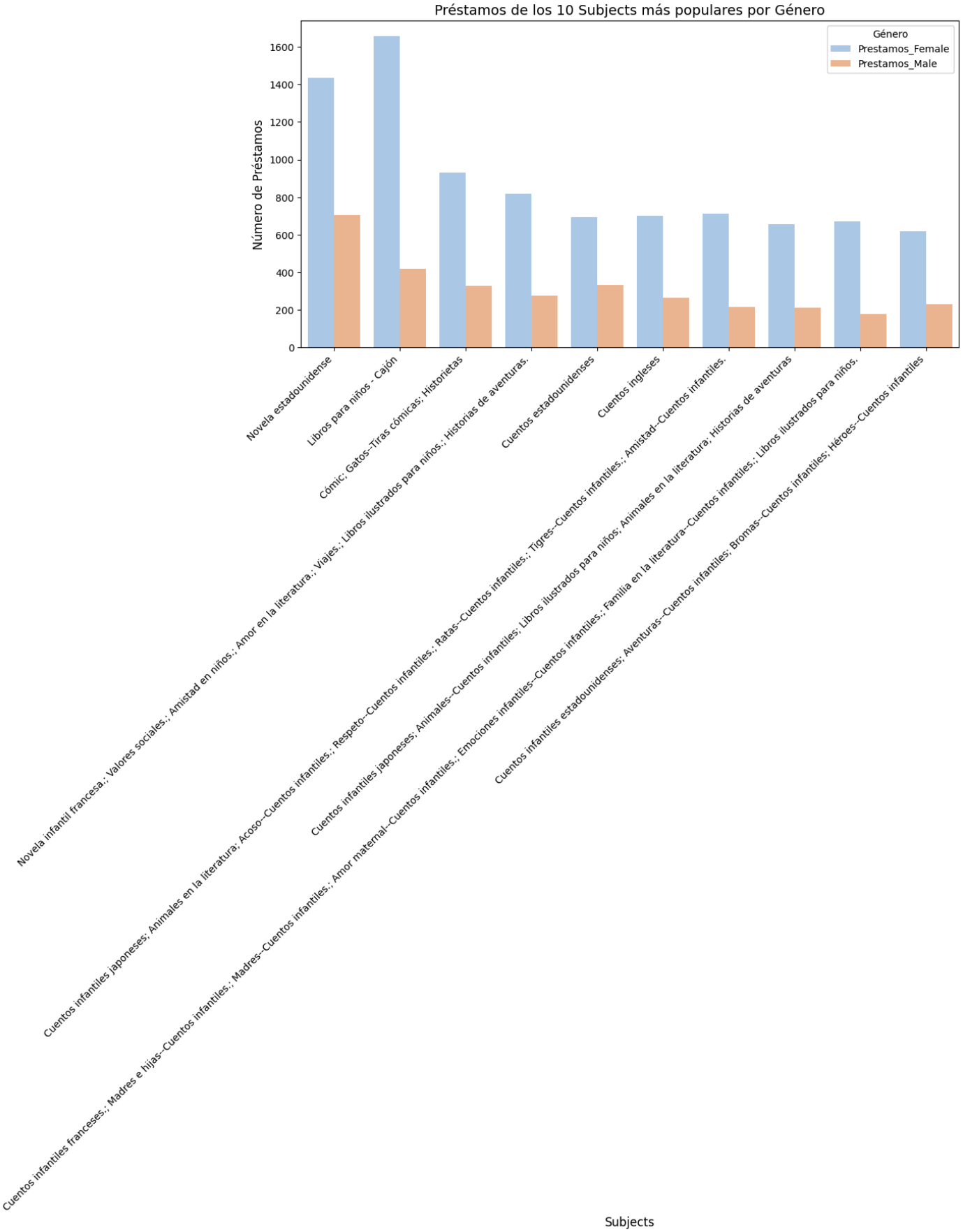



```
# 3. Préstamos por Ciudad
loans_by_city = data.groupby('City').size().reset_index(name='Total_Loans').sort_values(by='Total_Loans', ascending=False)
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Filtrar por género
prestamos_mujeres = data[data['Gender'] == 'Female']
prestamos_hombres = data[data['Gender'] == 'Male']

# Contar los Subjects más prestados para cada género
subjects_mujeres = prestamos_mujeres['Subjects'].value_counts().reset_index()
subjects_mujeres.columns = ['Subjects', 'Prestamos_Female']

subjects_hombres = prestamos_hombres['Subjects'].value_counts().reset_index()
subjects_hombres.columns = ['Subjects', 'Prestamos_Male']

# Combinar resultados para comparar en un solo DataFrame
subjects_totales = pd.merge(subjects_mujeres, subjects_hombres, on='Subjects', how='outer').fillna(0)

# Ordenar por cantidad de préstamos totales
subjects_totales['Total_Prestamos'] = subjects_totales['Prestamos_Female'] + subjects_totales['Prestamos_Male']
subjects_totales = subjects_totales.sort_values(by='Total_Prestamos', ascending=False).head(10)

# Visualización: Gráfico de barras agrupadas
plt.figure(figsize=(12, 6))
sns.barplot(data=subjects_totales.melt(id_vars='Subjects',
                                       value_vars=['Prestamos_Female', 'Prestamos_Male'],
                                       var_name='Género',
                                       value_name='Préstamos'),
            x='Subjects', y='Préstamos', hue='Género', palette='pastel')

# Configuración del gráfico
plt.title('Préstamos de los 10 Subjects más populares por Género', fontsize=14)
plt.xlabel('Subjects', fontsize=12)
plt.ylabel('Número de Préstamos', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.legend(title='Género')
```

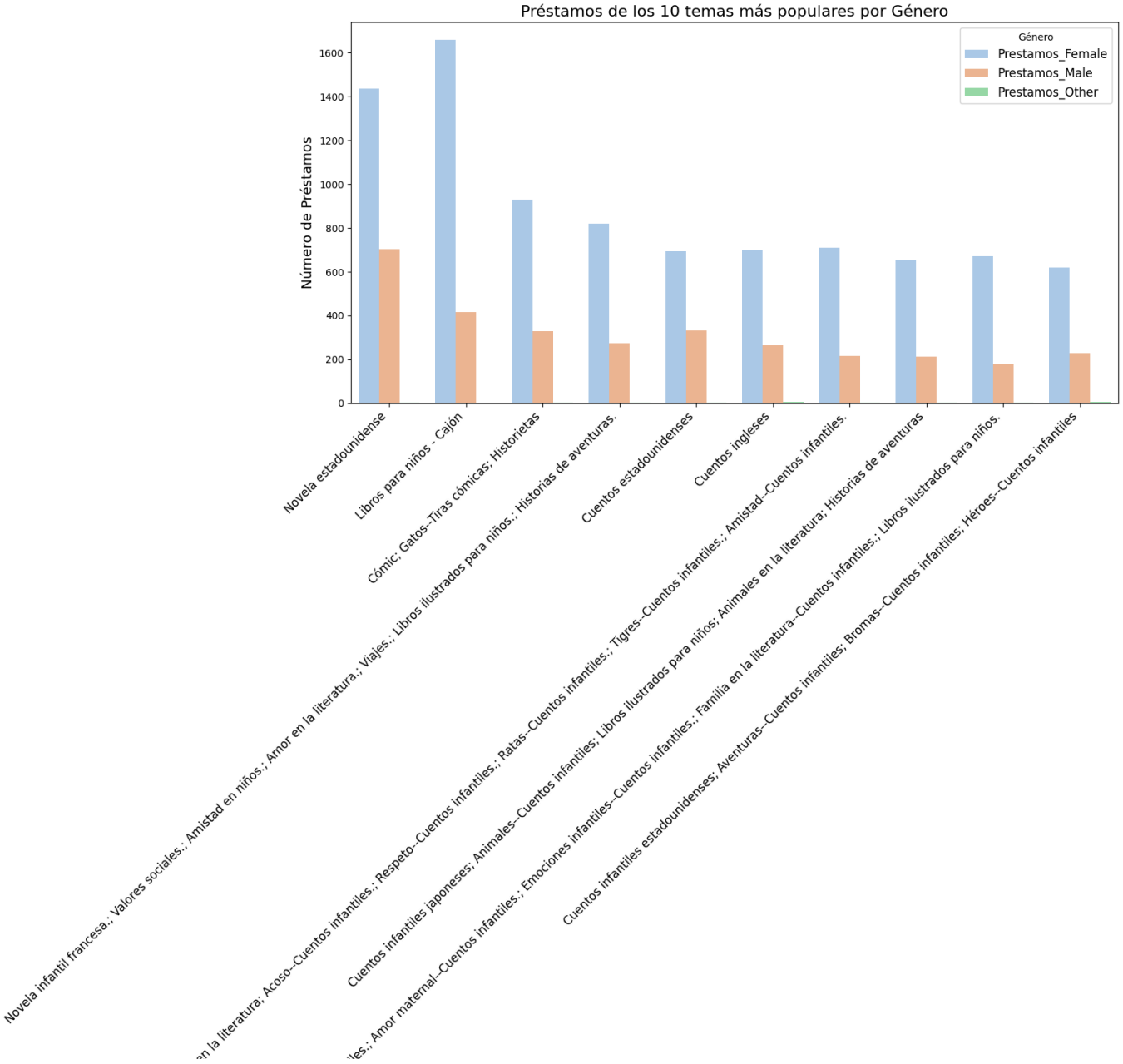
```
    plt.tight_layout()

    # Mostrar gráfico
    plt.show()
```

```
<ipython-input-17-ec7a5a710aec>:37: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to acc
  plt.tight_layout()
```



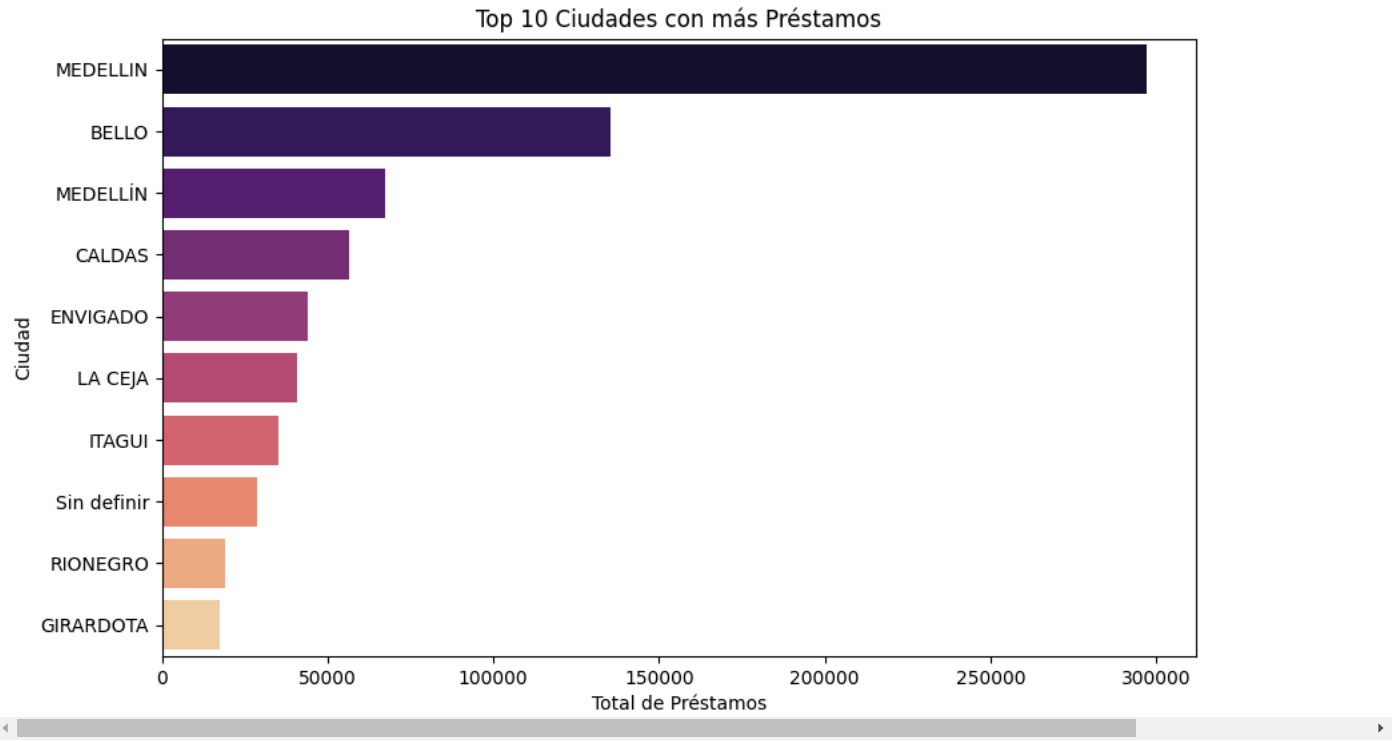
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Filtrar por género
prestamos_mujeres = data[data['Gender'] == 'Female']
prestamos_hombres = data[data['Gender'] == 'Male']
prestamos_other = data[data['Gender'] == 'Other']

# Contar los Subjects más prestados para cada género
subjects_mujeres = prestamos_mujeres['Subjects'].value_counts().reset_index()
subjects_mujeres.columns = ['Subjects', 'Prestamos_Female']

subjects_hombres = prestamos_hombres['Subjects'].value_counts().reset_index()
subjects_hombres.columns = ['Subjects', 'Prestamos_Male']

subjects_other = prestamos_other['Subjects'].value_counts().reset_index()
subjects_other.columns = ['Subjects', 'Prestamos_Other']

# Combinar resultados para comparar en un solo DataFrame
subjects_totales = pd.merge(subjects_mujeres, subjects_hombres, on='Subjects', how='outer').fillna(0)
subjects_totales = pd.merge(subjects_totales, subjects_other, on='Subjects', how='outer').fillna(0)

# Ordenar por cantidad de préstamos totales
subjects_totales['Total_Prestamos'] = (
    subjects_totales['Prestamos_Female'] +
    subjects_totales['Prestamos_Male'] +
    subjects_totales['Prestamos_Other']
)
subjects_totales = subjects_totales.sort_values(by='Total_Prestamos', ascending=False).head(10)

# Visualización: Gráfico de barras agrupadas
plt.figure(figsize=(14, 7))
sns.barplot(data=subjects_totales.melt(id_vars='Subjects',
                                       value_vars=['Prestamos_Female', 'Prestamos_Male', 'Prestamos_Other'],
                                       var_name='Género',
                                       value_name='Préstamos'),
            x='Subjects', y='Préstamos', hue='Género', palette='pastel')

# Configuración del gráfico
plt.title('Préstamos de los 10 temas más populares por Género', fontsize=16)
plt.xlabel('Subjects', fontsize=14)
plt.ylabel('Número de Préstamos', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.legend(title='Género', fontsize=12)
plt.tight_layout()

# Mostrar gráfico
plt.show()
```

```
<ipython-input-18-8622d17c21fe>:46: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to acc
  plt.tight_layout()
```




```
# Visualización: Gráfico de barras
plt.figure(figsize=(10, 6))
sns.barplot(x='Total_Loans', y='City', data=loans_by_city.head(10), palette='magma')
plt.title('Top 10 Ciudades con más Préstamos')
plt.xlabel('Total de Préstamos')
plt.ylabel('Ciudad')
plt.show()
```

```
<ipython-input-19-41886aa197a9>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend

  sns.barplot(x='Total_Loans', y='City', data=loans_by_city.head(10), palette='magma')
```



Organizar edades

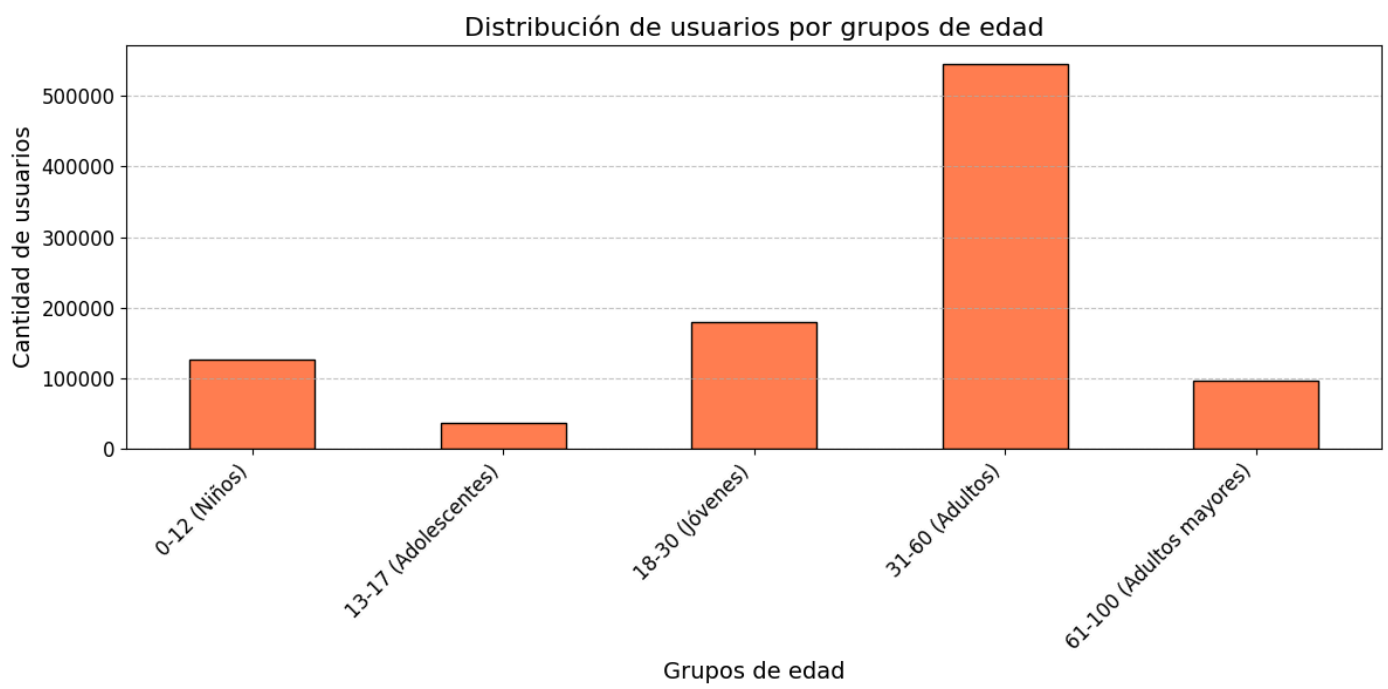
```
# Organizar edades
data['Birth Date'] = data['Birth Date'].astype(str)
data['Birth Date'] = data['Birth Date'].str.lstrip('0')
data['Birth Date'] = pd.to_datetime(data['Birth Date'], errors='coerce')

def calculate_age(birth_date):
    if pd.isnull(birth_date):
        return None
    today = datetime.today()
    age = today.year - birth_date.year - ((today.month, today.day) < (birth_date.month, birth_date.day))
    return age

data['Age'] = data['Birth Date'].apply(calculate_age)
data['Age'] = data['Age'].fillna(-1).astype(int)
```

```
# Selección de columnas relevantes
data_reco = data[['Primary Identifier', 'Full Name', 'Gender', 'Birth Date', 'Age', 'City', 'Loans (In House + Not In House)', 'Title', 'Sub
data_reco.head(30)
```

	Primary Identifier	Full Name	Gender	Birth Date	Age	City	Loans (In House + Not In House)	Title	Subjects	Author
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	1998-01-30	26	MEDELLÍN	1	Crea & divaga : vida y reflexiones de Jeff Bezos	Bezos, Jeff.--1964---Relatos personales; Amazo...	Bezos, Jeff. 1964-,
1	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	1998-01-30	26	MEDELLÍN	1	Sanación con cristales : las claves para inici...	Cristales--Uso terapéutico; Terapias de sanaci...	Cuellar B., Andrea
2	1035973109	RODRIGUEZ LAVERDE HAROLD	Female	2005-10-21	19	Itagui	1	Colombia una historia mínima : una mirada inte...	Violencia-Historia--Colombia; Colombia-Histori...	Melo, Jorge Orlando 1942-,
3	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Astronautas	Astronautas; Cohetes (Aeronáutica); Astronáuti...	Jung, Chang-hoon.
4	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	¿Qué es ese ruido?	Cuentos infantiles ingleses; Libros y lectura ...	Benjamín, A.H., 1950-,.
5	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Cuando el hielo se derrite	Cuentos infantiles ingleses; Osos polares--Cue...	Eve, Rosie, Autora e ilustradora
6	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	La enfermedad	Novela venezolana; Enfermedades--Novela; Padre...	Barrera Tyszka, Alberto. 1960-,
7	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Tetas : ¡mira! ¡mira! ¡¡un hombre con sujetador!!	Glándulas mamarias.; Lactancia materna.; Fisio...	Yagyu, Genichiro, 1943-,
8	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Dientes	Dientes--Cuidado e higiene.; Hábitos orales.; ...	Yagyu, Genichiro, 1943-,.
9	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Ombligo	Cuerpo humano.; Ombligo.; Cordón umbilical.; A...	Yagyu, Genichiro, 1943-,.
10	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Lom y los nudones	Cuentos infantiles venezolanos; Cabello - Cuid...	Kurusa, 1942-, (Carmen Diana Dearden)
11	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Un gato llamado Almohada	Cuentos infantiles españoles; Niños--Cuentos i...	Llinàs, Andreu, 1978- Autor e ilustrador
12	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Soy un cerdito	Cuentos infantiles españoles; Niños -	Santana, Eva, 1972-

```
import matplotlib.pyplot as plt


# Crear nuevos grupos de edad
bins = [0, 12, 17, 30, 60, 100]
labels = ['0-12 (Niños)', '13-17 (Adolescentes)', '18-30 (Jóvenes)',
          '31-60 (Adultos)', '61-100 (Adultos mayores)']
data_reco['Age_Group'] = pd.cut(data_reco['Age'], bins=bins, labels=labels, right=False)

# Contar usuarios en cada grupo
age_group_counts = data_reco['Age_Group'].value_counts().sort_index()

# Crear el gráfico
plt.figure(figsize=(12, 6))
age_group_counts.plot(kind='bar', color='coral', edgecolor='black')

# Configuración del gráfico
plt.title('Distribución de usuarios por grupos de edad', fontsize=16)
plt.xlabel('Grupos de edad', fontsize=14)
plt.ylabel('Cantidad de usuarios', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

```
# Mostrar gráfico
plt.tight_layout()
plt.show()
```




```
df_expanded = data_reco.explode('Title').reset_index(drop=True)

# Mostramos el DataFrame resultante
df_expanded.head(30)
```

	Primary Identifier	Full Name	Gender	Birth Date	Age	City	Loans (In House + Not In House)	Title	Subjects	Author	Age_Group
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	1998-01-30	26	MEDELLÍN	1	Crea & divaga : vida y reflexiones de Jeff Bezos	Bezos, Jeff.--1964---Relatos personales; Amazo...	Bezos, Jeff. 1964-,	18-30 (Jóvenes)
1	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	Male	1998-01-30	26	MEDELLÍN	1	Sanación con cristales : las claves para inici...	Cristales--Uso terapéutico; Terapias de sanaci...	Cuellar B., Andrea	18-30 (Jóvenes)
2	1035973109	RODRIGUEZ LAVERDE HAROLD	Female	2005-10-21	19	Itagui	1	Colombia una historia mínima : una mirada inte...	Violencia-Historia--Colombia; Colombia-Histori...	Melo, Jorge Orlando 1942-,	18-30 (Jóvenes)
3	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Astronautas	Astronautas; Cohetes (Aeronáutica); Astronáuti...	Jung, Chang-hoon.	31-60 (Adultos)
4	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	¿Qué es ese ruido?	Cuentos infantiles ingleses; Libros y lectura ...	Benjamín, A.H., 1950-,.	31-60 (Adultos)
5	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Cuando el hielo se derrite	Cuentos infantiles ingleses; Osos polares--Cue...	Eve, Rosie, Autora e ilustradora	31-60 (Adultos)
6	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	La enfermedad	Novela venezolana; Enfermedades--Novela; Padre...	Barrera Tyszka, Alberto. 1960-,	31-60 (Adultos)
7	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Tetas : ¡mira! ¡mira! ¡¡un hombre con sujetador!!	Glándulas mamarias.; Lactancia materna.; Fisio...	Yagyu, Genichiro, 1943-,	31-60 (Adultos)
8	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Dientes	Dientes--Cuidado e higiene.; Hábitos orales.; ...	Yagyu, Genichiro, 1943-,.	31-60 (Adultos)
9	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Ombligo	Cuerpo humano.; Ombligo.; Cordón umbilical.; A...	Yagyu, Genichiro, 1943-,.	31-60 (Adultos)
10	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Lom y los nudones	Cuentos infantiles venezolanos; Cabello - Cuid...	Kurusa, 1942-, (Carmen Diana Dearden)	31-60 (Adultos)
11	43205110	Velez Restrepo Alexandra	Female	1980-07-12	44	MEDELLIN	1	Un gato llamado Almohada	Cuentos infantiles españoles; Niños--Cuentos i...	Llinàs, Andreu, 1978- Autor e ilustrador	31-60 (Adultos)

```
# Cuantos datos faltantes hay por cada atributo
print("Datos faltantes por atributo:\n", data_reco.isnull().sum())
```

```
Datos faltantes por atributo:
 Primary Identifier                     0
Full Name                               0
Gender                                  0
Birth Date                         129926
Age                                     0
City                                    0
Loans (In House + Not In House)         0
Title                                   0
Subjects                                0
Author                                  0
Age_Group                          130544
dtype: int64
```

```
data_reco_cleaned = data_reco.dropna()
```

```
# Cuantos datos faltantes hay por cada atributo
print("Datos faltantes por atributo:\n", data_reco_cleaned.isnull().sum())
```

```
Datos faltantes por atributo:
 Primary Identifier                0
Full Name                          0
Gender                             0
Birth Date                         0
Age                                0
City                               0
Loans (In House + Not In House)    0
Title                              0
Subjects                           0
Author                             0
Age_Group                          0
dtype: int64
```

```
# Número de registros y atributos
shape = data_reco_cleaned.shape
print("Número de registros y atributos:", shape)
```

```
Número de registros y atributos: (984336, 11)
```

Organizar ciudades

```
!pip install nltk
import nltk
import ssl

try:
    _create_unverified_https_context = ssl._create_unverified_https_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context

# Download the 'wordnet' data using the download manager
nltk.download('wordnet', download_dir='/usr/local/share/nltk_data')
nltk.download('omw-1.4', download_dir='/usr/local/share/nltk_data')
nltk.download('stopwords', download_dir='/usr/local/share/nltk_data')
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package wordnet to
[nltk_data]     /usr/local/share/nltk_data...
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     /usr/local/share/nltk_data...
[nltk_data] Downloading package stopwords to
[nltk_data]     /usr/local/share/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import pandas as pd # Import pandas explicitly

def normalize_corpus_basic(corpus):
    normalized_corpus = []
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('spanish'))
    for text in corpus:
        if pd.notnull(text):
            text = text.lower()
            text = re.sub(r'[^a-zA-Z\s]', '', text)
            tokens = text.split()  # Divide palabras por espacios
            tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
            normalized_text = ' '.join(tokens)
            normalized_corpus.append(normalized_text)
        else:
            normalized_corpus.append('')
    return normalized_corpus
```

```
corpus = data_reco_cleaned['City']
norm_corpus = normalize_corpus_basic(corpus)
data_reco_cleaned['Clean_City'] = norm_corpus
data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Clean_City'])[0]
```

```
<ipython-input-29-b3884cbc6c13>:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Clean_City'] = norm_corpus
<ipython-input-29-b3884cbc6c13>:25: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Clean_City'])[0]
```

```
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import pandas as pd

# Función de normalización básica
def normalize_corpus_basic(corpus):
    normalized_corpus = []
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('spanish'))

    for text in corpus:
        if pd.notnull(text):  # Verificar si el texto no es nulo
            text = text.lower()  # Convertir a minúsculas
            text = re.sub(r'[^a-zA-Z\s]', '', text)  # Eliminar caracteres no alfabéticos
            tokens = text.split()  # Tokenizar
            tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]  # Lematización y eliminación de stopwords
            normalized_text = ' '.join(tokens)
            normalized_corpus.append(normalized_text)
        else:
            normalized_corpus.append('')  # Manejar valores nulos
    return normalized_corpus

# Normalizar la columna 'City'
corpus = data_reco_cleaned['City']
norm_corpus = normalize_corpus_basic(corpus)

# Asignar los resultados al DataFrame
data_reco_cleaned.loc[:, 'Clean_City'] = norm_corpus
data_reco_cleaned.loc[:, 'City Code'] = pd.factorize(data_reco_cleaned['Clean_City'])[0]

# Verificar columnas faltantes antes de imprimir
if 'Normalized_City' not in data_reco_cleaned.columns:
    data_reco_cleaned['Normalized_City'] = ''
if 'Corrected_City' not in data_reco_cleaned.columns:
    data_reco_cleaned['Corrected_City'] = ''

# Mostrar las primeras filas del DataFrame
print(data_reco_cleaned[['Clean_City', 'Normalized_City', 'Corrected_City', 'City Code']].head(20))
```

```
       Clean_City Normalized_City Corrected_City  City Code
0        medelln                                          0
1        medelln                                          0
2         itagui                                          1
3       medellin                                          2
4       medellin                                          2
5       medellin                                          2
6       medellin                                          2
7       medellin                                          2
8       medellin                                          2
9       medellin                                          2
10      medellin                                          2
11      medellin                                          2
12      medellin                                          2
13      medellin                                          2
14      medellin                                          2
15      medellin                                          2
16      medellin                                          2
```

```
17      medellin                                    2
18      medellin                                    2
17129   medellin                                    2
<ipython-input-30-d2cd828b77b6>:34: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Normalized_City'] = ''
<ipython-input-30-d2cd828b77b6>:36: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Corrected_City'] = ''
```

```
pip install unidecode
```

```
Collecting unidecode
  Downloading Unidecode-1.3.8-py3-none-any.whl.metadata (13 kB)
Downloading Unidecode-1.3.8-py3-none-any.whl (235 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 235.5/235.5 kB 3.5 MB/s eta 0:00:00
Installing collected packages: unidecode
Successfully installed unidecode-1.3.8
```

```
import unidecode

# Función para normalizar los nombres de las ciudades
def normalize_city_name(city_name):
    # Convertir a minúsculas y eliminar acentos
    city_name = unidecode.unidecode(city_name.lower())
    return city_name



# Reemplazar "medelln" por "medellín" en la columna 'Clean_City'
data_reco_cleaned['Clean_City'] = data_reco_cleaned['Clean_City'].replace('medelln', 'medellín')

# Normalizar los nombres nuevamente después de realizar la corrección
data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)

# Generar los códigos de las ciudades corregidas
data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Normalized_City'])[0]

# Verificar los resultados
print(data_reco_cleaned[['Clean_City', 'Normalized_City', 'City Code']].head(100))
```

```
<ipython-input-33-c6cac2381ac1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Clean_City'] = data_reco_cleaned['Clean_City'].replace('medelln', 'medellín')
<ipython-input-33-c6cac2381ac1>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)
<ipython-input-33-c6cac2381ac1>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Normalized_City'])[0]
           Clean_City  Normalized_City  City Code
0             medellín         medellin          0
1             medellín         medellin          0
2               itagui           itagui          1
3             medellin         medellin          0
4             medellin         medellin          0
...                ...              ...        ...
17212  medelln moravia  medelln moravia          7
17213  medelln moravia  medelln moravia          7
17214  medelln moravia  medelln moravia          7
17215  medelln moravia  medelln moravia          7
```

```
17216  medelln moravia  medelln moravia          7

[100 rows x 3 columns]
```

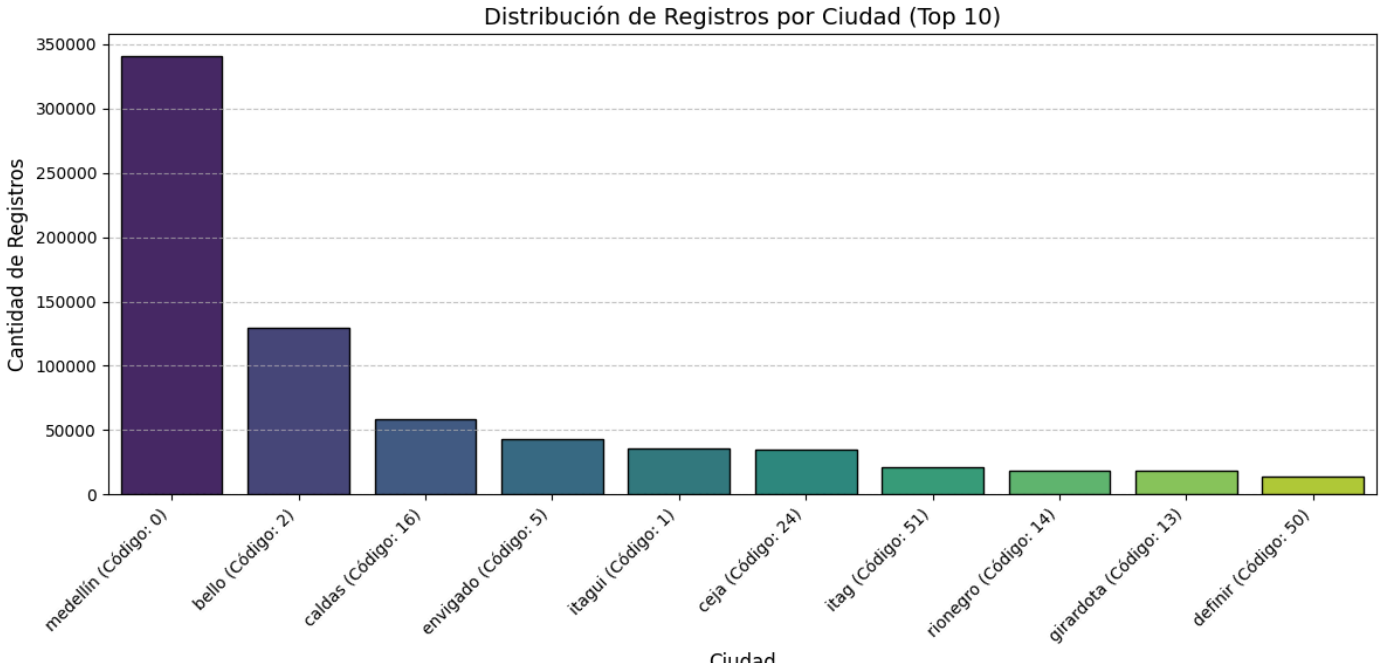
```
import pandas as pd
import re
import unicodedata

# Función para normalizar los nombres de las ciudades (eliminar tildes y caracteres especiales)
def normalize_city_name(city_name):
    # Eliminar tildes y convertir todo a minúsculas
    city_name = unicodedata.normalize('NFD', city_name).encode('ascii', 'ignore').decode('utf-8')
    return city_name.lower()

# Reemplazar todas las variantes de "Medellín" y sus barrios
def replace_medellin_variants(city_name):
    # Reemplazar cualquier variante de "Medellín" con "medellín"
    city_name = re.sub(r'medelln|medellin|medelin', 'medellín', city_name, flags=re.IGNORECASE)
    return city_name

# Aplicar el reemplazo a la columna 'Clean_City'
data_reco_cleaned['Clean_City'] = data_reco_cleaned['Clean_City'].apply(replace_medellin_variants)

# Normalizar los nombres después del reemplazo
data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)

# Generar los códigos de las ciudades corregidas
data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Normalized_City'])[0]

# Verificar los resultados
print(data_reco_cleaned[['Clean_City', 'Normalized_City', 'City Code']].head(100))
```

```
<ipython-input-34-3fa63f955325>:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Clean_City'] = data_reco_cleaned['Clean_City'].apply(replace_medellin_variants)
<ipython-input-34-3fa63f955325>:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)
<ipython-input-34-3fa63f955325>:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Normalized_City'])[0]
            Clean_City   Normalized_City  City Code
0              medellín          medellin          0
1              medellín          medellin          0
2                itagui            itagui          1
3              medellín          medellin          0
4              medellín          medellin          0
...                 ...               ...        ...
17212  medellín moravia  medellin moravia          7
17213  medellín moravia  medellin moravia          7
17214  medellín moravia  medellin moravia          7
17215  medellín moravia  medellin moravia          7
17216  medellín moravia  medellin moravia          7

[100 rows x 3 columns]
```

```
# Asegúrate de tener el DataFrame 'data_reco_cleaned' cargado correctamente
# Exportar el DataFrame a un archivo Excel
data_reco_cleaned[['Clean_City', 'Normalized_City', 'City Code']].to_excel('/content/ciudades_corregidas.xlsx', index=False)

# Si deseas descargar el archivo generado en tu máquina local, usa el siguiente código:
from google.colab import files
files.download('/content/ciudades_corregidas.xlsx')
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import unicodedata

# Función para normalizar los nombres de las ciudades (eliminar tildes y caracteres especiales)
def normalize_city_name(city_name):
    # Eliminar tildes y convertir todo a minúsculas
    city_name = unicodedata.normalize('NFD', city_name).encode('ascii', 'ignore').decode('utf-8')
    return city_name.lower()

# Mapeo para agrupar nombres similares
city_name_map = {
    'medelln': 'medellín',  # Agrupar "Medelln" y "Medellín" bajo "Medellín"
    # Aquí puedes agregar más variaciones si es necesario
}

# Función para corregir el nombre de la ciudad utilizando el mapeo
def correct_city_name(city_name):
    city_name = city_name.lower()  # Convertir a minúsculas
    return city_name_map.get(city_name, city_name)  # Si no está en el mapeo, mantener el nombre original

# Aplicar la normalización y corrección de nombres de ciudades
data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)

# Aplicar el mapeo para corregir variaciones de nombres de ciudades
data_reco_cleaned['Corrected_City'] = data_reco_cleaned['Normalized_City'].apply(correct_city_name)

# Reemplazamos el nombre de la ciudad en 'Corrected_City' con los valores agrupados (específicamente agrupamos "Medelln" con "Medellín")
data_reco_cleaned['Corrected_City'] = data_reco_cleaned['Corrected_City'].replace(city_name_map)

# Luego asignamos un único código a las ciudades corregidas
data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Corrected_City'])[0]

# Contar la cantidad de ocurrencias de cada código de ciudad basado en los nombres normalizados
city_code_counts = data_reco_cleaned['Corrected_City'].value_counts()

# Obtener los 10 nombres de ciudad más frecuentes
top_10_city_names = city_code_counts.head(10)

# Asociar los nombres de las ciudades con sus códigos
top_10_city_codes = data_reco_cleaned[data_reco_cleaned['Corrected_City'].isin(top_10_city_names.index)]
city_names_mapping = top_10_city_codes.drop_duplicates(subset='Corrected_City')[['Corrected_City', 'Clean_City', 'City Code']]

# Crear el gráfico
plt.figure(figsize=(12, 6))  # Aumentar el tamaño de la figura
sns.barplot(x=top_10_city_names.index, y=top_10_city_names.values, palette="viridis", edgecolor='black')

# Configuración del gráfico
plt.title('Distribución de Registros por Ciudad (Top 10)', fontsize=14)
plt.xlabel('Ciudad', fontsize=12)
plt.ylabel('Cantidad de Registros', fontsize=12)

# Reemplazar los nombres normalizados por los nombres originales de las ciudades y añadir el código
city_labels = [
    f"{city_names_mapping[city_names_mapping['Corrected_City'] == city]['Clean_City'].values[0]} (Código: {city_names_mapping[city_names_map
    for city in top_10_city_names.index
]

plt.xticks(ticks=range(10), labels=city_labels, rotation=45, ha='right')

plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

```
<ipython-input-36-d994ebc2d2d6>:23: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Normalized_City'] = data_reco_cleaned['Clean_City'].apply(normalize_city_name)
<ipython-input-36-d994ebc2d2d6>:26: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Corrected_City'] = data_reco_cleaned['Normalized_City'].apply(correct_city_name)
<ipython-input-36-d994ebc2d2d6>:29: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Corrected_City'] = data_reco_cleaned['Corrected_City'].replace(city_name_map)
<ipython-input-36-d994ebc2d2d6>:32: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['City Code'] = pd.factorize(data_reco_cleaned['Corrected_City'])[0]
<ipython-input-36-d994ebc2d2d6>:46: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x=top_10_city_names.index, y=top_10_city_names.values, palette="viridis", edgecolor='black')
```



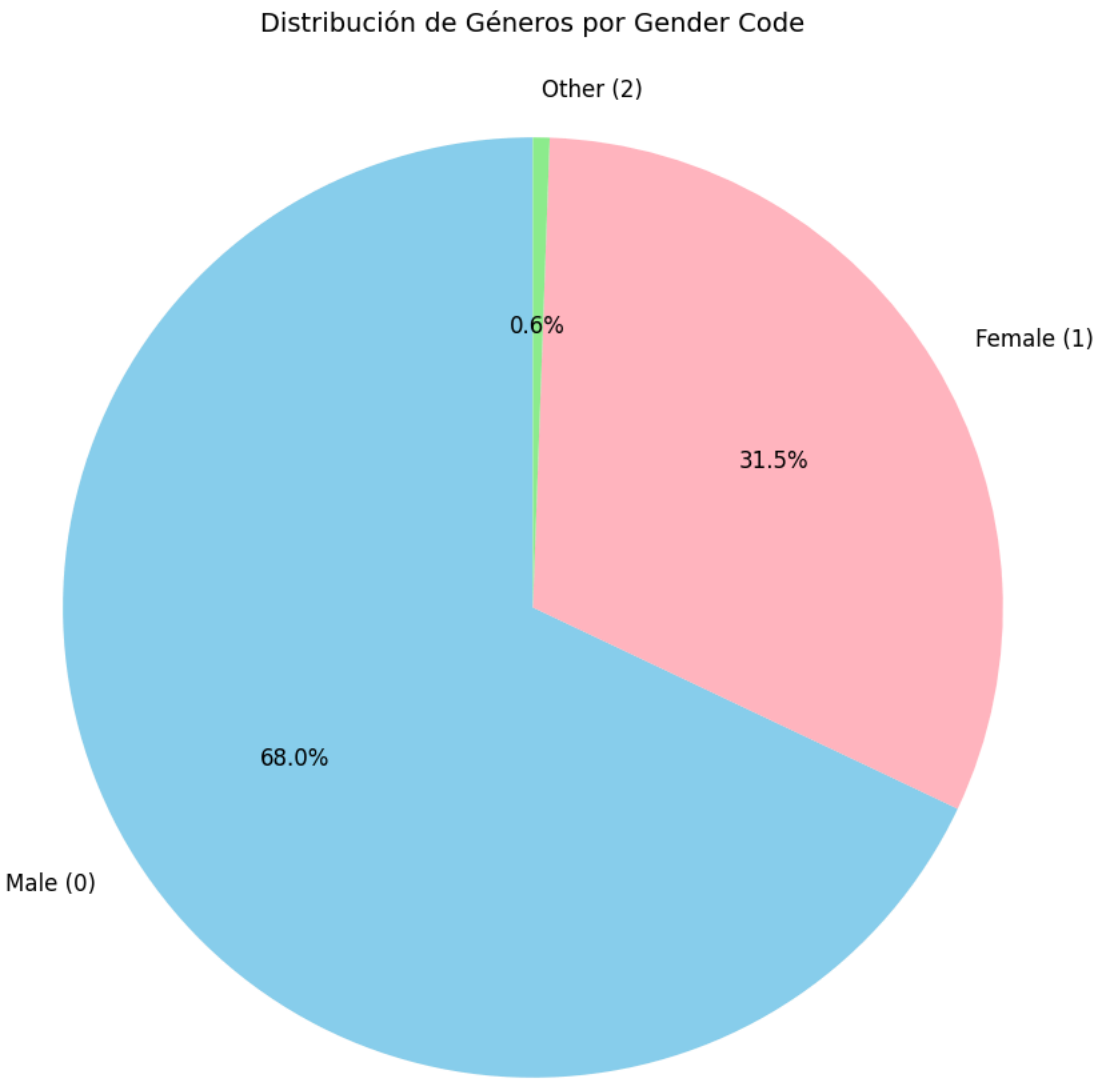
Organizar género

```
# Organizar género
gender_mapping = {'Female': 1, 'Male': 0, 'Other': 2}
data_reco_cleaned['Gender Code'] = data_reco_cleaned['Gender'].map(gender_mapping)
data_reco_cleaned['Gender Code'] = data_reco_cleaned['Gender Code'].fillna(0).astype(int)

# Selección de columnas para el modelo
data_reco_cleaned = data_reco_cleaned[['Primary Identifier', 'Full Name', 'Gender Code', 'Age', 'City Code', 'Loans (In House + Not In House
data_reco_cleaned.head(10)
```

```
<ipython-input-37-ce00ca5084f8>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Gender Code'] = data_reco_cleaned['Gender'].map(gender_mapping)
<ipython-input-37-ce00ca5084f8>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  data_reco_cleaned['Gender Code'] = data_reco_cleaned['Gender Code'].fillna(0).astype(int)
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Loans (In House + Not In House)	Title	Subjects	Author
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	1	Crea & divaga : vida y reflexiones de Jeff Bezos	Bezos, Jeff.--1964---Relatos personales; Amazo...	Bezos, Jeff. 1964-,
1	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	1	Sanación con cristales : las claves para inici...	Cristales--Uso terapéutico; Terapias de sanaci...	Cuellar B., Andrea
2	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	1	Colombia una historia mínima : una mirada inte...	Violencia-Historia--Colombia; Colombia-Histori...	Melo, Jorge Orlando 1942-,
3	43205110	Velez Restrepo Alexandra	1	44	0	1	Astronautas	Astronautas; Cohetes (Aeronáutica); Astronáuti...	Jung, Chang-hoon.
4	43205110	Velez Restrepo Alexandra	1	44	0	1	¿Qué es ese ruido?	Cuentos infantiles ingleses; Libros y lectura ...	Benjamín, A.H., 1950-,.
5	43205110	Velez Restrepo Alexandra	1	44	0	1	Cuando el hielo se derrite	Cuentos infantiles ingleses; Osos polares--Cue...	Eve, Rosie, Autora e ilustradora
6	43205110	Velez Restrepo Alexandra	1	44	0	1	La enfermedad	Novela venezolana; Enfermedades--Novela; Padre...	Barrera Tyszka, Alberto. 1960-,
							Tetas : ¡mira! ¡mira!	Glándulas mamarias :	Yagyu

```
import matplotlib.pyplot as plt

# Contar las ocurrencias de cada Gender Code
gender_counts = data_reco_cleaned['Gender Code'].value_counts()

# Etiquetas para el gráfico
labels = ['Male (0)', 'Female (1)', 'Other (2)']

# Colores para diferenciar las categorías
colors = ['skyblue', 'lightpink', 'lightgreen']

# Crear el gráfico de torta
plt.figure(figsize=(10, 11))
plt.pie(gender_counts, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors, textprops={'fontsize': 12})

# Configuración del gráfico
plt.title('Distribución de Géneros por Gender Code', fontsize=14)
plt.axis('equal')  # Asegura que el gráfico sea un círculo

# Mostrar el gráfico
plt.show()
```




```
# Guardar el DataFrame limpio
data_reco_cleaned.to_csv("/content/drive/MyDrive/datas/data_reco_cleaned_100.csv", index=False)
```

```
import pandas as pd
import sklearn as sk
import seaborn as sns
```

```
# Cargar los datos
data_reco_cleaned = pd.read_csv("/content/drive/MyDrive/datas/data_reco_cleaned_100.csv")
```

```
data_reco_cleaned
```

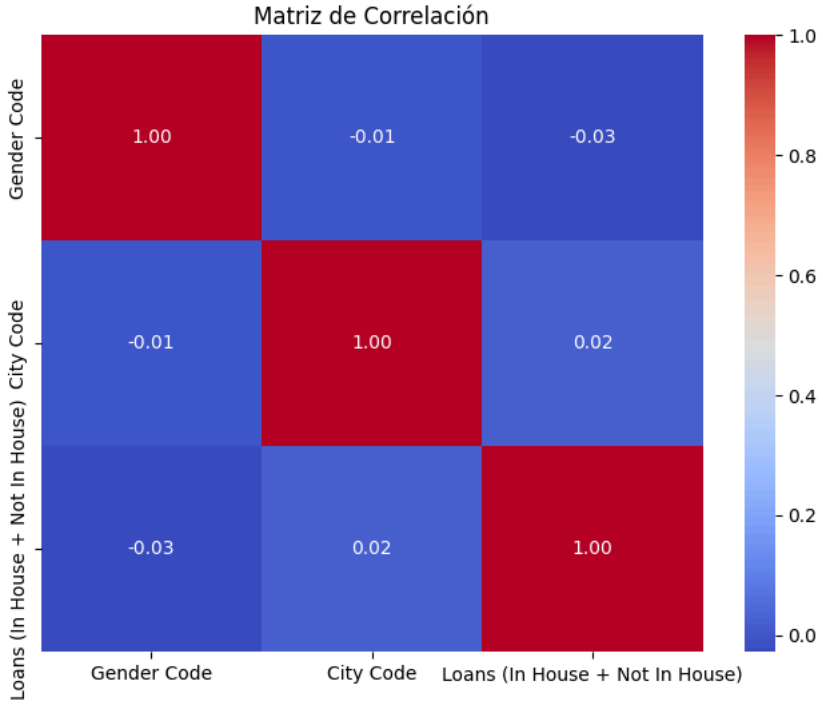
	Primary Identifier	Full Name	Gender Code	Age	City Code	Loans (In House + Not In House)	Title	Subjects	Author
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	1	Crea & divaga : vida y reflexiones de Jeff Bezos	Bezos, Jeff.--1964---Relatos personales; Amazo...	Bezos, Jeff. 1964-,
1	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	1	Sanación con cristales : las claves para inici...	Cristales--Uso terapéutico; Terapias de sanaci...	Cuellar B., Andrea
2	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	1	Colombia una historia mínima : una mirada inte...	Violencia-Historia--Colombia; Colombia-Histori...	Melo, Jorge Orlando 1942-,
3	43205110	Velez Restrepo Alexandra	1	44	0	1	Astronautas	Astronautas; Cohetes (Aeronáutica); Astronáuti...	Jung, Chang-hoon.
4	43205110	Velez Restrepo Alexandra	1	44	0	1	¿Qué es ese ruido?	Cuentos infantiles ingleses; Libros y lectura ...	Benjamín, A.H., 1950-,.
...	...	...	...	...	...	...	...	...	...
984331	XXPD4319487	GIRALDO DUQUE HERNAN	0	82	0	1	Código penal : ley 599 de 2000	Derecho penal - Legislación - Colombia; Proced...	Colombia. Leyes, decretos, etc.
984332	co1t1198465625	RESTREPO CANO JERONIMO	0	10	16	1	Cartas a quien pretende enseñar	Freire, Paulo, 1921-1997--Crítica e interpreta...	Freire, Paulo 1921-1997
984333	co1t1198465625	RESTREPO CANO JERONIMO	0	10	16	1	Una habitación en las nubes	Novela juvenil española; Familia--Novela juven...	Broseta Fandos,

```
print(data_reco_cleaned.columns)
```

```
Index(['Primary Identifier', 'Full Name', 'Gender Code', 'Age', 'City Code',
       'Loans (In House + Not In House)', 'Title', 'Subjects', 'Author'],
      dtype='object')
```

```
import pandas as pd
import numpy as np

# Crear una copia del dataset para trabajar con transformaciones
data_corr = data_reco_cleaned.copy()

# Seleccionar variables relevantes para la correlación
variables_corr = data_corr[['Gender Code', 'City Code', 'Loans (In House + Not In House)']]

# Calcular la matriz de correlación
correlation_matrix = variables_corr.corr()

# Mostrar la matriz de correlación
print("Matriz de Correlación:")
print(correlation_matrix)

# Visualización de la matriz de correlación
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Matriz de Correlación')
plt.show()
```

```
Matriz de Correlación:
                                Gender Code  City Code  \
Gender Code                        1.000000  -0.005433
City Code                         -0.005433   1.000000
Loans (In House + Not In House)   -0.027725   0.018014

                                Loans (In House + Not In House)
Gender Code                                           -0.027725
City Code                                              0.018014
Loans (In House + Not In House)                        1.000000
```



Eliminar duplicados

```
# Eliminar duplicados manteniendo títulos, materias y autores
unique_users = data_reco_cleaned.drop_duplicates(subset=['Primary Identifier', 'Full Name', 'Gender Code', 'Age', 'City Code'], keep='first'


# Agrupar títulos, materias y autores por usuario
grouped_data = data_reco_cleaned.groupby(['Primary Identifier', 'Full Name', 'Gender Code', 'Age', 'City Code']).agg({
    'Title': lambda x: list(x),
    'Subjects': lambda x: list(x),
    'Author': lambda x: list(x),
    'Loans (In House + Not In House)': 'sum'
}).reset_index()



# Verificar los resultados
grouped_data.head()
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Subjects	Author	Loans (In House + Not In House)
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	[Crea & divaga : vida y reflexiones de Jeff Be...	[Bezos, Jeff.--1964---Relatos personales; Amaz...	[Bezos, Jeff. 1964-,, Cuellar B., Andrea]	2
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	[Colombia una historia mínima : una mirada int...	[Violencia-Historia--Colombia; Colombia-Histor...	[Melo, Jorge Orlando 1942-,]	1
2	43205110	Velez Restrepo Alexandra	1	44	0	[Astronautas, ¿Qué es ese ruido?, Cuando el hi...	[Astronautas; Cohetes (Aeronáutica); Astronáut...	[Jung, Chang-hoon., Benjamín, A.H., 1950-,., E...	16
		HOLGUIN				[La campeona mundial de	[Cuentos infantiles	[Taylor, Sean, 1965-,.,	

```
print("Shape sin duplicados:", grouped_data.shape)
```

```
Shape sin duplicados: (107702, 9)
```

```
# Guardar el DataFrame limpio
grouped_data.to_csv("/content/drive/MyDrive/datas/grouped_data.csv", index=False)
```

```
import pandas as pd
import sklearn as sk
import seaborn as sns

# Cargar los datos
grouped_data = pd.read_csv("/content/drive/MyDrive/datas/grouped_data.csv")
grouped_data.head()
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Subjects	Author	Loans (In House + Not In House)
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	['Crea & divaga : vida y reflexiones de Jeff B...	['Bezos, Jeff.--1964---Relatos personales; Ama...	['Bezos, Jeff. 1964-,', 'Cuellar B., Andrea']	2
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	['Colombia una historia mínima : una mirada in...	['Violencia-Historia--Colombia; Colombia-Histo...	['Melo, Jorge Orlando 1942-,']	1
2	43205110	Velez Restrepo Alexandra	1	44	0	['Astronautas', '¿Qué es ese ruido?', 'Cuando ...	['Astronautas; Cohetes (Aeronáutica); Astronáu...	['Jung, Chang-hoon.', 'Benjamín, A.H., 1950-,....	16
						['La campeona			

Organizar materias

```
import nltk
import os
from nltk.data import find

# Eliminar el paquete punkt si existe
try:
    punkt_path = find('tokenizers/punkt')
    os.rmdir(punkt_path.path)
except LookupError:
    print("Punkt no estaba instalado o ya fue eliminado.")

# Forzar la re-descarga de punkt
nltk.download('punkt', force=True)
```

```
Punkt no estaba instalado o ya fue eliminado.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

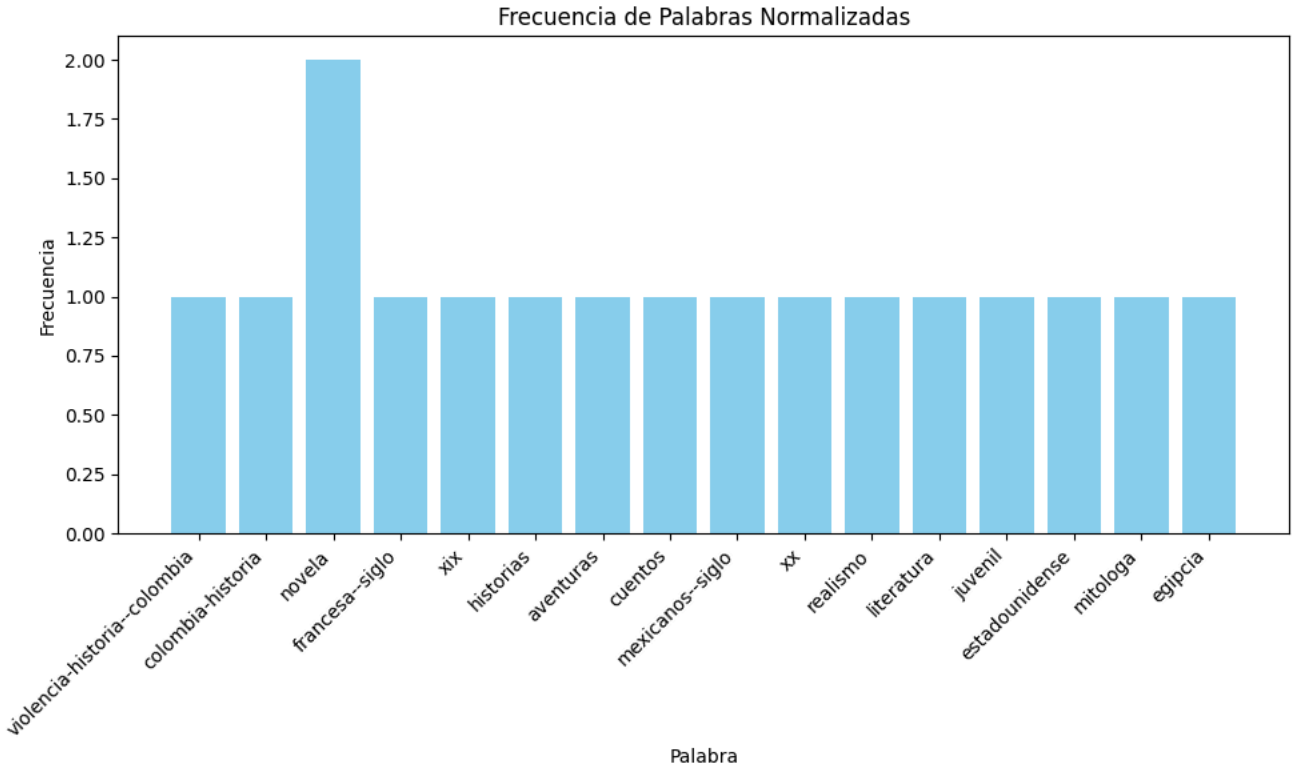
```
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Descargar stopwords y wordnet
import nltk
nltk.download('stopwords')
nltk.download('wordnet')

def normalize_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('spanish'))
```

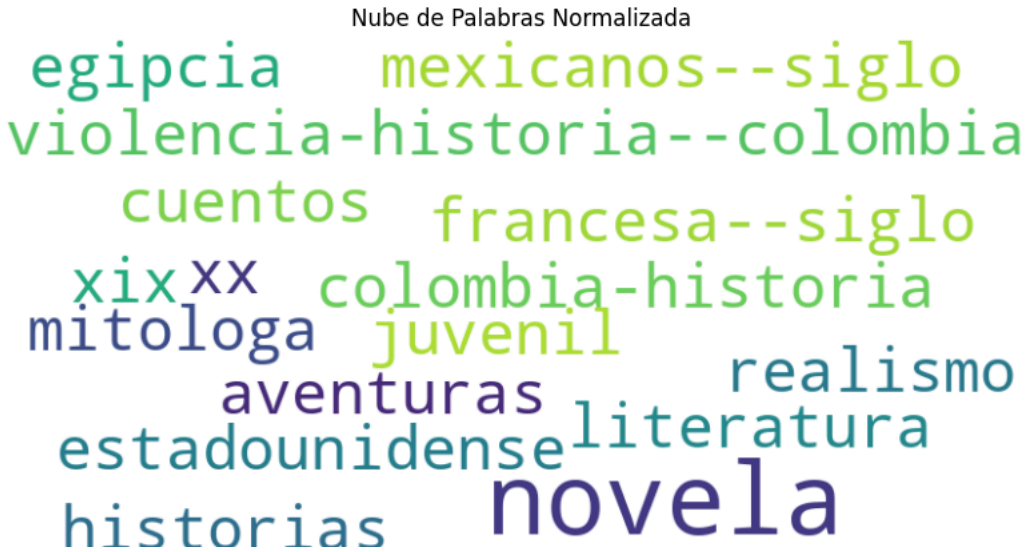
```
    # Convertir a minúsculas
    text = text.lower()

    # Mantener guiones en el texto
    text = re.sub(r'[^a-zA-Z\s-]', '', text)

    # Tokenizar usando split como alternativa
    tokens = text.split()

    # Lematizar y eliminar stopwords
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]

    return tokens

# Ejemplo de subjects
subjects = [
    "Violencia-Historia--Colombia; Colombia-Historia",
    "Novela francesa--Siglo XIX.; Historias de aventuras",
    "Cuentos mexicanos--Siglo XX.; Realismo en la literatura",
    "Novela juvenil estadounidense; Mitología egipcia"
]

# Aplicar la función de normalización
normalized_subjects = [normalize_text(subject) for subject in subjects]

# Imprimir los resultados
for original, normalized in zip(subjects, normalized_subjects):
    print(f"Original: {original}\nNormalizado: {normalized}\n")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
Original: Violencia-Historia--Colombia; Colombia-Historia
Normalizado: ['violencia-historia--colombia', 'colombia-historia']

Original: Novela francesa--Siglo XIX.; Historias de aventuras
Normalizado: ['novela', 'francesa--siglo', 'xix', 'historias', 'aventuras']

Original: Cuentos mexicanos--Siglo XX.; Realismo en la literatura
Normalizado: ['cuentos', 'mexicanos--siglo', 'xx', 'realismo', 'literatura']

Original: Novela juvenil estadounidense; Mitología egipcia
Normalizado: ['novela', 'juvenil', 'estadounidense', 'mitologa', 'egipcia']
```

Gráficos de materias

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from collections import Counter
# Contar la frecuencia de las palabras
all_tokens = [token for sublist in normalized_subjects for token in sublist]
word_counts = Counter(all_tokens)

# Graficar la frecuencia de las palabras más comunes
plt.figure(figsize=(10,6))
plt.bar(word_counts.keys(), word_counts.values(), color='skyblue')
plt.xlabel('Palabra')
plt.ylabel('Frecuencia')
plt.title('Frecuencia de Palabras Normalizadas')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Crear una nube de palabras
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_counts)

# Mostrar la nube de palabras
plt.figure(figsize=(10,6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Nube de Palabras Normalizada')
plt.show()
```






```
# Organizar materias
nltk.download('punkt')

def normalize_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('spanish'))
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    tokens = word_tokenize(text)
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
    return tokens
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
import nltk
nltk.download('punkt', force=True)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
import nltk
print(nltk.data.path)
```

```
['/root/nltk_data', '/usr/nltk_data', '/usr/share/nltk_data', '/usr/lib/nltk_data', '/usr/share/nltk_data', '/usr/local/share/nltk_data'
```

```
import re

def custom_sent_tokenize(text):
    # Reemplaza los separadores con puntos
    text = text.replace(';', '.')
    # Divide en oraciones usando expresiones regulares
    sentences = re.split(r'[.!?]', text)
    return [sentence.strip() for sentence in sentences if sentence.strip()]


import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
import re

def normalize_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('spanish'))

    # Convertir a minúsculas
    text = text.lower()

    # Mantener guiones en el texto
    text = re.sub(r'[^a-zA-Z\s-]', '', text)

    # Tokenizar usando expresiones regulares
    tokens = re.findall(r'\b\w+\b', text)

    # Lematizar y eliminar stopwords
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]

    return tokens


sentences = []
for subject in grouped_data['Subjects']:
    if isinstance(subject, list):
        subject = ' '.join(subject)
    subject_sentences = custom_sent_tokenize(subject.replace(';', '.'))
    for sentence in subject_sentences:
        normalized_sentence = normalize_text(sentence)
        sentences.append(normalized_sentence)


# Imprimir los resultados
for sentence in sentences:
    print(sentence)
```

```
Streaming output truncated to the last 5000 lines.
['literatura', 'infantil', 'croata']
['cuentos', 'aventuras']
['libros', 'ilustrados', 'nios', 'felicidad']
['satisfaccin', 'personal']
['relaciones', 'interpersonales', 'esencias', 'uso', 'teraputico']
['aceites', 'vegetales', 'uso', 'teraputico']
['enjuagues', 'bucales']
['dietoterapia', 'cuentos', 'infantiles', 'estadounidenses']
['cuentos', 'misterio']
['detective', 'literatura']
['cuentos', 'policacos', 'mitologa']
['mitologa', 'leyendas']
['mitologa', 'china']
```

```
['mitologa', 'oriental']
['dragones']
['dragones', 'literatura', 'cuentos', 'infantiles', 'colombianos']
['historias', 'aventuras']
['exploradores', 'cuentos']
['animales', 'cuentos', 'cuentos', 'infantiles', 'espaoles']
['asnos', 'cuentos', 'infantiles']
['mono', 'cuentos', 'infantiles']
['amistad', 'cuentos', 'infantiles']
['animales', 'cuentos', 'infantiles']
['tecnologa', 'cuentos', 'infantiles']
['libros', 'lectura', 'cuentos', 'infantiles', 'juegos', 'infantiles']
['pasatiempos']
['aficiones']
['ocio', 'animales', 'fantsticos']
['mitos']
['leyendas']
['seres', 'mgicos']
['libros', 'ilustrados', 'nios', 'chamanismo']
['alucingenos']
['brujera']
['ocultismo']
['magia']
['chamanes']
['ritos', 'ceremonias', 'profecas']
['profecas', 'bblicas']
['predicciones']
['adivinacin']
['clarividencia']
['magia']
['profecas', 'ocultismo', 'ajedrez', 'enseanza']
['ajedrez', 'reglamentos']
['ajedrez', 'historia']
['matemticas', 'recreativas']
['juegos', 'mesa']
['juegos', 'tablero', 'literatura', 'fantstica']
['hadas', 'literatura', 'infantil']
['hadas', 'leyendas']
['seres', 'fantsticos']
['libros', 'ilustrados', 'nios', 'libros', 'cocina']
['pasta', 'alimenticias', 'cocina']
['huevos', 'cocina']
['sopas']
['verduras', 'cocina']
```

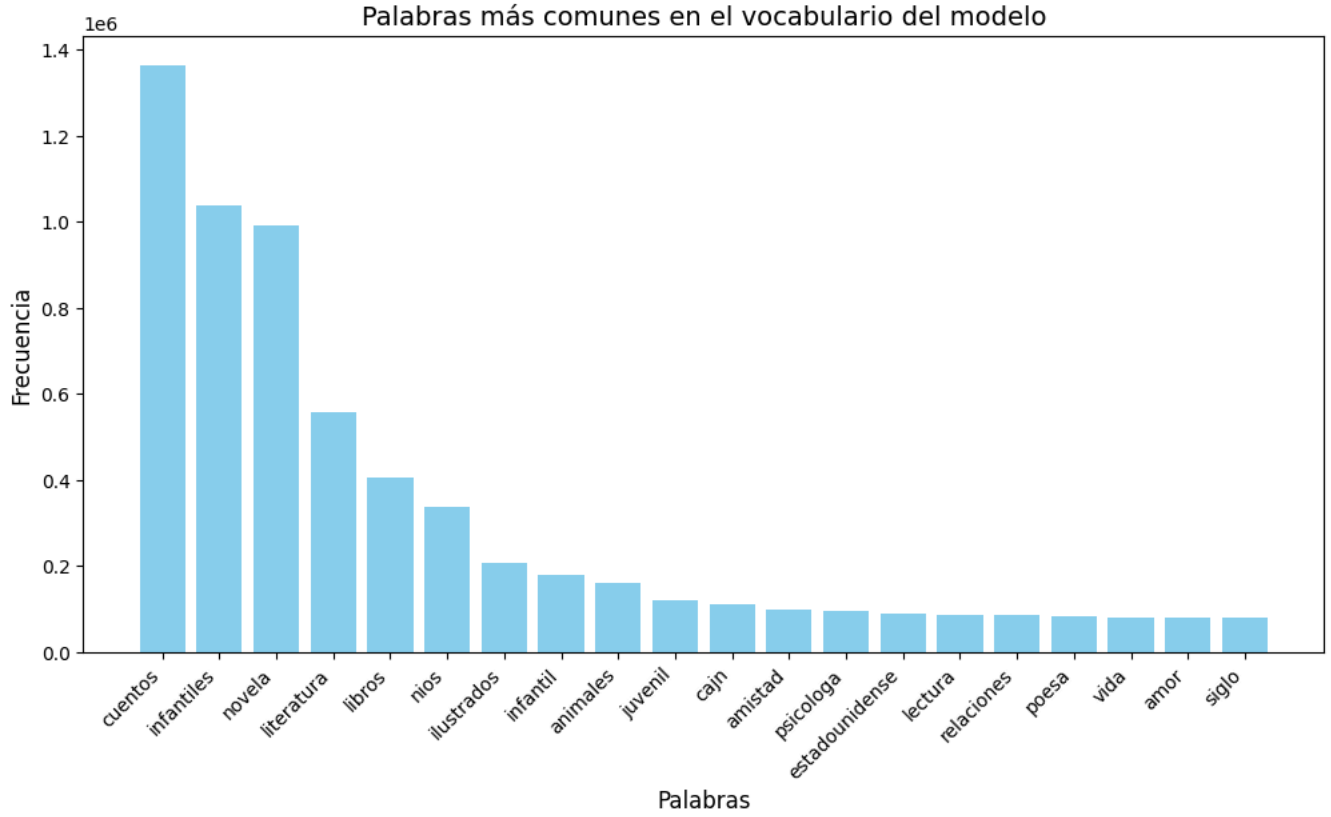
```
print("Oraciones tokenizadas:", sentences[:20])
```

```
# Crear y entrenar el modelo Word2Vec
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)
```

```
# Palabras más comunes en el vocabulario
print("Palabras más comunes en el vocabulario del modelo:", list(model.wv.key_to_index)[:20])
```

```
Palabras más comunes en el vocabulario del modelo: ['cuentos', 'infantiles', 'novela', 'literatura', 'libros', 'nios', 'ilustrados', 'in
```

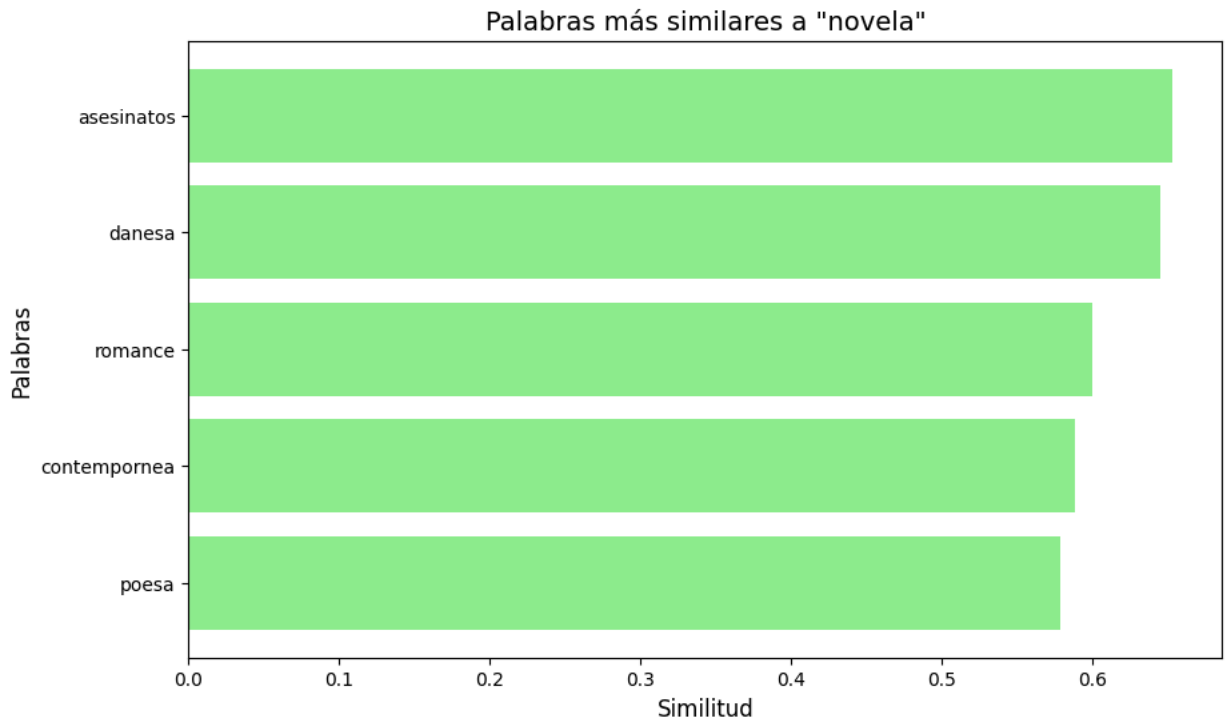
```
import matplotlib.pyplot as plt

# Obtener las palabras más comunes y sus frecuencias
vocab_words = list(model.wv.key_to_index.keys())[:20]
vocab_frequencies = [model.wv.get_vecattr(word, "count") for word in vocab_words]

# Crear el gráfico de barras
plt.figure(figsize=(12, 6))
plt.bar(vocab_words, vocab_frequencies, color='skyblue')
plt.xlabel('Palabras', fontsize=12)
plt.ylabel('Frecuencia', fontsize=12)
plt.title('Palabras más comunes en el vocabulario del modelo', fontsize=14)
plt.xticks(rotation=45, ha='right')
plt.show()
```




```
# Similaridad entre palabras en el modelo
print("Palabras similares a 'educación':", model.wv.most_similar('novela', topn=5))
```

```
Palabras similares a 'educación': [('asesinatos', 0.6537783145904541), ('danesa', 0.6453646421432495), ('romance', 0.6000727415084839),
```

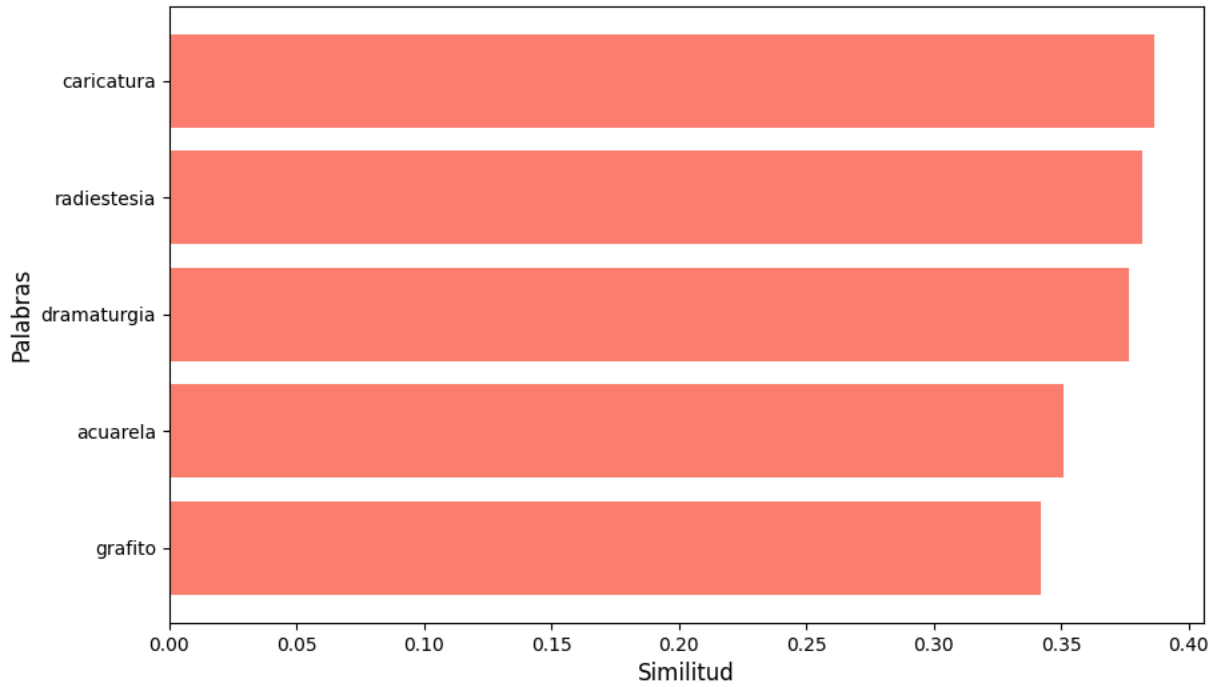
```
# Obtener las palabras más similares y sus similitudes
similar_words = model.wv.most_similar('novela', topn=5)
words = [word for word, similarity in similar_words]
similarities = [similarity for word, similarity in similar_words]

# Crear el gráfico de barras horizontales
plt.figure(figsize=(10, 6))
plt.barh(words, similarities, color='lightgreen')
plt.xlabel('Similitud', fontsize=12)
plt.ylabel('Palabras', fontsize=12)
plt.title('Palabras más similares a "novela"', fontsize=14)
plt.gca().invert_yaxis()  # Invertir el eje Y para mostrar la barra más alta arriba
plt.show()
```




```
# Pregunta de analogía
print("Respuesta a la analogía 'libro es a autor como pintura es a':", model.wv.most_similar(positive=['pintura', 'autor'], negative=['libro
```

```
Respuesta a la analogía 'libro es a autor como pintura es a': [('caricatura', 0.386505126953125)]
```

```
# Obtener el resultado de la analogía
analogy_result = model.wv.most_similar(positive=['pintura', 'autor'], negative=['libro'], topn=5)
analogy_words = [word for word, similarity in analogy_result]
analogy_similarities = [similarity for word, similarity in analogy_result]

# Crear el gráfico de barras horizontales
plt.figure(figsize=(10, 6))
plt.barh(analogy_words, analogy_similarities, color='salmon')
plt.xlabel('Similitud', fontsize=12)
plt.ylabel('Palabras', fontsize=12)
plt.title('Resultado de la analogía: "libro es a autor como pintura es a"', fontsize=14)
plt.gca().invert_yaxis()  # Invertir el eje Y para mostrar la barra más alta arriba
plt.show()
```




```
import re
import numpy as np

# Tokenización personalizada para oraciones
def custom_sent_tokenize(text):
    # Usamos una expresión regular para dividir el texto en oraciones
    # Simplemente separando por puntos, pero puedes mejorar esto para cubrir más casos.
    return re.split(r'(?<!\w\.\w.)(?<![A-Z][a-z]\.)(?<=\.|\?)\s', text)


def normalize_text(text):
    # Aquí va tu lógica de normalización
    # Por ejemplo: pasamos todo a minúsculas y eliminamos caracteres innecesarios
    normalized_text = text.lower()
    return normalized_text


def get_sentence_vector(model, sentence):
    # Tokenizamos la oración en palabras
    words = sentence.split()  # Ahora split() debería funcionar correctamente

    # Obtenemos el vector de cada palabra usando model.wv
    word_vectors = [model.wv[word] for word in words if word in model.wv]

    # Si hay palabras en la oración, promediamos sus vectores; si no, devolvemos un vector nulo
    if len(word_vectors) > 0:
        sentence_vector = np.mean(word_vectors, axis=0)
    else:
        sentence_vector = np.zeros(model.vector_size)  # Asumiendo que el modelo tiene un atributo vector_size
    return sentence_vector


subject_vectors = []
for subject in grouped_data['Subjects']:
    if isinstance(subject, list):
        subject = ' '.join(subject)

    # Reemplaza ';' con '.' y tokeniza en oraciones
    subject_sentences = custom_sent_tokenize(subject.replace(';', '.'))

    # Obtén el vector para cada oración
    sentence_vectors = [get_sentence_vector(model, normalize_text(sent)) for sent in subject_sentences]

    # Promediamos los vectores de las oraciones para obtener un vector único para cada materia
    subject_vector = np.mean(sentence_vectors, axis=0)
    subject_vectors.append(subject_vector)
```

```
subject_vectors_df = pd.DataFrame(subject_vectors)
data_reco_cleaned_with_vectors = pd.concat([grouped_data, subject_vectors_df], axis=1)


# Verificar el resultado
data_reco_cleaned_with_vectors.head()
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Subjects	Author	Loans (In House + Not In House)	0	...	90	91	
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	[Crea & divaga : vida y reflexiones de Jeff Be...	[Bezos, Jeff.--1964---Relatos personales; Amaz...	[Bezos, Jeff. 1964-,, Cuellar B., Andrea]	2	0.505917	...	-0.174399	0.069932	0.
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	[Colombia una historia mínima : una mirada int...	[Violencia--Historia--Colombia; Colombia--Histor...	[Melo, Jorge Orlando 1942-,]	1	0.000000	...	0.000000	0.000000	0.
2	43205110	Velez Restrepo Alexandra	1	44	0	[Astronautas, ¿Qué es ese ruido?, Cuando el hi...	[Astronautas; Cohetes (Aeronáutica); Astronáut...	[Jung, Chang-hoon., Benjamín, A.H., 1950-,, E...	16	-0.125397	...	-0.065630	-0.051234	0.
3	C01C1000654424	HOLGUIN RESTREPO DAHIANA	1	21	0	[La campeona mundial de mantenerse despierta, ...	[Cuentos infantiles ingleses.; Hora de dormir....	[Taylor, Sean, 1965-,, Dominguez, Celia., Bes...	8	-0.078558	...	-0.069577	-0.170189	0.
4	C01C1001033688	MESTRA MENESES GENESIS	0	24	0	[Romeo y Julieta, Orgullo y prejuicio, No hay ...	[Literatura inglesa; Teatro inglés; Tragedia i...	[Shakespeare, William 1564-1616, Austen, Jane,...	4	0.116022	...	-0.766553	-0.381200	-0.

5 rows × 109 columns

```
# Convertir columnas a tipo entero
cols_to_convert = ['Gender Code', 'Age', 'City Code', 'Loans (In House + Not In House)']



data_reco_cleaned_with_vectors[cols_to_convert] = data_reco_cleaned_with_vectors[cols_to_convert].fillna(0).astype(int)



data_reco_cleaned_with_vectors = data_reco_cleaned_with_vectors.drop(columns=['Subjects', 'Author'])


print("Shape después de eliminar columnas:", data_reco_cleaned_with_vectors.shape)
```

```
Shape después de eliminar columnas: (107744, 107)
```

```


# Guardar el DataFrame limpio
data_reco_cleaned_with_vectors.to_csv("/content/drive/MyDrive/datas/data_reco_cleaned_with_vectors21.csv", index=False)



import pandas as pd
import sklearn as sk
import seaborn as sns

# Cargar los datos
```

```
data_reco_cleaned_with_vectors = pd.read_csv("/content/drive/MyDrive/datas/data_reco_cleaned_with_vectors21.csv")
data_reco_cleaned_with_vectors.head()
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Loans (In House + Not In House)	0	1	2	...	90	91	92
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	['Crea & divaga : vida y reflexiones de Jeff B...	2	0.280615	-0.052416	0.307992	...	-0.032930	0.135688	0.498295
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	['Colombia una historia mínima : una mirada in...	1	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
2	43205110	Velez Restrepo Alexandra	1	44	2	['Astronautas', '¿Qué es ese ruido?', 'Cuando ...	16	-0.177818	-0.199111	-0.268580	...	0.051444	0.100448	0.224243
3	C01C1000654424	HOLGUIN RESTREPO DAHIANA	1	21	2	['La campeona mundial de mantenerse despierta'...	8	-0.263781	-0.385908	-0.466116	...	0.189849	-0.032371	0.396008
4	C01C1001033688	MESTRA MENESES GENESIS	0	24	0	['Romeo y Julieta', 'Orgullo y prejuicio', "No...	4	0.199063	-0.292346	-0.435913	...	-0.755373	0.079594	-0.833571

5 rows × 107 columns

```
# Normalización y clustering
clustering_columns = data_reco_cleaned_with_vectors.drop(columns=['Primary Identifier', 'Full Name'])
clustering_columns.columns = clustering_columns.columns.astype(str)



clustering_columns = data_reco_cleaned_with_vectors.drop(columns=['Primary Identifier', 'Full Name'])


# Verificar si hay listas en las columnas
for col in clustering_columns.columns:
    if clustering_columns[col].apply(lambda x: isinstance(x, list)).any():
        print(f"La columna '{col}' contiene listas")


# Convertir listas en columnas separadas
expanded_columns = []
for col in clustering_columns.columns:
    if clustering_columns[col].apply(lambda x: isinstance(x, list)).any():
        # Si la columna contiene listas, expandirlas
        expanded_col_df = pd.DataFrame(clustering_columns[col].tolist(), index=clustering_columns.index)
        expanded_col_df.columns = [f"{col}_{i}" for i in expanded_col_df.columns]
        expanded_columns.append(expanded_col_df)
    else:
        # Si la columna no contiene listas, mantenerla
        expanded_columns.append(clustering_columns[[col]])


# Concatenar las columnas expandidas en un solo DataFrame
clustering_columns_expanded = pd.concat(expanded_columns, axis=1)


clustering_columns_expanded.head()
```

	Gender Code	Age	City Code	Title	Loans (In House + Not In House)	0	1	2	3	4	...	90	91	92	9
0	0	26	0	['Crea & divaga : vida y reflexiones de Jeff B...	2	0.280615	-0.052416	0.307992	0.114421	-0.246577	...	-0.032930	0.135688	0.498295	-0.42671
1	1	19	1	['Colombia una historia mínima : una mirada in...	1	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
2	1	44	2	['Astronautas', '¿Qué es ese ruido?', 'Cuando ...	16	-0.177818	-0.199111	-0.268580	-0.104361	0.058935	...	0.051444	0.100448	0.224243	0.28488
3	1	21	2	['La campeona mundial de mantenerse despierta'...	8	-0.263781	-0.385908	-0.466116	-0.295928	0.014200	...	0.189849	-0.032371	0.396008	0.44455
4	0	24	0	['Romeo y Julieta', 'Orgullo y prejuicio', "No...	4	0.199063	-0.292346	-0.435913	-0.549494	-0.228809	...	-0.755373	0.079594	-0.833571	0.44821

5 rows × 105 columns

```
# Revisar las columnas y sus tipos de datos
print(clustering_columns_expanded.dtypes)
```

```
Gender Code                          int64
Age                                  int64
City Code                            int64
Title                               object
Loans (In House + Not In House)      int64
                                     ...
95                                 float64
96                                 float64
97                                 float64
98                                 float64
99                                 float64
Length: 105, dtype: object
```

```
# Eliminar la columna 'Title' ya que contiene valores no numéricos
clustering_columns_numeric = clustering_columns_expanded.drop(columns=['Title'])

# Verificar que la columna 'Title' ha sido eliminada
print(clustering_columns_numeric.dtypes)
```

```
Gender Code                          int64
Age                                  int64
City Code                            int64
Loans (In House + Not In House)      int64
0                                  float64
                                     ...
95                                 float64
96                                 float64
97                                 float64
98                                 float64
99                                 float64
Length: 104, dtype: object
```

```
from sklearn.preprocessing import StandardScaler
# Normalizar los datos
scaler = StandardScaler()
scaled_data = scaler.fit_transform(clustering_columns_numeric)
```

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

```
# Calcular SSE para diferentes valores de k
sse = []
silhouette_scores = []
k_values = range(2, 100000)
```

```
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    sse.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(scaled_data, kmeans.labels_))
```

Show hidden output

K-means

```
# Número de registros y atributos
shape = data_reco_cleaned_with_vectors.shape
print("Número de registros y atributos:", shape)
```

```
Número de registros y atributos: (107744, 107)
```

```
# K-Means
num_clusters = 30000
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
clusters = kmeans.fit_predict(scaled_data)
data_reco_cleaned_with_vectors['Cluster'] = clusters

print("Número de clusters y tamaño de cada uno:", data_reco_cleaned_with_vectors['Cluster'].value_counts())
```

```
Número de clusters y tamaño de cada uno: Cluster
3        215
608      206
49       153
14978    144
4603     135
        ...
14066      1
4531       1
27965      1
16891      1
27450      1
Name: count, Length: 30000, dtype: int64
```

```
# Verificar valores NaN
nan_values = data_reco_cleaned_with_vectors.isna().sum()
print("Valores NaN:\n", nan_values)
```

```
Valores NaN:
 Primary Identifier    0
Full Name              0
Gender Code            0
Age                    0
City Code              0
                      ..
96                     0
97                     0
98                     0
99                     0
Cluster                0
Length: 108, dtype: int64
```

```
data_reco_cleaned_with_vectors
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Loans (In House + Not In House)	0	1	2	...	91	92	
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	['Crea & divaga : vida y reflexiones de Jeff B...	2	0.280615	-0.052416	0.307992	...	0.135688	0.498295	-0.4
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	['Colombia una historia mínima : una mirada in...	1	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.0
2	43205110	Velez Restrepo Alexandra	1	44	2	['Astronautas', '¿Qué es ese ruido?', 'Cuando ...	16	-0.177818	-0.199111	-0.268580	...	0.100448	0.224243	0.2
3	C01C1000654424	HOLGUIN RESTREPO DAHIANA	1	21	2	['La campeona mundial de mantenerse despierta'...	8	-0.263781	-0.385908	-0.466116	...	-0.032371	0.396008	0.4
4	C01C1001033688	MESTRA MENESES GENESIS	0	24	0	['Romeo y Julieta', 'Orgullo y prejuicio', "No...	4	0.199063	-0.292346	-0.435913	...	0.079594	-0.833571	0.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
107739	XXAD985813541	VELEZ LOPERA DIEGO ALFREDO	0	53	193	['El poder invisible en acción', 'Ultimate Ele...	36	-0.325843	-0.368619	-0.336403	...	0.065907	0.340299	0.4
107740	XXAD98659045	APOYO DIDÁCTICO SERGIO OSVALDO RESTREPO JARAMILLO	0	48	1267	['Harry Potter y el misterio del príncipe (v6)...	10	-0.219635	0.050627	-0.497594	...	0.333386	-0.156405	0.3
107741	XXAD987158421	OTALVARO ALEXANDER	0	39	2	['Manual de educación canina guía completa de ...	253	-0.138680	-0.186923	-0.343813	...	0.047628	0.138010	0.2
107742	XXPD4319487	GIRALDO DUQUE HERNAN	0	82	2	['El gran libro de la mitología', 'Antología d...	8	-0.211005	0.236808	0.259652	...	-0.118466	0.340217	-0.0
107743	co1t1198465625	RESTREPO CANO JERONIMO	0	10	19	['Cartas a quien pretende enseñar', 'Una habit...	4	0.086940	-0.073065	-0.111857	...	-0.000559	0.008873	0.1

107744 rows × 108 columns

```
# Guardar el DataFrame limpio
data_reco_cleaned_with_vectors.to_csv("/content/drive/MyDrive/datas/data_reco_cleaned_with_vectors_30000.csv", index=False)
```

```
# Montar Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Cargar los datos
#La 22 es de 10000
#la 20000 es de 20000
# la 30000 es de 30000
data_reco_cleaned_with_vectors_2 = pd.read_csv("/content/drive/MyDrive/datas/data_reco_cleaned_with_vectors_30000.csv")
data_reco_cleaned_with_vectors_2.head()
```

	Primary Identifier	Full Name	Gender Code	Age	City Code	Title	Loans (In House + Not In House)	0	1	2	...	91	92	93
0	1007202128	MARTINEZ AGUIRRE FERNEY DUBAN	0	26	0	['Crea & divaga : vida y reflexiones de Jeff B...	2	0.280615	-0.052416	0.307992	...	0.135688	0.498295	-0.426712
1	1035973109	RODRIGUEZ LAVERDE HAROLD	1	19	1	['Colombia una historia mínima : una mirada in...	1	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
2	43205110	Velez Restrepo Alexandra	1	44	2	['Astronautas', '¿Qué es ese ruido?', 'Cuando ...	16	-0.177818	-0.199111	-0.268580	...	0.100448	0.224243	0.284885
3	C01C1000654424	HOLGUIN RESTREPO DAHIANA	1	21	2	['La campeona mundial de mantenerse despierta'...	8	-0.263781	-0.385908	-0.466116	...	-0.032371	0.396008	0.444555
4	C01C1001033688	MESTRA MENESES GENESIS	0	24	0	['Romeo y Julieta', 'Orgullo y prejuicio', "No...	4	0.199063	-0.292346	-0.435913	...	0.079594	-0.833571	0.448213

5 rows × 108 columns

```
# Contar la cantidad de usuarios por clúster
cluster_counts_example = data_reco_cleaned_with_vectors_2['Cluster'].value_counts().reset_index()

# Renombrar columnas para claridad
cluster_counts_example.columns = ['Cluster', 'Cantidad de Usuarios']

# Mostrar las primeras filas como ejemplo
cluster_counts_example.head(30000)
```

	Cluster	Cantidad de Usuarios
0	3	215
1	608	206
2	49	153
3	14978	144
4	4603	135
...	...	...
29995	14066	1
29996	4531	1
29997	27965	1
29998	16891	1
29999	27450	1

30000 rows × 2 columns

```
# Calcular la cantidad de usuarios por cluster
cluster_counts_example = data_reco_cleaned_with_vectors_2['Cluster'].value_counts()
```

```
# Calcular el promedio de usuarios por cluster
promedio_usuarios_por_cluster = cluster_counts_example.mean()

# Mostrar el promedio
print(f"Promedio de usuarios por cluster: {promedio_usuarios_por_cluster:.2f}")
```

```
Promedio de usuarios por cluster: 3.59
```

Títulos recomendados

```
import json
import ast
```

```
from sklearn.metrics.pairwise import cosine_similarity

# Función para recomendar libros considerando usuarios únicos
def recomendar_libros(usuario_id, num_recomendaciones=20):
    # Si el usuario existe en el dataset
    if usuario_id in data_reco_cleaned_with_vectors_2['Primary Identifier'].values:
        # Obtener el cluster del usuario
        cluster_usuario = data_reco_cleaned_with_vectors_2.loc[data_reco_cleaned_with_vectors_2['Primary Identifier'] == usuario_id, 'Cluste

        # Verificar si el clúster tiene solo un usuario
        usuarios_cluster = data_reco_cleaned_with_vectors_2[data_reco_cleaned_with_vectors_2['Cluster'] == cluster_usuario]['Primary Identif
        if len(usuarios_cluster) == 1:
            # Usuario único en el clúster: calcular similitudes con todos los usuarios
            vector_usuario = data_reco_cleaned_with_vectors_2[data_reco_cleaned_with_vectors_2['Primary Identifier'] == usuario_id].iloc[0,
            usuarios_vectores = data_reco_cleaned_with_vectors_2.iloc[:, -100:].values
            similitudes = cosine_similarity([vector_usuario], usuarios_vectores)[0]

            # Agregar las similitudes al dataset
            data_reco_cleaned_with_vectors_2['Similaridad'] = similitudes

            # Seleccionar los usuarios más similares (excluyendo el propio usuario)
            usuarios_similares = data_reco_cleaned_with_vectors_2.sort_values(by='Similaridad', ascending=False).head(50)

            # Obtener los libros más prestados por estos usuarios similares
            libros_similares = usuarios_similares.groupby('Title')['Loans (In House + Not In House)'].sum()
            recomendaciones = libros_similares.nlargest(num_recomendaciones).index.tolist()

        else:
            # Clúster con múltiples usuarios: recomendaciones estándar
            libros_similares = data_reco_cleaned_with_vectors_2[data_reco_cleaned_with_vectors_2['Cluster'] == cluster_usuario]
            libros_mas_prestados = libros_similares.groupby('Title')['Loans (In House + Not In House)'].sum()
            recomendaciones = libros_mas_prestados.nlargest(num_recomendaciones).index.tolist()

        libros = []
        for lista in recomendaciones:
            lista = ast.literal_eval(lista)
            libros.extend(lista)
        return libros, usuarios_cluster

    # Si el usuario no existe, recomendar libros populares
    else:
        libros_populares = data_reco_cleaned_with_vectors_2[['Title', 'Loans (In House + Not In House)']].sort_values(by='Loans (In House +
        return f"Libros recomendados para un nuevo usuario: {list(libros_populares)}"
```

```
data_reco_cleaned = pd.read_csv("/content/drive/MyDrive/datas/data_Comfama_20241.csv")
```

Usuarios de prueba

```
usuario_id = 'CO1C1128416568'
leidos_usuario = data_reco_cleaned[data_reco_cleaned['Primary Identifier'] == usuario_id]["Title"].to_list()
recomendaciones, usuarios_cluster = recomendar_libros(usuario_id)

# Filtrar recomendaciones que no han sido leídas por el usuario
```

```
recomendaciones = [recomendacion for recomendacion in recomendaciones if recomendacion not in leidos_usuario]

# Obtener los libros leídos por los usuarios del mismo cluster
leidos_cluster = data_reco_cleaned[data_reco_cleaned['Primary Identifier'].isin(usuarios_cluster)]["Title"].to_list()

# Contar la frecuencia de los libros leídos por los usuarios del cluster
conteo = {}
for libro in leidos_cluster:
    if libro in conteo:
        conteo[libro] += 1
    else:
        conteo[libro] = 1

# Filtrar los libros que están en las recomendaciones y ordenarlos por el conteo
conteo = {libro: conteo[libro] for libro in recomendaciones if libro in conteo}

# Convertir el diccionario de conteos a un DataFrame
conteo_df = pd.DataFrame(list(conteo.items()), columns=['Libro', 'Conteo'])

# Obtener los detalles de 'Author' y 'Subjects' para los libros recomendados
conteo_df['Author'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Author'].values[0] if x in data
conteo_df['Subjects'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Subjects'].values[0] if x in

# Ordenar por 'Conteo'
conteo_df = conteo_df.sort_values(by='Conteo', ascending=False).reset_index(drop=True)

# Mostrar las primeras 20 recomendaciones con author y subjects
conteo_df.head(20)
```

	Libro	Conteo	Author	Subjects
0	1984	6	NaN	Novela gráfica.; Literatura--Adaptaciones.; Gu...
1	Cien años de soledad	3	García Márquez, Gabriel, 1927-2014.	Novela colombiana--Siglo XX.; Realismo mágico ...
2	Las lágrimas de Shiva	2	Mallorquí, César, 1953-	Novela infantil española; Objetos perdidos--No...
3	Fahrenheit 451	2	Bradbury, Ray 1920-2012	Novela estadounidense--Siglo XX.; Censura--nov...
4	El futuro del espaciotiempo	2	Hawking, Stephen 1942-2018	Relatividad (Física); Teoría cuántica; Teoría ...
5	Delirio	2	Restrepo, Laura, 1950-,.	Novela colombiana--Siglo XXI.; Tráfico de drog...
6	Los juegos del hambre (v1)	2	Collins, Suzanne 1962-,	Literatura estadounidense--Sagas
7	El principito	2	Saint-Exupéry, Antoine de, 1900-1944, Autor e ...	Novela infantil francesa.; Valores sociales.; ...
8	El demonio y la señorita Prym	2	Coelho, Paulo 1947-,	Novela brasilera--Siglo XX; Literatura brasile...
9	El fin del mundo y un despiadado país de las m...	2	Murakami, Haruki, 1949-,.	Novela japonesa; Conciencia (Psicología)--Nove...
10	La divina comedia	2	Alighieri, Dante, 1265-1321.	Dante, Alighieri - La divina comedia - Crítica...
11	Cinco semanas en globo	1	Verne, Julio, 1828-1905.	Novela francesa; Aventuras--Novela; Viajes--No...
12	La Biblia perdida	1	Bergler, Igor. 1970-,	Novela rumana; Biblia--Novela; Vampiros--Novel...
13	Cuando seas mayor	1	Gane, Miguel, (Seudónimo de George Mihaita Gan...	Novela española; Pobreza--Novela; Rumanía--Nov...
14	Mitología para chicos	1	Catalano, Daniel	Mitología griega; Mitología romana; Literatura...
15	Colombia bizarra : los incidentes más insóli...	1	Pirry, 1970-,.	Crónicas periodísticas--Colombia.; Periodismo-...
	Las aventuras de tres rucos y tres ingleses			

```
usuario_id = 'CO1C1128389893'
leidos_usuario = data_reco_cleaned[data_reco_cleaned['Primary Identifier'] == usuario_id]["Title"].to_list()
recomendaciones, usuarios_cluster = recomendar_libros(usuario_id)

# Filtrar recomendaciones que no han sido leídas por el usuario
recomendaciones = [recomendacion for recomendacion in recomendaciones if recomendacion not in leidos_usuario]

# Obtener los libros leídos por los usuarios del mismo cluster
leidos_cluster = data_reco_cleaned[data_reco_cleaned['Primary Identifier'].isin(usuarios_cluster)]["Title"].to_list()

# Contar la frecuencia de los libros leídos por los usuarios del cluster
conteo = {}
```

```
for libro in leidos_cluster:
    if libro in conteo:
        conteo[libro] += 1
    else:
        conteo[libro] = 1

# Filtrar los libros que están en las recomendaciones y ordenarlos por el conteo
conteo = {libro: conteo[libro] for libro in recomendaciones if libro in conteo}

# Convertir el diccionario de conteos a un DataFrame
conteo_df = pd.DataFrame(list(conteo.items()), columns=['Libro', 'Conteo'])

# Obtener los detalles de 'Author' y 'Subjects' para los libros recomendados
conteo_df['Author'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Author'].values[0] if x in data
conteo_df['Subjects'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Subjects'].values[0] if x in

# Ordenar por 'Conteo'
conteo_df = conteo_df.sort_values(by='Conteo', ascending=False).reset_index(drop=True)

# Mostrar las primeras 20 recomendaciones con author y subjects
conteo_df.head(20)
```

	Libro	Conteo	Author	Subjects
0	Bailando juntos La cara oculta del amor en la ...	2	Garriga, Joan 1957-,	Relaciones de pareja; Amor - Aspectos psicológ...
1	Así habló Zaratustra; Más allá del bien y del mal	2	Nietzsche, Friedrich 1844-1900	Nietzsche, Friedrich, 1844-1900--Pensamiento f...
2	El extranjero	2	Camus, Albert, 1913-1960.	Novela francesa; Asesinatos en serie; Existenc...
3	Cincuenta sombras de Grey (v1)	2	James, Erika Leonard 1963-,	Novela inglesa; Literatura erótica; Sexo en la...
4	El arte de liderar	1	Alberoni, Francesco. 1929-,	Liderazgo; Líderes--Gestión administrativa; Po...
5	El mago de toledo	1	Vigil, Mercedes, 1957-	Novela uruguaya; Bien y mal--Novela; Masonería...
6	Cómo El Secreto cambió mi vida: Gente real. Hi...	1	Byrne, Rhonda, 1945-,.	Actitud (psicología); Actitud; Emociones
7	Cuentos de seducción	1	Vincenti, Carmen. 1943-,	Cuentos venezolanos; Relaciones de pareja--Cue...
8	¡Estoy agotada!	1	Parada Escribano, Alejandra. 1966-,	Novela chilena
9	El testamento de Judas	1	Easterman, Daniel. 1949-,	Novela irlandesa; Novela de misterio; Sociedad...
10	El penúltimo sueño	1	Becerra Acevedo, Ángela, 1957-,.	Novela colombiana; Relaciones de pareja--Novel...
11	Lo que no tiene nombre	1	Bonnett, Piedad, 1951-,.	Segura Bonnett, Daniel, 1983-2011--Suicidio; N...
12	El arte de vivir : elige la paz y la libertad....	1	Nhat Hanh, Thich, 1926-2022.	Vida espiritual--Budismo zen.; Budismo; Espiri...
13	Para otros es el cielo	1	Bonnett, Piedad, 1951-,.	Novela colombiana; Amor--Novela; Bondad--Novel...
14	El hombre que amaba las gaviotas y otros relatos	1	Osho, 1931-1990 (Chandra Mohan Jain)	Espiritualidad; Meditación; Paz del espíritu; ...
15	El códice secreto	1	Grossman, Lev. 1969-,	Novela estadounidense; Novela psicológica; Man...
16	La flor púrpura	1	Adichie, Chimamanda Ngozi 1977-,	Novela nigeriana; Literatura africana; Relacio...
17	El libro del ego liberarse de la ilusión	1	Osho, 1931-1990 (Chandra Mohan Jain)	Ego (Psicología); Éxito - Aspectos psicológico...
18	El rabino	1	Gordon, Noah 1926-2021	Novela estadounidense; Judíos--Novela estadoun...
19	De la estupidez a la locura : cómo vivir en un...	1	Eco, Umberto 1932-2016	Ensayo italiano--Siglo XXI; Artículos de prens...

```
usuario_id = 'CO1C43270431'
leidos_usuario = data_reco_cleaned[data_reco_cleaned['Primary Identifier'] == usuario_id]["Title"].to_list()
recomendaciones, usuarios_cluster = recomendar_libros(usuario_id)

# Filtrar recomendaciones que no han sido leídas por el usuario
recomendaciones = [recomendacion for recomendacion in recomendaciones if recomendacion not in leidos_usuario]

# Obtener los libros leídos por los usuarios del mismo cluster
leidos_cluster = data_reco_cleaned[data_reco_cleaned['Primary Identifier'].isin(usuarios_cluster)]["Title"].to_list()

# Contar la frecuencia de los libros leídos por los usuarios del cluster
conteo = {}
for libro in leidos_cluster:
    if libro in conteo:
        conteo[libro] += 1
    else:
        conteo[libro] = 1

# Filtrar los libros que están en las recomendaciones y ordenarlos por el conteo
```

```
conteo = {libro: conteo[libro] for libro in recomendaciones if libro in conteo}

# Convertir el diccionario de conteos a un DataFrame
conteo_df = pd.DataFrame(list(conteo.items()), columns=['Libro', 'Conteo'])

# Obtener los detalles de 'Author' y 'Subjects' para los libros recomendados
conteo_df['Author'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Author'].values[0] if x in data
conteo_df['Subjects'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Subjects'].values[0] if x in

# Ordenar por 'Conteo'
conteo_df = conteo_df.sort_values(by='Conteo', ascending=False).reset_index(drop=True)

# Mostrar las primeras 20 recomendaciones con author y subjects
conteo_df.head(20)
```

	Libro	Conteo	Author	Subjects
0	¡No más besos!	5	Chichester Clark, Emma 1955-	Cuentos infantiles ingleses; Besos--Cuentos in...
1	El más poderoso	4	Kasza, Keiko, 1951-, Autor e ilustrador	Cuentos infantiles; Literatura infantil japone...
2	¿Por qué lloramos?	3	Pintadera, Fran, 1982-,.	Cuentos infantiles españoles; Emociones--Cuent...
3	Nano y sus amigos	3	Da Coll, Ivar, 1962-,.	Cuentos infantiles colombianos; Amistad en la ...
4	Mi mamá	3	Browne, Anthony, 1946-, (Anthony Edward Tudor ...	Cuentos infantiles ingleses; Madre e hijo--Cue...
5	Una cena elegante	3	Kasza, Keiko, 1951-, Autor e ilustrador	Animales en la literatura; Cuentos infantiles;...
6	¿Cómo era yo cuando era un bebé?	3	Willis, Jeanne 1959-,	Cuentos infantiles ingleses; Niños--Cuentos in...
7	Mi mamá es mágica	3	Norac, Carl, 1960 -,.	Cuentos infantiles franceses.; Madres e hijas-...
8	¡Cuidado! ¡Palabra terrible!	3	Schreiber-Wicke, Edith 1943-,	Cuentos infantiles; Literatura infantil aleman...
9	Cuentos pintados	3	Pombo, Rafael, 1833-1912.	Cuentos infantiles colombianos; Cuentos popula...
10	El tigre y el ratón	3	Kasza, Keiko, 1951-, Autor Ilustrador	Cuentos infantiles japoneses; Animales en la l...
11	El perro que quiso ser lobo Keiko Kasza; Tradu...	3	Kasza, Keiko, 1951-, Autora e ilustradora	Cuentos infantiles; Literatura infantil japone...
12	Infinito : los ciclos mágicos del universo	3	Romero Mariño, Soledad	Evolución.; Historia natural; Cosmología.; Cos...
13	El monstruo de colores	3	Llenas, Anna, 1977-,.	Cuentos infantiles españoles; Monstruos--Cuent...
14	Mi día de suerte	2	Kasza, Keiko, 1951-, Autor e ilustrador	Cuentos infantiles japoneses; Inteligencia--Cu...
15	Franklin juega al fútbol	2	Bourgeois, Paulette 1951-,	Cuentos infantiles canadienses; Tortugas--Cuen...
16	Sofía y los lechuguitas	2	Green, Ilya, 1976-, Autor e ilustrador	Cuentos infantiles franceses; Verdad y mentira

```
#Marian Montoya
usuario_id = 'CO1C1128268344'
leidos_usuario = data_reco_cleaned[data_reco_cleaned['Primary Identifier'] == usuario_id]["Title"].to_list()
recomendaciones, usuarios_cluster = recomendar_libros(usuario_id)

# Filtrar recomendaciones que no han sido leídas por el usuario
recomendaciones = [recomendacion for recomendacion in recomendaciones if recomendacion not in leidos_usuario]

# Obtener los libros leídos por los usuarios del mismo cluster
leidos_cluster = data_reco_cleaned[data_reco_cleaned['Primary Identifier'].isin(usuarios_cluster)]["Title"].to_list()

# Contar la frecuencia de los libros leídos por los usuarios del cluster
conteo = {}
for libro in leidos_cluster:
    if libro in conteo:
        conteo[libro] += 1
    else:
        conteo[libro] = 1

# Filtrar los libros que están en las recomendaciones y ordenarlos por el conteo
conteo = {libro: conteo[libro] for libro in recomendaciones if libro in conteo}

# Convertir el diccionario de conteos a un DataFrame
conteo_df = pd.DataFrame(list(conteo.items()), columns=['Libro', 'Conteo'])

# Obtener los detalles de 'Author' y 'Subjects' para los libros recomendados
```

```
conteo_df['Author'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Author'].values[0] if x in dat
conteo_df['Subjects'] = conteo_df['Libro'].apply(lambda x: data_reco_cleaned[data_reco_cleaned['Title'] == x]['Subjects'].values[0] if x in

# Ordenar por 'Conteo'
conteo_df = conteo_df.sort_values(by='Conteo', ascending=False).reset_index(drop=True)

# Mostrar las primeras 20 recomendaciones con author y subjects
conteo_df.head(20)
```

	Libro	Conteo	Author	Subjects
0	Más fuerte que la adversidad : cómo afrontar l...	2	Riso, Walter, 1951-,.	Personalidad.; Conducta humana.; Ansiedad; Est...
1	Momo	2	Ende, Michael 1929-1995	Novela juvenil Alemana; Novela Fantástica; Ami...
2	Maravillosamente imperfecto, escandalosamente ...	2	Riso, Walter, 1951-,.	Superación personal; Autoayuda; Crecimiento pe...
3	Historia de la sexualidad. La voluntad de sabe...	2	Foucault, Michel 1926-1984	Psicología de la sexualidad; Conducta sexual; ...
4	Odisea	2	Homero.	Homero--Crítica e interpretación.; Literatura ...
5	Hazte dueño de ti : una guía para vivir con pr...	2	Martinez, Efrén.	Autorrealización (Psicología); Amor.; Conducta...
6	Mujeres conscientes : diez movimientos para la...	2	Echegoyen, Agustina. 1988-,	Mujeres - Conducta de vida; Psicología de la m...
7	Dioses sois	2	Miranda, Mercedesi	Jesucristo - Enseñanzas; Jesucristo--Vida cris...
8	Hombres sin mujeres	1	Murakami, Haruki, 1949-,.	Cuentos japoneses; Mujeres en la literatura; A...
9	Ilustración de moda : dibujo plano	1	Lleonart, Aitana	Diseño de vestidos; Moda--Dibujo; Corte y Conf...
10	Historia secreta de la música	1	Marín, Alejandro.	Música--Historia; Canciones--Historia; Música-...
11	Diciembre, otra vez	1	Cruz, Santiago, 1976-	Cruz, Santiago,1976--Autobiografía; Cruz, Sant...
12	Frankenstein	1	NaN	Shelley, Mary,--1797-1851.; Novela juvenil ing...
13	Black music : free jazz y conciencia negra 195...	1	Jones, Leroi	Música de Jazz--1957-1967--Estados unidos; Mús...
14	Verónika decide morir	1	Coelho, Paulo 1947-,	Literatura brasileña; Vida--Aspectos psicológi...
15	Ilustración de moda	1	Morris, Bethan	Diseño de modas; Moda--Dibujo; Dibujo de modas...
16	Punto a punto	1	Machado, Ana Maria, 1941-,.	Cuentos infantiles brasileños--Siglo XXI.; Muj...
17	La condición humana	1	Arendt, Hannah, 1906-1975.	Psicología social; interacción social; Filosof...
18	Guía completa de bordado : 500 combinaciones d...	1	Bothell, Valerie, 1962-	Bordado; Labores de aguja; Punto de cruz; Patc...
19	Y si me gustara morir	1	Cardoso Martins, Rui, 1967-,.	Novela portuguesa--Siglo XXI.; Suicidio--Novel...

```
#Maria Camila Molina
usuario_id = 'CO1C1017203312'
leidos_usuario = data_reco_cleaned[data_reco_cleaned['Primary Identifier'] == usuario_id]["Title"].to_list()
recomendaciones, usuarios_cluster = recomendar_libros(usuario_id)

# Filtrar recomendaciones que no han sido leídas por el usuario
recomendaciones = [recomendacion for recomendacion in recomendaciones if recomendacion not in leidos_usuario]

# Obtener los libros leídos por los usuarios del mismo cluster
leidos_cluster = data_reco_cleaned[data_reco_cleaned['Primary Identifier'].isin(usuarios_cluster)]["Title"].to_list()

# Contar la frecuencia de los libros leídos por los usuarios del cluster
conteo = {}
for libro in leidos_cluster:
    if libro in conteo:
```