
Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

# Desarrollo de un sistema de recomendaciones para un sitio de Comercio Electrónico

Trabajo especial de grado presentado ante la Ilustre

Universidad Central de Venezuela

Por el Br. Oscar Valecillos

Tutores:

Jesús Lares

Haydemar Nuñes

Caracas, febrero de 2019

## Acta

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado Desarrollo de un Sistema de Recomendaciones para un sitio de Comercio Electrónico presentado por El Br. Oscar Alberto Valecillos Girand (C.I. V-22343034) a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

Leído el trabajo por cada uno de los miembros del jurado, se fijó el día 20 de mayo, a las 11 horas de la mañana, para que el autor lo defendiera en forma pública, a través de videoconferencia, haciendo una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas.

El profesor y tutor Jesús Lares estará también presente via videoconferencia.

Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 20 de mayo de 2019.

Por Jesus

Prof. Jesús Lares (Tutor)

Prof. Andrés Sanoja (Jurado)

Profa. Mercy Ospina (Jurado)

# Resumen

Título: Desarrollo de un sistema de recomendación para el sitio de comercio electrónico Aprovecha.com

Autor: Oscar Valecillos

Tutores: Prof. Jesús Lares, Profa. Haydemar Nuñez

Desde el inicio de internet, el comercio electrónico ha estado creciendo de forma significativa, aprovechando todas las tecnologías que mejoren el negocio y tengan como fin que el usuario haga compras a través del sistema y se sienta más satisfecho con el proceso. Debido a la alta competencia que existe entre comercios electrónicos (conocido por su denominación en inglés como e-commerce), estos se ven en la necesidad de implementar soluciones que mejoren significativamente la experiencia de usuario, con el objetivo, de aumentar la intención de compra e incrementar sus ventas. Dejar de lado la atención de mejorar la satisfacción y personalización del usuario, puede tener un impacto negativo en el negocio y un bajo posicionamiento a diferencia de quienes sí lo tienen.

En el marco de los grandes volúmenes de datos, la ciencia de datos y el aprendizaje automático, una de las soluciones posibles para ofrecer una buena experiencia de usuario, es el desarrollo de un sistema de recomendación; con base en el comportamiento del usuario, sus compras, sus visitas y sus opiniones, se busca crear una experiencia más personalizada para el comprador y así incrementar las ventas. El objetivo de este proyecto es desarrollar un sistema de recomendación para el e-commerce Aprovecha.com.

Para lograr esto, se utilizó el lenguaje de programación Python, MongoDB como almacén de datos y siguiendo la metodología para la ciencia de datos de IBM. El problema fue analizado, los datos fueron recolectaros, preparados y así se realizó la predicción de las recomendaciones, mediante un filtrado colaborativo; se implementaron dos propuestas de algoritmos para las recomendaciones y cada uno fue evaluado con métricas de precisión estadística, siendo elegido como sistema recomendación el más preciso de ambos.

Finalmente, el sistema de recomendación fue desarrollado y evaluado, quedando para la integración con el e-commerce Aprovecha.com



Palabras Clave: Ciencias de datos, minería de datos, machine learning, aprendizaje automático, comercio electrónico, e-commerce, sistema de recomendación.

# Índice

# Índice de Figuras

# Introducción

El uso de los datos para la adquisición de información relevante, siempre ha estado presente en las empresas. Sin embargo, antiguamente, conseguir esta información oculta desde los datos, era bastante complicada y tediosa para las compañías. En los últimos años, esto ha mejorado gracias a una serie de herramientas que permiten la recolección, manipulación y almacenamiento de grandes volúmenes de datos de una forma eficiente y segura; estos datos son cada vez más grandes, son generados a mayor velocidad y poseen diferentes formatos, de manera que los medios tradicionales de almacenamiento no son capaces de procesar esta cantidad de datos; esto ha impulsado a la aparición de estas herramientas que hacen posible el manejo de estos grandes volúmenes de datos.

A partir de esta gran cantidad de datos, es posible conseguir información oculta que puede llevar a toma de decisiones importantes para una compañía.

La ciencia de datos utiliza estos grandes volúmenes de datos, para generar información relevante, esto se consigue haciendo uso de algoritmos de aprendizaje automático y la utilización de la computación paralela y distribuida para sacar el mejor provecho de dichos datos. La ciencia de datos es llevada a cabo por un científico de datos, un rol que es capaz de llevar a cabo los diferentes procedimientos para la explotación de estos datos desde un estado crudo, hasta información importante que se convierte en un valor para la empresa.

El uso de Big data y la ciencia de datos, ha incrementado en los últimos años en diferentes áreas. Un área donde se puede aprovechar de manera significativa es en el comercio electrónico, ya que es un medio donde se genera una gran cantidad de información, de las cual una gran parte se encuentra oculta en los datos generados por los usuarios. Existen una serie de aplicaciones posibles en este nicho que serán expuestas a lo largo de este informe al igual que las distintas herramientas para llevar a cabo las mismas.

Este trabajo especial se encuentra dividido en seis (6) capítulos. El capítulo uno hace el planteamiento del problema, expone los objetivos generales y específicos. El capítulo dos expone los diferentes principios teóricos y conceptuales utilizados para desarrollar este proyecto. En el capítulo tres se presenta la metodología de desarrollo en que se basó este proyecto para ser realizado. El capítulo 4 comprende la explicación detallada paso a paso del desarrollo del sistema de recomendación, siguiendo la metodología seleccionada. El capítulo 5 concluye este proyecto, se reconocen las contribuciones, se hacen las recomendaciones pertinentes y por último evaluación de trabajos futuros. Por último, en el capítulo 6 se pueden encontrar las referencias bibliográficas utilizadas para este proyecto.

# Capítulo 1: Planteamiento del problema

En este capítulo se expone el planteamiento y justificación del problema de este proyecto, como también el objetivo general y específicos del mismo.

## 1.1 Planteamiento del problema

Para un comercio electrónico, una de las formas de mostrarse ante los usuarios es creando y ofreciendo una mejor experiencia para los mismos, la cual se logra mediante diversas técnicas, por ejemplo: una buena interfaz de usuario, atención al cliente, usabilidad, accesibilidad; sumado a esto, se busca crear en el usuario la sensación que el sistema “lo conoce”, entendiendo sus gustos y preferencias, por lo que se le puede ofrecer productos y servicios que se adapten a sus intereses y necesidades, así como también, mejores opciones de compra, teniendo como resultado mejorar la experiencia de usuario y un mejor ambiente para el mismo.

Se puede entender que un *e-commerce* que no incluya una experiencia de usuario en su sitio tendría que competir ante otros que si lo implemente. Por consecuencia, surge la necesidad de aumentar la experiencia de usuario siendo un sistema de recomendación de productos una opción para lograr la satisfacción en el usuario e incrementar las ventas en el sitio.

Por lo tanto, se evidencia que existe una gran competencia en este nicho, en donde cada comercio electrónico (dependiendo de los productos o servicios que preste) trata de destacarse ante los usuarios para incrementar su productividad y a su vez aumentar su eficiencia.

Desde los inicios de internet, ha existido la necesidad o la disposición de ofrecer productos y servicios a través de este medio. Al principio, la popularidad de estos sitios que ofrecían la compra de dichos productos o servicios era bastante baja debido a la poca confianza de los usuarios hacia el sistema o por el poco conocimiento que tenían al respecto. No obstante, a medida que las tecnologías han evolucionado de manera constante y el uso de internet ha crecido, la confianza de compra y venta de productos a través de medios electrónicos ha aumentado significativamente.

Actualmente, existen grandes empresas que se dedican exclusivamente al comercio electrónico como Amazon, eBay, MercadoLibre, Groupon, así como también existen empresas de ventas al por menor que extienden sus modelos de negocio a una plataforma digital para ofrecer sus propios productos o servicios tales como Target, Pizza Hut, Nike, entre otros más. Al igual que estas compañías, existen pequeñas empresas emprendedoras que también deciden adentrarse en el negocio del *e-commerce*.

Aprovecha.com, siendo una de estas pequeñas empresas que inició como un proyecto emprendedor, actualmente no cuenta con un sistema de recomendación implementado, y por consecuencia lo deja atrás en las técnicas utilizadas por los comercios electrónicos de hoy en día para agregar experiencia al usuario, y al desarrollarlo le daría un valor agregado a esta compañía que está

en constante crecimiento y busca exhaustivamente ofrecer una mejor experiencia de usuario.

## 1.1 Justificación

Hoy en día, no contar con un sistema de recomendación representa un problema para cualquier compañía o negocio que decida adentrarse en el mundo del comercio electrónico, ya que los deja en una gran desventaja ante otras compañías que sí lo implementan. La existencia de un sistema de recomendación aumenta la producción y eficacia del negocio, garantizando un mayor valor del mismo, debido a que incrementa las ventas gracias a la mejora de la experiencia de usuario, todo esto teniendo como consecuencia ventajas al momento de competir con otras compañías.

Con el uso de los grandes volúmenes de datos, algoritmos de analítica predictiva y una metodología para la ciencia de datos, se hace posible abordar la problemática de la poca experiencia de usuario que yace actualmente en Aprovecha.com, al permitir realizar análisis sobre los datos teniendo como resultado un sistema de recomendación. Este proyecto busca desarrollar dicho sistema para la compañía y aportar todos los beneficios antes mencionados.

La solución a esta diatriba puede verse plasmada en grandes compañías de e-commerce como Amazon, que han tenido éxito debido a la buena experiencia de usuario que brindan, su sistema recomendador permite a los usuarios tener una buena experiencia de compra lo que incrementa y da valor al negocio.

## 1.2 Objetivos

### 1.2.1 Objetivo General

Desarrollar un sistema de recomendaciones que incremente la experiencia de usuario en el sitio de comercio electrónico Aprovecha.com, a partir del uso de los datos almacenados de la empresa utilizando algoritmos de analítica predictiva y almacenes de grandes volúmenes de datos.

### 1.2.2 Objetivos Específicos

- Definir el enfoque y alcance del sistema de recomendaciones de Aprovecha.com.
- Definir la(s) metodología(s) de desarrollo de aplicaciones para el sistema de recomendaciones de Aprovecha.com.
- Seleccionar las herramientas, marcos de trabajos (*Frameworks*), Almacenes de datos y lenguajes de programación a utilizar para el desarrollo del sistema de recomendaciones de Aprovecha.com.
- Analizar y seleccionar los datos que permitan definir e implementar los modelos de analítica predictiva para el desarrollo del sistema de recomendaciones de Aprovecha.com.

- Desarrollar un sistema de recomendaciones para Aprovecha.com sobre la base de la(s) metodología(s) y herramientas de desarrollo seleccionadas.
- Realizar pruebas para determinar la precisión del sistema de recomendaciones de Aprovecha.com.

## 1.5 Antecedentes

Los sistemas de recomendación existen de diversas maneras, desde recomendaciones básicas en revistas hasta los sofisticados que existen hoy en día. La utilización de estos sistemas en el comercio electrónico, han dado un valor agregado a este negocio ya que entra en contacto con los gustos de los usuarios y puede llevar a incremento de ventas y compras cruzadas.

Amazon, que es uno de los comercios electrónicos más grandes en la actualidad [134], se ha caracterizado por "vender para cada cliente" haciéndolos descubrir productos de su interés que no habrían conseguido de otra manera, esto con el fin de facilitar la experiencia al usuario e incrementar sus ventas. Amazon utiliza un filtrado colaborativo basado en ítem (se definirá más adelante en este trabajo) para hacer las recomendaciones, las mismas son de productos similares a los que compró o visualizó el usuario, productos que complementan o se juntan con el que está visualizando el usuario para así hacer ventas cruzadas, lo que usuarios similares han comprado, entre otros [135]. Gracias a este sistema de recomendación, Amazon se ha ganado la fama de ser un sistema pensado para los usuarios.

Otro caso notable es el de Netflix. Netflix no es un comercio electrónico, es un sistema de streaming de películas y series, sin embargo tiene un sistema de recomendación que dice a sus usuarios qué películas ver con base en las que han visto antes, este caso destaca ya que esta compañía creó un concurso abierto llamado "The Netflix Prize" [136] donde grupos de desarrolladores que intentarían crear el sistema de recomendación con datos proporcionados por la empresa por un premio final. Esto muestra la importancia para la empresa de contar con un motor de recomendación y así aumentar y mejorar la experiencia de usuario.

Al igual que Netflix y Amazon, diversas compañías, bien sean de comercio electrónico u otros rubros, han implementado un sistema de recomendación. En el caso de Aprovecha.com, no posee un sistema de recomendación de usuarios, siendo el objetivo de este trabajo de investigación crearlo.

News Dude es un sistema de noticias personalizadas que utiliza el habla sintetizada para leer nuevas noticias a los usuarios. Un modelo de frecuencia inversa de documento (TF-IDF) es utilizada para describir nuevas noticias con el fin de determinar las recomendaciones a corto plazo que luego son comparadas con la similitud del coseno y finalmente se suplen a un algoritmo de aprendizaje. CiteSeer es un sitio de indexación de citas (académicas) automático que utiliza varias heurísticas y algoritmos de aprendizaje automático para procesar los documentos. Actualmente, CiteSeer es uno los repositorios de documentos de investigación ampliamente más usados. [41] [130]

LIBRA es un sistema de recomendación basado en contenido de libros que utiliza información sobre los mismos obteniéndola de internet. Implementa un clasificador bayesiano ingenuo sobre la información extraída de la web para aprender un perfil de usuario para producir una lista calificada de títulos basadas en ejemplos de entrenamiento proporcionados por un usuario individual. [41] [131]

Ringo un sistema de filtrado colaborativo basado en usuario que hace recomendaciones de álbumes de música y artistas. En Ringo, cuando un usuario inicializa entrando al sistema, una lista de 125 artistas es dado al mismo para que califique según qué tanto le gusta escucharlos. La lista es hecha de dos secciones diferentes. La primera sesión consiste en los artistas más valorados y esto permite al usuario la oportunidad de calificar artistas que otros usuarios calificaron igualmente, por lo que hay un nivel de similitudes entre los perfiles de diferentes usuarios. La segunda es generada en una selección al azar de ítems de la matriz usuario-ítem, por lo que todos los artistas y álbumes son eventualmente valorados en algún punto de las fases iniciales de calificación. [41] [ 132]

GroupLens es un sistema de filtrado colaborativo basado en la arquitectura cliente-servidor; el sistema recomienda noticias Usenet, que es un servicio de listas de discusión de gran volumen en internet. La corta vida de las Netnews y el subyacente esparcimiento de las matrices de calificación son los dos retos principales enfrentados por este sistema. Los usuarios y las Netnews son agrupados basados en los grupos de noticias existentes en el sistema y los *ratings* implícitos son calculados midiendo el tiempo que el usuario pasa leyendo cada noticia. [41] [ 133]

## 1.6 Alcance

El alcance de este trabajo de investigación es crear un sistema de recomendación para Aprovecha.com hasta la fase de evaluación, considerando

que la integración con la aplicación en la fase de despliegue comprende un desarrollo extenso y paralelo que puede comprender otras metodologías y puede ser implementado en el marco de otro TEG.

En este trabajo se implementará un sistema de recomendación en el cual se analizará el negocio, recolectarán datos, se prepararán para un posterior modelado y evaluación en el que quedará listo para su despliegue e integración con Aprovecha.com.

# Capítulo 2: Marco Teórico

En este proyecto se desarrolla un sistema de recomendación para un comercio electrónico, por consiguiente, es necesario definir y conceptualizar ambos términos, así como también resaltar sus características y tipos.

## 2.1 Comercio electrónico

### 2.1.1 Definición

Consiste en la compra y venta de productos o servicios a través de medios electrónicos, tales como internet y otras redes informáticas. Es una combinación de una estrategia, una tecnología, un sistema y un enfoque de ventas que incluye el intercambio electrónico de bienes físicos o intangibles. Como se trata de un intercambio comercial, comprende todas las etapas de una transacción (marketing, pedidos, pago y soporte para la distribución), incluye también servicios post venta y colaboración entre empresas. [34]

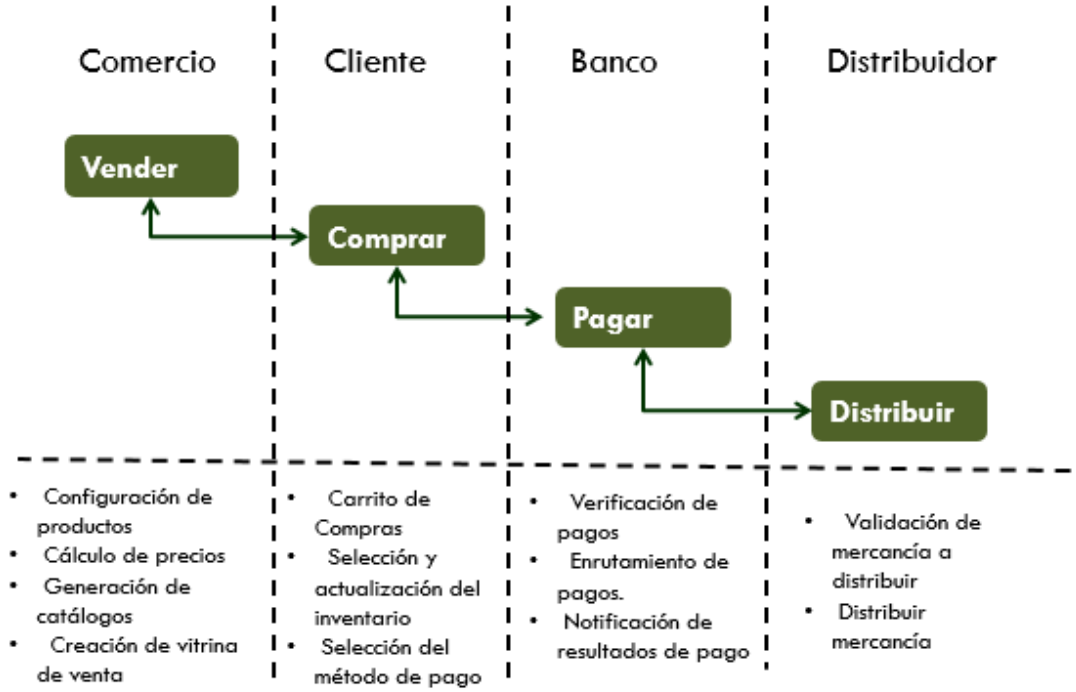


*Figura 1 - Flujo de un comercio electrónico [Clases de Comercio Electrónico - Prof. Aparicio Peña - UCV]*

### 2.1.2 Tipos

El comercio electrónico puede tener diversos tipos, dependiendo de su clasificación.

- **Por tipo de bien:**
  - ➢ Indirecto: productos
  - ➢ Directo: servicios
- **Por tipo de alcance**
  - ➢ Local
  - ➢ Regional

- ➢ Nacional
- ➢ Global

- **Por tipo de actor**
  - ➢ B2C: conocido como *business to consumer* (negocio a consumidor), es aquel que se lleva a cabo entre el negocio o, una tienda virtual, y una persona interesada en comprar un producto o adquirir un servicio. [35]
  - ➢ B2B: es la abreviación de *business to business* (negocio a negocio), es aquel en donde la transacción comercial únicamente es realizada entre empresas que operan en internet o algún medio electrónico, lo que quiere decir que no intervienen consumidores. [35]
  - ➢ B2E: *business to employee* (negocio a empleado), se centra principalmente entre una empresa y sus empleados, es decir, son las ofertas que la propia empresa ofrece a sus empleados, con propuestas atractivas que servirán de impulso para una mejora en el desempeño laboral. [35]
  - ➢ C2C: es el que se conoce como *consumer to consumer* (consumidor a consumidor), es cuando una persona ya no necesita algún producto y busca ofrecerlo a través de un comercio electrónico como medio para realizar esta transacción con otro consumidor, en otras palabras, son dos consumidores finales que intervienen en este tipo. [35]
  - ➢ G2C: es cuando un gobierno municipal, estatal o federal permite que los ciudadanos realicen sus trámites en línea a través de un portal o cualquier medio electrónico, se conoce como *goverment to consumer* (gobierno a consumidor) y se considera comercio porque se paga por un trámite o un servicio ofrecido por la entidad. [35]

## 2.2 Sistema de Recomendación

### 2.2.1 Definición

Los sistemas de recomendación son herramientas y técnicas de software que brindan sugerencias sobre ítems que serán de utilidad para el usuario. Estas sugerencias se relacionan a varios procesos de toma de decisiones, tales como qué ítems comprar, qué música escuchar o qué noticias leer. Un ítem es generalmente el término usado para denotar lo que el sistema recomienda al usuario. [137]

Basado en el historial de compras del usuario, los comercios pueden predecir qué pueden desear los clientes comprar la próxima vez. Para esto, existen los Sistemas de Recomendación. Un sistema de recomendación (*recommender system*) es un tipo de sistema de filtrado de información que busca predecir la evaluación o preferencia que un usuario daría a un elemento o producto. [40][42]

Estos sistemas son muy usados en el comercio electrónico; si un usuario compra o visita un producto, este maximiza el valor tanto para el comprador como para el vendedor en un determinado momento del tiempo. Para hacer las recomendaciones, el sistema analiza y procesa información histórica de los usuarios (edad, compras previas, calificaciones), de los productos o de los contenidos (marcas, modelos, precios, contenidos similares, categorías) y la transforma en conocimiento accionable, es decir, puede predecir qué producto puede ser interesante para el usuario y para la empresa. [40][41][42][43]

### 2.2.2 Funciones de un sistema de recomendación

Es necesario primero, distinguir entre el rol que desempeña el sistema de recomendación en nombre del proveedor del servicio y la del usuario que hace uso de él. Existen varias razones del porqué un proveedor de servicio podría querer utilizar esta tecnología [137]:

- **Incrementar el número de ítems vendidos:** esta es probablemente la función más importante para un sistema de recomendación comercial, es decir, ser capaces de vender un conjunto adicional de artículos comparado a aquellos que usualmente venden sin ningún tipo de recomendación. Este objetivo es alcanzado porque es probable que los ítems recomendados se ajusten a las necesidades y deseos del usuario. Aplicaciones no comerciales tienen objetivos similares, incluso si no existe ningún costo para el usuario que es asociado con dicho ítem. Por ejemplo, una red de contenido apunta a incrementar el número de noticias en su sitio. [137]
- **Vender ítems más diversos:** otra gran función de un sistema de recomendación es permitir al usuario conseguir artículos que podrían ser difíciles de encontrar sin una recomendación precisa. [137]
- **Incrementar la satisfacción de usuario:** un sistema de recomendación bien diseñado puede incluso mejorar la experiencia de usuario en el sitio o la aplicación. El usuario encontrará las recomendaciones interesantes, relevantes y con una interacción humano-computador bien diseñada, podrá disfrutar de usar el sistema. La combinación de recomendaciones efectivas y precisas y una interfaz usable incrementará la evaluación subjetiva del usuario del sistema. Esto a su vez aumentará el uso del sistema y la probabilidad de que se acepten las recomendaciones. [137]
- **Incrementar la fidelidad del usuario:** un usuario debería ser leal a un sitio web que, cuando lo visite, lo reconozca como antiguo cliente y lo trate como un visitante valioso. Esto es una característica común de un sistema de recomendación, ya que la mayoría de ellos computan las recomendaciones, aprovechando la información adquirida en interacciones previas del usuario, ejemplo, sus *ratings* de los ítems. Consecuentemente, mientras más interactúe el usuario con el sitio, más refinado el modelo de usuario será, es decir, la representación del sistema de las preferencias del usuario, y más se puede personalizar

efectivamente la salida del recomendador para que coincida con las preferencias del usuario. [137]

- **Mejor entendimiento de lo que el usuario quiere:** otra función importante de los sistemas de recomendación, que puede ser aprovechar para muchas otras aplicaciones, es la descripción de las preferencias del usuario, ya sea recolectado explícitamente o predichas por el usuario. El proveedor de servicio podría entonces decidir reusar este conocimiento para varios objetivos tales como mejorar la gestión del stock o producción del artículo. [137]

Como se mencionó antes, los sistemas de recomendación tienen un rol del lado del usuario, por lo tanto, el mismo debe balancear las necesidades de ambos "jugadores" (el proveedor y el usuario) y ofrecer un servicio que sea valioso para ambos. Algunas de las funciones de un sistema de recomendación del lado del usuario son: [137]

- **Encontrar algunos ítems buenos:** recomendar al usuario algunos ítems como una lista calificada junto con las predicciones de que tanto el usuario le gustará. Es una de las técnicas de sistema recomendación más comunes. [137]
- **Encontrar todos los ítems buenos:** recomendar todos los ítems que pueden satisfacer las necesidades de algún usuario. En tal caso es insuficiente solo encontrar algunos ítems buenos. Esto es cierto especialmente cuando el número de ítems es relativamente pequeño o cuando el sistema de recomendación es de misión crítica, como en aplicaciones médicas o financieras. En estas situaciones, adicionalmente al beneficio derivado de examinar cuidadosamente todas las posibilidades, el usuario puede también beneficiarse del *ranking* de estos ítems del sistema de recomendación o de explicaciones adicionales que el recomendador puede generar. [137]
- **Recomendar en secuencia:** en vez de enfocarse en la generación de una recomendación simple, la idea es recomendar una secuencia de ítems que satisfaga en conjunto. [137]
- **Recomendar en bulto:** recomendar un grupo de ítems que encajan bien juntos. Por ejemplo, ofrecer un plan de viaje que puede estar compuesto de varias atracciones, destinos y servicios de alojamiento que esta localizados en un área delimitada. Desde el punto de vista del usuario estas alternativas pueden ser consideradas y seleccionadas como un destino de viaje único. [137]
- **Solo navegar:** en esta tarea, el usuario navega el catálogo sin ninguna intención inminente de comprar un ítem. La tarea del recomendador es ayudar al usuario a navegar en los ítems que son más probable que caigan en alcance de los intereses del usuario. [137]
- **Mejorar el perfil de usuario:** esto se relaciona con la capacidad del usuario de proveer información al sistema de recomendación sobre qué le gusta o no. Esta es una tarea fundamental que es estrictamente necesaria para proveer recomendaciones. [137]

- **Expresarse:** a algunos usuarios pueden no importarle las recomendaciones en absoluto. Más bien, lo que es importante para ellos es que se les permita contribuir con sus calificaciones y así expresar sus opiniones y creencias. La satisfacción de esta actividad puede todavía actuar como palanca para sujetar al usuario firmemente a la aplicación. [137]
- **Ayudar a otros:** algunos usuarios se sienten a gusto de contribuir con información, por ejemplo, su evaluación de un ítem, porque cree que la comunidad se beneficia de su contribución. [137]

### 2.2.3 Técnicas para los sistemas de recomendación

A continuación, se listarán las diferentes técnicas para desarrollar un sistema de recomendación, para más información, referir a la documentación.

#### 2.2.3.1 Métodos de minería de datos

Los sistemas de recomendación típicamente aplican técnicas y metodologías de otras áreas vecinas, tales como la Interacción humano-computador y recuperación de información. Sin embargo, la mayoría de estos sistemas tienen en su núcleo un algoritmo que puede entenderse como una instancia particular de una técnica de minería de datos. [137]

El proceso de minería de datos consiste en tres etapas, llevadas en sucesión *Procesamiento de datos, análisis de datos e interpretación de resultados*. [137] Cabe destacar, que más adelante en este proyecto se hablará a fondo de la minería de datos.

- Procesamiento de datos: los datos de la vida real necesitan ser pre-procesados con el fin de ser usados en las técnicas de aprendizaje automático en la fase de análisis. Cuando se trata de un sistema de recomendación, existen tres cuestiones importantes para su desarrollo: definir las medidas de distancia o similitud. Es necesario un muestreo, para así reducir el número de elementos en grandes colecciones de datos, pero conservando sus características principales. Por ultimo reducir la dimensionalidad, para evitar problemas con los datos dispersos, tales como: Análisis de Componentes Principales, descomposición de valores singulares, reducción de ruido. [137]
- Análisis de datos: esto se pueden dividir en dos grupos, la clasificación y el análisis de grupos. La clasificación es la correspondencia entre un espacio de entidad y un espacio de etiqueta, donde las características son pertenecientes al elemento a clasificar y las etiquetas representan las clases. Existes diferentes algoritmos para el método de clasificación entre ellos: *Nearest Neighbors,* árboles de decisión, clasificadores basado en reglas, clasificadores bayesianos, redes neuronales artificiales, máquinas de soporte vectoriales, conjunto de clasificadores. En cuanto al *Cluster analysis,* consiste en asignar grupos para que los elementos en los mismos, sean más similares

entre sí que los elementos en diferentes grupos: el objetivo es descubrir grupos naturales (o significativos) que existen en los datos. K-medias es el algoritmo más utilizado para este método. [137]

#### 2.2.3.2 Recomendaciones basadas en restricciones

Este tipo de recomendaciones son conocida como recomendaciones basada en conocimientos, estos se basan en un proceso de recopilar los requisitos de los usuarios, reparaciones para requerimientos inconsistentes son automáticamente propuestos en situaciones donde no se encuentra soluciones y los resultados de la reconciliación son explicados. Otra característica importante es que hace uso de reglas definidas previamente en cómo relacionar los requerimientos del usuario con las características de los artículos. Típicamente se definen dos conjuntos de variables (Propiedades del clientes y Propiedades del producto) y tres conjuntos diferentes de restricciones (restricciones, filtrado de contenidos y productos), estas variables son utilizadas para generar las recomendaciones. [137]

#### 2.2.3.3 Sistema de recomendación de contexto

La mayoría de los sistemas de recomendación se concentran en recomendar los ítems más relevantes a usuarios individuales y no considerar ninguna información contextual, tal como tiempo, lugar o la compañía de otras personas. Sin embargo, en muchas aplicaciones, tales como recomendar un paquete de vacaciones, contenido personalizado en un sitio web, o una película, pueden no ser suficiente solo considerar usuarios e ítems, es importante también incorporar información contextual dentro del proceso de recomendación con el fin de recomendar ítems a usuario bajo *ciertas circunstancias.* Los sistemas de recomendación basados en contexto utilizan el dominio de la aplicación y los datos disponibles, para obtener alguna información contextual que puede ser utilizado para proveer mejores recomendaciones. Para esto, se utilizan diferentes enfoques, entre ellos son mencionados: Pre-filtrado Contextual, Post-filtrado Contextual y Modelado Contextual. [137]

#### 2.2.3.4 Sistemas de recomendación basado en filtrados

Son las técnicas más comunes para crear un sistema de recomendación [137].
En la sección 2.7 se profundiza en esta técnica.

En la siguiente sección se definirán los términos de Ciencia de datos, elemento fundamental para la realización de este proyecto, ya que la Metodología elegida para el desarrollo está estrechamente relacionada con la ciencia de datos.

## 2.3 Ciencias de datos

### 2.3.1 Definición

Ciencias de datos es la generación de conocimiento a partir de grandes volúmenes de datos, aplicando técnicas de procesamiento paralelo y distribuido para implementar algoritmos que permitan predecir o detectar patrones sobre los datos almacenados. [1] [2]

A partir de los resultados obtenidos se podrán construir herramientas que permitan analizar los resultados y realizar procesos de toma de decisiones. [2]

### 2.3.2 Grandes Volúmenes de Datos (*Big data)*

### 2.3.3 Definición

*Big data* es un campo dedicado al análisis, procesamiento y almacenamiento de grandes colecciones de datos que frecuentemente se originan de fuentes distintas. Normalmente, las soluciones y prácticas *big data* son requeridas cuando el análisis de datos, las tecnologías y técnicas de procesamiento y almacenamiento convencionales son insuficientes. Específicamente, *big data* gestiona distintos requerimientos, tales como la combinación de múltiples conjuntos de datos no relacionados, procesamiento de grandes cantidades de datos no estructurados y la recolección de información oculta a tiempo real. [3]

Adicionalmente al enfoque tradicional de análisis basado en la estadística, *big data* agrega nuevas técnicas que aprovechan recursos computacionales y se enfoca en ejecutar algoritmos analíticos. Mientras que los enfoques estadísticos han sido usados para aproximar mediciones de una población mediante el muestreo, avances las ciencias de la computación han permitido el procesamiento de conjunto de datos enteros, haciendo el muestreo innecesario. [3]

El análisis de grandes volúmenes de datos es un esfuerzo interdisciplinario que incluye matemáticas, estadística, ciencias de la computación y experiencia en el tema. [3]

*Big data* puede ser descrito por 5 elementos que engloban su definición y características. Estas características son conocidas como las 5 V's. Estas son:

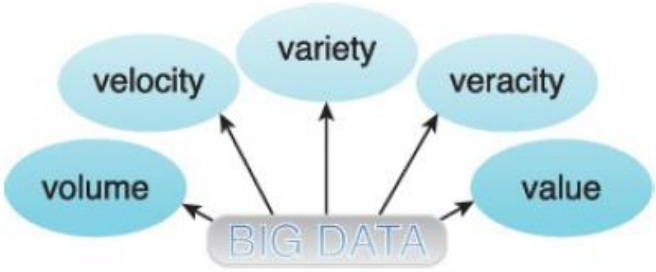


*Figura 2 - Las 5 V's [3]*

- **Volumen:** se refiere a la cantidad de datos que son manipulados por la aplicación, partiendo por unidades de Terabytes llegando hasta Zettabytes. Debido a esto el uso de bases de datos relacionales desmejora la eficiencia del programa, ya que el rendimiento de las mismas es deficiente y no es posible tener particiones de las mismas. Por esto es necesario considerar la preparación de los datos, limpieza y gestión. [4] [5]
- **Velocidad**: se refiere no solo a la alta frecuencia con la que se generan nuevos datos, sino a la necesidad de dar respuesta a la información en tiempo real. Dependiendo de la fuente, la velocidad puede variar de intensidad. Por ejemplo, un escaneo de imágenes por resonancia magnética no es generado tan frecuentemente como un log de entradas en un servidor web de alto tráfico. Como se ve en la figura 2, se puede ver que la velocidad se pone en perspectiva considerando la fuente de los datos. [4] [5]

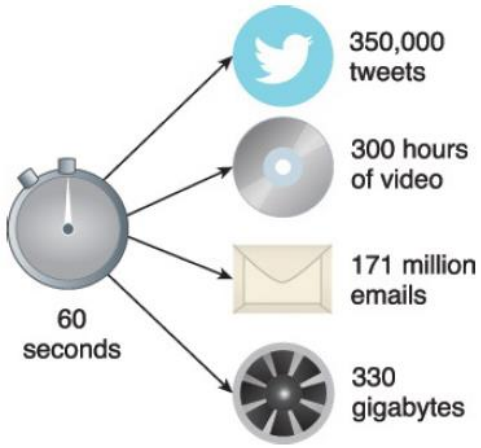


*Figura 3 - Velocidad [3]*

- **Variedad:** se refiere a la naturaleza diversa de la información a manejar. Venimos de información estructurada que encajaba perfectamente en el modelo relacional pero ahora nos encontramos con información semi- y des-estructurada (video, audio, imágenes, redes sociales, etc.) que requiere de nuevos métodos de persistencia y consulta. Esta variedad trae consigo retos para la integración, transformación, procesamiento y almacenamiento de los datos y que *big data* busca solucionar. [4] [5]

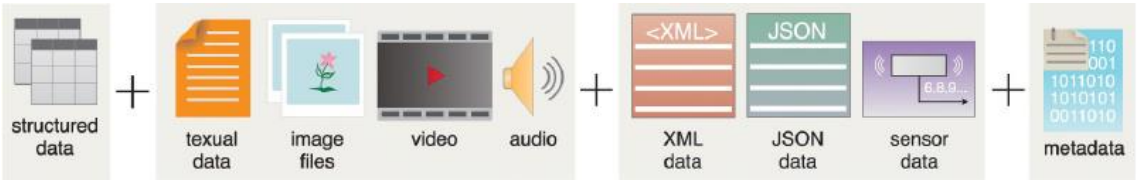


*Figura 4 - Variedad [3]*

- **Veracidad:** se refiere a la calidad o veracidad de los datos. Los datos que ingresan a un ambiente *big data*, necesitan tener

evaluaciones de calidad, que llevan a realizar mecanismos de procesamientos de los datos para remover datos inválidos y valores ruidosos, que vienen de las diferentes fuentes y las distintas estructuras. Se conoce como ruido a aquellos datos que no pueden ser convertidos en información y por lo tanto no tiene valor, mientras que datos limpios y veraces tienen valor y llevan a información significativa. Los datos que han pasado por una limpieza y manipulación tienen son más veraces que datos ruidosos. [5]

- **Valor:** es la habilidad de convertir los datos en información valiosa. Con *big data* se busca encontrar el valor oculto y darle significado más allá del valor representativo del mismo. El valor está estrechamente relacionado con la veracidad de los datos, cuanto mayor sea la fidelidad de los mismos, más valor tiene para el negocio. [5]

### 2.3.4 Campos de aplicación

Actualmente, los grandes volúmenes de datos son generados en diversos campos y ambientes, por lo que su utilización se ha extendido con el paso de los años, a continuación, se hará referencia a algunos de los campos donde se hace uso de *big data* y es comprobable su éxito:

- **Banca comercial:** con grandes cantidades de datos transmitidos en tiempo real desde diversas fuentes, los bancos se han visto forzados a encontrar nuevas e innovadoras formas de gestionar soluciones *big data*. Mientras que es importante entender a los clientes y aumentar su satisfacción, es igualmente importante minimizar los riesgos y los fraudes, manteniendo regulaciones en el cumplimiento normativo. [6]
- **Educación:** educadores armados con una visión basada en datos pueden crear un impacto significante en el sistema escolar, estudiantes y programa educativo. Analizando grandes volúmenes de datos, pueden identificar diversos riesgos en los estudiantes, asegurarse de que los mismos están progresando adecuadamente y pueden implementar un mejor sistema para la evaluación apoyando maestros y directivos. [6]
- **Gobiernos:** cuando las agencias gubernamentales son capaces de aprovechar y aplicar análisis a sus grandes volúmenes de datos, ganan un terreno significativo en cuanto a la gestión de servicios públicos, agencias de gestión, mejorar la congestión del tráfico o la prevención de delincuencia. Pero, mientras hay muchas ventajas que aporta *big data*, los gobiernos también deben abordar lo relacionado con transparencia y privacidad. [6]
- **Salud y medicina:** records de pacientes. Planes de tratamiento. Información de prescripciones. Cuando se trata de salud, todo necesita hacerse de manera rápida, precisa, y en algunos casos,

con suficiente transparencia para satisfacer las regulaciones de la industria. Cuando los grandes volúmenes de datos son gestionados eficientemente, los proveedores de atención médica pueden descubrir información oculta que mejoran la atención al paciente. [6]

- **Industria manufacturera:** equipados con el conocimiento que *big data* puede proveer, las industrias y fabricantes pueden aumentar la calidad y salida de productos minimizando los gastos (procesos que son clave en el mercado competitivo hoy en día). Cada vez, más fabricantes están trabajando con una cultura basada en analítica de datos, que significa que puede resolver problemas más rápido y tomar mejores decisiones. [6]
- ***Retail*:** crear una relación con el cliente es crítico en la industria de ventas al por menor, y la mejor manera para lidiar con eso, es analizando grandes volúmenes de datos. Minoristas (*retailers*) necesitan conocer la mejor manera de comercializar hacia los clientes, manejar las transacciones de forma eficiente y la mejor estrategia para traer de vuelta a los clientes. Con *big data* se encuentra la solución a todos esos problemas. [6]

### 2.3.5 Aplicaciones de *big data* en el comercio electrónico

Como se ha visto, con el análisis y gestión de grandes volúmenes podemos aprovechar mejoras en diferentes campos y negocios, ya que aporta información y conocimientos que previamente no se tenía y que ayudan a la toma de mejores decisiones. En el caso de comercio electrónico no es una excepción, existen numerosas técnicas y usos donde la ciencia de datos y los grandes volúmenes de datos son usados para incrementar ventas, mejorar procesos, satisfacer a clientes, entre otros. A continuación, se verán las técnicas más importantes y con más éxito. [37]

- **Sistema de recomendaciones y personalización:**

Se busca crear recomendaciones basadas en los comportamientos de los usuarios, se hablará de esto en profundidad durante el resto de este trabajo.

- **Email Retargeting:**

Está relacionado con los sistemas de recomendación, se trata del envío automatizado de correos electrónicos personalizados en función de del comportamiento del usuario: correos electrónicos para la recuperación de carritos de compra cuando este se abandona, *newsletters* que contienen productos seleccionados para cada usuario en función del nivel de actividad del usuario en el sitio y también recomendación de productos similares. [44]

- **Dinamización de precios:**

Se basa en analizar los datos de navegación y tiempo de permanencia en la página para determinar nuevos precios en los productos [36]. Puede ser usado con diferentes enfoques, por ejemplo:

- generar urgencia de compra, el precio de un producto aumenta si este es abandonado y luego revisado posteriormente por el usuario. También se puede ofrecer un descuento que solo estará disponible los primeros 30 minutos de permanencia en la página, esta guardará los datos de navegación y si se ingresa luego de transcurrido este tiempo ya no existirá esta oferta. Esto creara en el usuario la necesidad de compra inmediata para tener una mejor ganancia, [36][37]
- combinación de productos haciendo ofertas especiales, haciendo uso de los sistemas de recomendación es posible ofrecer ventas cruzadas (ofrecer productos complementarios al que está intentando adquirir el cliente) y así hacer un descuento en el precio total que debe pagar el cliente, [36][37]
- supervisión del *stock* y gestión de promociones, observar los productos que se tienen y si es posible cambiar los precios o hacer promociones para mejorar las ventas, [36]37]
- mejores ofertas entre los competidores, utilizado *web scrapping* que se trata de analizar con minería de datos o aprendizaje automático contenido en las páginas web de los competidores para mejorar los precios. [36][37]

- **Análisis de sentimientos en las redes sociales:**

Escuchar y analizar lo que los clientes comentan de los productos y la empresa puede beneficiarlos en mejorar sus estrategias. Plataformas como Hadoop facilitan el análisis de grandes cantidades de datos no estructurados. El uso de procesamiento de lenguajes naturales (NLP) es utilizado para extraer información de las redes sociales. El NLP es un campo de la inteligencia artificial y la lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. Esto ligado con algoritmos aprendizaje automático, da sentido a los comentarios de los clientes y puede ser usado para la toma de decisiones. [46]

- **Servicio al cliente:**

Almacenar un historial de los comportamientos del cliente, pueden ayudar a conocerlo al momento de proveerle servicio de atención al cliente, teniendo en cuenta su historial y haciendo un análisis sobre el mismo permitirá recomendarle productos o hacer más fácil la ayuda y solventar sus problemas. Esto puede lograrse también con el análisis de redes sociales, la minería de texto, clasificando correos electrónicos en reclamos o preguntas para agilizar su resolución, entre otros. Todo esto se busca con la intención de generar confianza y cercanía con el cliente y la empresa. [37]

- **Toma estratégica de decisiones basadas en los datos:**

Almacenando, gestionando y analizando grandes volúmenes de datos se puede encontrar información oculta beneficiosa para el negocio. Creando visualizaciones (gráficos, tableros interactivos) de los historiales de ventas, productos, clientes, ingresos, etc., y luego analizándolas,

contribuirá a una mejor toma de decisiones a los directivos de la empresa y hacer cambios para mejorar o mantener el éxito de la misma. [36]

### 2.3.6 Casos de éxito

Algunos casos de éxitos destacables en que comercios electrónicos han aplicado *big data* son:

- **Alibaba:** esta empresa ha demostrado que con el buen uso de los datos se puede conseguir cualquier producto en cualquier parte del mundo. En esta plataforma puede que el vendedor y el comprador nunca tengan un contacto más allá de la compra, aun así, se realizan transacciones de miles de millones de dólares al día. Haciendo uso de recomendaciones y analizando los hábitos de los clientes, se le ofrece lo que busca el mejor precio. [37]
- **eBay:** esta plataforma ofrece una experiencia personalizada a sus clientes, envía promociones basada en datos recolectados, hace sugerencias basadas en la navegación de su página, etc. [37]
- **Groupon:** se basa en la información de clientes y proovedores para hacer transacciones diarias, toma en cuenta los intereses del usuario, las búsquedas, su historial de navegación, las compras realizadas, su localización para así generar recomendaciones al usuario como también envío de correos electrónicos personalizados con ofertas basado en todo en todo lo mencionado anteriormente. [37]
- **CVS:** es una gran cadena de farmacias en Estados Unidos que a través de grandes volúmenes de datos descubrió que un tercio de los consumidores dejan de tomar la medicina prescrita después de un mes, y que el 50 % la deja después del año. Para mejorar esto, la compañía inició un programa automatizado de mensajes, llamadas telefónicas y correos electrónicos para recordad a la gente que debían comprar sus medicamentos nuevamente. También añadieron una alerta especial a los perfiles de los clientes que permiten a los farmacéuticos hablar con los clientes sobre la toma de sus medicinas y así recolectar datos para su posterior análisis. [37]
- **Amazon:** es uno de los casos de éxito más utilizados en *big data* y la ciencia de datos. Utilizan el análisis predictivo de datos en áreas tales como la personalización de cada interacción del cliente, predicción de tendencias de compra, su sistema de recomendación para generar ventas cruzadas es bastante exacto por lo que incrementa las mismas, análisis de ventas para ofrecer el mejor producto y precio, envío de correos personalizados etc. [37]

Los datos, como se observó anteriormente, son fundamentales para la ciencia de datos, por lo tanto, es necesario definir los diferentes formatos, las bases de datos, los diferentes modelos y tipos de bases de datos y sus arquitecturas.

## 2.4 Organización de los datos

Los datos procesados por una solución *big data* puede ser generados por interacción humana o maquinas. Estos datos pueden ser arrojados en diferentes formatos, que dependiendo de sus características son clasificados y almacenados de distinta forma y con base en esto es elegido el modelo de datos. [3]

### 2.4.1 Estructurados

Los datos estructurados conforman un esquema donde son almacenados de forma tabular. Es usado para capturar relaciones entre diferentes entidades y comúnmente son almacenados en una base de datos relacional. Estos son generados frecuentemente por aplicaciones corporativas y sistemas de información como ERP y sistemas CRM. Debido a la abundancia de herramientas y bases de datos que soportan nativamente datos estructurados, raramente requieren de consideraciones especiales para el procesamiento y almacenamiento. [3]

*Figura 5 - Datos estructurados [3]*

### 2.4.2 Semi-estructurados

Estos datos tienen un nivel definido de estructura implícita y consistencia. Los datos semiestructurados son jerárquicos o basados en grafos. Este tipo de datos es comúnmente almacenado en archivos que contienen texto. Tienen la capacidad de autodescribirse mediante metadatos que describen los objetos y la relaciones entre ellos. Algunos de estos son JSON, XML o HTML. [3]


*Figura 6 - Datos semi-estructurados [3]*

### 2.4.3 No-estructurados

Son datos que no conforman ningún modelo de datos conocido ni tienen una forma establecida. Están en el formato en que fueron recolectados. Se estima que representan al menos el 80 % de los datos dentro de cualquier empresa. Estos datos pueden ser tanto textuales como binarios y a menudo se transmiten a través de archivos que son autónomos y no relacionales. Este tipo de datos pueden ser generados por diferentes fuentes, tales como: videos, audios, sensores, archivos PDF, correos electrónicos, entre otros. [3]

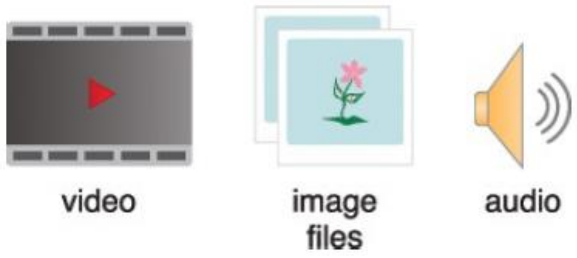


*Figura 7 - Datos No-estructurados [3]*

## 2.5 Base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. [7]

### 2.5.1 Modelo de datos

Un modelo de datos es un tipo de lenguaje, una representación abstracta de datos orientado a hablar de las relaciones que estos datos tienen entre sí, nos permite describir el tipo de datos que hay en la base de datos o todos los elementos reales que intervienen en un problema o situación y la forma en que se relacionan entre sí. El propósito de cualquier modelo de datos es a la vez que representa los datos, que esta representación sea comprensible. [3]

Una base de datos puede clasificarse según su modelo de datos en:

#### 2.5.1.1 Relacional

El modelo relacional constituye una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que, además de proporcionarnos los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de álgebra relacional) para su manipulación, sin ambigüedad posible. [8]

Este modelo permite el cumplimiento de una serie de propiedades llamadas *ACID*, estas son utilizadas por las transacciones, que son un conjunto de órdenes que se ejecutan formando una unidad de trabajo para su correcto funcionamiento. Estas serán definidas a continuación: [8][9]

- **Atomicidad:** cualquier cambio de estado que produce una transacción es atómico, es decir, ocurren todos o no ocurre ningunos, por lo que ninguna operación queda incompleta. [9]
- **Consistencia:** esta establece que solo los valores o datos válidos serán escritos en la base de dato, para asegurar esto se debe garantizar que las transacciones que se ejecutaron en la base de datos fueron válidas para llegar a un estado válido. Si por algún motivo, se ejecuta una transacción que viole esta propiedad, se aplicará un *rollback* a toda transacción dejando la base de datos en un estado consistente. [9]
- **Aislamiento:** esta propiedad asegura que no sean afectadas entre sí las transacciones, en otras palabras, esto asegura que la realización de dos o más transacciones sobre la misma información sea independiente y no generen ningún tipo de error. [9]
- **Durabilidad:** una vez finalizada la ejecución de una transacción, sus resultados son permanentes a pesar de otras consecuencias, por ejemplo, si falla el disco duro el sistema aún será capaz de recordar todas las transacciones que han sido realizadas en el sistema. [9]

#### 2.5.1.2 **No relacionales o NOSQL**

Son modelos que defieren del modelo tradicional relacional, sino que definen su propio esquema

##### 2.5.1.2.1 **Familia de columnas**

Estas bases de datos siguen un modelo de datos basado en columna. Una columna es un concepto análogo a las bases de datos relacionales. Esta es una tripleta compuesta por nombre de columna, valor, marca de tiempo. Una familia de columna es similar a una tabla en una base de datos relacional. Podemos encontrar dos tipos [10]:

- **Familia de columna estándar:** es un objeto que contiene columnas de datos relacionados y una tupla es un par clave/valor donde la calve está asociada a un conjunto de columnas.
- **Super familia de columnas:** este es un objeto que contiene una familia de columnas por lo que en este caso la tupla es un par clave/valor donde la clave está asociada a una familia de columnas.

Un ejemplo de este modelo es el manejador de bases de datos Cassandra.

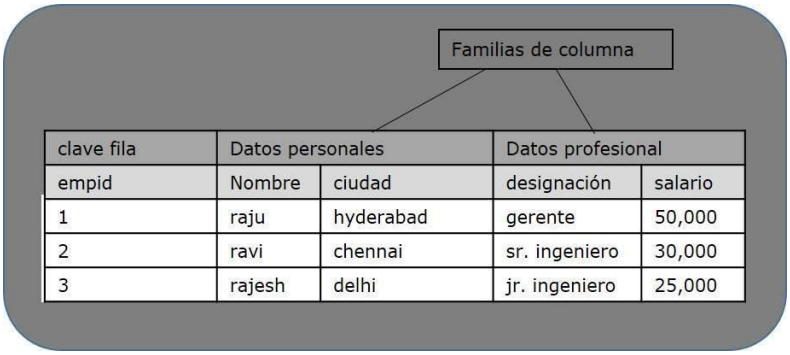


*Figura 8 - Base de datos orientado a columna[https://www.tutorialspoint.com/es/hbase/images/table.jpg]*

### 2.5.1.2.2 **Clave-Valor**

Estas bases de datos utilizan colecciones asociativas que pueden verse como una estructura de tipo diccionario o tabla de hashes. Almacenan los datos en pares de clave y valor, de esa forma puede obtenerse el valor cuando se conoce la clave. Dependiendo del manejador, la clave puede ser única o no. Se caracterizan por ser notablemente rápidos en lectura. [12] [13]

Un ejemplo de este modelo es el sistema manejador Redis, el cual misma es un servicio que se mantiene en memoria RAM con la capacidad de mantener la persistencia de las operaciones, entre otras características. [11]

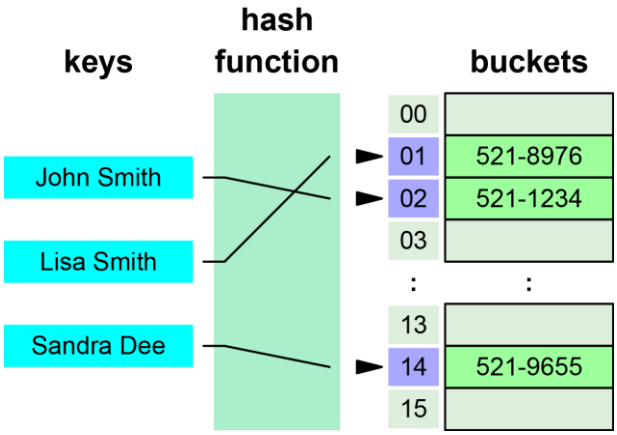


*Figura 9 - Base de datos Clave-valor [https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Hash_table_es.svg/1200px-Hash_table_es.svg.png]*

### 2.5.1.2.3 **Orientado a documentos**

Una base de datos documental o base de datos orientada a documentos está constituida por un conjunto de programas que almacenan, recuperan y gestionan datos de documentos o datos de algún modo estructurados o semiestructurados. A diferencia de las bases de datos relacionales, estos documentos se describen así mismos y por lo tanto son libres de esquema. Las

codificaciones usadas por objetos que la componen están XML, YAML, JSON y BSON. [14]

Estos documentos contienen alguna información similar y otra diferente. En el lado contrario una base de datos relacional todos los registros deben tener los mismos atributos que pueden estar vacíos. [14]

Los documentos se suelen recuperar a través de consultas dinámicas e impredecibles. Así, las bases de datos de documentos por lo general pueden asociar cualquier número de campos de cualquier longitud en un documento. De esta manera se puede almacenar por ejemplo junto con el nombre de una imagen médica de un paciente los datos de nacimiento. En otro momento se puede agregar también el sexo y la profesión incluso si no se concibió originalmente. [14]

Estas bases de datos permiten la escalabilidad horizontal debido a su naturaleza distribuida. Para hacer la replicación de la misma se maneja un esquema de maestro-esclavo. También presentan excelente comportamiento de concurrencia de procesos.

Una de las más utilizadas y estables actualmente es MongoDB.


*Figura 10 - Base de datos orientada a documento [https://sg.com.mx/sites/default/files/images/stories/sg43/sg43-tecnologia-basesdedatos-figura1.jpg]*

### 2.5.1.2.4 **Orientada a grafos**

Esta base de datos utiliza una estructura de grafos para representar y almacenar los datos. Estas representan la información como nodos de un grafo y las relaciones como las aristas del mismo. Esto permite a los datos estar conectados directamente y poder obtenerlos con operaciones. [15]

Están basadas en la teoría de grafos, por lo que emplean nodos, aristas y propiedades y se pueden utilizar los distintos teoremas para recorrerlo.

- Nodos, representan entidades, como personas, empresas o cualquier otro elemento que pueda ser rastreado. Son comparados a un registro en una base de datos relacional. [15]
- Aristas, son las líneas que conectan los grafos entre sí, representan las relaciones entre ellos. Representan el componente principal en las bases de datos orientadas a grafos, son una abstracción que no es implementada en otros sistemas. [15]

- Propiedades, son información que corresponde a la relación entre nodos. [15]

Son normalmente más rápidas comparadas con las bases de datos relacionales cuando se trata de conjunto de datos asociativos. Pueden escalar mejor con grandes conjuntos de datos ya que no necesitan hacer operaciones JOIN costosas. [15]

De las más importantes se tiene el manejador Neo4J, que se caracteriza de ser robusta, escalable y de alto rendimiento. [16]

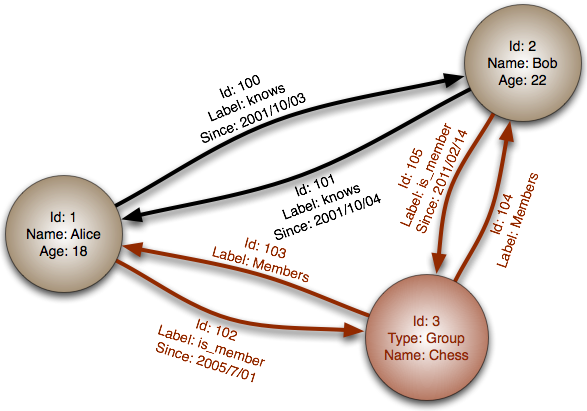


*Figura 11 - Base de datos orientada a grafos [https://en.wikipedia.org/wiki/Graph_database#/media/File:GraphDatabase_PropertyGraph.png]*

En la siguiente sección se exponen a fondo las técnicas para el análisis y detección de patrones para conseguir información relevante dentro de los datos (minería de datos), así como las técnicas de filtrado para un sistema de recomendación.

## 2.6 Minería de datos

### 2.6.1 Definición

Es el proceso de detectar información y patrones en grandes conjuntos de datos. Utiliza el análisis matemático, la inteligencia artificial, el aprendizaje automático y la estadística para deducir los patrones y tendencias que existen en estos datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque hay demasiados datos. [23][24]

#### 2.6.1.1 Tareas de la minería de datos

Las tareas de la minería de datos se dividen en dos dependiendo el modelo que sigan, estas serán definidas a continuación:

- Tareas predictivas: usan variables para predecir valores futuros o desconocidos de otras variables. En esta encontramos la clasificación para valores categóricas y la regresión para variables numéricas. Son generados modelos predictivos y se realizan con aprendizaje supervisado. [25][26]

- Tareas descriptivas: buscan patrones interpretables para describir los datos. Estas incluyen: la agrupación (*clustering*), análisis de asociación y detección de anomalías. En este se generan modelos descriptivos y se utiliza el aprendizaje no supervisado. [25][26]

### 2.6.2 Proceso KDD

Proceso no trivial de identificar, a partir de datos, patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles. No es un proceso automático, es un proceso iterativo que explora exhaustivamente grandes volúmenes de datos para determinar relaciones. [26]

#### 2.6.2.1 **Etapas del proceso KDD**

a) **Selección de datos**, consiste en buscar el objetivo y las herramientas del proceso de minería, identificando los datos y las fuentes de donde serán extraídos. Esto quiere decir, primero se debe tener en cuenta lo que se quiere obtener y cuáles son los datos que nos facilitarán esa información para poder llegar a nuestra meta, antes de comenzar el proceso en tal. [24]
b) **Limpieza de datos o preprocesamiento,** en esta etapa consiste en preprocesar y limpiar los datos obtenidos de las diversas fuentes de datos; estos datos incluyen datos sucios, datos incompletos, el ruido y datos incompletos, que son manejados mediante diversas estrategias que dependerán de las decisiones tomadas por la persona que esté realizando la minería, esto con el fin de obtener una estructura adecuada para su posterior transformación. [24]
c) **Transformación,** consiste en el tratamiento preliminar de los datos, transformación y generación de nuevas variables a partir de las ya existentes con una estructura de datos apropiada para así generar finalmente una vista minable. Aquí se realizan operaciones de agregación o normalización, consolidando los datos de una forma necesaria para la fase siguiente. [24]
d) **Minería de datos,** es la búsqueda de los patrones de interés, datos ocultos o información nueva que pueden expresarse como un modelo o simplemente que expresen dependencia de los datos. Esto se realiza mediante un algoritmo de aprendizaje previamente seleccionado. [24]
e) **Evaluación e interpretación de resultados,** consiste en entender los resultados, analizarlos, conocer sus implicaciones para luego llegar a la toma de decisiones. Las medidas de evaluación dependen del tipo de tarea a realizar, por lo que esto puede llevar a realizar a uno de los pasos anteriores nuevamente. Es importante contrastar el conocimiento adquirido con cualquier conocimiento previo que esté disponible, así como la verificación con los expertos, para asegurar que se tienen resultados confiables. [24]

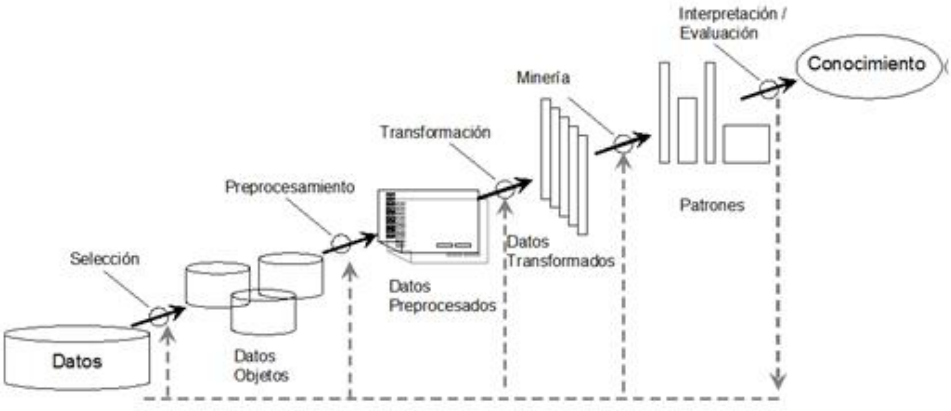


*Figura 12 - Proceso KDD [24]*

### 2.6.3 Campos de aplicación

Al igual que el análisis de grandes volúmenes de datos y la inteligencia artificial, la minería de datos puede ser usada en campos similares para ser aprovechada

- **Educación:** contribuye a la construcción de perfiles de estudiante que facilita la asignación de programas, predicción de rendimiento estudiantil, así como detección de fraudes en trabajos de investigación. [26]
- **Mercado e inteligencia de negocios:** se puede conseguir la segmentación de clientes, hacer evaluaciones de campañas publicitarias, hacer análisis de mercados, predecir ventas, entre otras. [26]
- **Sistemas de recomendación:** dependiendo del negocio, se pueden crear modelos de recomendación, así como la construcción de perfiles de usuarios que contribuye al incremento de ventas. [26]
- **Banca:** puede ser utilizado para el análisis en riesgo de asignación de créditos, detección de fraudes, mejoras en los sistemas para ofrecer a los clientes. [26]
- **Medicina:** diagnóstico de enfermedades a partir de muestras, gestión hospitalaria, recomendación de medicinas para pacientes con enfermedades similares, entre otras. [26]
- **Análisis de texto:** es una abstracción de la minería de datos conocida como minería de texto (*text mining*), que consiste en la clasificación de documentos, construcción automática de resúmenes, análisis de sentimientos u opiniones, identificación de tópicos. [26]
- **Web:** con relación a las aplicaciones web, permite el análisis de comportamiento de usuarios, clasificación de sitios web, patrones de clics hechos por usuarios, etc. [26]

## 2.7 Técnicas de filtrado

Como se mencionó anteriormente, esta es una de las diversas técnicas existentes para desarrollar un sistema de recomendación. Esta fue la elegida para desarrollar el sistema de este proyecto, en la siguiente sección se hablará en profundidad de ellas.

### 2.7.1 Fases del proceso de recomendación

**- Fase de recolección de información**

En esta fase se recolecta información relevante del usuario para generar un perfil de usuario o un modelo para tareas predictivas incluyendo los atributos del usuario, comportamiento o los contenidos al que el mismo accede. Un sistema de recomendación no podrá funcionar con precisión hasta que el modelo/perfil de usuario haya sido bien construido. El sistema necesita conocer lo suficiente del usuario a fin de proporcionar recomendaciones razonables. Los sistemas de recomendación dependen de diferentes tipos de entrada, tales como el muy conveniente *feedback* explícito, que incluye entradas explícitas por el usuario con respecto al elemento de su interés o el *feedback* implícito que infiere las preferencias del usuario indirectamente observando el comportamiento del mismo. Un *feedback* híbrido puede ser también obtenido con la combinación de los antes mencionados. El éxito de cualquier sistema de recomendación depende altamente en su habilidad para representar los intereses actuales del usuario. A continuación, se explicará con más detalle los tipos de *Feedback* [41]:

- **Feedback explícito:** el sistema normalmente solicita al usuario a través de una interfaz del sistema a proveer *ratings* o calificaciones a un ítem con el fin de construir y mejorar su modelo. La precisión de la recomendación depende en la cantidad de calificaciones dadas por el usuario. El único fallo de este método es que requiere un esfuerzo del usuario y también que el mismo no está dispuesto a suministrar información suficiente. A pesar del hecho que el *feedback* explícito requiere más esfuerzo del usuario, todavía se considera que proporciona datos más fiables ya que no involucra extraer preferencias desde acciones y comportamientos, y también provee transparencia y también proporciona transparencia en el proceso de recomendación que da como resultado una calidad de recomendación percibida ligeramente mayor y más confianza en las recomendaciones. [41]
- **Feedback implícito:** el sistema automáticamente infiere las preferencias del usuario supervisando las diferentes acciones del él tales como el historial de compras, historial de navegación y tiempo dedicado en algunas páginas web, enlaces navegados, contenido de correos electrónicos, clics de contenidos, entre otros. Este método reduce la carga en los

usuarios al inferir sus preferencias de sus comportamientos con el sistema. A pesar de que el sistema no requiera de un esfuerzo del usuario, es menos preciso. También, se ha argumentado que este *feedback* implícito puede ser realmente más objetico, ya que no hay sesgos derivados de que los usuarios respondan manera socialmente deseable ni ninguna necesidad de mantener una buena imagen ante otros. [41]

- **Feedback híbrido:** las fortalezas de ambos *feedbacks* (implícito y explícito) pueden ser combinadas en un sistema híbrido con el fin de minimizar sus debilidades y obtener un sistema con mejor rendimiento. Esto se puede lograr usando datos implícitos y una comprobación con calificaciones o permitiendo al usuario dar *feedback* explícito solo cuando escoja expresar interés de forma explícita. [41]

**- Fase de aprendizaje**

Se aplica un algoritmo de aprendizaje para filtrar y aprovechar las características del usuario obtenidas en la fase de recolección. En esta fase se construye un modelo predictivo. [41]

**- Fase de predicción/recomendación**

En esta etapa es cuando se recomienda o predice que tipo de elemento puede preferir el usuario. Cabe destacar que estas fases son iterativas para mejorar constantemente el sistema. [41]

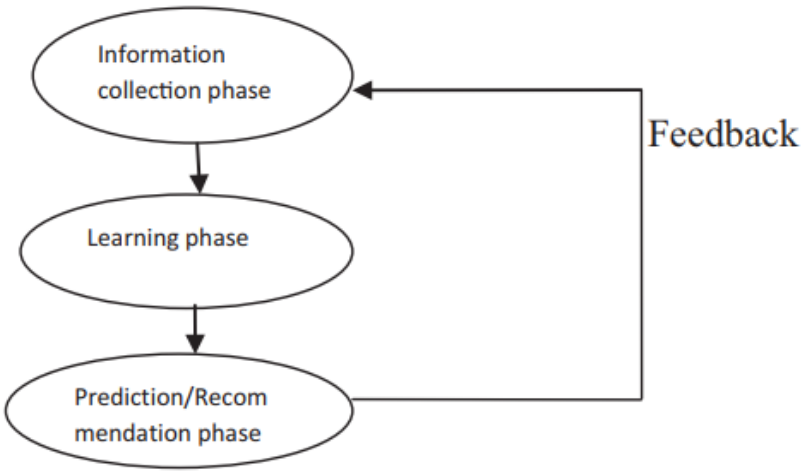


*Figura 13 - Fases de sistema de Recomendación [41]*

### 2.7.2 Técnicas de filtrado de recomendaciones

El uso de técnicas de recomendación eficientes y precisas es muy importante para que el sistema pueda proveer una buena y útil recomendación a sus usuarios. [41]

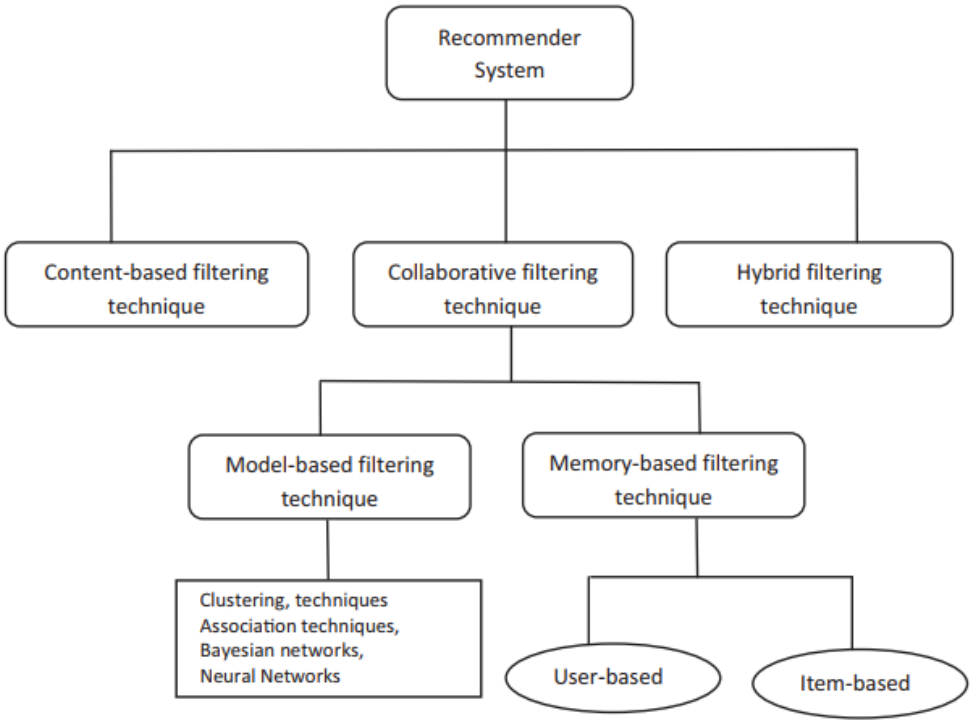


*Figura 14 - Técnicas para un Sistema de Recomendación [41]*

**- Filtrado basado en contenido (*content-based filtering*):** la técnica basada en contenidos es un algoritmo dependiente del dominio y hace énfasis en el análisis de los atributos de los elementos con el fin de generar predicciones. En esta técnica, la recomendación es hecha basada en los perfiles de usuarios utilizando características extraídas del contenido de los elementos que el usuario ha evaluado, visitado o comprado en el pasado. Los elementos que principalmente estén relacionados con los elementos que el usuario haya calificado de manera positiva o comprado frecuentemente, son los que le serán recomendados. En otras palabras, el filtrado basado en contenido construye su recomendación basado en el comportamiento del usuario enfocado en los elementos relacionados. Son usados diferentes tipos de modelos para hallar la similitud entre los elementos con el fin de generar recomendaciones significantes. Podría usarse un Modelo de espacio Vectorial como la Frecuencia inversa de Documentos (TF/IDF) o modelos probabilísticos como el Clasificador Bayesiano Ingenuo (Naive Bayes), Árboles de Decisión o Redes Neuronales. Estas técnicas hacen las recomendaciones aprendiendo el modelo subyacente usando análisis estadístico y técnicas de aprendizaje automático. El filtrado basado en contenido no necesita explorar ni conocer los perfiles de otros usuarios ya que estos no influencian en la recomendación. También si el comportamiento del usuario cambia, esta técnica tiene el potencial de ajustar la recomendación en un corto período de tiempo. La mayor desventaja que encontramos en este tipo de filtrado es la necesidad de tener un conocimiento profundo y la descripción de las características de los elementos en el perfil. [41]

*** Pros y contras del filtrado basado en contenido:** el filtrado basado en contenido tiene la habilidad de recomendar nuevos ítems aún si no hay *ratings* proporcionados por los usuarios. Entonces si la base de datos no contiene las preferencias de los usuarios, la precisión de las recomendaciones no se ve afectada. También, si las preferencias del usuario cambian, tiene la capacidad de ajustar sus recomendaciones en corto período de tiempo. Puede manejar situaciones donde diferentes usuarios no comparten el mismo ítem, solo ítems idénticos según sus características intrínsecas. Los usuarios pueden recibir recomendaciones sin compartir su perfil (perfil de usuario), lo que asegura privacidad. Sin embargo, esta técnica sufre varios problemas. El filtrado basado en contenido es dependiente de los metadatos de los ítems, lo que requiere una rica descripción de los ítems y un perfil de usuario bien organizado antes de que la recomendación pueda ser hecha. Esto es llamado *análisis de contenido limitado*. Por esto, la efectividad de esta técnica depende en la disponibilidad de los datos descriptivos. Contenido sobre especialización es otro serio problema del filtrado basado en contenido, los usuarios están restringidos a obtener recomendaciones similares a los ítems definidos en sus perfiles. [41]

**- Filtrado colaborativo (*Collaborative filtering*):** es una técnica de predicción independiente del dominio para contenidos que no pueden ser fácilmente ni adecuadamente descritos por metadatos como películas y música. El filtrado colaborativo trabaja en construir una base de datos (matriz usuario-ítem) con preferencias de elementos por usuarios. Empareja usuarios con preferencias e intereses relevantes calculando similitudes entre sus perfiles para crear recomendaciones. Estos usuarios construyen un grupo llamado vecindad. Un usuario obtiene una recomendación a un ítem que aún no ha calificado, comprado o visitado pero que usuarios miembros de su vecindad sí lo han hecho. Las recomendaciones que son producidas por un CF (*collaborative filtering*) pueden ser: una Predicción o una Recomendación. Una Predicción es un valor numérico, $R_{ij}$, que expresa la puntuación predicha del ítem *j* para el usuario *i*, mientras que una Recomendación es una lista de los mejores N elementos que al usuario le podrían gustar más, esto se muestra en la figura 16. Este tipo de filtrado puede ser dividido en dos categorías: basada en memoria (*memory-based)* y basado en modelo (*model-based)*. [41]

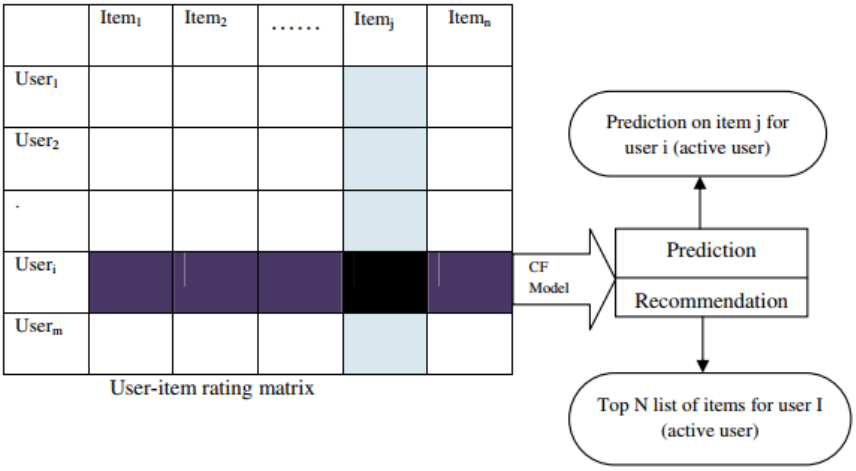


*Figura 15 - Filtrado Colaborativo [41]*

- **Técnica basada en memoria:** Los ítems que ya han sido calificados por el usuario anteriormente, desempeñan un papel importante en la búsqueda de un vecino que comparta características con él. Una vez que se encuentra el vecino de un usuario, diferentes algoritmos pueden ser usados para combinar las preferencias de los vecinos para generar recomendaciones. El filtrado colaborativo basado en memoria puede lograrse a través de dos técnicas: basada en usuario (*user-based*) y basada en ítem (*item-based*). El filtrado colaborativo basado en usuario calcula la similitud entre los usuarios al comparar sus calificaciones en el mismo ítem, y luego predice la calificación de un ítem para el usuario activo como un promedio ponderado de las calificaciones del elemento por usuarios similares al usuario activo donde las ponderaciones son las similitudes de estos usuarios con el elemento objetivo. La técnica basada en ítem (*ítem-based*) calcula las predicciones usando la similitud entre los elementos y no entre los usuarios. Construye un modelo de similitudes entre elementos al recuperar todos los elementos calificados por un usuario activo de la matriz *user-item*, determina que tan similar son los ítems recuperados con el ítem objetivo, luego selecciona los *k* ítems más similares y sus correspondientes similitudes también se determinan. La predicción es hecha al tomar un promedio ponderado de las calificaciones de los usuarios activos en los *k* ítems similares. Existen diversas medidas para calcular la similitud entre ítem/usuario. Las dos más populares son la basada en correlación y la del coseno. [41]
- **Técnica basada en modelo:** esta técnica utiliza calificaciones previas para aprender un modelo con el fin de mejorar el rendimiento del filtrado colaborativo. El modelo puede ser construido utilizando técnicas de aprendizaje automático o minería de datos. Estas técnicas pueden recomendar rápidamente un conjunto de ítems por el hecho de que usan un modelo pre-computado y han demostrado que producen recomendaciones que

son similares a las técnicas de recomendación basados en vecindad. Algunas de estas técnicas son de Reducción de Dimensionalidad tales como Descomposición en Valores Singulares (SVD por sus siglas en inglés), Completación de Matriz, métodos de Semántica Latentes, regresión y agrupamiento (*clustreging)*. Las técnicas basadas en modelo analizan la matriz *user-item* para identificar relaciones entre los ítems, usan estas relaciones para comparar la lista de las principales *N* recomendaciones. Algunos algoritmos usados en esta técnica son: reglas de asociación, agrupamiento (*clustering)*, árboles de decisión, redes neuronales, regresión, clasificadores bayesianos, entre otros. [41]

*** Pros y contras de las técnicas de filtrado colaborativo:** este filtrado tiene algunas ventajas importantes en comparación al filtrado basado en contenido, por ejemplo, que puede ejecutarse en dominios donde no hay mucho contenido asociado con los ítems y donde este contenido es difícil para un sistema computacional analizar. Esta técnica tiene la habilidad de proporcionar recomendaciones fortuitas, lo que significa que puede recomendar ítems que son relevantes para el usuario aun sin estar contenido en el perfil del usuario. A pesar del éxito de las técnicas de filtrado colaborativo, su extendido uso ha revelado algunos problemas potenciales [41]:

1) **Problema *cold-start*:** esto se refiere a la situación donde el recomendador no tiene la información adecuada sobre el usuario o un ítem en orden de hacer predicciones relevantes. Este es uno de los mayores problemas que reducen el desempeño del sistema de recomendación. El perfil de un usuario nuevo o un ítem estará vacío ya que no se ha empezado a proveer *ratings*. Esto también ocurre cuando el sistema es nuevo. [41]

2) **Problema de datos esparcidos:** este problema ocurre como resultado de falta de suficiente información, que es, cuando solo un poco del total de ítems disponibles en la base de datos fueron calificados por los usuarios. Esto siempre lleva a una matriz usuario-ítem muy esparcida, la inhabilidad de localizar exitosamente a los vecinos y finalmente, la generación de recomendaciones débiles. Sumado a esto, los datos dispersos siempre llevan a problemas de cobertura, que es el porcentaje de ítems en el sistema que el recomendador puede considerar. [41]

3) **Escalabilidad:** este es otro problema asociado con los algoritmos de recomendación porque el cómputo normalmente crece linealmente con el número de usuarios e ítems. Una técnica de recomendación que es eficiente cuando el número del conjunto de datos es limitado, puede ser incapaz de generar recomendaciones satisfactorias cuando el volumen de datos crece. Por lo que es importante considerar la escalabilidad de un sistema de recomendación. [41]

**- Filtrado híbrido:** combina diferentes técnicas de recomendación con el fin de obtener un sistema más optimizado y así evitar limitaciones y problemas que puedan generar los sistemas de recomendación puro. La idea detrás las técnicas híbridas es que las combinaciones de algoritmos proveerán recomendaciones más precisas y efectivas que un solo algoritmo, ya que las desventajas de un algoritmo son superadas por otro. La combinación de enfoques puede hacerse en cualquiera de las siguientes formas: implementación separada de algoritmos y combinación de resultados, utilizando algo de filtrado basado en contenido en un enfoque colaborativo, utilizando algo de filtrado colaborativo en un enfoque basado en contenido, creando un sistema de recomendación unificado que junta ambos enfoques [41]. Los tipos son:

- **Ponderado:** combina los resultados de diferentes sistemas de recomendación para generar una lista de recomendación o predicción mediante la integración de los resultados de cada una de las técnicas en uso por una fórmula lineal. [41]
- ***Switching hybridization:*** El sistema cambia a una de las técnicas de recomendación según una heurística que refleja la capacidad de recomendación para producir una buena calificación. [41]
- **Cascada:** aplica un proceso de refinamiento iterativo para construir un orden de preferencia entre diferentes elementos. Las recomendaciones generadas por una técnica, son refinadas por otra y así sucesivamente. [41]
- **Hibridación mixta:** combina recomendaciones resultantes de diferentes técnicas al mismo tiempo en vez de tener solo una recomendación por ítem. Cada elemento tiene múltiples recomendaciones asociadas con diferentes técnicas de recomendación. [41]
- ***Feature-combination:*** las características derivadas de diferentes fuentes de conocimientos son combinadas y dadas a un solo algoritmo de recomendación. Por ejemplo, la calificación de usuarios similares que es una característica del filtrado colaborativo es usada en otra técnica de recomendación como una de las características para determinar la similitud entre ítems. [41]
- ***Feature-augmentation:*** hace uso de una técnica de recomendación para determinar características como

calificaciones u otras para luego ser juntada con información adicional para otra técnica. [41]

- ***Meta-level:*** el modelo generado por una técnica de recomendación es usada como entrada para otro. [41]

### 2.7.3 Métricas de evaluación

La calidad de un algoritmo de recomendación puede ser evaluado usando diferentes tipos de medidas las cuales pueden ser precisión y cobertura. El tipo de métricas usado depende de la técnica de filtrado. La precisión es la fracción de las correctas recomendaciones sobre el total de posibles recomendaciones, mientras que la cobertura mide la fracción de objetos en el espacio de búsqueda del sistema que es capaz de proporcionar recomendaciones. Las métricas para la medición de la precisión del sistema de recomendación están divididas en estadísticos y métricas de precisión de soporte de decisión. La adaptación de cada métrica depende en las características del conjunto de datos y los tipos de tareas que el sistema de recomendación hará.

- **Medidas de precisión estadística:** evalúan la precisión de una técnica de filtrado comparando los *ratings* predichos directamente con el verdadero *rating* dado del usuario. Las métricas utilizadas son la *Mean Absolute Error Mean Absolute* y la *Root Mean Squere Error*.

- **Métricas de precisión de soporte de decisión:** Estas métricas ayudan a los usuarios a seleccionar elementos que son de muy alta calidad del conjunto de elementos disponibles. Las métricas ven el procedimiento de predicción como una operación binaria que distingue los artículos buenos de aquellos que no son buenos. Las más utilizadas son: *Precision, Recall* y *F-measure*.

- **Cobertura:** tiene que ver con el porcentaje de ítems y usuarios que un sistema de recomendación puede proveer predicciones. Algunas predicciones pueden ser imposibles de hacer si ningún usuario o muy pocos usuarios han calificado un ítem. La cobertura puede ser reducida definiendo pequeñas medidas de vecindad entre usuarios o ítems.

## 2.8 Lenguaje de Programación

### 2.8.1 Definición

Es un lenguaje formal que especifica un conjunto de instrucciones finitas que pueden ser utilizadas para producir varios tipos de salida. Generalmente consisten en instrucciones para ser interpretadas por una computadora. Los lenguajes de programación pueden ser usados también para crear programas que implementen algoritmos. [27]

### 2.8.2 Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Es multiparadigma, ya que soporta programación orientaba a objetos, programación imperativa, y en menor medida, programación funcional. Es un lenguaje funcional de tipado

dinámico y es multiplataforma. Posee una licencia de código abierto, denominada Python Software Fundation License. [32][33]

Entre sus principales características y paradigmas encontramos:

- Usa conteo de referencias para la administración de memoria. [32][33]
- Resolución dinámica de nombres, es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa. [32][33]
- La facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. [32][33]
- Utiliza la identación para separar y agrupar, es decir no usa llaves ni punto y coma. [32[33]

# Capítulo 3: Marco Metodológico

## 3.1 Foundational Methodology

Al igual que los científicos tradicionales, los científicos de datos requieren de una metodología fundamental, que pueda servir como una guía estratégica para la resolución de problemas.

Esta metodología tiene algunas similitudes con otras metodologías reconocidas para la minería de datos, pero hace énfasis en las nuevas prácticas en la ciencia de datos tales como el uso de grandes volúmenes de datos, la incorporación de analítica de texto en un modelo predictivo y la automatización de algún proceso. Con esto en mente, IBM ha propuesto la "*Foundational Methodology for Data Science"*, desarrollada por el científico de datos John Rollins.

La metodología consiste de 10 pasos que forman un proceso iterativo para utilizar datos y descubrir información oculta. Cada etapa juega un papel vital en el contexto de la metodología completa. Van desde la concepción de la solución hasta la implementación de esta, incluyendo el un *Feedback* y refinamiento. [114]

**Etapas**

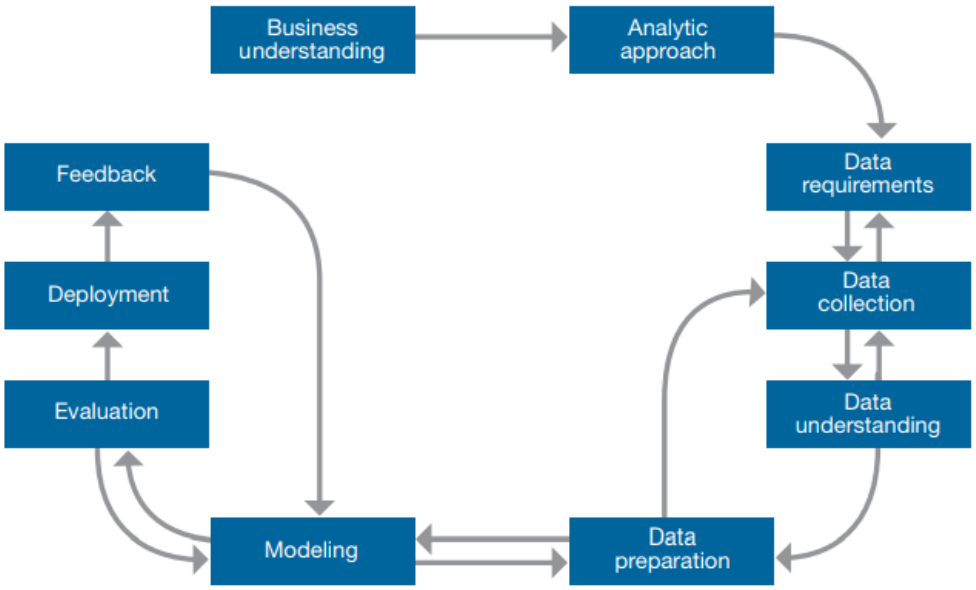


*Figura 16 - Diagrama de Foundational Methodology for Data Science [114]*

1) **Entendimiento del negocio (*Business understanding*):** todo proyecto empieza con el entendimiento del negocio. Los representantes del negocio que necesitan la solución analítica desempeñan el papel más crítico en esta etapa al definir el problema, objetivos del proyecto y los requisitos de la solución desde una perspectiva comercial. Para garantizar el éxito del proyecto, los representantes del negocio pueden estar involucrados en todo el proyecto para proporcionar experticia en el

dominio, revisar los hallazgos intermedios y asegurar que el trabajo se mantenga en buen camino para generar la solución deseada. [114]

2) **Enfoque analítico (*Analitic approach*):** una vez el problema del negocio ha sido claramente definido, el científico de datos puede definir el enfoque analítico para resolver el problema. Esta etapa involucra expresar el problema el contexto de técnicas de aprendizaje automático y estadístico, y así el científico de datos pueda identificar cuáles son las técnicas que mejor se adapta para conseguir los resultados deseados. [114]
3) **Requerimiento de datos (*Data requirements*):** El enfoque analítico elegido, determina los requerimientos de datos. Específicamente, los métodos analíticos a usar requieren ciertos contenidos de datos, formatos y representaciones, guiados por el conocimiento de dominio. [114]
4) **Recolección de datos (*Data collection*):** en la etapa inicial de recolección de datos, el científico de datos identifica y reúne las fuentes de datos disponibles (estructurados, semiestructurados y no estructurados) relevantes para el dominio del problema. Normalmente, deben elegir si realizar inversiones adiciones para obtener elementos de datos menos accesibles. Puede ser mejor postergar la inversión hasta que se conozca más sobre los datos y el modelo. En caso de encontrar brechas en la recolección de datos, podría ser necesario revisar los requerimientos de datos y recolectar más. [114]
5) **Entendimiento de los datos (*Data understanding*):** después de la recolección de datos, generalmente el científico de datos utiliza la estadística descriptiva y técnicas de visualización para entender el contenido de los datos, evaluar la calidad de los datos y descubrir patrones iniciales acerca de los datos. [114]
6) **Preparación de los datos (*Data preparation*):** esta etapa engloba todas las actividades para la construcción del conjunto de datos que será usado en la etapa de modelado. Las actividades de preparación de datos incluyen: limpieza de datos (tratar con valores faltantes o inválidos, eliminación de duplicados, formateo correcto), combinar datos de múltiples fuentes (archivos, tablas, plataformas) y trasformar datos en variables más usables. La preparación de datos es generalmente la etapa que más consume tiempo en un proyecto de ciencia de datos. [114]
7) **Modelado (*Modeling*):** iniciando con la primera versión del conjunto de datos preparado, la etapa de modelado se enfoca en desarrollar un modelo predictivo o descriptivo acorde al enfoque analítico definido previamente. Con modelos predictivos, el científico de datos usa un conjunto de entrenamiento (datos históricos en los cuales el resultado es conocido) para construir el modelo. El proceso de modelado es altamente iterativo. Para una técnica dada, los científicos de datos pueden utilizar múltiples algoritmos con sus respectivos parámetros para encontrar el mejor modelo que se adapte a las variables disponibles. [114]
8) **Evaluación (*Evaluation*):** durante el desarrollo del modelo y antes del despliegue, el científico de datos evalúa la calidad del modelo y asegurarse si dirige de forma completa y apropiada el problema del negocio. Esta etapa comprende varias medidas de diagnóstico

computado y otras salidas como tablas y gráficos, permitiendo al científico de datos interpretar la calidad del modelo y su eficacia para resolver problemas. Para un modelo predictivo, el conjunto de prueba es usado, el mismo es independiente del conjunto de entrenamiento, pero sigue la misma distribución de probabilidad y tiene una salida conocida. El conjunto de prueba es usado para evaluar el modelo y ser refinado de ser necesario. [114]

9) **Despliegue (*Deployment*):** luego del desarrollo de un modelo satisfactorio y su aprobación por los representantes del negocio, el mismo es desplegado dentro del ambiente de producción o un ambiente de prueba equiparable. Usualmente es desplegado de una forma limitada hasta que su rendimiento sea totalmente evaluado. [114]
10) **Retroalimentación (*Feedback*):** mediante la recolección de resultados del modelo implementado, la organización recibe una retroalimentación sobre el rendimiento del modelo y su impacto en el ambiente en el cual fue desplegado. Analizando esta retroalimentación, los científicos de datos puede refinar el modelo, incrementar su precisión y así su usabilidad. [114]

# Capítulo 4: Marco aplicativo

Para la aplicación y puesta en práctica de este trabajo, se tuvo como base la *Foundational Methodology for Data Science* de IBM con el fin de desarrollar el sistema de recomendación para Aprovecha.com siguiendo las fases de la misma. A continuación, se describe los pasos tomados en cada una de ellas:

## 4.1 Entendimiento del negocio

Aprovecha.com es un sitio de comercio electrónico que ofrece cupones de descuento en distintas categorías tales como gastronomía, belleza, viajes, etc. Estos cupones los ofrece mediante establecimientos que se asocian al sitio y deciden publicar sus ofertas. El usuario obtiene estos cupones de su preferencia desde la página y luego se dirige a los establecimientos que lo ofrecen y así disfrutan de los mismos.

El principal objetivo de esta empresa es entonces, tener más ventas y así más ganancias.

Para esta primera fase, se tuvo como objetivo recopilar y entender las necesidades del proyecto desde un enfoque comercial, por lo que se realizó una reunión con los representantes del negocio donde estos dieron a conocer el problema y sus requisitos. De esta reunión se obtuvieron los requerimientos, de los que se partió para desarrollar la solución de este proyecto.

Los representantes del negocio plantearon sus problema y objetivos, los cuáles, en resumen, fueron: aumentar el valor del sitio, incrementar las ventas y ampliar la intención de compra, esto, incorporando al sistema actual alguna funcionalidad que les ayudara a alcanzar dichos objetivos. Con base en el problema planteado en esta investigación, se determinó que un sistema de recomendación podía satisfacer estas necesidades.

Luego de analizar los objetivos del cliente, se propusieron crear un sistema de recomendación para satisfacer los requerimientos del cliente.

## 4.2 Enfoque analítico

Una vez entendido el negocio, se procedió a expresar el problema en el contexto de sistemas de recomendación para así determinar la mejor técnica para satisfacer los requerimientos.

Como se ha mencionado antes, se quiere desarrollar un sistema de recomendación para aprovecha.com, el objetivo a recomendar son los cupones de descuento ofrecidos en la página. Un cupón, conocido en el negocio como *deal*, tiene un título, un precio, y una categoría asociada, como atributos principales, entre otros. Los deals son agrupados por categorías en la página, siendo esta su característica principal para ser identificado por los usuarios.

Aprovecha.com no ofrece un sistema de calificaciones por lo que no se cuentan con *ratings* explícitos, sin embargo, el sistema posee un historial de ventas en

su base de datos (811366 registros), esta tabla de ventas guarda relación con los *deals,* de manera que es posible determinar cuáles *deals* han sido comprados por un usuario; dichas compras pueden ser inferidas como *ratings* implícitos gracias al comportamiento del usuario con el sistema y así una posible preferencia por parte del mismo.

Teniendo este largo historial de compras por usuario (considerándolos como *ratings* implícitos) será posible agrupar usuarios con intereses y preferencias similares y a partir de la similitud entre estos, generar las recomendaciones. Con esto en cuenta se estaría trabajando con un enfoque de filtrado colaborativo.

Además de esto, si se utilizarán las categorías de los *deals* para agrupar a los mismos, las recomendaciones se podrían hacer a partir de dicha categoría, es decir, determinar las recomendaciones a partir de los usuarios con preferencias similares en ítems de la misma categoría. Planteado de esta manera, se tiene un filtrado colaborativo, combinando el basado en usuarios y basado en ítems.

Para alcanzar los objetivos planteados, se hará uso de las siguientes tecnologías:

- Python como lenguaje de programación,
- MongoDB, como base de datos para la recolección

Luego de analizar y determinar el enfoque, se ideó y propuso la siguiente arquitectura de solución:

### 4.2.1 Arquitectura de solución propuesta

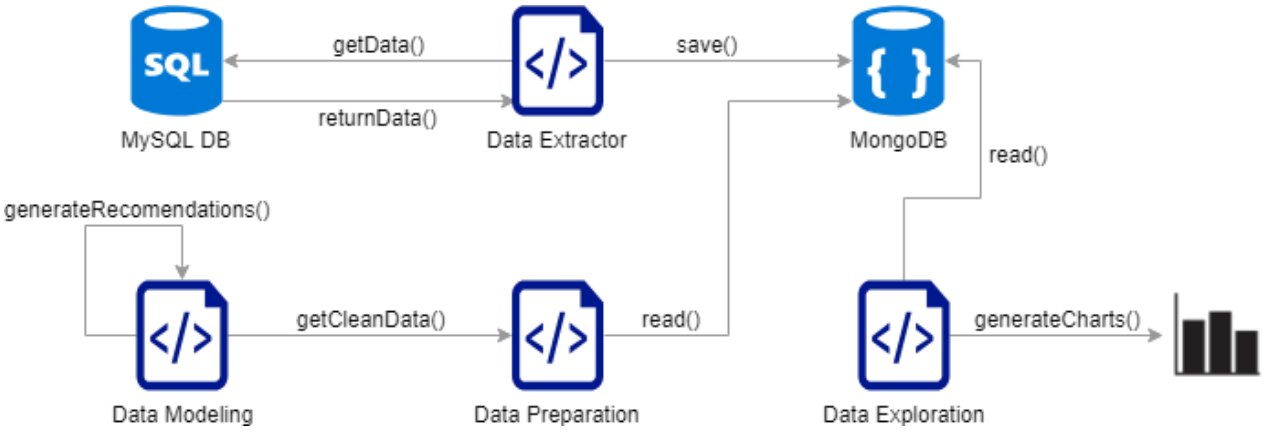


*Figura 17 - Arquitectura de solución*

Como se puede observar, se propone la utilización de dos bases de datos y cuatro scripts encargados de hacer los procedimientos. A continuación, se describen los elementos del diagrama presentado:

- **MySQL DB:** base de datos relacional donde se encuentran los datos de muestra para el desarrollo del sistema recomendador.

- **MongoDB:** base de datos no relacional documental donde estarán almacenados los datos recolectados para este estudio.
- **Data Extractor:** este script es el encargado de recolectar los datos de la base de datos MySQL, procesarlos y formatearlos para luego guardarlos en la base de datos MongoDB.
- **Data Exploration:** este script leerá los datos de MongoDB y utilizar diferentes métricas para explorar las características de los datos, así como también generar gráficos con información relevante de los mismos.
- **Data Preparation:** al igual que data exploration, obtiene los datos de Mongo y aplica toda la limpieza, normalización y reglas pertinentes para un correcto manejo de los datos.
- **Data Modeling:** obtiene los datos del script anterior (ya preparados) y utiliza las técnicas de filtrado para generar las recomendaciones.

### 4.2.2 Tecnologías utilizadas

Las tecnologías seleccionadas fuero las siguientes:

- **Python 3.6 de 64bits:** Este lenguaje de programación fue el utilizado para el desarrollo del sistema de recomendación. Python es utilizado con frecuencia en proyectos de ciencias de datos y en los que se hace uso de grandes conjuntos de datos debido a las siguientes razones:

    - **Lenguaje multipropósito:** Python puede ser usado para cualquier tipo de programación, proyectos web, scripts, ciencia de datos, etc.
    - **Fácil de instalar:** python es muy fácil de instalar y agregar librerías para usarlo, comparándolo con otras plataformas.
    - **Legibilidad:** la estructura y sintaxis de un programa python, es bastante legible, lo que permite que sea sencillo de procesar y mantener.
    - **Gran cantidad de bibliotecas para ciencia de datos, *big data*, *machine learning*:** existe un gran número de librerías optimizadas utilizadas para estos propósitos que facilitan el desarrollo.

    Esta última, es la principal razón de la elección de Python como lenguaje de programación.

- **Bibliotecas**: las bibliotecas utilizadas fueron las siguientes:

    - **Padas:** es una librería que ofrece fácil manipulación y análisis de grandes estructuras de datos, es muy usado para proyectos de ciencias de datos. [119]
    - **Numpy:** agrega soporte a vectores y matrices, constituyendo un conjunto de funciones matemáticas para operar entre ellos. [120]
    - **Scikit-Learn:** incluye herramientas para minería de datos y aprendizaje automático. Está construida sobre numpy, scipy y matplotlib. [121]
    - **Scipy:** comprende herramientas y algoritmos matemáticos. [122]

    - **Pymongo:** interfaz para manipular y trabajar con MongoDB en python. Es la recomendada por la documentación de MongoDB [123].
    - **matplotlib:** es una biblioteca de gráficos 2-D. [124]
    - **seaborn:** biblioteca de visualización de datos basada en matplotlib, incorporando mejoras atractivas en los gráficos. [125]
    - **pymysql:** interfaz entre python y MySQL para obtener los datos de dicha base de datos. [126]

- **MongoDB 3.6.3:** base de datos utilizada para la recolección de datos. La misma fue elegida debido a su buen *performance* con grandes volúmenes de datos. [129]
- **Anaconda:** es una distribución de Python y R, que proporciona una capa de bibliotecas y un manejador de paquetes enfocado en el desarrollo de aplicaciones de ciencias de datos y *big data* [127]. También proporciona el IDE utilizado en este trabajo especial:
    - **Jupyter Notebook**: es un ambiente interactivo en el que se permite combinar código, texto enriquecido, funciones matemáticas y gráficos [128].

## 4.3 Requerimiento de los datos

Aprovecha.com tiene almacenados sus datos en una base de datos relacional utilizando MySQL como sistema manejador. La misma cuenta con más de 10 millones registros y 145 tablas. Entre las tablas más importantes se encuentran la de usuarios, *deals*, compras, categorías, pagos, entre otras.

Como se puede apreciar, los datos que se utilizarán para desarrollar el sistema de recomendación son estructurados, ya que están almacenados de forma tabular y poseen una estructura definida.

En la fase anterior se determinó que el enfoque para desarrollar el sistema será un filtrado colaborativo, utilizando el historial de compras de los usuarios, los ítems comprados y las categorías. Para resolver estos requerimientos es necesario primero, obtener datos del usuario, más importante un identificador, ya que no es necesario más características del mismo, por lo tanto, se puede usar el *id* de la tabla usuarios como dicho identificador.

Luego es necesario obtener qué ítems ha adquirido cada usuario, el monto pagado puede ser relevante como dato informativo. Para cada ítem es fundamental un identificador único también, por lo que servirá de igual forma el *id* de la tabla *deals* y la fecha de inicio como dato referencial. Por último, se necesita la categoría de cada ítem, siendo el nombre la característica más importante para identificarlo.

Siendo estos los datos requeridos, se procederá a recolectarlos.

## 4.4 Recolección de datos

Como se mencionó anteriormente, se tiene como fuente la base de datos de Aprovecha.com, la misma es relacional y utiliza el sistema manejador MySQL. Con base en la fase anterior, es necesario utilizar las siguientes tablas:

- *purchases*: contiene el historial de compras por usuario.
- *deals*: los ítems con toda su información.
- *purchase_deals*: tabla relación entre *purchases* y *deal*.
- *categories*: almacena las categorías.

Las relaciones entre dichas tablas, se expresa en el siguiente gráfico.

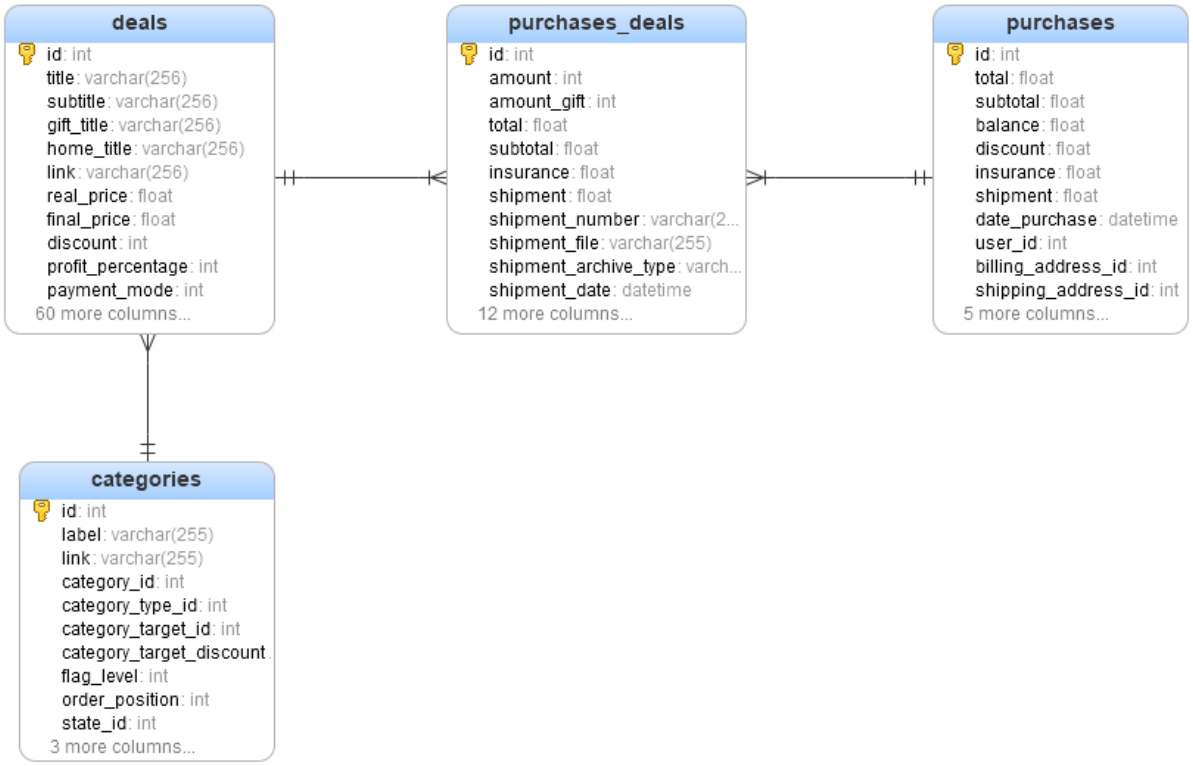


*Figura 18 - Modelo Relacional de Aprovecha.com*

Ya teniendo esto claro, los datos necesarios para cubrir los requerimientos serán los siguientes:

- *user_id*: identificador único del usuario.
- *purchase_id*: identificador de la compra.
- *total*: total de la compra.
- *date_purchase*: fecha de la compra.
- *deal_id:* identificador del *deal* (ítem).
- *category_id*: identificador de la categoría.
- *category_label*: nombre de la categoría.

Otros aspectos a considerar son los siguientes. Las compras que se quieren obtener deben estar en estado concretado (purchases_deals_state_id=9), así como también el ítem debe estar activo (d.state_id=1). La página ofrece tres tipos de deals: promociones, escapes y productos, siendo los dos primeros los cupones de descuento, los cuales son el objetivo a recomendar (deal_type_id in (1,2)). Además, hay dos formas de definir un *deal,* simple, el cual tiene sus propias características y no dependen de otro, padre, el cual define una estructura inicial con ciertas características que finalmente serán compartidas con sus hijos, quienes definen sus propias características que los diferencian entre los otros; siendo esto una lógica de negocio, para un usuario esto es transparente, y al hacer una compra este puede ser un *deal* simple o hijo, ya que un *deal* padre no es más que una abstracción y no puede ser adquirido directamente, de esta forma se realiza el filtro (*deal_multiplicity_id in* (1,3)). Por último, se tomaron los *deals* con fecha de inicio a partir del 01-01-2016.

Para obtener estos datos se utilizó la siguiente consulta y fueron contenidos en un *data frame* de la librería pandas de Python.

```
SELECT p.user_id as userID, p.id as PurchaseID, p.total as totalPurchase,
p.date_purchase as datePurchase, d.id as dealID, c.id as categoryID, c.label as category
FROM purchases p INNER JOIN purchases_deals pd
ON pd.purchase_id=p.id
INNER JOIN deals as d
ON d.id=pd.deal_id
INNER JOIN categories c
ON c.id=d.l1_category_id
WHERE pd.purchases_deals_state_id=9 and
d.state_id=1 and d.deal_type_id in (1,2) and
d.deal_multiplicity_id in (1,3)
and d.date_begin  >= '2016-01-01' and
c.category_type_id=1
ORDER BY userID;
```

*Figura 19 - Query para obtención de datos*

Para almacenar los datos recolectados y facilitar su estudio, se utilizó la base de datos MongoDB, el modelo de como serán almacenados los documentos, es el siguiente:

```
{
    user_id: user_id,
    purchases: [
        {
            purchaseID: purchaseId,
            purchaseTotal: total,
            purchaseDate: purchaseDate,
            dealID: dealId,
            category: categoryLabel,
            categoryID: categoryID
        }
        ...
    ]
}
```

*Figura 20 - Modelo de datos de MongoDB*

Luego de obtener los datos a través de la query y tenerlos en el *data frame* se procede a abrir una conexión con MongoDB e insertar los datos en un documento, para de esta forma, tener los datos en un repositorio accesible para este estudio.

Una vez insertados todos los datos, la base de datos Mongo quedó con 29,251 documentos para ser analizados.

## 4.5 Entendimiento de los datos

Esta fase comprende usar técnicas de visualización de datos para conocer los datos con que se estarán trabajando.

Para lograr esto, primero se tuvo que obtener los datos almacenados en la base de datos MongoDB, esto se hizo a través de la librería **pymongo** de la siguiente manera:

```
cursor=mongodb.aggregate([
    {"$unwind":"$purchases"},
    {"$project":
            {
                "_id":0,
                "purchaseID":"$purchases.purchaseID",
                "purchaseTotal":"$purchases.purchaseTotal",
                "purchaseDate":"$purchases.purchaseDate",
                "dealID":"$purchases.dealID",
                "categoryID":"$purchases.categoryID",
                "category":"$purchases.category",
                "userID":1,
            }
    }
])
```

*Figura 21 - Inserción de los datos en MongoDB*

Luego este *cursor* es convertido en un *data frame* por la librería *pandas*.

Con los datos ya en el *data frame* se procedió al estudio de los datos. En primer lugar, se obtuvo cómo están contenidos los datos en el *data frame,* es decir los *metadatos*. Se obtuvo lo siguiente:

```
# metadata
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52275 entries, 0 to 52274
Data columns (total 7 columns):
category         52275 non-null object
categoryID       52275 non-null int64
dealID           52275 non-null int64
purchaseDate     52275 non-null object
purchaseID       52275 non-null int64
purchaseTotal    52275 non-null int64
userID           52275 non-null int64
dtypes: int64(5), object(2)
memory usage: 2.4+ MB
```

*Figura 22 - Info de los datos cargados*

Vemos que el objeto donde están almacenados es un DataFrame de la librería *pandas*, el mismo tiene 52275 entradas y 7 columnas que corresponden a los datos requeridos en las fases anteriores. Ninguna de las columnas tiene valores nulos, lo que facilita su estudio y por último, cinco de los datos son de tipo *int64* (*categoryID, dealID, purchaseID, purchaseTotal, userID*) y dos de tipo *object* (*category, purchaseDate*). Cabe destacar que los datos no están ni indexados ni agrupados de ninguna manera; para facilitar las operaciones, se decidió indexar los datos por *user_id* y *purchase_id*, siendo estos datos los que permiten diferenciar entre usuarios y compras.

Con los datos ya indexados y teniendo en cuenta que las recomendaciones se harán con base en las categorías, se generaron los siguientes gráficos:

- **Cantidad de deals vendidos por categoría entre 2016 y 2017**

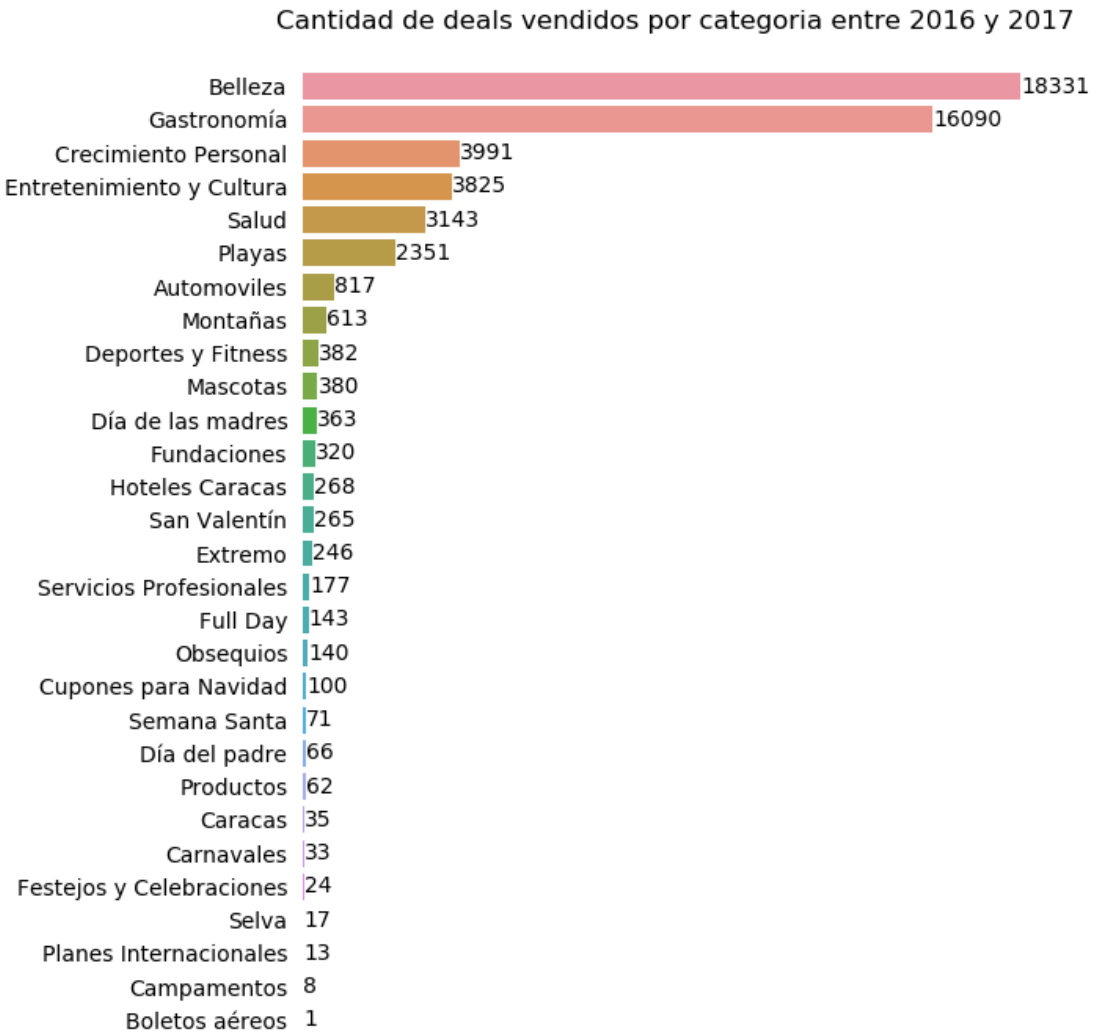


*Figura 23 - Deals vendidos por categoría 2016 y 2017*

Las categorías Belleza, Gastronomía, Crecimiento Personal, Entretenimiento y Cultura, Salud y Playas son las más vendidas en el período comprendido entre 2016 y 2017, a partir de esto se puede pensar que serán las categorías con más recomendaciones, mientras que el resto de las categorías, que son menos comunes entre los usuarios, se esperaría que tengan menos recomendaciones.

Para ver esto en más detalle, se hizo el mismo gráfico por cada año.

- **Cantidad de deals vendidos por categoría en el 2016**

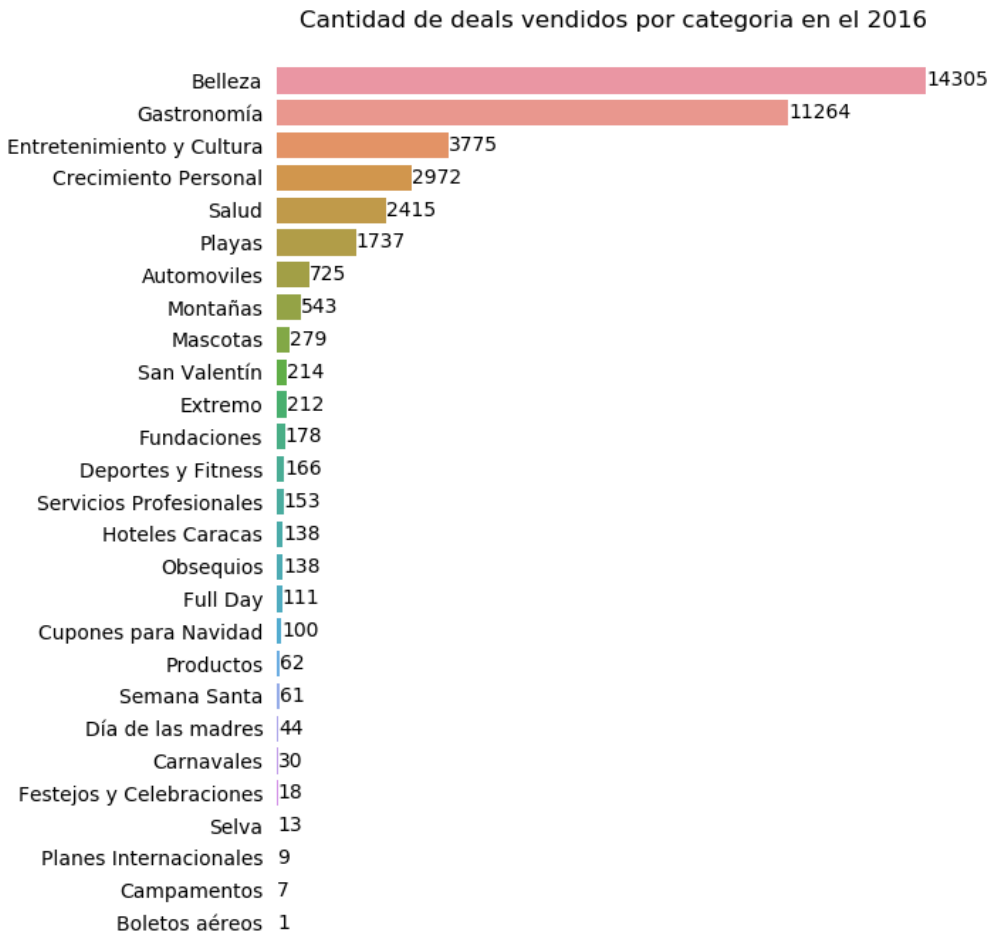


*Figura 24 - Deals vendidos por categoría 2016*

Podemos ver que este gráfico es similar al anterior, donde las primeras categorías son las más populares entre los usuarios por lo que se puede esperar que estén entre las más recomendadas.

- **Cantidad de deals vendidos por categoría en el 2017**

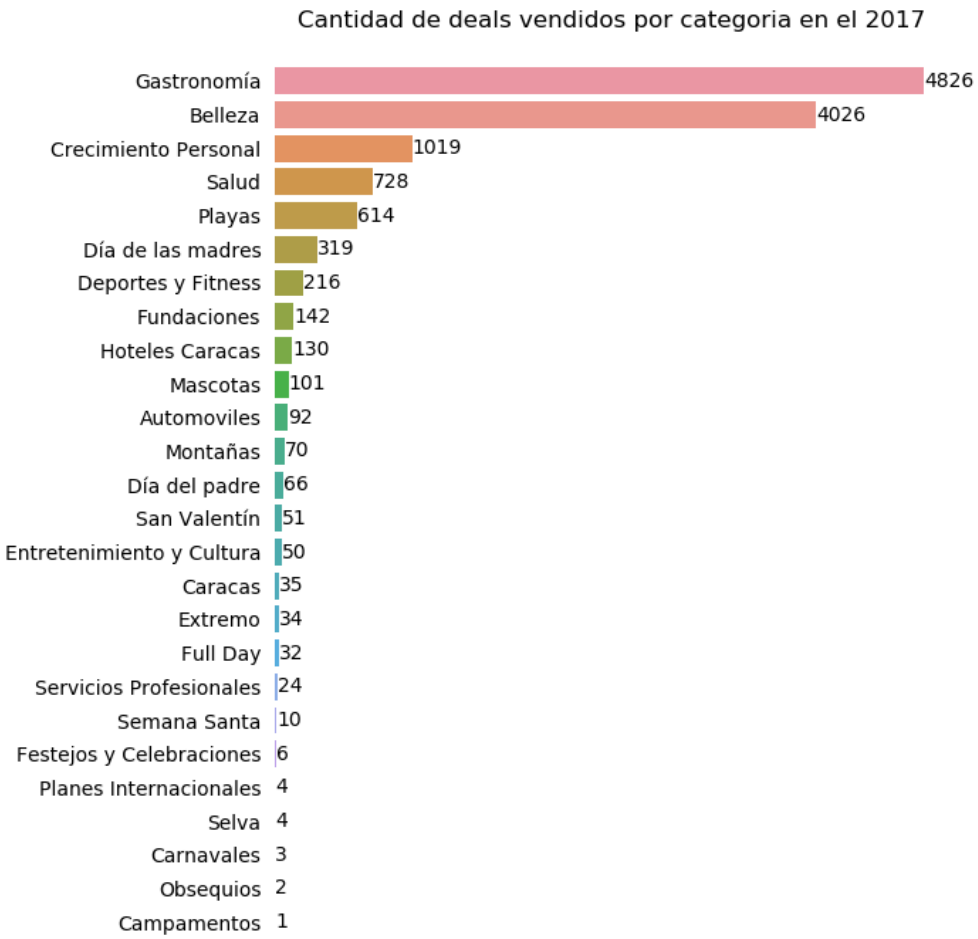


*Figura 25 - Deals vendidos por categoría en el 2017*

En este caso podemos ver que hubo un aumento en la categoría Gastronomía, siendo la principal en este caso, así como la disminución de algunas (Entretenimiento y Cultura) así como la desaparición de otras como (Boletos aéreos), sin embargo, en ambos años vemos que las categorías más populares entre los usuarios se mantienen.

- **Total de Bs. por categoría**

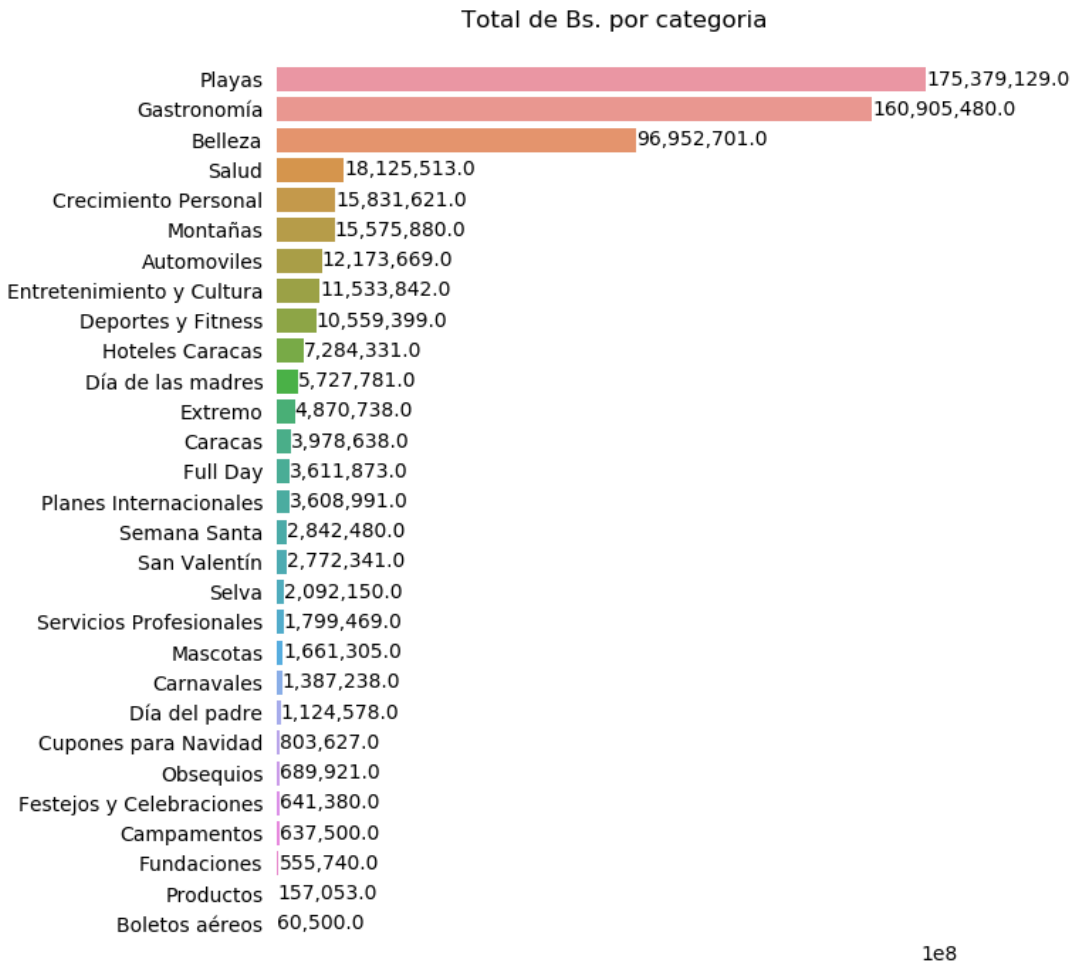


*Figura 26 - Total de bs. por categoría*

En cuanto al total de Bs. recaudados por categoría, se puede ver que una vez más las categorías que se mostraron populares en los casos anteriores están entre las que más recaudan Bs., por otra parte, hay que resaltar, que algunas categorías como Playas, Montañas, Automóviles que aparecen entre las primeras en este gráfico, sus *deal* asociados suelen ser costosos porque involucran viajes, planes, full-days, etc. No obstante, no debería confundirse esto como una categoría que tuvo mucha cantidad de ventas ni tampoco que es muy popular entre los usuarios ya que como se vio anteriormente el número de *deals* vendidos está por debajo de Belleza y Gastronomía.

- **Cantidad de Bs. recaudados entre Enero 2016 a Agosto 2017**

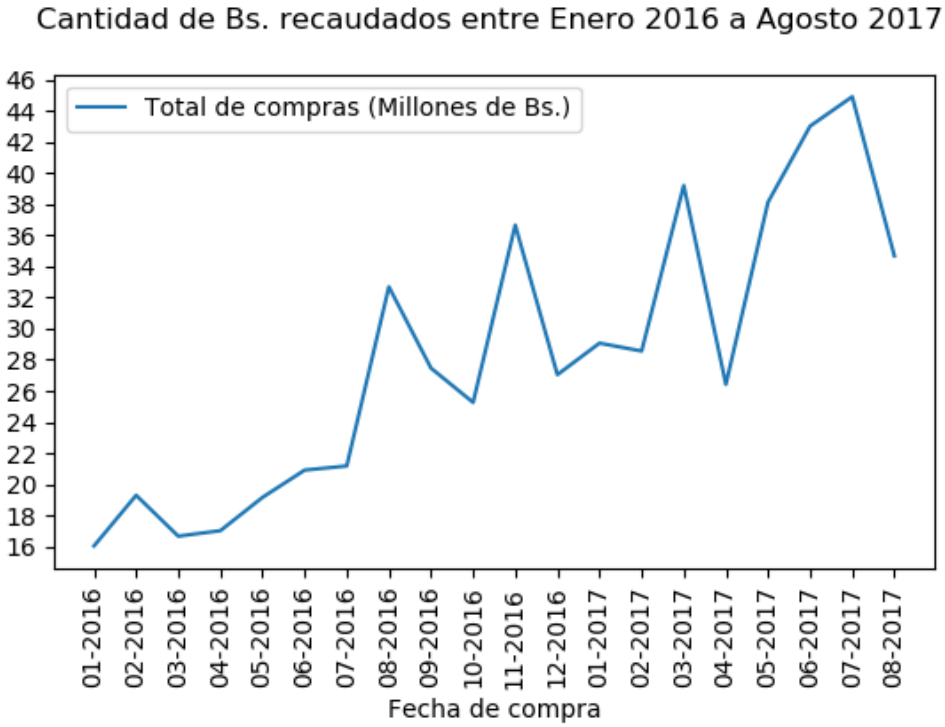


*Figura 27 - Bs. recaudados entre Enero 2016 a Agosto 2017*

En este gráfico podemos ver cómo fueron evolucionando las ventas desde enero de 2016 hasta agosto de 2017 (período de los datos a analizar). Se puede apreciar un crecimiento con diversos picos de altos y bajos, esto se puede deber tanto al aumento de precios como caída en la actividad de la página, sin embargo, el entendimiento de dicho comportamiento está fuera de los objetivos de este proyecto; la finalidad de este gráfico es tener noción de cómo han evolucionado las ventas en este período de tiempo.

## 4.6 Preparación de los datos

Ya que se pretende realizar un filtrado colaborativo, es necesario crear una matriz usuario/ítem donde cada valor representa el *rating* que cada usuario ha dado a cada ítem. En esta caso la matriz a construir es usuario/categoría y cada valor corresponde a cuántos *deals* de dicha categoría ha comprado el usuario, siendo este valor el *rating* implícito.

Esta fase también involucra la limpieza de los datos, sin embargo, debido a que los datos provienen de una fuente de datos estructurados, los mismos están bastante limpios y acordes para el estudio. Como vimos en la fase anterior, no hay datos nulos o vacíos, y los datos duplicados son necesarios en este caso ya que representan múltiples compras del mismo ítem. Teniendo ya los datos listos se procedió a crear la matriz.

Como la matriz será usuario/categoría, se deben indexar los datos por *user_id* y *category*, en este caso se usará el nombre de la categoría para que esta manera sea más legible.

```
data_aux = data.set_index(["userID","category"])
```

*Figura 28 - Definir indices*

Luego, utilizando funciones de la librería Pandas, podemos generar la matriz de la siguiente manera:

```
user_item=data_aux.groupby(["userID","category"])["categoryID"].count().unstack(fill_value=0)
```

*Figura 29 - Creación de la matriz usuario/categoría*

Siendo la matriz resultante la siguiente:

| category | Automoviles | Belleza | Boletos aéreos | Campamentos | Caracas | Carnavales | Crecimiento Personal | Cupones para Navidad | Deportes y Fitness | Día de las madres | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| userID | | | | | | | | | | | |
| 13 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 103 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | ... |
| 110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 131 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 197 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 204 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 206 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 219 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 222 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 223 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 230 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 239 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 275 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... |
| 286 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 293 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |

*Figura 30 - Matriz usuario/categoría*

Los valores en cero (0) significa que el usuario no ha comprado ítems de esa categoría por lo que son candidatos a ser recomendados. Esta matriz resultante, es muy dispersa debido a su naturaleza que implica que los ítems no calificados, sean denotados con cero.

Para facilitar el acceso a las columnas y evitar problemas con los espacios y caracteres especiales (acentos, eñes, etc.) se sustituyeron por guión bajo (_) y fueron removidos respectivamente.

```
user_item.columns=user_item.columns.str.replace(" ","_")
user_item.columns=user_item.columns.str.replace("á","a")
user_item.columns=user_item.columns.str.replace("é","e")
user_item.columns=user_item.columns.str.replace("í","i")
user_item.columns=user_item.columns.str.replace("ó","o")
user_item.columns=user_item.columns.str.replace("ú","u")
user_item.columns=user_item.columns.str.replace("ñ","n")
```

```
user_item.columns.values
```

```
array(['Automoviles', 'Belleza', 'Boletos_aereos', 'Campamentos',
       'Caracas', 'Carnavales', 'Crecimiento_Personal',
       'Cupones_para_Navidad', 'Deportes_y_Fitness', 'Dia_de_las_madres',
       'Dia_del_padre', 'Entretenimiento_y_Cultura', 'Extremo',
       'Festejos_y_Celebraciones', 'Full_Day', 'Fundaciones',
       'Gastronomia', 'Hoteles_Caracas', 'Mascotas', 'Montanas',
       'Obsequios', 'Planes_Internacionales', 'Playas', 'Productos',
       'Salud', 'San_Valentin', 'Selva', 'Semana_Santa',
       'Servicios_Profesionales', 'Media_Rating'], dtype=object)
```

*Figura 31 - Preparación de los datos*

En este punto, los datos están preparados para realizar las recomendaciones.

## 4.7 Modelado

Esta fase comprende realizar las recomendaciones, como se ha mencionado previamente, las mismas se realizarán con el algoritmo filtrado colaborativo, específicamente el basado en usuario.

Para este trabajo de grado se realizarán 2 iteraciones con ajustes al algoritmo para determinar el más óptimo.

Para realizar las recomendaciones basadas en usuario, es necesario calcular la similitud entre dos usuarios comparando el *rating* dado a un mismo ítem. Estas medidas de similitud pueden ser: correlación de Pearson, distancia euclidiana y similitud del coseno. Sin embargo, cada medida de similitud tiene su ventaja con respecto a los datos, algunas recomendaciones son las siguientes [117]:

- Correlación de Pearson: cuando existe sesgo del usuario en los datos.
- Similitud del coseno: cuando los datos son muy dispersos y muchas recomendaciones faltan.
- Distancia Euclidiana: si los datos no son dispersos y la magnitud del valor es significativa.

Con esto en mente, la similitud del coseno es la que mejor se ajusta a nuestros datos debido a que la matriz usuario/categoría es bastante dispersa. Esta mide

la similitud entre dos ventores de n-dimensiones basado en el coseno del ángulo entre ellos [41], es decir, entre los vectores de usuario y de ítems. La fórmula es la siguiente:

$$s(\vec{u},\vec{v}) = \cos\theta = \frac{\vec{u}.\vec{v}}{|\vec{u}| \times |\vec{v}|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2} \times \sqrt{\sum_i r_{v,i}^2}} \qquad [41]$$

Donde:

- $s(\vec{u},\vec{v})$, es la similitud entre el vector *u* y el vector *v*, este caso los usuarios *u* y *v*.
- $r_{u,i}$ y $r_{v,i,}$ son los *rating* dados por ambos usuarios al ítem *i.*

Una vez obtenido la similitud entre los usuarios, para predecir el *rating*, en este caso implícito, se calcula la suma ponderada de los *ratings* de dados por los usuarios a un ítem *i*, representada por la siguiente formula:

$$p_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}} \qquad [118]$$

Donde:

- $p_{u,i}$, es la predicción de *rating* de un ítem *i* del usuario *u*.
- $r_{v,i}$, es el *rating* dado por un usuario *v* al ítem *i*.
- $s_{u,v}$, es la similitud entre ambos usuarios.

Esta fórmula puede ser generalizada y mejorada agregando la media de *ratings* dada por el usuario y agregando a la suma ponderada la diferencia de la media de *ratings* del resto de los usuarios, definido por la siguiente formula:

$$p_{u,i} = \bar{u} + \frac{\sum_i (r_{v,i} - \bar{r_v}) \times s_{uv}}{\sum_v s_{u,v}} \qquad [41]$$

Donde:

- $\bar{u}$, es la media de *ratings* dada por el usuario *u.*
- $\bar{r_v}$, es la media de *ratings* dada por el usuario *v.*

Una vez obtenida esta predicción, esta se tomará como recomendación si supera un umbral seleccionado previamente.

Una vez definido el algoritmo de filtrado colaborativo, la implementación se muestra a continuación.

Para la primera iteración se utilizará la fórmula de predicción simple.

**Iteración 1**

Como primer paso, se separaron los datos en dos conjuntos, uno de entrenamiento y otro de evaluación, esto con el fin de no sobreajustar los datos y no evaluar el sistema con los mismos datos que fueron entrenados. Para esto se dividió un 70% el conjunto de entrenamiento y un 30% los datos de prueba. Se utilizó la librería *sklearn* para realizar esta división de la siguiente manera:

```
# split data
train_set, test_set = train_test_split(user_item, test_size=0.30, random_state=100)
```

*Figura 32 - Separación de los conjuntos de datos*

Debido a que los datos son muy dispersos, es altamente costoso realizar cálculos sobre esta matriz, no obstante, la librería *scipy* ofrece un objeto llamado *csr_matrix* (*Compressed Sparse Row matrix*) una matriz dispersa comprimida que mejora la eficiencia en cuánto a los cálculos, el conjunto *train_set* fue convertido en dicho tipo de matriz:

```
train_set_aux=csr_matrix(train_set.values)
train_set_aux
```

```
<18720x29 sparse matrix of type '<class 'numpy.int64'>'
        with 22662 stored elements in Compressed Sparse Row format>
```

*Figura 33 - Matriz comprimida*

Se puede ver que queda una matriz de 18720x29 con 22662 elementos almacenados en Compressed Sparse Row format.

Con la matriz en este formato, se procedió a calcular la similitud del coseno entre los usuarios haciendo uso de la función *cosine_similarity* de la librería *sklearn*:

```
similarity=cosine_similarity(train_set_aux)
```

*Figura 34 - Cálculo de la similitud*

Esto retornará una matriz con las similitudes ente [0,1] donde 0 es nada similares y mientras más cercano a 1 es más similar:

```
[[1. 0. 0. ... 0. 0. 0.]
 [0. 1. 1. ... 1. 1. 0.]
 [0. 1. 1. ... 1. 1. 0.]
 ...
 [0. 1. 1. ... 1. 1. 0.]
 [0. 1. 1. ... 1. 1. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

*Figura 35 - Matriz de similitud*

Es evidente como en la diagonal siempre es 1 debido ya que un usuario es similar a sí mismo.
Teniendo las similitudes, ahora es posible realizar las predicciones usando la primera fórmula, implementada de la siguiente manera:

```
def prediction(users, similarity):
    pred = similarity.dot(users) / np.array([np.abs(similarity).sum(axis=1)]).T
    return pred
```

*Figura 36 - Implementación de primera fórmula*

Esta función recibe la matriz usuario/ítem y la matriz de similitudes. Se utiliza la función *dot* de la librería *numpy* para realizar el producto punto entre cada vector de la matriz y se divide entre la suma de las similitudes. Se utiliza el valor absoluto para evitar errores, y se suma de fila en fila, generando un vector de una sola dimensión, debido a esto, se convierte en un nuevo *array* y al hacerle T se convierte en la transpuesta y queda en la misma dimensión que la otra matriz. Esta división de matrices será entre cada de los vectores.

Así, la matriz de predicciones, queda de la siguiente manera:

```
[[1.59616555e-02 1.29689717e-01 0.00000000e+00 ... 1.26291444e-04
  7.79189440e-04 8.47779805e-04]
 [7.77098314e-03 1.63454079e+00 8.60563353e-05 ... 6.08510183e-05
  5.35288831e-04 2.10415040e-03]
 [1.36752547e-02 1.56192869e-01 0.00000000e+00 ... 0.00000000e+00
  5.84003097e-03 5.84003097e-03]
 ...
 [8.10507872e-03 1.34689011e-01 0.00000000e+00 ... 0.00000000e+00
  3.75093321e-04 2.30882194e-03]
 [7.77098314e-03 1.63454079e+00 8.60563353e-05 ... 6.08510183e-05
  5.35288831e-04 2.10415040e-03]
 [1.59616555e-02 1.29689717e-01 0.00000000e+00 ... 1.26291444e-04
  7.79189440e-04 8.47779805e-04]]
```

*Figura 37 - Matriz de predicciones*

Para verlo con mayor comodidad y entender mejor qué se tiene, se puede convertir esta matriz en un *data frame.*

```
pd.DataFrame(recommendations).head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.007992 | 1.620095 | 0.000078 | 0.000078 | 0.000068 | 0.000617 | 0.041915 | 0.001540 | 0.005453 | 0.008860 | ... |
| 1 | 0.016217 | 0.130693 | 0.000000 | 0.000094 | 0.000469 | 0.000420 | 0.027899 | 0.001710 | 0.004313 | 0.004312 | ... |
| 2 | 0.016217 | 0.130693 | 0.000000 | 0.000094 | 0.000469 | 0.000420 | 0.027899 | 0.001710 | 0.004313 | 0.004312 | ... |
| 3 | 0.016217 | 0.130693 | 0.000000 | 0.000094 | 0.000469 | 0.000420 | 0.027899 | 0.001710 | 0.004313 | 0.004312 | ... |
| 4 | 0.010026 | 0.443214 | 0.000000 | 0.000000 | 0.000351 | 0.000000 | 0.044159 | 0.002781 | 0.007584 | 0.007527 | ... |

5 rows × 29 columns

*Figura 38 - Matriz de recomendaciones en formato Dataframe*

Tenemos entonces una predicción por usuario a cada categoría. Ahora que con las predicciones/recomendaciones, se evalúa el modelo.

## 4.8 Evaluación

La evaluación del sistema de recomendación es uno de los pasos más importantes debido a que determinará, qué tan eficiente serán las recomendaciones.

Las métricas utilizadas para evaluar la precisión en los sistemas de recomendación se dividen en dos tipos:

- Métricas de precisión estadística: evalúan la precisión de un sistema de filtrado comparando los *ratings* predichos con los *ratings* reales dados por el usuario. Estas comprenden:
    - Mean Square Error (MSE): es una medida de la desviación de la recomendación del valor específico del usuario. Cuanto más bajo sea el MSE, más preciso será el sistema de recomendación.

$$MSE = \frac{1}{N}\sum_{u,i}\left(p_{u,i} - r_{u,i}\right)^2 \qquad [41]$$

Donde:

- $p_{u,i}$, es la predicción del ítem *i* para el usuario *u*
- $r_{u,i}$, es el *rating* otorgado por el usuario *u* al ítem *i.*

- Root Mean Square Error (RMSE): pone más énfasis en el error absoluto más grande e igualmente al MSE, mientras más bajo, mejor el sistema de recomendación. Este viene dado por:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}\left(p_{u,i} - r_{u,i}\right)^2} \quad [41]$$

- Métricas de precisión de soporte de decisión: Estas métricas ayudan a los usuarios a seleccionar elementos que son de muy alta calidad del conjunto de elementos disponibles. Las métricas ven el procedimiento de predicción como una operación binaria que distingue los artículos buenos de aquellos que no son buenos. Los más usados son:

  - Precision: es la fracción de los elementos recomendados que son realmente relevantes para el usuario.

$$Precision = \frac{Correctly\ recommended\ items}{Total\ recommende\ items} \quad [41]$$

  - Recall: es la fracción de elementos relevantes que también forman parte del conjunto de elementos recomendados.

$$Recall = \frac{Correctly\ recommende\ items}{Total\ useful\ recommended\ items} \quad [41]$$

Para este trabajo utilizarán las métricas de precisión estadística, debido a la naturaleza de las recomendaciones, las mismas se basan en una predicción en un *rating* implícito que puede ser comparado con un *rating* real dado por el usuario para determinar la precisión.

Para lograr esto, los datos de evaluación fueron separados en dos conjuntos, uno donde se sustituirá el *rating* de un usuario a un ítem por cero (0), es decir, ausencia de *rating*, y otro conjunto, del mismo tamaño que el anterior, pero solo con los valores (*ratings)* que fueron removidos del primero en el correspondiente usuario/ítem, esto con el fin de predecir un valor conocido y luego calcular la precisión del mismo comparándolos. La elección de estos valores será aleatoria, siempre y cuando el usuario tenga más de un (1) *rating*, sino, al usuario queda con su único valor. Esto se realizó de la siguiente manera:

```
def calculate_target_set(val_set):
    target = np.zeros(val_set.shape)
    val_set_aux = val_set.copy()
    for user in range(val_set.shape[0]):
        if(len(val_set[user, :].nonzero()[0]) > 1):
            rats = np.random.choice(val_set[user, :].nonzero()[0], replace=False)
            val_set_aux[user, rats] = 0.
            target[user, rats] = val_set[user, rats]
    return val_set_aux, target
```

*Figura 39 - Creación de target set*

Podemos ver que *target* y *val_set_aux* son dos matrices del mismo tamaño, pero la primer solo con ceros, luego se recorren todos los usuarios y se toma un *rating* al aleatorio que se remueve de la copia del conjunto original (*val_set_aux*) y se asigna al que contendrá los removidos (*target*).

```
test_set_b, target_set = calculate_target_set(test_set.values)
```

*Figura 40 - Separación de los test de evaluaciones*

Así se crean estos dos conjuntos de datos desde los datos de evaluación originales. Con *test_set_b* se hará las predicciones, por lo que se debe sacar primero la similitud entre los usuarios:

```
test_set_aux=csr_matrix(test_set_b)
test_set_aux
```

```
similarity_val=cosine_similarity(test_set_aux)
```

*Figura 41- Matriz comprimida*

Una vez obtenidas las similitudes, se pueden hacer las recomendaciones.

```
recommendations_val=prediction(test_set_b, similarity_val)
```

*Figura 42 - Ejecución de la recomendación para evaluar*

Con estos resultados, se evaluará el sistema, existe una serie de valores que fueron predichos cuyo valor real es conocido y está almacenado en *target_set*. Por lo tanto, se calculará el error en dichos valores. Para calcular el MAE, se utilizará la función del mismo nombre de la biblioteca *scikit-learn*, y para el RMSE, la raíz cuadrada del anterior:

```
mean_squared_error(target_set[target_set.nonzero()], recommendations_val[target_set.nonzero()])
```

```
4.748760383651609
```

*Figura 43 - MSE iteración 1*

```
sqrt(mean_squared_error(target_set[target_set.nonzero()], recommendations_val[target_set.nonzero()]))
```

```
2.1791650657193475
```

*Figura 44 - RMSE iteración 1*

Se puede ver que en ambos los errores son considerablemente bajos, por lo que se puede decir que el modelo está haciendo recomendaciones precisas.

**Iteración 2**

## 4.9 Modelado

En esta iteración, la fórmula de predicción será ajustada para remover la parcialidad de los usuarios hacia ciertos ítems, se pueden dar casos donde los usuarios tengan cierta parcialidad hacia un ítem lo que hará que tenga más *ratings* que otros. Esta parcialidad puede ser removida restando la media de *ratings* de cada usuario al hacer las sumas ponderadas y luego sumar la media de *ratings* del usuario a predecir. Esto lo que comprende en la segunda fórmula.

Para esta iteración, se volvió a separar los datos en entrenamiento y evaluación para evitar el sobreajuste:

```
# split data
train_set, test_set = train_test_split(user_item, test_size=0.30, random_state=2000)
```

*Figura 45 - División para la segunda iteración*

La implementación de la fórmula antes mencionada, quedó de la siguiente manera:

```
def prediction_general(user, similarity):
    rating_means = user.mean(axis=1)
    mean_diff = (user - rating_means[:,np.newaxis])
    pred = rating_means[:,np.newaxis] + similarity.dot(mean_diff) / np.array([np.abs(similarity).sum(axis=1)]).T
    return pred
```

*Figura 46 - Implementación fórmula 2*

Se puede notar que se calcula la diferencia entre los ratings y la media por cada usuario. Ahora se determina la similitud entre los usuarios y luego se procede a hacer las recomendaciones.

```
train_set_aux=csr_matrix(train_set.values)
train_set_aux
```

```
<20475x29 sparse matrix of type '<class 'numpy.int64'>'
        with 24754 stored elements in Compressed Sparse Row format>
```

```
similarity_2=cosine_similarity(train_set_aux)
```

*Figura 47 - Compresión de la matriz y cálculo de la similitud*

```
recommendations_2 = prediction_general(train_set.values, similarity_2)
```

*Figura 48 - Generación de recomendaciones*

Vemos las recomendaciones generadas:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.025625 | 1.575742 | -0.032688 | -0.032688 | -0.032535 | -0.032245 | 0.007107 | -0.031539 | -0.027635 | -0.023961 | ... |
| 1 | -0.018702 | 0.094631 | -0.035514 | -0.035420 | -0.035012 | -0.035239 | -0.006241 | -0.034281 | -0.032054 | -0.032382 | ... |
| 2 | -0.025625 | 1.575742 | -0.032688 | -0.032688 | -0.032535 | -0.032245 | 0.007107 | -0.031539 | -0.027635 | -0.023961 | ... |
| 3 | -0.025625 | 1.575742 | -0.032688 | -0.032688 | -0.032535 | -0.032245 | 0.007107 | -0.031539 | -0.027635 | -0.023961 | ... |

*Figura 49 - Matriz de predicciones*

Podemos que ver que en este caso hay valores negativos, los mismos representan que no es un ítem a recomendar gracias a la substracción de la parcialidad del usuario.

Una vez definido el modelo, se procede a evaluar.

## 4.10 Evaluación

En esta etapa, al igual que en la iteración 1, es necesario dividir el conjunto de evaluación para tener en un conjunto valores conocidos faltantes para ser predichos y en el otro, dichos valores conocidos. Se realiza de la misma forma, seleccionando *ratings* aleatorios de cada usuario, siempre y cuando tenga más de un *rating*.

```
test_set_c, target_set = calculate_target_set(test_set.values)
```

*Figura 50 - División de matriz para evaluación*

La matriz *test_set_c* será la usada para hacer las predicciones, mientras que *target_set* contiene los valores conocidos.

Nuevamente es necesario calcular las similitudes entre los usuarios para hacer las predicciones.

```
test_set_aux=csr_matrix(test_set_c)
test_set_aux
```

```
<8776x29 sparse matrix of type '<class 'numpy.int64'>'
        with 9175 stored elements in Compressed Sparse Row format>
```

```
similarity_eval_2=cosine_similarity(test_set_aux)
```

```
recommendations_eval_2=prediction_general(test_set_c, similarity_eval_2)
```

*Figura 51 - Compresión de la matriz, generación de la similitud y la matriz de recomendaciones*

Por último, con las recomendaciones predichas, se procede a evaluar nuevamente comprando con los datos conocidos:

```
sqrt(mean_squared_error(target_set[target_set.nonzero()], recommendations_eval_2[target_set.nonzero()]))
```

```
2.095435116084294
```

```
mean_squared_error(target_set[target_set.nonzero()], recommendations_eval_2[target_set.nonzero()])
```

```
4.3908483257191975
```

*Figura 52 - MSE y RMSE de iteración 2*

En este caso, ambos errores también dan bajo. Corresponde ahora a comprar ambos modelos para determinar cuál tiene mejor *performance*.

## 4.11 Comparación de resultados

El siguiente gráfico corresponde a un cuadro comparativo entre los resultados de la evaluación de cada modelo en cada iteración:

| | MSE | RSME |
|---|---|---|
| **Iteración 1** | 4.748760383651609 | 2.1791650657193475 |
| **Iteración 2** | 4.3908483257191975 | 2.095435116084294 |

*Figura 53 - Cuadro comparativo entre ambas iteraciones*

Se puede ver que la diferencia entre el MSE modelo de la iteración 1 y la iteración 2 es, aproximadamente, 0.358, mientras que con respecto al RSM, la diferencia es de 0.084. A pesar que ambos errores son bajos y que la diferencia entre ellos es pequeña también, el modelo desarrollado en la iteración 2, donde se resta la parcialidad de los usuarios, tiene mejor rendimiento, por lo que se considera que es la mejor opción para ser utilizada como sistema de recomendación.

## 4.12 Interpretación de los resultados obtenidos en la evaluación

Para describir qué significa los resultados obtenidos en la evaluación, tomaremos como referencia el RSME obtenido en la segunda iteración.

Como se dijo anteriormente, el RSME calcula la raíz cuadrada de la diferencia cuadrática entre el valor predicho y el valor actual entre el número de observaciones, por esto puede ser considerada como la desviación estándar y la mejor para interpretar su significado.

Teniendo esto en cuenta, el valor RSME ≅ 2.095 **es la diferencia promedia entre los *ratings* implícitos reales y los *ratings* predichos**, esto se interpreta como que para cada recomendación, se puede esperar un error de aproximadamente 2.095 en el *rating* implícito que daría un usuario a dicho elemento, en esta caso las compras, recordemos que en este trabajo el *rating* implícito comprendía cada compra a un ítem.

Esto implica que puede haber una diferencia de lo esperado en cuántas veces compre un usuario un ítem recomendado y el número de veces que realmente lo haga.

# Capítulo 5: Conclusiones

En el presente trabajo especial de grado, se desarrolló un sistema de recomendación para Aprovecha.com. Esto se hizo siguiendo la metodología *Foundational Methodology for data science*, analizando los requerimientos del negocio, para así hacer un enfoque analítico del problema, analizar los datos requeridos, recolectar dichos datos, explorarlos para su entendimiento, modificar y hacer las operaciones pertinentes para dejar los datos preparados y luego se realizar el modelado y por último la evaluación del sistema de recomendación. En ese proceso se realizaron dos iteraciones donde se probaron 2 algoritmos de recomendación propuestos para luego comparar el rendimiento de ambos, siendo elegido el mejor.

Se logró desarrollar el sistema de recomendación que se había planteado utilizando el conjunto de datos de aprovecha.com mediante herramientas que facilitaron la implementación del mismo por su eficacia con manejo de grandes volúmenes de datos y funcionalidades que proveen algoritmos para ciencia de datos y aprendizaje automático. Las recomendaciones se lograron prediciendo el *rating* de un usuario a un ítem calculando las sumas ponderadas de las similitudes entre los usuarios y la diferencia entre el *rating* del resto de los usuarios al mismo ítem y la media de *ratings* de los mismos. Las medidas de evaluación utilizadas fueron las métricas de evaluación estadística, debido a que las recomendaciones se hicieron como predicciones con un cálculo estadístico.

Durante el proceso de desarrollo del sistema de recomendación, se observó que debido al alto volumen de datos y su dispersión el procesamiento es lento y es costoso a nivel de cómputo.

Este trabajo especial fue realizado hasta la fase de Evaluación de la metodología *Foundational Methodology for data science,* pretendiendo dejar listo el sistema de recomendación para su futura integración como fue definido en los objetivos específicos y en alcance. Las fases de Despliegue y Feedback, involucran esta integración. Esto debido a lo complejo y prolongado que involucraría integrar dos tecnologías y plataformas distintas sumado a lo ya realizado en este TEG.

Los objetivos planteados en el capítulo 1, fueron alcanzados de la siguiente manera:

***Definir el enfoque y alcance del sistema de recomendaciones de Aprovecha.com.**

El alcance del desarrollo del sistema de recomendación fue definido en el capítulo 1 de este trabajo especial, estableciendo que fuese hasta la fase evaluación, para de esta manera dejar finalizado el sistema de recomendación. En cuanto al enfoque, se definió y desarrolló un filtrado colaborativo para realizar las recomendaciones.

***Definir la(s) metodología(s) de desarrollo de aplicaciones para el sistema de recomendaciones de Aprovecha.com.**

La metodología seleccionada para desarrollar el sistema de recomendación, como se indica en el capítulo 3, fue la *Foundational Methodology for Data Science*. Para el desarrollo del sistema de recomendación, se siguió cada fase de esta metodología lo que permitió dividir y enfocar cada problema en dichas etapas facilitando el flujo de trabajo y desarrollo de este TEG.

***Seleccionar las herramientas, marcos de trabajos (Frameworks), Almacenes de datos y lenguajes de programación a utilizar para el desarrollo del sistema de recomendaciones de Aprovecha.com.**

Se realizó la selección exitosa de las herramientas, lenguajes de programación, almacén de datos, como se indica en el capítulo 4 del presente trabajo, dichas herramientas seleccionadas se adaptaron correctamente para la implementación del sistema de recomendación y facilitaron su desarrollo.

***Analizar y seleccionar los datos que permitan definir e implementar los modelos de analítica predictiva para el desarrollo del sistema de recomendaciones de Aprovecha.com.**

Se hizo el análisis y selección de datos en la fase Requerimiento de Datos de la metodología seleccionada, así se determinaron aquellos datos necesarios para predecir las recomendaciones para los usuarios en aprovecha.com.

***Desarrollar un sistema de recomendaciones para Aprovecha.com sobre la base de la(s) metodología(s) y herramientas de desarrollo seleccionadas.**
Siguiendo la metodología mencionada previamente y haciendo uso de las herramientas de desarrollo seleccionadas, se logró desarrollar de forma exitosa el sistema de recomendaciones de aprovecha.com, prediciendo las recomendaciones a través de los datos recolectados. Se propusieron dos algoritmos para hacer las recomendaciones, quedando al final el que tuvo una mejor evaluación.

***Realizar pruebas para determinar la precisión del sistema de recomendaciones de Aprovecha.com.**

Las medidas de evaluación seleccionadas fueron las métricas de predicción estadística, específicamente, el Error Cuadrático Medio (MSE, por sus siglas en inglés) y el Error de Raíz Cuadrática Media (RMSE). Las mismas fueron fundamentales para determinar cuál algoritmo fue más preciso entre ambos para el sistema de recomendación, siendo este el que menor error arrojó.

## 5.1 Inserción de la solución en el sitio Aprovecha.com

Como se ha mencionado anteriormente, la integración e implementación del sistema de recomendación dentro del sitio se recomienda como trabajo a futuro debido a la complejidad de integrar dos plataformas y tecnologías distintas, sin embargo, se recomiendan las siguientes consideraciones para la inserción:

- Para las recomendaciones, se debe definir un umbral de *rating* y tomar todas aquellas recomendaciones que superen ese umbral.

- Actualizar constantemente la base de datos Mongo para tener siempre los últimos datos para realizar las recomendaciones.

- Definir una interfaz de programación de aplicación (API) que se comunique con el sistema recomendador y retorne los ítems a recomendar para un usuario.

- Al tener las recomendaciones, hacer una puesta en valor de los datos, normalizándolos en información relevante para el usuario y distribuirlos.

Será decisión de quienes realicen esta implementación seleccionar las arquitecturas, *frameworks*, metodologías para el correcto funcionamiento del mismo.

## 5.2 Contribuciones

Se desarrolló un sistema de recomendación que pretende aumentar la experiencia de usuario en Aprovecha.com, contribuyendo a dar más valor e importancia en el negocio y da la posibilidad de incrementar las ventas. Este TEG mostró el desarrollo de un sistema de recomendación utilizando una metodología de ciencias de datos, dejando una referencia y guía para proyectos futuros que quieran seguir el mismo tópico.

La idea es que esta solución sea utilizada por la empresa para obtener las posibles recomendaciones a un usuario y así conseguir más ganancias, a través de ventas cruzadas, sugerencias de compras a través de correo electrónico, alcance por redes sociales, etc. Todo esto para asegurar la personalización del sitio en la perspectiva de cada usuario.

## 5.3 Recomendaciones

Para ejecutar este proyecto se debe usar Python 3 de 64 bits en una máquina que tenga igual o más de 8GB de memoria RAM debido a que las operaciones de matrices con muchos datos requieren de bastantes recursos. Se recomienda también utilizar la plataforma Anaconda 3, la cual proporciona un ambiente de desarrollo con la mayoría de bibliotecas necesarias para ejecutar este proyecto además del mismo lenguaje de programación previamente mencionado. Por último, utilizar el *notebook* Jupyter para correr el proyecto y ver los resultados de forma interactiva.

## 5.4 Trabajos futuros

Como se ha mencionado previamente, el presente TEG, fue implementado hasta la fase de Evaluación del sistema de recomendación, quedando las fases de Despliegue y *Feedback* propuestas para un TEGs futuros, estas fases comprenden la integración del sistema de recomendación con el sistema de Aprovecha.com. Dicho proceso de desarrollo puede implicar la utilización de otra metodología, definir reglas de negocio para el uso del sistema de recomendación, entre otros requerimientos que encajan para el desarrollo de una TEG.

# Capítulo 6: Referencias Bibliográficas


1. What is Data Science?. (s.f). [On-line]. Disponible en (10/10/2017): https://datascience.berkeley.edu/about/what-is-data-science/
2. Rouse, M. (Octubre, 2017). *Definition: data science*. [On-line]. Disponible en (10/10/2017): http://searchcio.techtarget.com/definition/data-science
3. Erl, T; Khattak, W y Buhler, P. (Diciembre, 2015). *Big Data Fundamentals: Concepts, Drivers & Techniques*. Indiana, Estados Unidos. Prentice Hall.
4. Casado, R. (Octubre, 26 de 2015). Big Data: volumen, velocidad y variedad. [On-line]. Disponible en (10/10/2017): http://www.dataprix.com/blog-it/data-science/big-data-volumen-velocidad-variedad
5. Marr, B. (Marzo, 19 de 2015). *Why only one of the 5 Vs of big data really matters.* [On-line]. Disponible en (10/10/2017): http://www.ibmbigdatahub.com/blog/why-only-one-5-vs-big-data-really-matters
6. Big Data, What is and why it matters. (s.f.). [On-line]. Disponible en (10/10/2017): https://www.sas.com/en_us/insights/big-data/what-is-big-data.html
7. Base de datos. (s.f.). En Wikipedia. [On-line]. Disponible en (10/10/2017): https://es.wikipedia.org/wiki/Base_de_datos
8. El Modelo Relacional. (s.f). Disponible en (10/10/2017): http://docencia.lbd.udc.es/bdd/teoria/tema2/2.3.1.-ElModeloRelacional.pdf
9. Herrera, W. (Agosto, 29 de 2012). *Bases de Datos (ACID, Reglas de Codd e Integridad de datos).* [On-line]. Disponible en (10/10/2017): https://www.slideshare.net/W4L73R/bases-de-datos-acid-reglas-de-codd-e-integridad-de-datos
10. Ospina, Mercy. (s.f). *BD NoSQL: Familia de Columnas* [On-Line-Diapositivas]. Disponible en (21/04/2018): http://slideplayer.es/slide/10253277/
11. RedisLabs. (s.f.). Redis. [On-line]. Disponible en (10/10/2017): https://redis.io/
12. Aeropike. (s.f.). *What is a Key-Value Store?.* [On-line]. Disponible en (10/10/2017): https://www.aerospike.com/what-is-a-key-value-store/
13. Key-value database. (s.f.). En Wikipedia. [On-line]. Disponible en (10/10/2017): https://en.wikipedia.org/wiki/Key-value_database
14. Bases de datos orientadas a objetos y documentos. (Marzo, 14 de 2013). [On-line]. Disponible en (10/10/2017): https://www.zainex.es/tags/nosql/bases-datos-orientadas-objetos-documentos-ddbb-nosql

15. Graph database. (s.f.). En Wikipedia. [On-line]. Disponible en (10/10/2017): https://en.wikipedia.org/wiki/Graph_database

16. Neo4J. (s.f.). [On-line]. Disponible en (10/10/2017): https://neo4j.com/

17. Russell, S. y Norvig, P. (Diciembre, 11 de 2009). *Artificial Intelligence: A Modern Approach (3rd Edition)*. Estados Unidos. Prentice Hall.

18. Rouse, M. (Diciembre, 2016). *Definition: AI (Artificial Intelligence).* [On-Line]. Disponible en (11/10/17): http://searchcio.techtarget.com/definition/AI

19. Artificial Intelligence (AI). (s.f.). [On-line]. Disponible en (11/10/17): https://www.techopedia.com/definition/190/artificial-intelligence-ai

20. Machine Learning. (s.f.). En Wikipedia. [On-line]. Disponible en (11/10/17): https://en.wikipedia.org/wiki/Machine_learning

21. Rouse, M. (Junio, 2017). *Definition: machine learning*. [On-line]. Disponible en (11/10/17): http://whatis.techtarget.com/definition/machine-learning

22. Microsoft. (Marzo, 03 de 2017). *Conceptos de minería de datos.* [On-line]. Disponible en (11/10/17): https://docs.microsoft.com/es-es/sql/analysis-services/data-mining/data-mining-concepts

23. ¿Qué es el data mining? La definición de la minería de datos. (Junio, 03 de 2016). [On-line]. Disponible en (11/10/17): https://clinic-cloud.com/blog/data-mining-que-es-definicion-mineria-de-datos/

*24.* Iribarra, F. (s.f.). *Descubrimiento del Conocimiento (KDD) : "El Proceso de minería".* [On-line]. Disponible en (11/10/17): http://mineriadatos1.blogspot.com.ar/2013/06/descubrimiento-del-conocimiento-kdd-el.html

25. Fayyad, U., Piatetsky-Shapiro, G., Smyth. P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine.

26. Han, J., Kamber, M., Pei, J. (2012). *Data Mining concepts and techniques.* Estados Unidos. Morgan Kaufmann.

27. Programming Lenguage. (s.f.). En Wikipedia. [On-line]. Disponible en (11/10/17): https://en.wikipedia.org/wiki/Programming_language

28. R-Project. (s.f.). *What is R?.* [On-line]. Disponible en (11/10/17): https://www.r-project.org/about.html

29. RStudio. (s.f.). RStudio. [On-line]. Disponible en (11/10/17): https://www.rstudio.com/products/rstudio/

30. Java (lenguaje de programación). (s.f.). En Wikipedia. [On-line]. Disponible en (11/10/17): https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

31. Oracle. (s.f.). *What is Java?*. [On-line]. Disponible en (11/10/17): https://www.java.com/es/download/faq/whatis_java.xml

32. Python. (s.f.). [On-line]. Disponible en (11/10/17): https://www.python.org/

33. Python. (s.f.). En Wikipedia. On-line]. Disponible en (11/10/17): https://es.wikipedia.org/wiki/Python

34. Comercio Electrónico. (s.f.). En Wikipedia. En Wikipedia. On-line]. Disponible en (12/10/17): https://es.wikipedia.org/wiki/Comercio_electr%C3%B3nico

35. Lane, A. (s.f.). *Los 5 tipos de comercio electrónico*. Disponible en (12/10/17): https://es.shopify.com/blog/12621205-los-5-tipos-de-comercio-electronico

36. Trescinski, N., Van Roey, G. (Noviembre, 25 de 2014). *Data Science for e-commerce.* [On-line]. Disponible en (16/10/17): https://es.slideshare.net/infofarm/data-science-for-ecommerce

37. Rodríguez, R. (Julio, 29 de 2016). *Cómo mejorar la conversión de tu eCommerce con Big Data Marketing.* [On-line]. Disponible en (16/10/17): https://www.iebschool.com/blog/big-data-tienda-online-e-commerce/

*38.* Cortizo, J. (s.f.). *Big Data, un "must" para el éxito en ECommerce*. [On-line]. Disponible en (16/10/17): https://www.brainsins.com/es/blog/big-data-exito-ecommerce/108597

*39.* Virmani, A. (Febrero, 23 de 2017). *How Big Data is Transforming Retail Industry.* [On-line]. Disponible en (16/10/17): https://www.simplilearn.com/big-data-transforming-retail-industry-article

40. Jones, M.T.( Diciembre, 12 de 2013). *Introduction to approaches and algorithms.* [On-line]. Disponible en (16/10/17): https://www.ibm.com/developerworks/library/os-recommender1/

41. Isinkaye, F., Folajimi, Y., Ojokoh, B. (Agosto, 20 de 2015). *Recommendation systems: Principles, methods and evaluation.* Egyptian Informatics Journal. [PDF]. Disponible en (16/10/17): http://www.sciencedirect.com/science/article/pii/S1110866515000341

42. Gonzáles, A. (Septiembre, 19 de 2014). *Sistemas de recomendación de contenido con Machine Learning.* [On-line]. Disponible en (16/10/17): http://cleverdata.io/sistemas-recomendacion-machine-learning/

43. Hristakeva, M. (Noviembre, 16 de 2015). *A practical guide to building recommender systems.* [On-line]. Disponible en (16/10/17): https://buildingrecommenders.wordpress.com/2015/11/16/overview-of-recommender-algorithms-part-1/

44. Cardona, L. (s.f.). *¿Qué es el email retargeting?.* [On-line]. Disponible en (16/10/17): https://www.cyberclick.es/numerical-blog/que-es-el-email-retargeting

45. Web scraping. (s.f.). En Wikipedia. [On-line]. Disponible en (16/10/17): https://es.wikipedia.org/wiki/Web_scraping

46. Procesamiento de lenguajes naturales. (s.f.). En Wikipedia. [On-line]. Disponible en (16/10/17): https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales

47. Apache Hadoop. (Octubre, 2017). *What is Apache Hadoop?.* En Wikipedia. [On-line]. Disponible en (17/10/17): http://hadoop.apache.org/

48. Hortonworks. (s.f.). *Apache Hadoop*. [On-line]. Disponible en (17/10/17): https://hortonworks.com/apache/hadoop/

49. MapR. (s.f.). *Hadoop & Big Data*. [On-line]. Disponible en (17/10/17): https://mapr.com/products/apache-hadoop/

50. Cloudera. (s.f.). *Apache Hadoop Ecosystem.* [On-line]. Disponible en (17/10/17): https://www.cloudera.com/products/open-source/apache-hadoop.html

51. MongoDB. (s.f.). *Hadoop and MongoDB.* [On-line]. Disponible en (17/10/17): https://www.mongodb.com/hadoop-and-mongodb

52. Hadoop Common. (s.f.). [On-line]. [On-line]. Disponible en (17/10/17): https://www.techopedia.com/definition/30427/hadoop-common

53. Apache Hadoop. (s.f.). *HDFS Architecture.* [On-line]. Disponible en (17/10/17): https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

54. Apache Hadoop. (s.f.). *MapReduce Tutorial*. [On-line]. Disponible en (19/10/17): https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

55. Hortonworks. (s.f.). *Apache Hadoop MapReduce.* [On-line]. Disponible en (19/10/17): https://es.hortonworks.com/apache/mapreduce/

56. MapReduce. (s.f.). En Wikipedia. [On-line]. Disponible en (19/10/17): https://es.wikipedia.org/wiki/MapReduce

57. HortonWorks. (s.f.). *Apache Hadoop YARN.* [On-line]. Disponible en (30/10/17): https://es.hortonworks.com/apache/yarn/

58. Apache Hadoop. (s.f.). *Apache Hadoop YARN.* [On-line]. Disponible en (30/10/17): https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

59. Hortonworks. (s.f.). *Apache Hadoop YARN - Concepts and Applications.* [On-line]. Disponible en (30/10/17): https://es.hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/

60. Hortonworks. (s.f.). *Apache Accumulo*. [On-line]. Disponible en (30/10/17): https://es.hortonworks.com/apache/accumulo/

61. Apache Accumulo. (s.f.). *User Manual*. [On-line]. Disponible en (30/10/17): https://accumulo.apache.org/1.7/accumulo_user_manual

62. Apache HBase. (s.f.). HBase. [On-line]. Disponible en (31/10/17): https://hbase.apache.org/index.html

63. Apache HBase. (s.f.). *HBase: Reference Guide*. [On-line]. Disponible en (31/10/17): https://hbase.apache.org/book.html

64. MapR. (s.f.). *An In-Depth Look at the HBase Architecture.* [On-line]. Disponible en (31/10/17): https://mapr.com/blog/in-depth-look-hbase-architecture/

65. Hortonworks. (s.f.). *Apache HBase.* [On-line]. Disponible en (31/10/17): https://es.hortonworks.com/apache/hbase/

66. Leverenz, L. (Octubre, 16 de 2017). *Apache Hive.* [On-line]. Disponible en (31/10/17): https://cwiki.apache.org/confluence/display/Hive/Home

67. Leverenz, L. (Junio, 15 de 2017). *Lenguage Manual*. [On-line]. Disponible en (31/10/17): https://cwiki.apache.org/confluence/display/Hive/LanguageManual

68. Leverenz, L. (Noviembre, 08 de 2015). *Hive Design.* [On-line]. Disponible en (31/10/17): https://cwiki.apache.org/confluence/display/Hive/Design

69. Apache Spark. (s.f.). *Spark Overview.* [On-line]. Disponible en (01/11/17): https://spark.apache.org/docs/latest/

70. Apache Spark. (s.f.). [On-line]. Disponible en (01/11/17): http://spark.apache.org/

71. Pérez, M. (s.f.). *Apache Spark: qué es y cómo funciona.* [On-line]. Disponible en (01/11/17): https://geekytheory.com/apache-spark-que-es-y-como-funciona/

72. Spark Architecture. (s.f.). [On-line]. Disponible en (01/11/17): https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-architecture.html

73. Niño, M. (s.f.). *Claves principales de los "Resilient Distributed Datasets" (RDD) de Apache Spark.* [On-line]. Disponible en (01/11/17): http://www.mikelnino.com/2016/03/claves-principales-resilient-distributed-datasets-rdd-apache-spark.html

74. Apache Storm. (s.f.). [On-line]. Disponible en (01/11/17): http://storm.apache.org/index.html

75. Apache Storm. (s.f.). [On-line]. Disponible en (01/11/17): http://storm.apache.org/releases/2.0.0-SNAPSHOT/Concepts.html

76. Hortonworks. (s.f). *Apache Storm.* [On-line]. Disponible en (01/11/17): https://hortonworks.com/apache/storm/

77. Apache Pig. (Junio, 2017). [On-line]. Disponible en (01/11/17): https://pig.apache.org/

78. Apache Pig. (Junio, 2017). *Pig Setup*. [On-line]. Disponible en (01/11/17): http://pig.apache.org/docs/r0.17.0/start.html

79. Hortonworks. (s.f.). *Apache Pig.* [On-line]. Disponible en (01/11/17): https://es.hortonworks.com/apache/pig/

80. Apache Pig Architecture. (s.f.). [On-line]. Disponible en (01/11/17): http://www.w3ii.com/en-US/apache_pig/apache_pig_architecture.html

81. Apache Mohout. (s.f.). [On-line]. Disponible en (02/11/17): http://mahout.apache.org/

82. Hortonworks. (s.f.). *Apache Mahout*. [On-line]. Disponible en (02/11/17): https://es.hortonworks.com/apache/mahout/

83. Ingersoll, G. (Septiembre, 8 de 2009). *Introducing Apache Mahout.* [On-line]. Disponible en (02/11/17): https://www.ibm.com/developerworks/java/library/j-mahout/

84. Apache Pheonix. (s.f.). [On-line]. Disponible en (02/11/17): https://phoenix.apache.org/index.html

85. Hortonworks. (s.f.). *Apache Pheonix*. [On-line]. Disponible en (02/11/17): https://es.hortonworks.com/apache/phoenix/

86. Apache Drill. (s.f.). [On-line]. Disponible en (02/11/17): https://drill.apache.org/

87. Apache Drill. (s.f.). *Architecture*. [On-line]. Disponible en (02/11/17): https://drill.apache.org/architecture/

88. Apache Falcon. (s.f.). [On-line]. Disponible en (03/11/17): https://falcon.apache.org/

89. Hortonworks. (s.f.). *Apache Falcon.* [On-line]. Disponible en (03/11/17): https://es.hortonworks.com/apache/falcon/

90. Apache Sqoop. (s.f.). [On-line]. Disponible en (03/11/17): http://sqoop.apache.org/

91. Hortonworks. (s.f.). *Apache Sqoop.* [On-line]. Disponible en (03/11/17): https://es.hortonworks.com/apache/sqoop/

92. Sqoop Architecture. (s.f.). [On-line]. Disponible en (03/11/17): https://www.hdfstutorial.com/sqoop-architecture/

93. Apache Know. (s.f.). [On-line]. Disponible en (03/11/17): https://knox.apache.org/

94. Hortonworks. (s.f.). *Apache Knox Gateway.* [On-line]. Disponible en (03/11/17): https://es.hortonworks.com/apache/knox-gateway/

95. IBM. (s.f.). *Descripción general de la pasarela de Apache Knox.* [On-line]. Disponible en (03/11/17):

https://www.ibm.com/support/knowledgecenter/es/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/knox_overview.html

96. Apache Ranger. (s.f.). [On-line]. Disponible en (03/11/17): https://ranger.apache.org/

97. Hortonworks. (s.f.). *Apache Ranger.* [On-line]. Disponible en (03/11/17): https://es.hortonworks.com/apache/ranger/

98. Apache Amabri. (s.f.). [On-line]. Disponible en (04/11/17): https://ambari.apache.org/

99. Hortonworks. (s.f.). *Apache Amabri.* [On-line]. Disponible en (04/11/17): https://es.hortonworks.com/apache/ambari/

100. Hortonworks. (s.f.). *Amabri Architecture.* [On-line]. Disponible en (04/11/17): https://docs.hortonworks.com/HDPDocuments/Ambari-2.5.2.0/bk_ambari-operations/content/architecture.html

101. Apache Oozie. (s.f.). [On-line]. Disponible en (04/11/17): http://oozie.apache.org/

102. Hortonworks. (s.f.). *Apache Oozie.* [On-line]. Disponible en (04/11/17): https://hortonworks.com/apache/oozie/

103. Apache Zookeeper. (s.f.). [On-line]. Disponible en (04/11/17): https://zookeeper.apache.org/

104. Hortonworks. (s.f.). *Apache Zookeeper.* [On-line]. Disponible en (04/11/17): https://es.hortonworks.com/apache/zookeeper/

105. Zookeeper - Fundamentals. (s.f.). [On-line]. Disponible en (04/11/17): https://www.tutorialspoint.com/zookeeper/zookeeper_fundamentals.htm

106. Hortonworks. (s.f.). [On-line]. Disponible en (06/11/17): https://es.hortonworks.com/

107. Hortonworks. (s.f.). *Hortonworks Data Plataforms.* [On-line]. Disponible en (06/11/17): https://es.hortonworks.com/products/data-platforms/hdp/

108. Cloudera. (s.f.). *CDH Overview.* [On-line]. Disponible en (06/11/17): https://www.cloudera.com/documentation/enterprise/latest/topics/cdh_intro.html

109. Cloudera. (s.f.). *Apache Hadoop Ecosystem.* [On-line]. Disponible en (06/11/17): https://www.cloudera.com/products/open-source/apache-hadoop.html

110. MapR. (s.f.). [On-line]. Disponible en (06/11/17): https://mapr.com/

111. MapR. (s.f.). *Plataform*. [On-line]. Disponible en (06/11/17): https://maprdocs.mapr.com/home/MapROverview/c_overview_intro.html

112. MapR. (s.f.). *MapR-FS*. [On-line]. Disponible en (06/11/17): https://maprdocs.mapr.com/home/MapROverview/c_maprfs.html

113. MapR. (s.f.). *MapR Control System*. [On-line]. Disponible en (06/11/17): https://maprdocs.mapr.com/52/MapROverview/c_mcs.html

114. IBM. (Julio, 2015). *Foundational Methodology for Data Science*. [On-line-PDF]. Disponible en (07/11/17): http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IMW14824USEN

115. Cross Industry Standard Process for Data Mining. (s.f.). En Wikipedia. [On-line]. Disponible en (07/11/17): https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining

116. Goicochea, A. (Agosto, 11 de 2009). *CRISP-DM, Una metodología para proyectos de Minería de Datos*. [On-line]. Disponible en (07/11/17): https://anibalgoicochea.com/2009/08/11/crisp-dm-una-metodologia-para-proyectos-de-mineria-de-datos/

117. Saluja, C. *Collaborative Filtering based Recommendation Systems exemplified.* [On-line]. Disponible en (03/09/2018): https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1

118. Sharma, P. (Junio, 21 de 2018). *Comprehensive Guide to build a Recommendation Engine from scratch (in Python)*. [On-line]. Disponible en (03/09/2018): https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/

119. Pandas (*s.f.). pandas.* [On-line]. Disponible en (Septiembre 10, 2018): https://pandas.pydata.org/

120. Numpy (*s.f.*). *Numpy*. [On-line]. Disponible en (Septiembre, 10 de 2018): http://www.numpy.org/

121. Scikit-Learn (*s.f.*). *Scikit-learn*. [On-Line]. Disponible en (Septiembre, 10 de 2018): http://scikit-learn.org/stable/

122. Scipy (*s.f.*). *Scipy*. [On-Line]. Disponible en (Septiembre, 10 de 2018): https://www.scipy.org/

123. PyMongo (*s.f.*). PyMongo 3.7.1 Documentation. [On-Line]. Disponible en (Septiembre, 10 de 2018): https://api.mongodb.com/python/current/

124. Matplotlib (*s.f.)*. Matplotlib. [On-Line]. Disponible en (Septiembre, 10 de 2018): https://matplotlib.org/

125. Seaborn *(s.f.)*. seaborn: statistical data visualization. [On-Line]. Disponible en (Septiembre, 10 de 2018): https://seaborn.pydata.org/

126. PyMSQL (*s.f.*). [On-Line]. Disponible en (Septiembre, 10 de 2018): https://pymysql.readthedocs.io/en/latest/

127. Anaconda *(s.f.). Anaconda.* [On-Line]. Disponible en (Septiembre, 10 de 2018): https://www.anaconda.com/

128. IPython (*s.f.*). *The Jupyter Notebook.* [On-Line]. Disponible en (Septiembre, 10 de 2018): http://ipython.org/notebook.html

129. MongoDB (*s.f.*). *MongoDB*. . [On-Line]. Disponible en (Septiembre, 10 de 2018): https://www.mongodb.com/

130. Billsus D, Pazzani MJ. *User modeling for adaptive news access. User Model User-adapted Interact*. Septiembre, 2000. (2–3):147–80.4

131. Mooney RJ, Roy L*. Content-based book recommending using learning for text categorization. En: Proceedings of the fifth ACM conference on digital libraries*. ACM; Año: 2000. p. 195–204.

132. Shardanand U, Maes P. *Social information filtering: algorithmsfor automating "word of mouth". En: Proceedings of the SIGCHIconference on human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co.; Año: 1995. p. 210–7.

133. Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J. *Applying collaborative filtering to usenet news*. Commun ACM Año: 1997;40(3):77–87.

134. Loeb, W. Who Are The Top 10 U.S. Online Retailers?. Forbes. [On-line]. Disponible en (Febrero, 19 de 2019): https://www.forbes.com/sites/walterloeb/2018/08/06/who-are-the-top-10-u-s-online-retailers/

135. Linden, F; Smith, B; York, J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. [On-line]. Disponible en (Febrero, 19 de 2019: https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf

136. Netflix. Netflix Prize. [On-line]. Disponible en (Febrero, 19 de 2019): https://www.netflixprize.com/rules.html

137. Ricci, F; Rokach, L; Shapira, B; Kantor, P. B. (Octubre, 2010) *Recommender Systems Handbook*. Estados Unicos. Springer.