

Guatemala 31 de Marzo de 2020
Algoritmos y Estructuras de Datos
Facultad de Ingeniería
Docente: Douglas



Angel David Cuellar Bautista - 18382
Augusto Alonso Ascencio - 181085
Pablo Alejandro Méndez Morales - 19195

TEMA: Recomendación de productos

Definición de problema:

A menudo cuando vamos al supermercado para comprar productos nos encontramos que al final siempre compramos lo que no necesitamos, gastamos demás, se nos olvidan productos y no terminamos por sentirnos satisfechos. Esto es debido a que nuestra memoria es muy frágil y nuestro cerebro es muy vulnerable a la publicidad del supermercado y la disposición de los pasillos, además de otras maniobras de marketing.

Por eso se considera necesario un sistema con el cual el usuario recibe recomendaciones de productos para su hogar según sus verdaderas necesidad y lo que busca sin hacer que se vea perjudicado alcanzando su mejor satisfacción. Con este sistema además prevemos futuras necesidades que el usuario pueda tener y tratar de solucionarlas de la mejor manera.

Usuarios:

1. Personas encargadas de comprar los productos para el hogar:

El enfoque principal en primera instancia es para aquellas personas que en el hogar a menudo son las que se encargan de ir a comprar los productos que son necesarios para el día a día, debido a que son estas personas las que más expuestas a los supermercados y las más importantes puesto que deben de llenar todas las necesidades de un hogar

2. Cualquier persona

En algún punto de nuestra vida hemos encontrado algún producto que no sabíamos que necesitábamos hasta que lo encontramos esto puede ser desde un simple quita grapas, una tijera para zurdos hasta una navaja multifuncional. Hay infinidad de gustos y necesidades como personas lo que provoca que haya una necesidad de saber que productos venderle a cada tipo de persona.

Entrevistas

Algunas preguntas que se realizaron:

1. ¿Usted cuando va al supermercado se siente satisfecho con su compra?
2. ¿Considera que usted siempre se lleva a su hogar los mejores productos?
3. ¿Usted compra su producto basado en su experiencia o en recomendación?
4. ¿Considera que la publicidad influye en su compra?
5. ¿Ha comprado alguna vez por internet? En caso afirmativo ¿Cómo fue su experiencia en cuanto a recomendaciones?

Los audios de las entrevistas se pueden encontrar en el siguiente link:

https://drive.google.com/drive/folders/1050oVO28N5bdpb-_39ugnqOgDSypJe2v?usp=sharing

Algoritmos existentes de recomendación

1) Filtrado colaborativo:

Se caracteriza porque se utiliza la información de los mismos usuarios que utilizan el algoritmo para producir recomendaciones para otros. Cosa que parece bastante intuitiva puesto que a la gran mayoría les interesa lo mismo. Este sistema está basado en memoria y en modelo, este último lo que significa es que genera o aprende mediante un modelo mientras que en memoria se utilizan los datos en tiempo real. Este tipo de filtrado se puede dividir en dos básicamente:

a) Vecinos próximos:

Este método selecciona un número k de vecinos similares, de tal manera que con estos usuarios que se parecen bastante se trata de hacer una predicción de rating. A partir de esta predicción podemos sacar el rating de ítems a recomendar. Para seleccionar a los vecinos hay un mínimo de similitud a considerar para que clasifique como vecino. Este mismo algoritmo se ha desarrollado por usuario y por ítem (Na, 2019).

Como dato interesante en el 2008 se demostró que los modelos de factorización matrices son superiores a esto en términos de la predicción del error.

2) Basados en contenido:

Estos algoritmos se basan en la descripción de cada elemento que queremos recomendar sin utilizar la información de otros usuarios para recomendar al usuario que la está usando en ese preciso momento. Por ejemplo para recomendar publicaciones en facebook podemos considerar las etiquetas, fotos, personas, etc como el contenido del ítem objetivo y el contenido de las publicaciones hechas por el usuario como base para inferir gustos de este. Este algoritmo a su vez se divide en dos modelos:

a) Rocchio:

Se basa en el cálculo de centroides para cada usuario donde cada vector representa a un usuario. Por ejemplo las etiquetas del contenido de las publicaciones serían vectores que representan al usuario con la relación de las etiquetas (Na, 2019).

b) Item kNN:

Se basa en el mismo principio que el basado en usuarios pero este se diferencia en la forma de calcular la similitud entre ítems. Mientras que el

basado en usuarios utiliza ratings de otros usuarios este utiliza la descripción de los ítems (Na, 2019).

3) Algoritmos no personalizados

Estos recomiendan ítems a los usuarios sin conocer ningún dato del usuario. Estos se utilizan para dar un punto de referencia estándar a las métricas. Los más conocidos son:

a) Popularidad:

Recomienda ítems por la popularidad en el sistema, da a cada participante del programa el mismo rating. Por popularidad de ítem se refiere a cuántos usuarios han interactuado con ese ítem. Por ejemplo esto se puede ver en las grandes industrias como el cine donde a todos les gusta las malas películas de Marvel pero como les gusta a todos entonces eso es lo que hacen, también ocurre con la música urbana y youtube (Na, 2019).

b) Random:

Se recomiendan los ítems de manera aleatoria y su precisión se relaciona con la densidad del conjunto de datos (Na, 2019).

4) Híbridos

Estos algoritmos utilizan información tanto de usuarios como de los ítems así mismo como la interacción de las cosas.

Base de datos basada en grafos Neo4j

El problema con bases de datos convencionales como SQL y OracleDataBase es buscar las relaciones entre diferentes datos. Se suele tener que realizar operaciones costosas en recursos computacionales para la búsqueda de un dato interconectado. Bases de datos normales no son eficientes para relacionar información (Cña, 2015).

Neo4j es una base de datos de grafos. Esto quiere decir que a diferencia de SQL y OracleDataBase, la información no es guardada dentro de una tabla. Por el contrario, es guardada a través de grafos.. En otras palabras, Neo4j guarda cada dato junto a los punteros que hacen referencias a los nodos a los que se encuentra conectado (Neo4j, s.f.). Esto quiere decir que la búsqueda de información relacionada es mucho más rápida que una base de datos convencional

Neo4j se utiliza en base de datos de grandes empresas. Incluye, pero no se limita a:

- Comcast
- Walmart
- Adobe

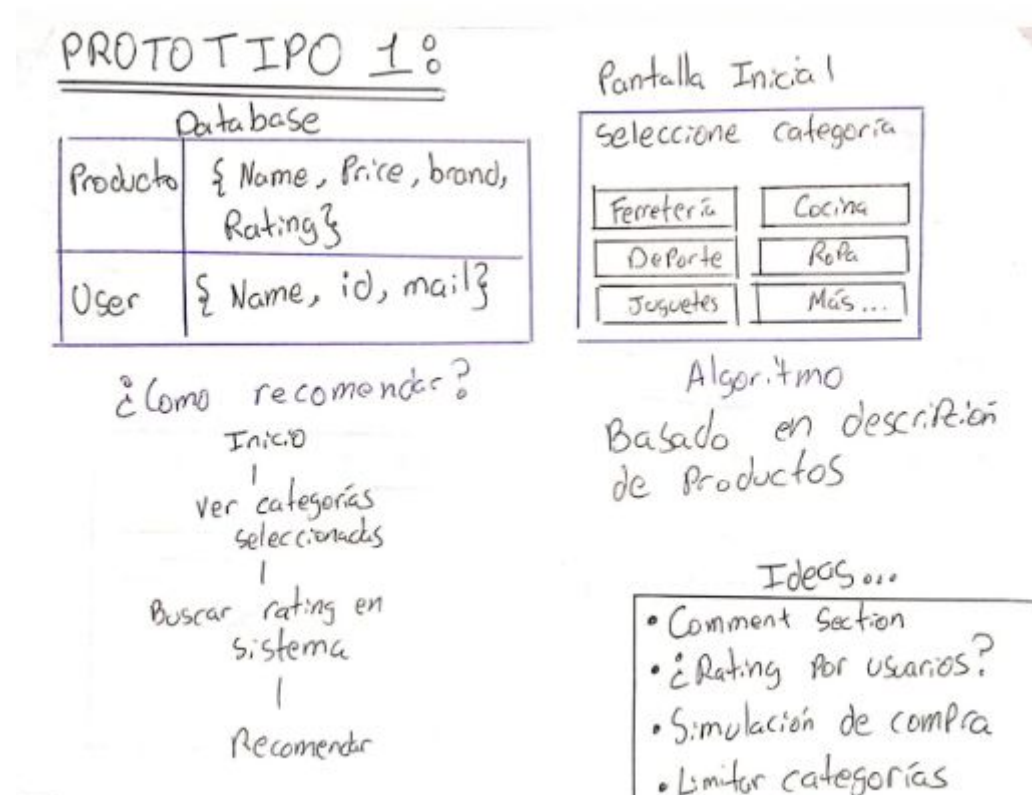
- Ebay
- Lyft
- IBM

Neo4j lanzó su primera versión en febrero del 2020. Inicialmente, solo funcionaba en programas hechos en Java. Más, adelante, se sacó una versión para Python, Ruby, PHP, C#, JavaScript y Go (Cía, 2015).

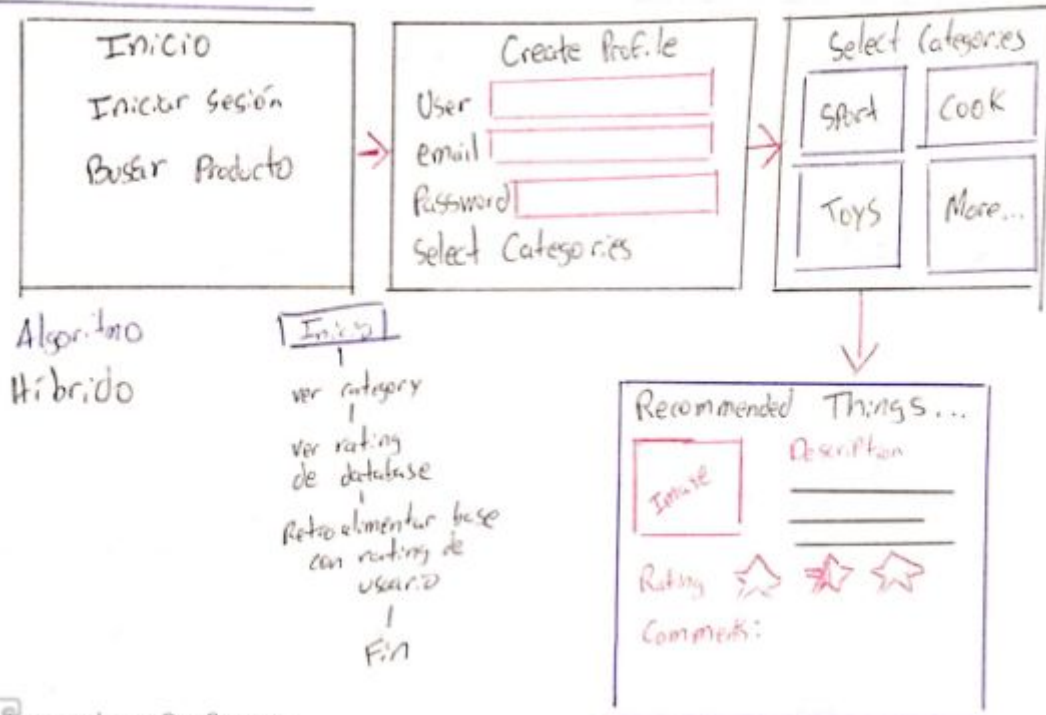
Actualmente hay tres diferentes versiones de Neo4j: *Community*, *Enterprise* y *Aura*.

1. **Community Edition:** La edición más simple de Neo4j. Está dirigida a pequeños proyectos personales y para aprender como utilizar la base datos fuera del estrés de un área profesional (Neo4j, s.f.).
2. **Enterprise Edition:** Cuenta con todas las características de Community Edition. diferencia de Aurora y Community, está hecha para empresas que tienen la infraestructura necesaria para tener una base de datos completamente independiente. Debido a esto tiene características únicas como un tamaño ilimitado para cada grafo, cantidad ilimitada de base de datos y diferentes opciones de seguridad como Kerberos (Neo4j, s.f.).
3. **Aura Edition:** Una base de datos simplificada y completa. A diferencia de las otras dos ediciones, está totalmente automatizada. El manejo de infraestructura es hecho por ingenieros de Neo4j. Aunque esto permite concentrarse en otras partes del proyecto, proporciona una menor cantidad de control de la base de datos (Neo4j, s.f.).

Prototipos de baja fidelidad



PROTO TIPO 2º



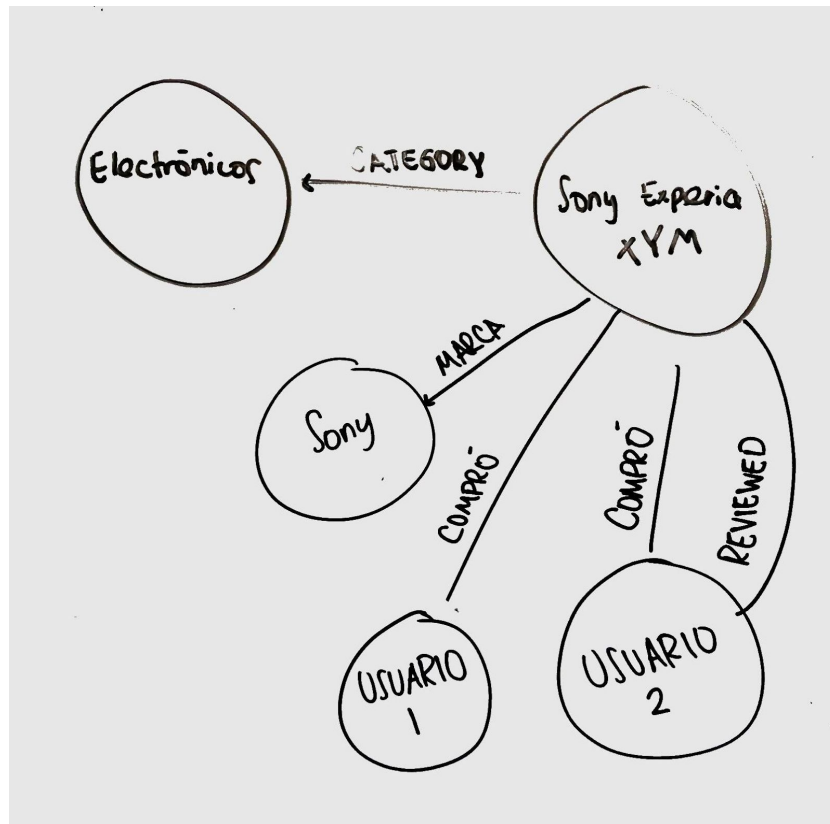
Pseudocódigo del algoritmo de recomendación

1. Inicio
2. Pedirle al usuario que inicie sesión
3. Buscar en la base de datos al usuario
4. Darle al usuario la opción de buscar o salir
 - a. Si desea filtrar
 - i. Preguntarle cómo desea filtrar los datos, ej. por ratings, por marca, por categoría, etc..
 - ii. Buscar en la base de datos usando el filtro
 - iii. Imprimir los resultados en un formato agradable
 - iv. Volver al paso 4
 - b. Si desea Salir
 - i. Se termina el programa

Diseño y explicación de la base de datos inicial

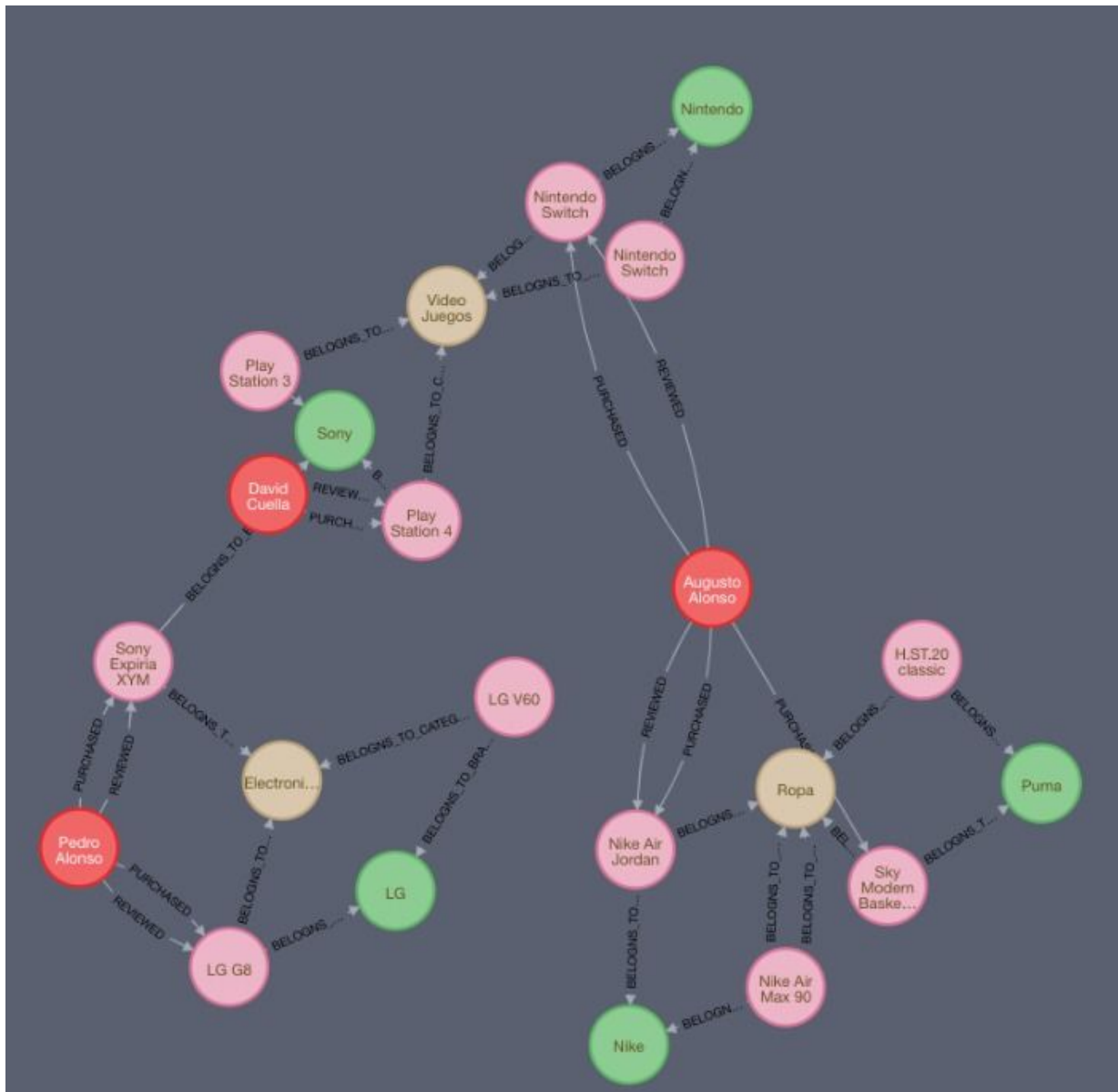
La base de datos está compuesta de una serie de productos que tienen relación con unos nodos de tipo Categoría. Asimismo, tiene una relación con las marcas, los usuarios que han comprado el producto y los usuarios que han dado una calificación.

Ejemplo:



Cabe mencionar que entre los nodos producto cuentan con el nombre, id y precio de si mismos; mientras que los usuarios cuentan con su nombre, correo y contraseña.

La base de datos inicia con usuarios *mockup* y con algunos productos como la Nintendo Switch, PS3 y NikeAirJordan.



Prototipos programados y bases de datos

El link del repositorio es: <https://github.com/aeaa1998/neo4j-project.git>

```
CREATE (Augusto:User {id: 1, name: 'Augusto Alonso', email: 'a@a.com', password: 'secret'})
CREATE (PedroAlonso:User {id: 2, name: 'Pedro Alonso', email: 'p@a.com', password: 'secret'})
CREATE (Margarita:User {id: 3, name: 'Margarita Ascencio', email: 'm@a.com', password: 'secret'})
CREATE (David:User {id: 4, name: 'David Cuella', email: 'd@c.com', password: 'secret'})
CREATE (PedroJimenez:User {id: 5, name: 'Pedro Jimenez', email: 'p@j.com', password: 'secret'})
CREATE (JulianGabriel:User {id: 6, name: 'Julian Gabriel', email: 'j@g.com', password: 'secret'})
```



```
CREATE (NikeAirJordan:Product {id: 1, name: 'Nike Air Jordan', price: 800})
CREATE (AirMax90:Product {id: 2, name: 'Nike Air Max 90', price: 9000})
CREATE (SkyModern:Product {id: 3, name: 'Sky Modern Basketball', price: 1000})
CREATE (HST20:Product {id: 4, name: 'H.ST.20 classic', price: 3200})
CREATE (NintendoSwitch:Product {id: 5, name: 'Nintendo Switch', price: 3400})
CREATE (NintendoSwitchLite:Product {id: 6, name: 'Nintendo Switch Lite', price: 2400})
CREATE (PS4:Product {id: 7, name: 'Play Station 4', price: 3000})
CREATE (PS3:Product {id: 8, name: 'Play Station 3', price: 2000})
CREATE (SonyXperia:Product {id: 9, name: 'Sony Expiria XYM', price: 900})
CREATE (LGV60:Product {id: 10, name: 'LG V60', price: 1700})
CREATE (LGG8:Product {id: 11, name: 'LG G8', price: 1900})
```

Referencias

Pepa S. (2014). *Suite de algoritmos de recomendación en aplicaciones reales*. Universidad autónoma de Madrid Escuela Politécnica superior, Madrid.

Cia, F. (2015). *Neo4j: What a graph database is and what it is used for*. Recuperado de <https://bbvaopen4u.com/en/actualidad/neo4j-what-graph-database-and-what-it-used>

Neo4j, (s.f.) *INTRODUCING Neo4j 4.j*. Recuperado de <https://neo4j.com/whats-new-in-neo4j/?ref=home-banner>

Na, J. (2019). *Sistemas de recomendación*. Recuperado de <https://www.aprendemachinelearning.com/sistemas-de-recomendacion/>