UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de ingeniería.

CC2016



**Proyecto 2. Sistema de recomendaciones**

Javier Alejandro Chávez León – 23132

Pavel Huerta

Catedrático: Pablo Alejandro Godoy Díaz

Algoritmos y estructuras de datos

Sección 30

9/5/2025

# Investigación

Los algoritmos de recomendación son sistemas con los que los usuarios interactúan con plataformas de comercio electrónico mayormente. Su capacidad para filtrar información masiva y predecir preferencias individuales los convierte en herramientas indispensables para mejorar la experiencia del usuario y optimizar decisiones. Estos algoritmos funcionan a través de diferentes métodos, como el filtrado colaborativo, en el cual se toman en cuenta las valoraciones y preferencias de otros usuarios con gustos similares, y el filtrado basado en contenido, en el cual se analizan las características y descripciones de los artículos o servicios para hacer recomendaciones. Estos algoritmos se han vuelto cada vez más comunes en plataformas como Netflix, Amazon y Spotify, mejorando la experiencia del usuario y aumentando las ventas de productos o servicios.

Existen distintos Tipos de Algoritmos de Recomendación:

1.Algoritmo Basado en Contenidos  
 Estos Algoritmos Recomiendan ítems similares a los que el usuario ha interactuado previamente, utilizando características intrínsecas de los ítems.

Ventajas:

* Efectivos para nichos específicos (ej: música indie, productos técnicos).
* Evitan el problema del cold start para ítems nuevos, ya que no dependen de interacciones previas.
* Transparencia: el usuario entiende por qué se recomienda un ítem (ej: "basado en tus preferencias por X").

Limitaciones:

* Poca diversidad: Tienden a recomendar ítems muy similares, limitando la exploración.
* Dependen de metadatos de calidad, lo que requiere un etiquetado exhaustivo.
* Ignoran el comportamiento colectivo de otros usuarios.

2. Filtrado Colaborativo

Este Algoritmo utiliza patrones de interacción de usuarios (valoraciones, compras, clicks) para predecir preferencias. Se puede Basar en los usuarios de este modo encuentra usuarios similares y recomienda ítems que ellos han preferido. También se puede Basar en ítems, de esta forma recomienda ítems similares con los que el usuario ya ha interactuado.

Ventajas:

* Captura preferencias implícitas (ej: "usuarios como tú también compraron esto").
* Funciona bien en dominios con alta densidad de datos (ej: plataformas de streaming).
* No requiere metadatos detallados de los ítems.

Limitaciones:

* Cold start: No funciona con usuarios o ítems nuevos (sin historial).
* Sparsity: En sistemas con millones de ítems, la matriz de interacciones suele ser dispersa.
* Riesgo de sobre especialización: repite recomendaciones obvias.

3. Sistemas Híbridos

Combina técnicas como filtrado colaborativo y basado en contenido para mitigar sus limitaciones. Un ejemplo podría ser: Netflix, usa colaborativo para tendencias y basado en contenido para géneros.

Ventajas:

* Mayor cobertura (funciona con usuarios nuevos y existentes).
* Equilibra personalización y diversidad.
* Reduce el impacto del cold start y la sparsity.

Limitaciones:

* Complejidad de implementación: Integrar múltiples algoritmos requiere ajuste fino.
* Mayor costo computacional.
* Dificultad para priorizar qué técnica usar en cada escenario.

4.Algoritmo Basado en Grafos

Modelan relaciones entre entidades (usuarios, ítems, características) como nodos y aristas en un grafo. Usa técnicas comunes como: PageRank, caminos más cortos, propagación de etiquetas.

Ventajas:

* Manejan relaciones complejas (ej: "usuarios que prefieren X también visitaron Y").
* Ideales para dominios con conexiones multicapa (ej: ubicaciones, redes sociales).
* Escalables en bases de datos como Neo4j.

Limitaciones:

* Requieren un diseño cuidadoso del grafo (estructura de nodos y relaciones).
* Consultas complejas pueden ser lentas en grafos muy grandes.
* Menos intuitivos para usuarios sin experiencia en teoría de grafos.

5. Sistemas Basados en Conocimiento

Usan reglas predefinidas o ontologías para generar recomendaciones (ej: "si un usuario busca laptops, priorizar marcas con buen soporte técnico").

Ventajas:

* Útiles en dominios con requisitos técnicos (ej: medicina, ingeniería).
* Evitan sesgos de popularidad (no dependen de interacciones masivas).
* Explicabilidad clara ("recomendado por la regla Z").

Limitaciones:

* Rigidez: Las reglas deben actualizarse manualmente ante cambios en el dominio.
* No se adaptan a preferencias dinámicas de los usuarios.
* Limitados a escenarios con reglas bien definidas.

# Explicación Tema Proyecto

El proyecto consiste en desarrollar un sistema de recomendaciones, en nuestro caso nuestro tema serán automóviles utilizando una base de datos basada en grafos. El objetivo es conectar las preferencias de los usuarios con las características de los vehículos para generar recomendaciones personalizadas.

Viéndolo desde el punto de Vista de un Grafo Nuestros Nodos podrían ser los siguientes:

* Automóvil: Modelos específicos (ej. Toyota Corolla, Ford Mustang).
* Marca: Fabricantes (ej. Toyota, Ford, Tesla).
* Característica: Atributos técnicos o funcionales (ej. "Bajo consumo", "Alta seguridad", "Eléctrico").
* Tipo de Vehículo: Categorías (ej. Sedán, SUV, Deportivo).
* Usuario: Perfiles de usuarios con preferencias.
* Concesionario: Ubicaciones o puntos de venta.
* Ubicación: Ciudades o regiones (ej. Ciudad de Guatemala, Quetzaltenango).

Nuestro concepto de Nodos o entidades representaran las entidades Fundamentales para el sistema de nuestro proyecto. En el contexto de la recomendación, son los elementos clave que interactúan entre sí.

# Design Thinking

Definición de problema:

Hay demasiadas opciones de carros nuevos y usados para comprar, y puede ser difícil tomar una decisión sobre cuál es la mejor opción, en especial para personas que van a comprar su primer carro o no saben mucho sobre mecánica y automovilismo.

Propuesta de solución:

Desarrollar un sistema que recomiende carros a los usuarios dependiendo de sus gustos y necesidades, utilizando bases de datos con grafos y algoritmos de recomendación precisos y confiables en los que mientras más información proporcione el usuario sobre sus actividades, obtendrá resultados más específicos.

Un sistema que recomiende opciones de carros según el presupuesto, gustos y necesidades de cada persona.

Guión para entrevistas:

* Cuántos años tienes?
* Tienes carro? (si/no, porqué)
* Si tienes carro, cómo fue tu experiencia al comprarlo? (fácil, difícil, confuso, etc.)
* Piensas comprar un carro en un futuro cercano?
* Qué características consideras más importantes en un carro y porqué?
* Tienes alguna marca o modelo favorito?
* Cuál es el carro de tus sueños y porqué?
* Crees que hay suficiente información accesible que te ayude a decidir qué carro comprar?
* Si existiera un sistema de recomendaciones para compra de carros, lo usarías o recomendarías a conocidos? Porqué?
* Qué factores hacen que te decidas por un carro u otro?
* Te importa más la estética de un carro o sus especificaciones?
* Qué tipo de vehículo de gusta más y porqué?
* Cuántas veces al día te transportas por vehículo propio o transporte público?
* Prefieres carros automáticos o mecánicos?
* Prefieres carros de gasolina, eléctricos o híbridos?

Mapa de empatía:



# Prototipo:

Dd

# 

# 

# pseudocódigo

Crear nodos para usuarios, autos, marcas, tipo de auto, tipo de combustible y tipo de transmisión.

Preguntar al usuario qué características le gustan en un carro, mostrar autos al usuario para que elija opciones que le gustan de acuerdo con las características que indicó, buscar qué tienen en común los autos que le gustan al usuario para recomendaciones más precisas, contar cuántas coincidencias hay entre los autos, ordenarlos de acuerdo con el número de coincidencias que tengan.

Relaciones de la base de datos:

Usuario -> auto (los autos que le gustan al usuario).

Usuario -> edad (la edad del usuario).

Usuario -> marca (Las marcas que más le gusten al usuario).

Usuario -> tipo (Los tipos de vehículos que más le gusten al usuario).

Usuario -> combustible (Qué tipo de combustible o fuente de energía prefiera el usuario en un carro).

Usuario -> Transmisión (qué tipo de transmisión prefiera el usuario).

Usuario -> presupuesto (El presupuesto que tiene el usuario).

Auto -> marca (de qué marca es el auto).

Auto -> tipo (qué tipo de carro es).

Auto -> combustible (qué tipo de combustible o fuente de energía usa).

Auto -> transmisión -> (si es automático o mecánico).

Auto -> Precio (El precio de un carro).

Precio – Presupuesto (Se recomiendan autos de precios parecidos al presupuesto).

# Repositorio de Github

Link: [JAVIERCLO/Proyecto\_Recomendaciones: Proyecto 2 de algoritmos y estructuras de datos sobre un sistema de recomendaciones de carros](https://github.com/JAVIERCLO/Proyecto_Recomendaciones)