

Algoritmos y Estructura de Datos
Proyecto 02 "Recommendation Engines"

Instrucciones:

- Utilice la metodología de Design Thinking para definir correctamente el problema que desea resolver por medio de este proyecto. Recuerde que las etapas principales son:
 - Empatía
 - Definición
 - Ideación
 - Prototipos
 - Testing
- Investigue sobre algoritmos de recomendación (recommendation engines1), especialmente los que utilizan bases de datos basados en grafos.
- Diseñe su algoritmo, basado en representar la información en grafos, para resolver el problema que usted identificó con las etapas del Design Thinking. Debe incluir también el diseño inicial de su base de datos.
- Suba el algoritmo a un repositorio de github, recuerde que debe haber varios commits de todos los miembros del equipo, durante todo el tiempo de disponibilidad de la tarea.

Link de repositorio GitHub:

<https://github.com/lfmendoza/recommendation-engine>

1. Algoritmos de recomendación:

Los algoritmos de recomendación, también conocidos como recommendation engines, son herramientas para el análisis de datos y recientemente con inteligencia artificial. Estos algoritmos se utilizan para predecir la preferencia o interés de un usuario en un objeto específico, como un producto, una película o una canción, y se utilizan ampliamente en plataformas de comercio electrónico, servicios de streaming de contenido, redes sociales y más.

Para este proyecto el sistema de recomendación es para la búsqueda de vehículos por lo que algoritmos de recomendación basados en grafos para este sistema pueden ser:

- **Filtrado colaborativo basado en grafos:** Este enfoque utiliza las conexiones entre usuarios y elementos para generar recomendaciones. Si dos usuarios tienen gustos similares y han interactuado con los mismos elementos de manera positiva en el pasado, es probable que un elemento que uno de los usuarios haya disfrutado también sea apreciado por el otro.

```
import networkx as nx

# Crear un grafo de ejemplo
G = nx.Graph()
G.add_edges_from([(1, 'A'), (1, 'B'), (2, 'A'), (2, 'C'), (3, 'B'), (3, 'C')])

# Definir usuarios y elementos
usuarios = {n for n, d in G.nodes(data=True) if d['bipartite'] == 0}
elementos = set(G) - usuarios

# Generar recomendaciones para un usuario dado
usuario = 1
elementos_recomendados = {}
for elemento in elementos:
    if elemento not in G[usuario]:
        suma_puntuaciones = sum(G[vecino][elemento]['weight'] for
vecino in G[usuario])
        elementos_recomendados[elemento] = suma_puntuaciones

print("Elementos recomendados para el usuario", usuario, ":",
elementos_recomendados)
```

- **Algoritmos de propagación de etiquetas (Label Propagation):** Estos algoritmos asignan etiquetas a los nodos del grafo y propagan esas etiquetas a través de las relaciones de vecindad en el grafo. Esto puede utilizarse para inferir la preferencia de un usuario sobre un elemento en función de la preferencia de otros usuarios con intereses similares.

```

import networkx as nx

# Crear un grafo de ejemplo
G = nx.Graph()
G.add_edges_from([(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)])

# Asignar etiquetas iniciales a los nodos
for nodo in G.nodes():
    G.nodes[nodo]['label'] = nodo

# Propagar etiquetas a través de las aristas
for i in range(10): # Iterar varias veces para la propagación
    for u, v in G.edges():
        if G.nodes[u]['label'] < G.nodes[v]['label']:
            G.nodes[v]['label'] = G.nodes[u]['label']

# Obtener las etiquetas finales de los nodos
etiquetas_finales = {nodo: G.nodes[nodo]['label'] for nodo in G.nodes()}
print("Etiquetas finales de los nodos:", etiquetas_finales)

```

- **Algoritmos de recomendación basados en caminos más cortos:** Estos algoritmos utilizan la longitud de los caminos más cortos en el grafo para determinar la similitud entre usuarios o elementos. Si dos usuarios están conectados por un camino corto en el grafo, es probable que tengan preferencias similares.

```

import networkx as nx

# Crear un grafo de ejemplo
G = nx.DiGraph()
G.add_edges_from([(1, 2), (1, 3), (2, 1), (3, 1), (3, 2)])

# Calcular el PageRank
pagerank = nx.pagerank(G)

# Obtener los nodos ordenados por su PageRank
nodos_ordenados = sorted(pagerank.items(), key=lambda x: x[1],
reverse=True)

```

```
print("Nodos ordenados por PageRank:", nodos_ordenados)
```

- **Algoritmos de recomendación basados en comunidades (Community-based recommendation):** Estos algoritmos identifican comunidades de usuarios con intereses similares en el grafo y generan recomendaciones en función de las preferencias de la comunidad a la que pertenece un usuario.

```
import networkx as nx

# Crear un grafo de ejemplo
G = nx.Graph()
G.add_edges_from([(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)])

# Calcular las caminatas aleatorias
caminatas = nx.random_walk.G
print("Caminatas aleatorias:", caminatas)
```

2. Etapas de Design Thinking:

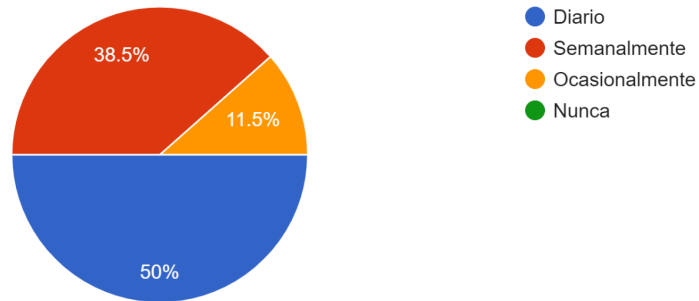
a. Entrevistas

Enlace de la documentación recabada en las entrevistas de los usuarios:
https://docs.google.com/forms/d/e/1FAIpQLSeHUAk-2Tlr_AJyerk-pt8YIFPONjwDF9USkHsS0sQY31nJTQ/viewform?usp=sharing

Resultados:

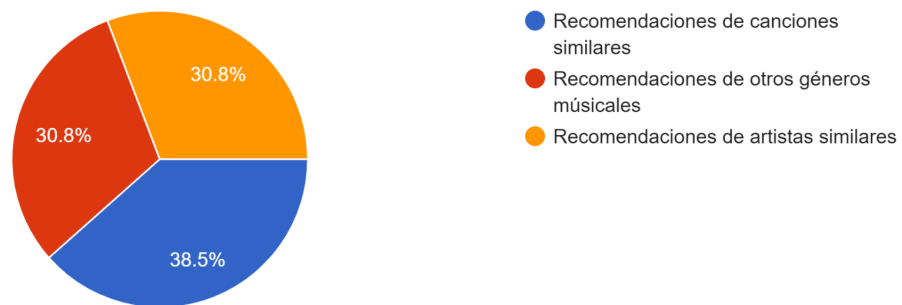
¿Con qué frecuencia utilizas servicios de streaming de música como Spotify?

26 respuestas



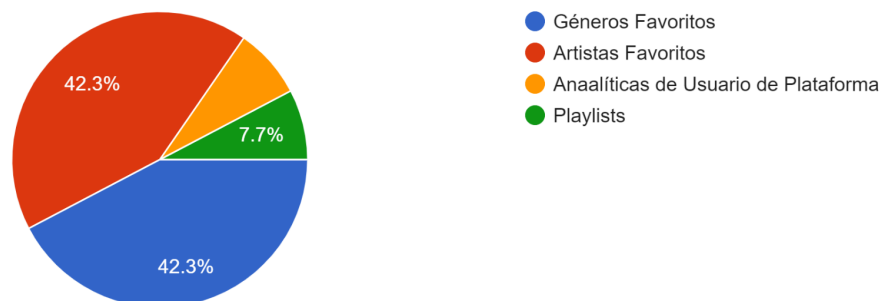
¿Qué características encuentras más útiles en las funciones de recomendación de música de estas plataformas?

26 respuestas



¿Qué tipo de información estarías dispuesto a compartir para mejorar las recomendaciones de música?

26 respuestas



¿Qué aspectos crees que podrían mejorarse en los sistemas de recomendación de música actuales?

26 respuestas

Mejora de aspecto aplicación

Personalización más precisa basada en el estado de ánimo del usuario.

Incorporación de algoritmos de aprendizaje profundo para entender mejor las preferencias del usuario.

Mayor diversidad en las recomendaciones para evitar la sobreexposición a un género o artista.

Integración de información contextual, como ubicación o actividad, para recomendaciones más relevantes.

Mejora en la capacidad de descubrimiento de música nueva y menos conocida.

Inclusión de opciones de retroalimentación más detalladas para refinar las recomendaciones.

Implementación de filtros avanzados para que los usuarios puedan ajustar sus preferencias con mayor precisión.

¿Qué aspectos crees que podrían mejorarse en los sistemas de recomendación de música actuales?

26 respuestas

Implementación de filtros avanzados para que los usuarios puedan ajustar sus preferencias con mayor precisión.

Incorporación de recomendaciones colaborativas, donde los usuarios puedan influir en las recomendaciones de otros.

Integración de funciones sociales para compartir y descubrir música con amigos.

Mejora en la detección de gustos cambiantes del usuario con el tiempo.

Mayor transparencia en los algoritmos utilizados para generar recomendaciones.

Personalización basada en el momento del día para adaptarse a los diferentes estados de ánimo.

Inclusión de recomendaciones basadas en la historia de escucha del usuario.

Incorporación de la opción de crear listas de reproducción automáticas según el estado de ánimo o la actividad.

¿Qué aspectos crees que podrían mejorarse en los sistemas de recomendación de música actuales?

26 respuestas

Incorporación de recomendaciones de conciertos y eventos musicales en función de las preferencias del usuario.

Personalización basada en la estacionalidad y eventos especiales, como vacaciones o festividades.

Mejora en la capacidad de recomendación cruzada entre diferentes plataformas de música.

Integración de la retroalimentación en tiempo real del usuario durante la reproducción para ajustar las recomendaciones en el momento.

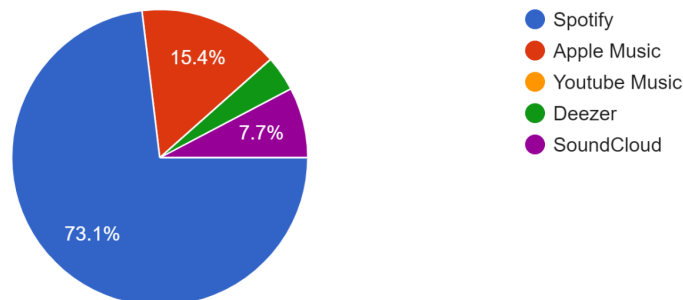
Inclusión de recomendaciones basadas en la actividad física o el ritmo cardíaco del usuario.

Personalización basada en la instrumentación y estructura de las canciones, no solo en el género o el artista.

Incorporación de la capacidad de recomendar música en función del clima o las condiciones meteorológicas.

¿Qué plataformas o aplicaciones de música crees que tienen el mejor sistema de recomendación en la actualidad?

26 respuestas



Dylan Alvarez

Piensa:

Considera que la información detallada de los vehículos es importante para su búsqueda.

Dice:

Expresa que las recomendaciones en plataformas de búsqueda de vehículos son poco acertadas.

Siente:

Se siente insatisfecho con las

Hace:

Propone mejorar la precisión del

recomendaciones actuales.	sistema de recomendación con más opciones de filtros de personalización.
---------------------------	--

Cecilia Chamale	
Piensa: Valora la marca como aspecto importante en la búsqueda de vehículos.	Dice: No está segura si ha utilizado un sistema de recomendación antes.
Siente: Se siente poco satisfecha con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

Dana Valdez	
Piensa: Cree que las recomendaciones precisas y relevantes son clave en un sistema de recomendación.	Dice: Evalúa las recomendaciones actuales como acertadas.
Siente: Se siente satisfecha con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

Telvy Yorleny	
Piensa: Considera que la capacidad de adaptarse a sus preferencias es crucial en un sistema de recomendación.	Dice: Valora la información detallada de los vehículos en las recomendaciones.
Siente: Se siente poco satisfecha con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

Esaú Melgoza	
Piensa: Cree que la marca es un aspecto	Dice: Evalúa las recomendaciones actuales

importante en la búsqueda de vehículos.	como poco acertadas.
Siente: Se siente insatisfecho con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

Gustavo Menéndez	
Piensa: Piensa que la información detallada de los vehículos es esencial en la búsqueda.	Dice: No está seguro si ha utilizado un sistema de recomendación antes.
Siente: Se siente poco satisfecho con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

Diego Gómez	
Piensa: Considera que la marca es un factor relevante en la búsqueda de vehículos.	Dice: No está seguro si ha utilizado un sistema de recomendación antes.
Siente: Se siente insatisfecho con las recomendaciones actuales.	Hace: Sugiere mejorar la precisión del sistema de recomendación con más opciones de filtros de personalización.

b. Definición del problema

Basándonos en el análisis de las respuestas proporcionadas por los usuarios, podemos proponer la siguiente definición del problema:

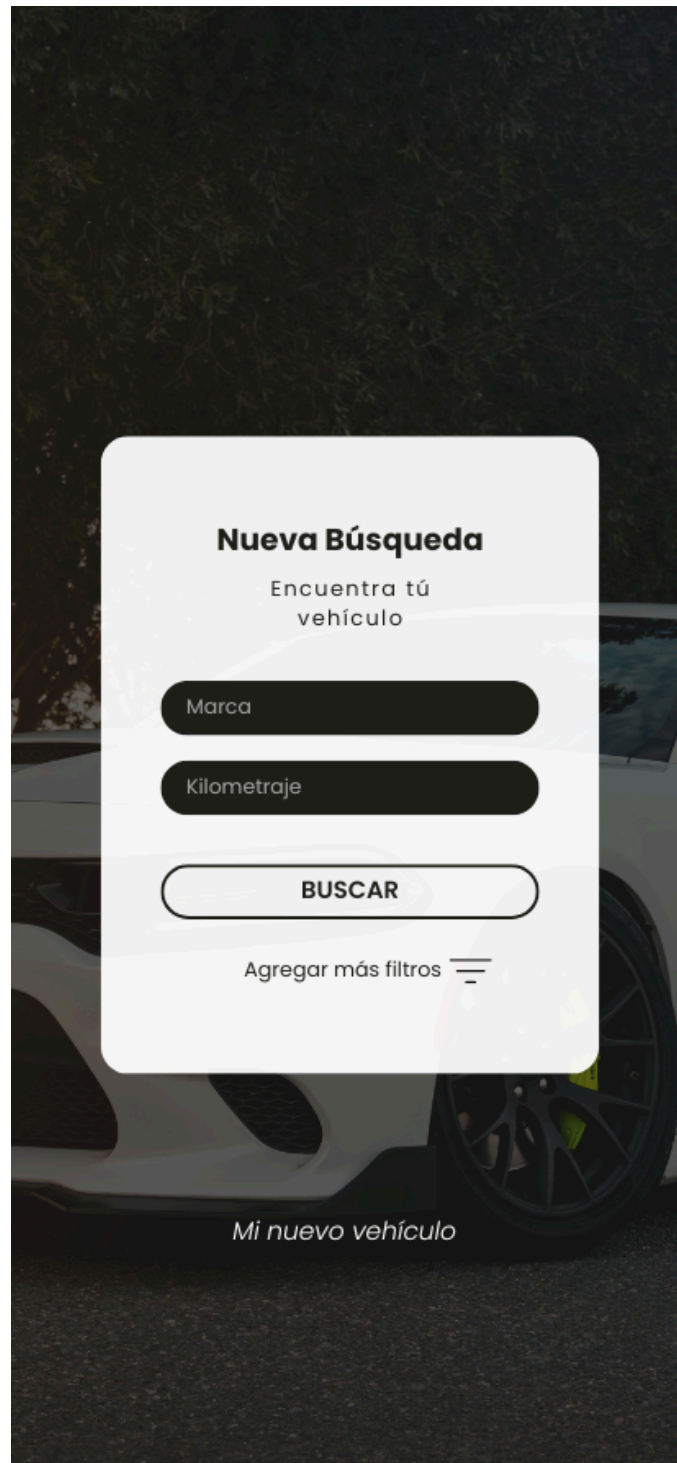
Los usuarios de plataformas de búsqueda de vehículos enfrentan dificultades al buscar información relevante y comparar diferentes opciones. Las recomendaciones actuales en estas plataformas se perciben como poco acertadas y no satisfacen completamente las necesidades y preferencias de los usuarios. Existe una demanda de sistemas de recomendación de vehículos que sean más precisos, personalizados y capaces de adaptarse a las preferencias individuales de

los usuarios. Además, se requiere una interfaz de usuario intuitiva y con información detallada de los vehículos recomendados para mejorar la experiencia de búsqueda y selección de vehículos en línea.

c. Propuesta de ideas de solución del problema

1. **How might we? Mejorar la precisión de las recomendaciones de vehículos.** Utilizar algoritmos de filtrado colaborativo basados en grafos. Estos algoritmos pueden analizar las interacciones entre usuarios y vehículos para generar recomendaciones más personalizadas.
2. **How might we? identificar grupos o comunidades de vehículos similares para realizar la recomendación.** Aplicar algoritmos de propagación de etiquetas en el grafo de vehículos. Esto puede ayudar a mejorar la relevancia de las recomendaciones al agrupar vehículos con características similares.
3. **How might we? Encontrar conexiones entre vehículos que puedan no ser evidentes a simple vista para los usuarios.** Utilizar técnicas para hallar los caminos más cortos en un grafo de recomendación de vehículos. Esto puede ayudar a descubrir opciones alternativas y sugerir vehículos que se ajusten mejor a las preferencias del usuario.
4. **How might we? Mejorar las recomendaciones.** Identificar comunidades de usuarios con preferencias similares y utilizar esta información. Al agrupar usuarios con intereses comunes, el sistema puede ofrecer recomendaciones más precisas y relevantes.
5. **How might we? Ofrecer una interfaz con buena experiencia de usuario.** Diseñar un sistema interactivo y personalizable que permita a los usuarios ver información detallada de los vehículos recomendados, incluyendo descripciones, precios y características técnicas.
6. **How might we? Implementar un sistema de filtrado que tenga en cuenta las visitas recurrentes de un usuario a ciertos vehículos.** Esto puede indicar un interés particular en esos vehículos y ayudar a mejorar las recomendaciones.
7. **How might we? Permitir a los usuarios especificar sus preferencias de manera detallada, como marca, precio, características técnicas, etc.** Utilizar esta información para personalizar las recomendaciones de acuerdo con las preferencias individuales de cada usuario y asignar un peso a cada uno de estos parámetros de búsqueda.
8. **How might we? Mejora en la precisión de las recomendaciones.** Continuar mejorando la precisión del sistema de recomendación mediante la incorporación de más opciones de filtros de personalización y la optimización de los algoritmos utilizados.

d. Prototipo de baja fidelidad:

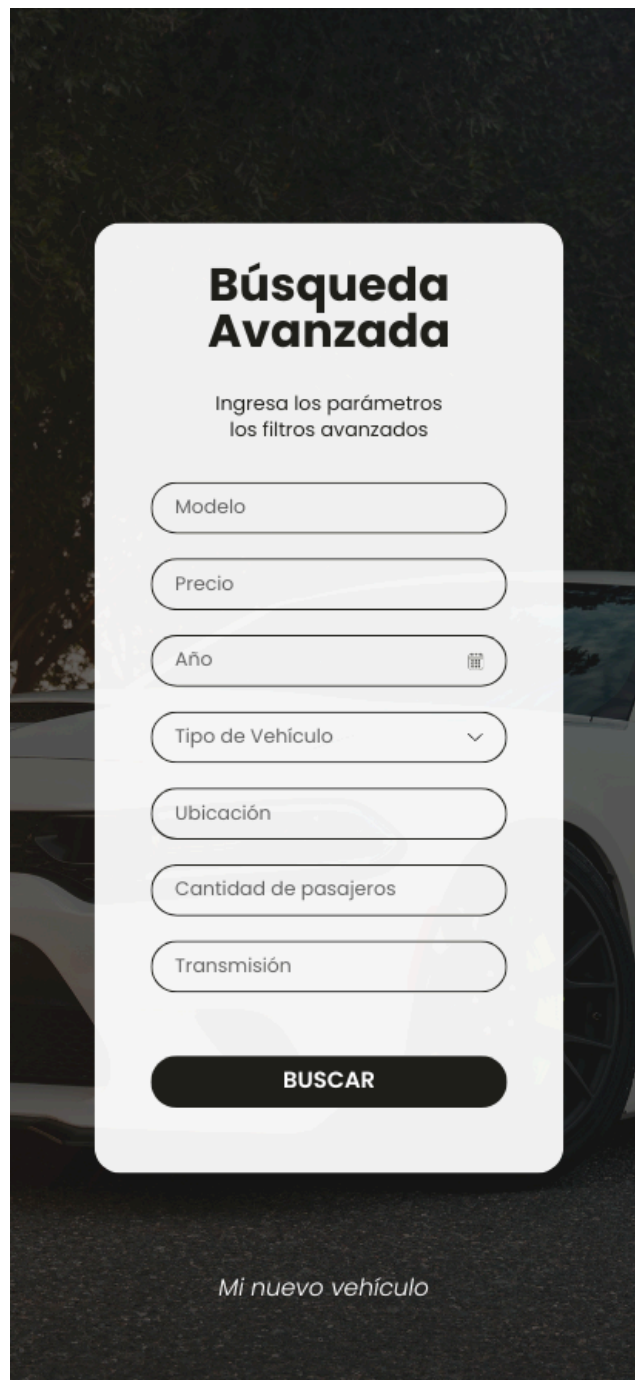


The image shows a low-fidelity prototype of a car search interface. It is a white, rounded rectangle centered over a dark background image of a car's front end. The interface contains the following elements:

- Nueva Búsqueda**: A bold title at the top of the form.
- Encuentra tú vehículo**: A subtitle below the title.
- Marca**: A text input field with a dark background and white text.
- Kilometraje**: A text input field with a dark background and white text.
- BUSCAR**: A button with a white border and dark text.
- Agregar más filtros**: A text label followed by a hamburger menu icon (three horizontal lines).

At the bottom of the image, below the white form, the text *Mi nuevo vehículo* is written in a light, italicized font.

Iniciando la recomendación se solicitarán los parámetros más relevantes según las entrevistas realizadas, la marca y el kilometraje son los aspectos más importantes para los usuarios.

A mobile app interface for an advanced search feature. The background is a dark, blurred image of a car. A white, rounded rectangular card is centered on the screen. At the top of the card, the title 'Búsqueda Avanzada' is written in a bold, black, sans-serif font. Below the title, a subtitle 'Ingresa los parámetros los filtros avanzados' is displayed in a smaller, regular font. The card contains eight input fields, each with a light gray border and rounded corners. The first four fields are 'Modelo', 'Precio', 'Año', and 'Tipo de Vehículo'. The 'Año' field has a small calendar icon on its right side, and the 'Tipo de Vehículo' field has a downward-pointing chevron icon. The next three fields are 'Ubicación', 'Cantidad de pasajeros', and 'Transmisión'. At the bottom of the card is a large, black, rounded rectangular button with the word 'BUSCAR' in white, uppercase letters. Below the card, the text 'Mi nuevo vehículo' is written in a small, italicized font.

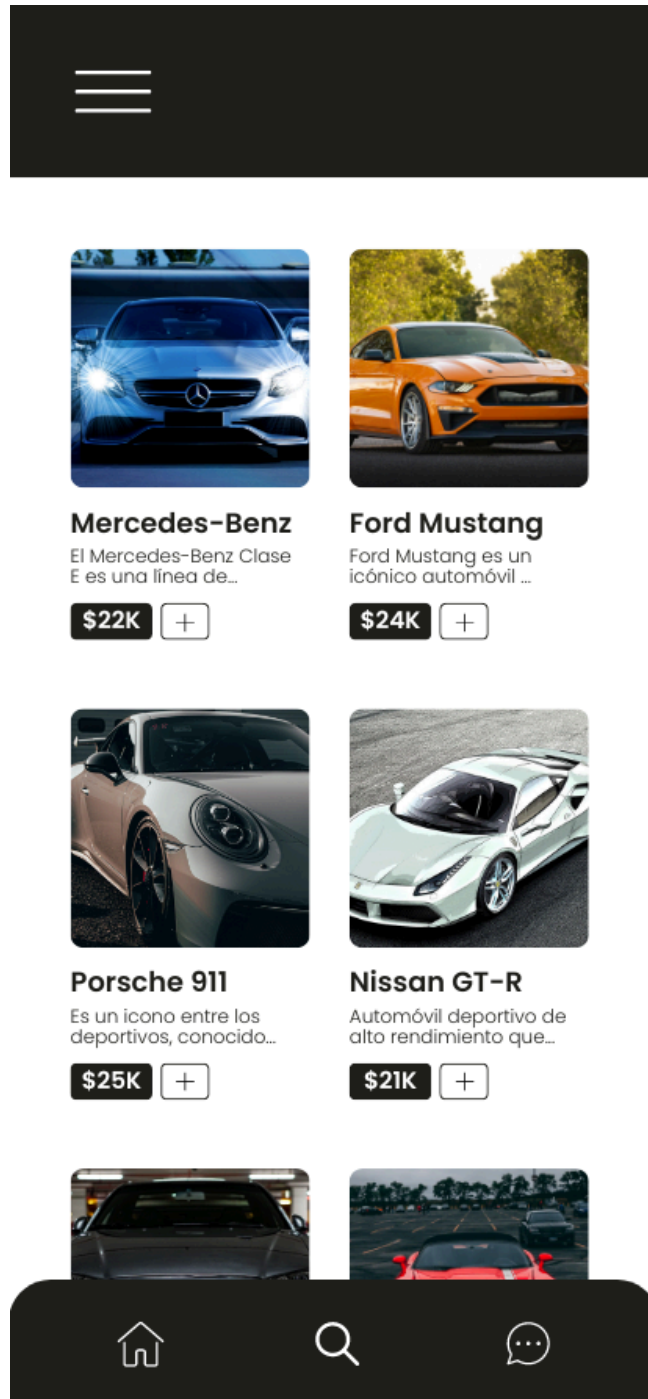
Búsqueda Avanzada

Ingresa los parámetros los filtros avanzados

BUSCAR

Mi nuevo vehículo

Para mejorar la precisión de la recomendación el usuario puede utilizar la búsqueda avanzada según las características definidas en el nodo vehículo y que posteriormente se utilizarán para determinar la conexión entre los vehículos y las preferencias del usuario



Luego de seleccionar los filtros se muestran la lista de vehículos recomendados ordenados por los que tengan mayor relevancia basados en las características de

los usuarios y el historial de conexiones que tenga con otros usuarios que estén interesados en vehículos con características similares.



La información detallada del vehículo aparece una vez el usuario selecciona el vehículo en el que está interesado. Esta relación ayuda a encontrar posibles recomendaciones a otros usuarios que tengan intereses similares y crea conexiones entre los vehículos revisados según los intereses de los usuarios

3. Pseudocódigo del algoritmo:

```
function recommendVehicle(user):  
    // Obtener todos los usuarios conectados al usuario dado
```

```

connectedUsers = getConnectedUsers(user)

// Inicializar un diccionario para almacenar las preferencias
de los usuarios conectados
preferences = {}

// Recorrer todos los usuarios conectados
for each connectedUser in connectedUsers:
    // Obtener las preferencias del usuario conectado
    userPreferences = getUserPreferences(connectedUser)

    // Actualizar las preferencias acumuladas
    for each preference in userPreferences:
        if preference not in preferences:
            preferences[preference] = 0
        preferences[preference] += 1

// Ordenar las preferencias por cantidad de votos en orden
descendente
sortedPreferences = sortPreferencesByVotes(preferences)

// Recorrer las preferencias ordenadas
for each preference in sortedPreferences:
    // Verificar si el usuario actual no ha evaluado esta
preferencia
    if preference not in getUserPreferences(user):
        return preference

// Si el usuario ha evaluado todas las preferencias, devolver
una recomendación aleatoria
return getRandomVehicle()

```

4. Base de datos inicial que soporta la operación del sistema de recomendaciones:

Nodos:

- **Usuario:** Representa a un usuario y puede contener información como el ID del usuario y sus preferencias de vehículos.

- **Vehículo:** Representa un vehículo y puede contener información como el ID del vehículo, la marca, el modelo, el año, el tipo de vehículo y otras características relevantes.

Relaciones:

- **CONEXIÓN:** Una relación entre dos nodos de usuario que indica que están conectados de alguna manera, por ejemplo, son amigos, pertenecen a la misma comunidad, etc.
- **PREFERENCIA:** Una relación entre un nodo de usuario y un nodo de vehículo que indica que el usuario tiene una preferencia por ese vehículo.

Para crear los nodos de usuario y vehículo:

```
CREATE (:Usuario {id: 1, nombre: 'Usuario1'})
CREATE (:Usuario {id: 2, nombre: 'Usuario2'})
CREATE (:Usuario {id: 3, nombre: 'Usuario3'})

CREATE (:Vehiculo {id: 101, marca: 'Toyota', modelo: 'Corolla',
año: 2022, tipo: 'Sedán'})
CREATE (:Vehiculo {id: 102, marca: 'Honda', modelo: 'CR-V', año:
2022, tipo: 'SUV'})
CREATE (:Vehiculo {id: 103, marca: 'Ford', modelo: 'Mustang', año:
2022, tipo: 'Deportivo'})
CREATE (:Vehiculo {id: 104, marca: 'Mercedes-Benz', modelo: 'Clase
E', año: 2022, tipo: 'Sedán'})
```

Para las relaciones de conexión entre usuarios:

```
MATCH (u1:Usuario {id: 1}), (u2:Usuario {id: 2})
CREATE (u1)-[:CONEXIÓN]->(u2)

MATCH (u1:Usuario {id: 1}), (u3:Usuario {id: 3})
CREATE (u1)-[:CONEXIÓN]->(u3)
```

Para las relaciones de preferencia de usuarios por vehículos:

```
MATCH (u1:Usuario {id: 1}), (v1:Vehiculo {id: 101})
CREATE (u1)-[:PREFERENCIA]->(v1)

MATCH (u2:Usuario {id: 2}), (v2:Vehiculo {id: 102})
CREATE (u2)-[:PREFERENCIA]->(v2)

MATCH (u3:Usuario {id: 3}), (v3:Vehiculo {id: 103})
```


CREATE (u3) -[:PREFERENCIA]->(v3)

The screenshot shows the Neo4j workspace interface. On the left, the 'Database information' panel displays 'Nodes (7)' with labels 'Usuario' and 'Vehiculo', 'Relationships (0)', and 'Property keys' including 'año', 'data', 'id', 'marca', 'modelo', 'nombre', 'nodes', 'nombre', 'relationships', 'style', 'tipo', and 'visualisation'. The main query editor shows a successful execution of the query: `CREATE (:Usuario {id: 1, nombre: 'Usuario1'}) CREATE (:Usuario {id: 2, nombre: 'Usuario2'}) CREATE (:Usuario {id: 3})`. A confirmation message states: 'Created 7 nodes, set 26 properties, added 7 labels. Completed after 82ms'.

Pruebas de satisfacción:

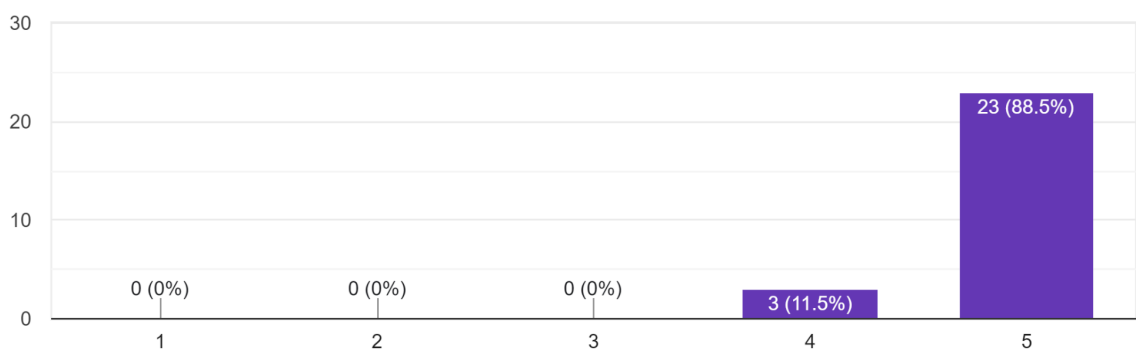
Encuesta realizada para obtener el punteo de satisfacción de los usuarios que utilizaron el programa y sus posibles recomendaciones:

<https://forms.gle/t1P3BJgW8NgeBT3C6>

Resultados:

Púntea tu nivel de satisfacción

26 respuestas



¿Tienes alguna recomendación? Esta pregunta no es obligatoria.

2 respuestas

Tal vez en lo que viene siendo la interfaz con el usuario, eso es algo que podría mejorarse simplemente por razones estéticas.

Sí, agregar una mayor cantidad de opciones sería algo positivo. O etiquetas distintas no relacionadas a carros