**Planorama — Documento de justificación**

**1) Problema y relevancia**

**Necesidad real:** personas en Bogotá quieren decidir qué plan hacer según sus preferencias tiempo presupuesto, edad y si lo desean cercanía

**Objetivo**: construir un agente racional que dadas percepciones del usuario (mensajes en lenguaje natural, ubicación) y un entorno dinámico (listado de eventos) elija acciones (preguntar, filtrar, rankear y recomendar) para maximizar una medida de rendimiento: mostrar planes relevantes que el usuario considere útiles

**Según Russel**: un agente racional actúa para lograr el mejor resultado (o el mejor esperado) con base en sus percepciones, conocimiento y la medida de rendimiento definida por el diseñador.

**2) Enfoque de agente (según Russell & Norvig)**

**Tipo de agente:** agente basado en objetivos/utilidad sencilla que combina:

- **Percepción**: texto del usuario más clic en mapa.

**- Estado interno:** fecha o rango, categorías, gratis, presupuesto, cercanía, edad/, parte del día.

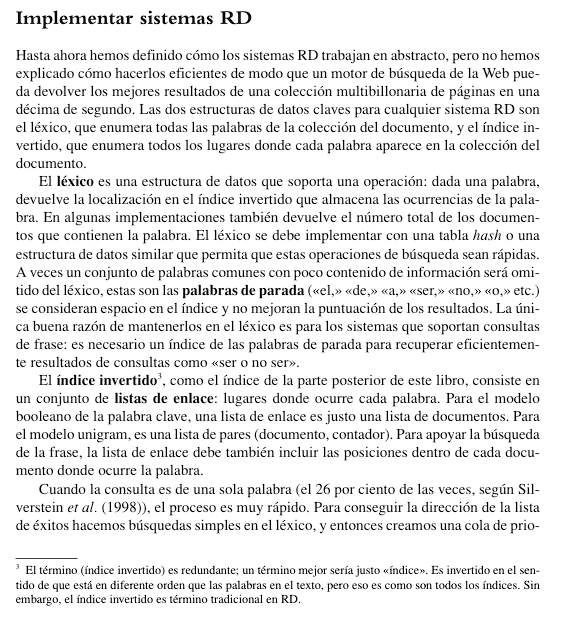
**- Acciones**: (i) preguntar lo mínimo necesario (ii) filtrar (iii) rankear (iv) recomendar Top-N (v) registrar interacción.

**3) ¿Qué datos usamos o recolectamos?**

- Dataset plano (Planorama\_BD.csv): título, descripción, tags, categoría, fecha/hora, lat/lon, precio, is\_free, age\_min, urls, imagen.

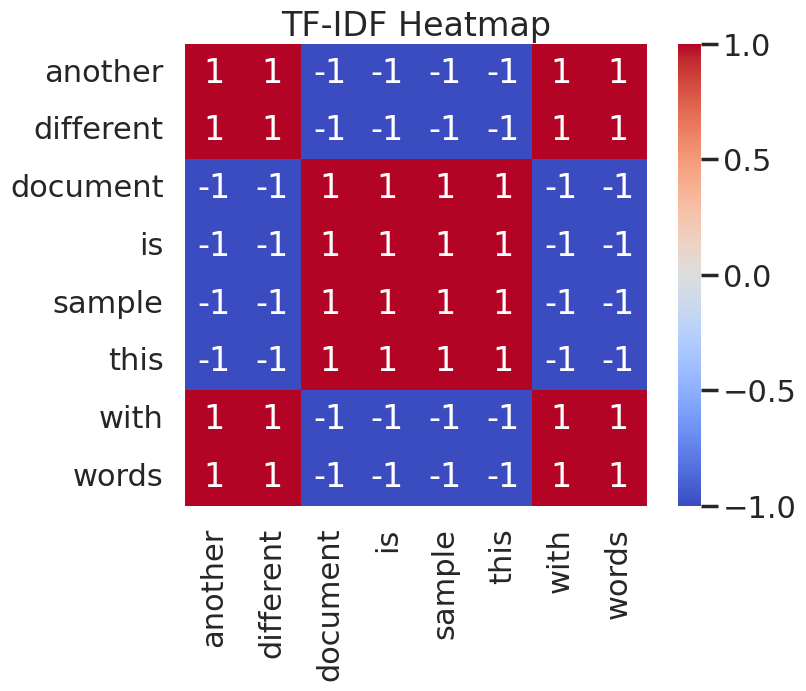
**4) ¿Qué técnica de IA empleamos?**

Recomendación basada en contenido con TF-IDF más similitud del coseno.



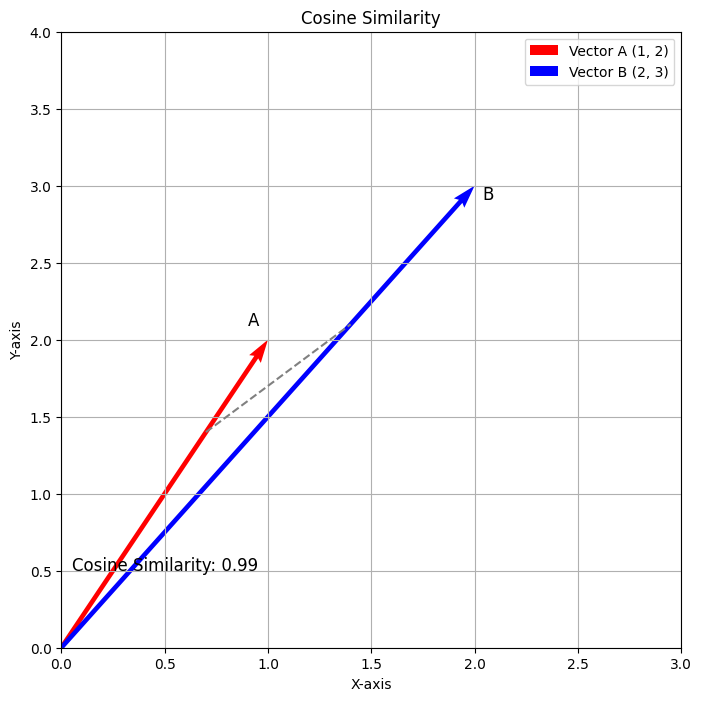
Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.



[Mastering TF-IDF Calculation with Pandas DataFrame in Python - GeeksforGeeks](https://www.geeksforgeeks.org/machine-learning/mastering-tf-idf-calculation-with-pandas-dataframe-in-python/)

Representamos eventos y la consulta derivada del perfil como vectores TF-IDF, comparamos por coseno afinidad semántica.



Añadimos señales de negocio como precio respecto al presupuesto, fecha, parte del día, edad mínima, distancia (Haversine)

**Resultado**: score final = combinación de afinidad de contenido + precio + fecha/horario + distancia = ranking Top-N.

NLP en el bucle comprensión de lenguaje: Usamos Gemini para procesamiento del lenguaje natural (NLP), que transforma lenguaje cotidiano en un JSON normalizado (categorías, rango temporal). *Gemini no recomienda solo entiende y normaliza. La recomendación la hace el motor de contenido + reglas.*

**5) ¿Es Machine Learning?**

Sí, encaja en Machine Learning clásico

1) Representación vectorial y similitud (TF-IDF entrena sus pesos en los eventos es no supervisado).

2) Decisión por cercanía: la idea de distancias/similitudes es central en K-NN y clustering; nuestro clasificador es un ranker por similitud con más datos de interacción

**6) Estrategias de búsqueda/heurísticas**

Aunque no resolvemos un “laberinto” el sistema optimiza sobre candidatos aplicando heurísticas:

- Filtro duro (**factibilidad**): ciudad=BOG, fecha/rango, edad, gratis/pago/presupuesto.

- Función objetivo (**score**): combina señales (contenido, distancia, precio, horario).

- **Selección**: “primero los mejores” por score (búsqueda informada donde la heurística guía hacia los mejores candidatos).

**7) Diseño y desarrollo técnico (resumen)**

- **Agente** (Streamlit): diálogo, estado de perfil, decisión de pedir mapa, render de resultados.

**- Procesamiento del Lenguaje Natural NLP:** Gemini más un perfil estructurado.

- **Core**: TF-IDF + coseno

- **Datos**: Planorama\_BD.csv normalizado (fechas, lat/lon, precios).

- **Salida**: tarjetas Top-N con precio, fecha, ubicación aproximada, enlaces.

**9) Resumen técnico**

Integra agentes racionales, NLP, búsqueda heurística (ranking), aprendizaje clásico (TF-IDF y similitud) y razonamiento práctico (reglas de negocio/edad/fecha).