**¿Qué va a ser Planorama?**

Un recomendador conversacional de planes/eventos en Bogotá. La persona habla en lenguaje natural (“mañana algo de rock cerca de mí, ojalá barato”), el sistema entiende lo que pide, filtra los eventos del dataset y rankea los mejores según afinidad de contenido, precio, fecha/horario y, si el usuario quiere, cercanía geográfica.

¿Cómo lo hace, paso a paso?

Conversación natural

El usuario escribe libremente. Usamos Gemini solo para entender y normalizar lo dicho (categorías, fecha o rango, si quiere gratis/pago, presupuesto, si le importa la cercanía, etc.). Gemini no recomienda; solo transforma texto libre a un formulario estructurado (JSON).

Filtrado “duro” sobre el dataset (CSV)

Con esas preferencias filtramos: ciudad = Bogotá, fecha/rango, categoría, gratis/pago y presupuesto, edad adecuada. Si pidió cercanía, pedimos un clic en el mapa para saber su ubicación aproximada.

Señal de afinidad de contenido (IA clásica)

Convertimos los textos de los eventos (título + descripción + tags) a vectores con TF-IDF y comparamos la consulta del usuario contra cada evento con similitud del coseno. Eso nos da cuán “parecido” es el evento a lo que pidió.

Contexto y reglas de negocio

Sumamos puntajes por precio (que no se pase del presupuesto), parte del día (si pidió mañana/tarde/noche) y distancia (con Haversine, si le importa la cercanía).

Ranking final

Combinamos esas señales en un score y mostramos el Top-N (2–10 según cuántos calcen). Tarjetas con título, fecha, zona, precio en COP, imagen y enlace.

¿Qué técnica de IA estamos utilizando?

Recomendación basada en contenido con TF-IDF + similitud del coseno (más distancia geográfica si aplica).

TF-IDF crea un “vector” para cada evento ponderando las palabras por su importancia en el corpus.

Similitud del coseno mide qué tan alineado está el vector del evento con el vector de la consulta del usuario.

Luego aplicamos reglas (precio, fecha, parte del día, edad, cercanía) para ajustar el orden.

Por qué cuadra con el curso:

Es un enfoque de IA clásica basada en distancias/similitudes. La idea de “parecidos” en un espacio vectorial es el mismo principio que se enseña con KNN o clustering: comparar por distancia/similitud y escoger los más cercanos. Aquí, en vez de “vecinos” de puntos, comparamos consulta vs. eventos y nos quedamos con los más similares.

Dicho simple: metemos los textos en un “mapa” numérico y escogemos lo que queda más cerca de lo que el usuario pidió.

¿Qué “modelo” entrenamos para que funcione?

Entrenamiento ya presente:

TF-IDF se ajusta (fit) al corpus de eventos: aprende qué términos son frecuentes o raros y construye el vocabulario y sus pesos.

No necesita etiquetas ni historial; es un entrenamiento no supervisado/auto-supervisado sobre el texto disponible.

Cálculo en tiempo real:

Tomamos la consulta, la vectorizamos con ese TF-IDF y calculamos coseno con cada evento.

En otras palabras, el “modelo” mínimo sí existe y sí se entrena (el TF-IDF sobre el CSV). No hay una red neuronal ni un árbol complejo; es IA clásica de representación + comparación.

¿Esto es “Machine Learning”, “Matching Learning” u otra cosa?

Es Machine Learning clásico centrado en representación vectorial y medidas de similitud. Está en la misma familia conceptual que KNN (aprendizaje por instancias): no aprendemos reglas complejas, sino que comparamos elementos en un espacio numérico y elegimos los más parecidos.

Si oyeron “Matching Learning” como término informal, aquí el “matching” es precisamente emparejar la consulta con los eventos más similares según TF-IDF + coseno.

¿Cómo aplicamos todo esto con los datos que tenemos?

Fuente: data/Planorama\_BD.csv con columnas como título, descripción, fecha/hora, lat/lon, precio, edad mínima, etc.

Preproceso:

Parseo de fechas/horas y corrección de rangos relativos (“próxima semana/mes”) al futuro correcto.

Normalización de categorías (concierto/teatro/experiencia).

Cálculo de text\_blob (texto combinado) para TF-IDF.

Vectorización: TF-IDF sobre text\_blob.

Consulta: se arma desde lo que normaliza Gemini (p. ej. “concierto rock noche”), se vectoriza con el mismo TF-IDF.

Scoring: similitud del coseno + reglas (precio, parte del día, distancia si hay pin).

Salida: top de eventos con tarjetas.

¿Por qué este enfoque es adecuado ahora?

Frío-inicio: no necesitamos historial de usuarios para funcionar bien desde el día 1.

Interpretabilidad: podemos explicar por qué rankeó alto (coinciden palabras/temas, está dentro del presupuesto, está cerca, cae de noche).

Eficiencia: rápido de implementar y ejecutar.

Escalable a futuro: si luego tenemos clics/guardados, podemos entrenar un ranker supervisado (tipo Learning-to-Rank con LightGBM/XGBoost) usando como features lo que ya calculamos (similitud, distancia, precio, etc.). Pero no es requisito para la versión académica actual.

Resumen en una frase

Planorama es un recomendador conversacional que usa IA clásica basada en similitud: TF-IDF + coseno para encontrar los eventos más parecidos a lo que el usuario describe, y luego ajusta con reglas de negocio (fecha, precio, cercanía, edad) para mostrar los mejores planes de forma clara y natural.