



Introducción a IA

Proyecto #1

David Estevan Rodríguez Jurado

Sergio Felipe Barreto Cifuentes

Juan Esteban Linares

Alejandro Castelblanco

10/03/2025

1. Análisis y diseño de variables, estados y acciones.

1.1 Variables

Usuario

Representa a los clientes que utilizan el sistema de recomendación, elegimos el uso del nombre en vez del uso del identificador, para mejor comprensión del sistema. La función de esta variable en el sistema es identificar a los clientes y almacenar sus interacciones con los productos, para después poder recomendarles otros productos.

Ejemplo en prolog:

usuario(juan).

usuario(maria).

usuario(carlos).

Producto y Categoría

Representa los artículos disponibles en la tienda que pueden ser recomendados y la categoría a la cual pertenece se definen 3 categorías (electrónica, ropa, alimento), se va a representar con el nombre del producto y nombre de la categoría. La función de estas variables en el sistema es permitir la agrupación de productos y generar recomendaciones basadas en la categoría de productos que un usuario ya ha comprado.

Ejemplo en prolog:

producto(laptop, electronica).

producto(camisa, ropa).

producto(chocorrano, alimento).

Compra

Representa el historial de compras de los usuarios, este tendrá el nombre del usuario y el nombre del producto que compró.

La función de esta variable es permitir el análisis de los productos que ha adquirido cada usuario para generar recomendaciones basadas en su historial de compras.

Ejemplo en prolog:

compra(juan, laptop).

compra(maria, camiseta).

Calificación

Representa la puntuación que un usuario otorga a un producto después de comprarlo, es una variable distinta, ya que no siempre que un usuario compre un producto, este lo califica. Va a representar la relación entre el usuario, el producto que compró y la calificación que le dio de 1 a 5. 5 es le gusta 1 es no le gusta para centralizar mejor la información.

La función de esta variable es capturar las preferencias de los usuarios y generar recomendaciones basadas en productos con calificaciones altas.

Ejemplo en prolog:

calificacion(juan, laptop, 5).

calificacion(maria, camiseta, 3).

Similares

Define aquellos productos que sean similares esto con el fin de que la recomendación tenga sentido, para no recomendar productos solo porque pertenecen a la misma categoría sino que tengan algo que ver entre sí.

1.2 Estados

Se define como estado a el conjunto de todas las variables ya presentadas, un ejemplo de estado inicial del sistema en prolog es el siguiente:

```
estado_inicial([
    usuario(juan),
    usuario(maria),
    usuario(carlos),
    producto(laptop, electronica),
    producto(smartphone, electronica),
    producto(zapatos, ropa),
    compra(juan, laptop),
    compra(maria, camiseta),
    calificacion(juan, laptop, 5),
    calificacion(maria, camiseta, 3)
]).
```

1.3 Acciones

Describir todas las acciones del sistema de recomendacion

% Acción: agregar_compra(Usuario, Producto, EstadoActual, NuevoEstado)

agregar_compra(Usuario, Producto, EstadoActual, NuevoEstado) :-

 % Verificar que el usuario y el producto existan member(element,list)

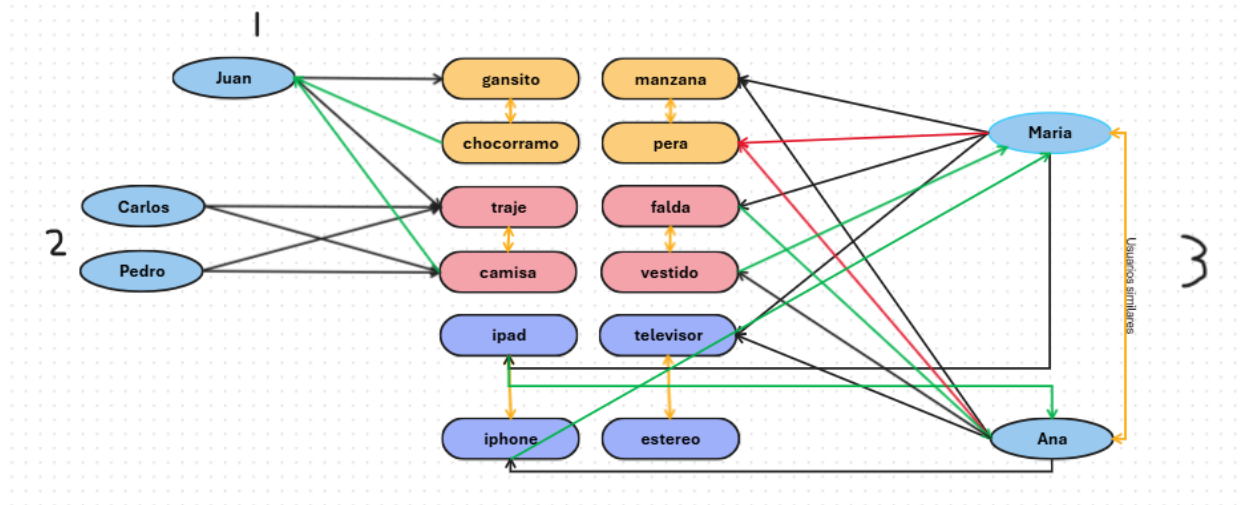
 member(usuario(Usuario), EstadoActual),

 member(producto(Producto, _), EstadoActual),

 % Agregar la nueva compra al estado

 NuevoEstado = [compra(Usuario, Producto) | EstadoActual].

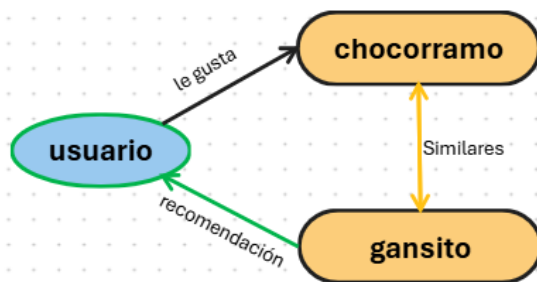
2. Generación del grafo



¿Cómo se va a desarrollar? El sistema va a tener 3 formas distintas para recomendar productos:

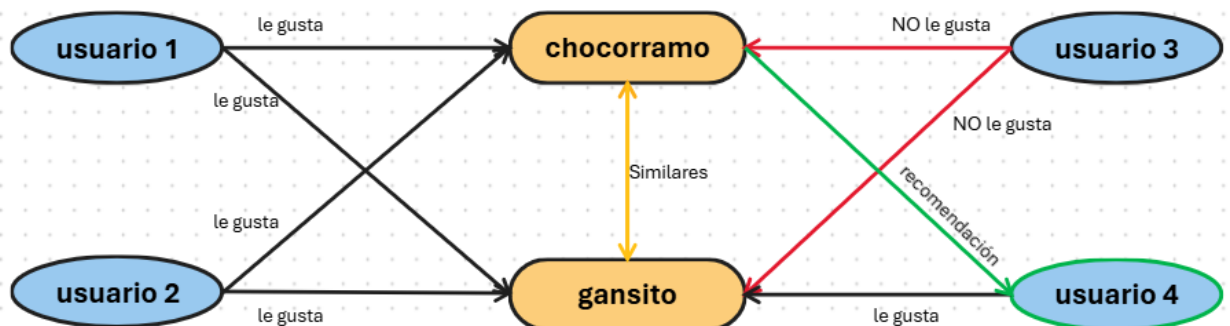
1. Filtros Basados en contenido Query 1

“Tienen como base de la predicción el producto y sus características, es decir, para un usuario concreto se analizan sus gustos o compras mostrando productos de características similares.”



2. Árboles de venta/recomendación cruzada Query 2

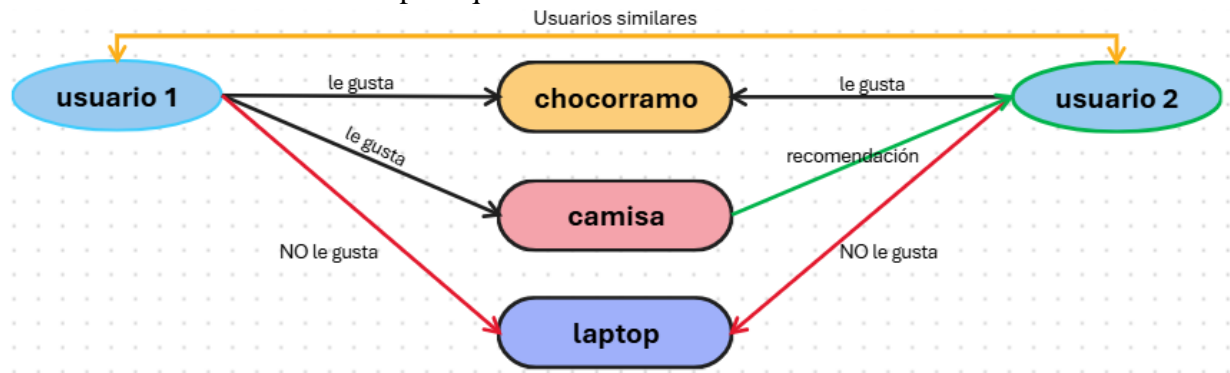
“Su funcionamiento es similar al de los filtros basados en contenido. En este caso, se agrupan los productos que usualmente se compran juntos o que son valorados de manera análoga por los usuarios.”



3. Filtros colaborativos Query 3

“Estos filtros basan su funcionamiento en la información que se tiene acerca de los usuarios. El sistema analiza las compras, gustos o calificaciones de todos los usuarios y los agrupa, empleando dicha información en conjuntos de usuarios similares o con

los mismos gustos. Para un usuario dado, se recomiendan productos que han gustado a usuarios similares pero que éste aún no ha consumido.”



Realización:

- dado un usuario, el sistema le recomienda un ítem:

Query 1:

```
recomendar_contenido(juan, Item).
```

Item = chocorramo
Item = camisa

```
?- recomendar_contenido(juan, Item).
```

Query 2 :

```
%recomendar_contenido(juan, Item). recomendar_cruzada(juan, Item).
```

Item = camisa

Next 10 100 1,000 Stop

```
?- %recomendar_contenido(juan, Item).  
recomendar_cruzada(juan, Item).
```

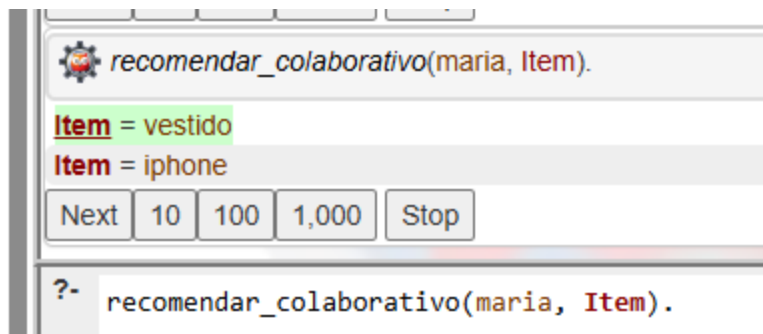
Query 3

```
recomendar_colaborativo(ana, Item).
```

Item = falda
Item = ipad

Next 10 100 1,000 Stop

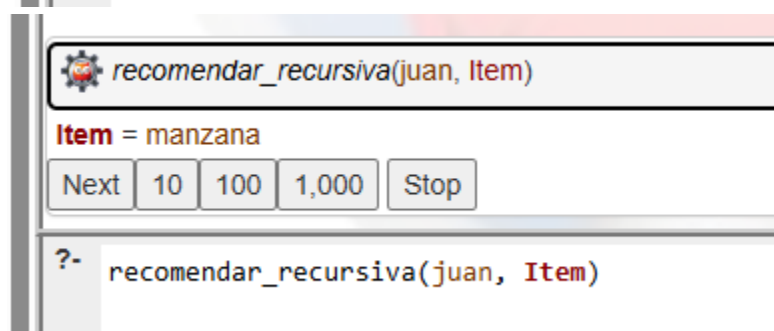
```
?- recomendar_colaborativo(ana, Item).
```



- Dado un usuario, el sistema le recomienda una lista de ítems:
Retorna los productos que un usuario ha calificado con más de 3 estrellas.



- Query que genere una recomendación y que involucre el uso de la recurrencia: Mira que ha comprado el usuario y le recomienda algo de la categoría bien calificado.



- Query que, dado una lista de usuarios, retorne un top 10 de la lista de ítems:

```
top_items_usuarios([juan, maria], Resultados).  
  
Resultados = [(juan,[gansito, traje]), (maria,[falda, ipad, manzana, televisor])]  
  
?- top_items_usuarios([juan, maria], Resultados).
```

Conclusiones

- Las decisiones tomadas durante el diseño del sistema, como representar usuarios y productos mediante hechos y reglas, influyeron directamente en la calidad de las recomendaciones. La definición precisa de categorías y el análisis del historial de compras nos permitieron obtener sugerencias coherentes, pero también descubrimos que un análisis más profundo de las interacciones a lo largo del tiempo podría mejorar aún más la precisión de las recomendaciones.
- El uso de recurrencia en las consultas para generar recomendaciones nos ofreció una forma elegante de manejar listas de usuarios y productos, evitando el recorrido manual de cada elemento.
- Aunque el sistema funcionó adecuadamente con un conjunto limitado de datos, anticipamos que su escalabilidad podría verse comprometida con un mayor número de usuarios y productos.
- Aprendimos que la calidad de las sugerencias depende en gran medida de la precisión de los datos y de la claridad en la definición de las reglas.

Referencias

Ibm. (2025, 21 enero). Topics. IBM. <https://www.ibm.com/topics>

Prolog tutorial. (s. f.). <https://www.tutorialspoint.com/prolog/index.htm>

SWI-Prolog. (s. f.). <https://www.swi-prolog.org/>

[Modelos de recomendación: recomendando qué recomendar | Quanam](#)