

“Taco-ntento”

Un Sistema de Recomendaciones

El tema a desarrollar se basa en el mundo de la gastronomía, una industria enorme en la que todo tipo de personas participan, ya sea buscando nuevas opciones culinarias o regresando a disfrutar de sus platillos favoritos. Nuestro objetivo es crear una aplicación que ofrezca recomendaciones de restaurantes a los usuarios, basándose en sus pedidos anteriores.

El propósito de la aplicación está diseñado para facilitar el proceso de decisión y elección sobre qué comer. A través de ella, los usuarios recibirán opciones que tienen una alta probabilidad de ser de su agrado, todo ello fundamentado en su historial de pedidos. Esto no solo ahorra tiempo al usuario, sino que también enriquece su experiencia gastronómica, incrementando la probabilidad de que elijan comidas que realmente les gusten.

Investigación

Bases de Datos con Grafos

Tipos de filtrado por relación útiles

En un grafo, las relaciones son fundamentales para conectar nodos de usuarios, restaurantes, tipos de comida, reseñas, etc.

1. Filtrado basado en relaciones directas (EXPLICIT FEEDBACK)

Se usa cuando el usuario tiene interacciones registradas como pedidos, calificaciones o favoritos.

Relación: (:Usuario)-[:PIDIÓ]->(:Restaurante)

- Puedes sugerir restaurantes similares a los que el usuario ya pidió.
- También puedes priorizar los que **otros usuarios similares también pidieron**.

Ejemplo de Cypher:

```
MATCH (u:Usuario {nombre: 'Sofía'})-[:PIDIÓ]->(r:Restaurante)<-[:PIDIÓ]-
(otro:Usuario)-[:PIDIÓ]->(sugerencia:Restaurante)
WHERE NOT (u)-[:PIDIÓ]->(sugerencia)
RETURN sugerencia.nombre, count(*) AS relevancia
ORDER BY relevancia DESC
LIMIT 5
```

2. Filtrado basado en contenido (CONTENT-BASED)

Usa propiedades de los restaurantes que coincidan con las preferencias del usuario.

- Si el usuario suele pedir comida saludable, se filtran lugares con tipo: saludable.
- Puedes usar propiedades como tiempo_entrega, precio, calificación, menú_infantil, etc.

Ejemplo de Cypher:

```
MATCH (r:Restaurante)
WHERE r.tipo CONTAINS 'saludable'
  AND r.calificacion >= 4.0
  AND r.tiempo_entrega <= 30
RETURN r.nombre
LIMIT 10
```

3. Filtrado colaborativo basado en relaciones comunes (USER-USER COLLABORATIVE FILTERING)

“Usuarios que pidieron lo mismo que tú también pidieron...”

Relaciones como:

- (:Usuario)-[:PIDIÓ]->(:Restaurante)
- (:Usuario)-[:CALIFICÓ]->(:Restaurante)

Ideal para detectar usuarios "parecidos" a través de patrones de comportamiento.

Ejemplo:

```
MATCH (u:Usuario {nombre: 'Luis'})-[:PIDIÓ]->(r:Restaurante)<-[:PIDIÓ]-
(otro:Usuario)-[:PIDIÓ]->(sugerido:Restaurante)
WHERE NOT (u)-[:PIDIÓ]->(sugerido)
RETURN sugerido.nombre, count(*) AS score
ORDER BY score DESC
```

4. Filtrado geográfico (GEO-FILTERING)

Si tienes ubicación (o zona) en tu modelo, puedes recomendar lugares cerca del usuario.

Relación: (:Usuario)-[:VIVE_EN]->(:Zona) y (:Restaurante)-[:ESTÁ_EN]->(:Zona)

Ejemplo:

```
MATCH (u:Usuario {nombre: 'Camila'})-[:VIVE_EN]->(z:Zona)<-[:ESTÁ_EN]-
(r:Restaurante)
WHERE r.tiempo_entrega <= 20
RETURN r.nombre
```

5. Filtrado por comunidad o segmentación de perfiles

Usa etiquetas como :Estudiante, :Padre, :Nutricionista y genera reglas de negocio.

Esto te permite personalizar más aún usando inferencias como:

- Estudiantes → filtros por precio bajo y entrega rápida.
- Padres → comida familiar, menú infantil.
- Influencers → platillos visuales, lugares nuevos.

Ejemplo:

```
MATCH (u:Usuario {nombre: 'Valeria'})-[:PERTENECE_A]->(:Perfil {tipo:
'Influencer'})
MATCH (r:Restaurante)
WHERE r.nuevo = true AND r.fotos IS NOT NULL AND r.calificacion >= 4
RETURN r.nombre
```

Recomendación híbrida

Puedes combinar varios filtros:

- Relaciones pasadas + tipo de comida + distancia + promociones.
- Ideal para usuarios con poco historial (cold start) o para sorprender con algo nuevo.

Algoritmo de recomendación

(recomendación de lenguaje, ChatGPT)

Para desarrollar un **algoritmo de recomendación** que se conecte con **Neo4j** y esté optimizado para trabajar con **bases de datos de grafos**, te recomiendo **Python**, por varias razones:

Por qué Python es ideal:

1. Excelente soporte para Neo4j:

- Tiene un driver oficial: [neo4j](#)
- También puedes usar la librería py2neo si prefieres una API más sencilla.
- Fácil conexión, envío de queries Cypher y manejo de resultados.

2. Muchas librerías de recomendación y grafos:

- NetworkX: para algoritmos de grafos (PageRank, centrality, etc.)
- scikit-learn: si quieres combinar con modelos clásicos (filtrado colaborativo, clustering).
- Pandas/NumPy: para procesar datos fácilmente.
- graphdatascience: si usas el plugin GDS (Graph Data Science) en Neo4j.

3. Rapidez para prototipar:

- Es muy expresivo, lo cual es útil para la fase de pruebas y diseño iterativo.

4. Facilidad para conectar con interfaces web o APIs:

- Puedes integrarlo con Flask o FastAPI si luego quieres exponer el algoritmo como servicio web.
- También puedes usarlo en notebooks (Jupyter) para testing y análisis.

¿Alternativas?

- **Java o Kotlin:** si estás trabajando con una arquitectura más robusta o backend en Spring, Neo4j tiene muy buen soporte en ese ecosistema también.
- **JavaScript (Node.js):** útil si tu backend ya está en JS, pero tiene menos herramientas para trabajar con algoritmos de grafos en comparación con Python.

Recomendación final:

Usa Python + Neo4j driver + NetworkX

→ Es flexible, poderoso y bien documentado para lo que necesitas.

Además, estaremos usando **virtualenvwrapper** de manera personal para **manejar** de manera limpia todo tipo de **librerías** necesarias para la creación de la aplicación.

Planeación

Opiniones/Entrevistas

Sofía, 22 años, estudiante universitaria

"Normalmente cuando quiero pedir algo para comer, lo primero que pienso es en lo que tengo antojo. Si se me mete que quiero hamburguesa, ya no hay vuelta atrás. Después abro la app y busco los restaurantes que tengan buena calificación. Si veo menos de 4 estrellas, ni lo considero. También me fijo que no tarde tanto, porque a veces ando estudiando y no puedo esperar una hora para comer."

Andrés, 28 años, oficinista

"Yo soy súper práctico. Busco lo que esté cerca y que tenga entrega rápida. Paso de los restaurantes nuevos o que tienen pocas reseñas, porque ya me ha tocado pedir y que llegue frío o mal hecho. Además veo si tienen combos o promociones, porque me gusta ahorrar. No me importa tanto el tipo de comida, mientras sea rápido y no pase de cierto precio."

Mariana, 25 años, creativa freelance

"Para mí comer es toda una experiencia, así que me guío un montón por las fotos del menú. Si veo un platillo que se ve brutal, ya estoy medio convencida. Después leo los comentarios de otros usuarios, sobre todo si dicen si llegó caliente, si el empaque era bueno, esas cosas. No siempre pido lo mismo; me gusta probar lugares nuevos, aunque arriesgue un poco."

Luis, 20 años, estudiante universitario

"Yo sinceramente pido lo más barato que haya. Siempre filtro por 'precio más bajo' o busco cupones antes de pedir. A veces sí me fijo en la calificación, pero si tiene hambre de verdad, hasta un 3.5 estrellas pasa. Prefiero pedir de lugares que ya conozco, porque ya sé qué esperar y no me arriesgo a quedarme con hambre."

Camila, 30 años, ama de casa

"Cuando pido comida, lo primero es que llegue rápido. Con un niño pequeño no puedo estar esperando tanto. Busco opciones que tengan menú infantil o porciones que sé que a él le van a gustar. También prefiero restaurantes que ya haya probado antes o que tengan buenas reseñas sobre la limpieza y presentación de la comida."

Esteban, 24 años, deportista aficionado y oficinista

"Siempre trato de pedir algo relativamente sano, aunque esté en apps de comida rápida. Busco wraps, ensaladas o bowls. Uso los filtros de la app para ver opciones de 'comida saludable'. También reviso si ponen la información nutricional, aunque no todos lo hacen. Me guío poco por el precio, para mí es más importante que no sea comida súper pesada."

Valeria, 27 años, influencer

"Yo casi siempre elijo por tendencia. Si veo que en Instagram todo el mundo está probando un nuevo lugar, me da curiosidad. Busco en la app si está y si tiene buenas fotos, pido. También leo reviews, pero me influye mucho el nombre del restaurante y qué tan 'instagrameable' es la comida. Sí, lo admito."

Pablo, 35 años, ingeniero

"La verdad yo soy bien rutinario. Pido de los mismos 4 o 5 lugares siempre. Comida rápida, confiable y sin sorpresas. Si de verdad quiero variar, me guío por la calificación primero, después veo qué tan lejos está. No me importa tanto el precio, pero sí me importa el tiempo de entrega, que no pase de 30-40 minutos."

Victoria, 19 años, estudiante universitaria

"Normalmente busco lo que esté más conveniente dependiendo de mi localización y cuánto tiempo tengo, como vivo relativamente cerca de la universidad y por ende, cayalá, en donde hay muchos más lugares de comida a los que podría ir"

(Integrante) Esteban, 19 años, estudiante universitario

"No suelo comer en la universidad en días que me quedo muy poco tiempo y cuando lo hago suelo elegir lo más accesible, menos ocupado y más barato que encuentre"

(Integrante) Héctor, 23 años, estudiante universitario

“Depende del ánimo con el que este y la situación. Normalmente, solo compro comida afuera pocas veces y cuando lo hago pensando probar algo que todavía no haya probado. La otra situación recurrente es comprar por emergencia, caso en el que trato de encontrar la mejor oferta de los restaurantes que tenga más cerca, buscando la mayor cantidad por el menor precio posible. Por lo demás, lo que más suelo consumir es café y snacks, ya que de vez en cuando me encanta leer o estudiar en cafeterías.”

Daniela, 26 años, diseñadora gráfica

"Me gusta que la app recuerde mis pedidos anteriores, porque a veces olvido el nombre del restaurante. Si me gustó algo, quiero repetirlo sin perder tiempo buscando. Pero también agradezco cuando me da sugerencias parecidas. No me molesta pagar un poco más si la calidad lo vale."

Carlos, 31 años, profesor de secundaria

"Generalmente pido comida los viernes en la noche, es como mi 'premio' de la semana. Valoro mucho que el restaurante cumpla con el tiempo estimado. Si ya me hicieron esperar una vez más de lo prometido, difícilmente les doy una segunda oportunidad."

Ana Lucía, 21 años, estudiante y barista

"Yo soy muy indecisa, así que la función de 'lo que otros usuarios con tus gustos pidieron' me ayudaría un montón. Me cuesta elegir entre tantas opciones. Siempre leo los comentarios con fotos, porque a veces lo que ves en el menú no es lo que recibes."

Mateo, 33 años, emprendedor en tecnología

"Para mí la experiencia del usuario es clave. Si la app es lenta o me cuesta encontrar lo que busco, me frustra. Me interesan los lugares que tienen menús claros, opciones vegetarianas y buena calificación general. Me fijo en la consistencia de los comentarios."

Isabel, 40 años, contadora y madre de dos hijos

"Cuando pido comida, pienso en toda la familia. La app debería tener filtros por tipo de menú familiar o por porciones grandes. Me gusta que me recomienden opciones que ya han sido bien valoradas por otras familias, no solo por solteros o jóvenes."

Adrián 22 años, estudiante universitario

"No suelo pedir comida rápida pero cuando pido comida suelo escoger que quiero comer en conjunto con mi familia, y solamente pido comida de lugares que he escuchado que son buenos. Por lo general solo mi hermano me recomienda lugares nuevos."

Emilio 19 años, estudiante universitario

"Solamente pido comida unas dos o tres veces por mes, pero yo no decido que comer generalmente mis papas escogen lo que más les antoje, y ellos solo suelen comer en los mismos lugares que siempre comen."

Andy 25 años, Nutricionista

"Yo personalmente pido mucha comida rápida, de lugares saludables si un menú tiene una opción saludable generalmente la voy a escoger, me decido que comer dependiendo de que comí en la mañana. Creo que simplemente no me gusta cocinar"

Design-Thinking (solution-based and user-centric=

Cuadro empatía

Nombre	Que Piensa y siente	Que Ve	Que Oye	Que Dice y hace
Sofía	Valora la rapidez y la buena calificación; se guía por antojos	Calificaciones, tiempo de entrega	Comentarios de otros usuarios	Evita restaurantes con menos de 4 estrellas
Andrés	Busca practicidad, ahorro y rapidez	Promociones, ubicación cercana	Experiencias negativas pasadas	Evita lugares con pocas reseñas
Mariana	Valora la experiencia visual y sensorial de la comida	Fotos del menú	Comentarios sobre empaque, temperatura	Le gusta probar lugares nuevos
Luis	Busca lo más barato posible	Filtros por precio	Calificaciones si está muy hambriento	Pide de lugares conocidos
Camila	Necesita rapidez por tener un hijo pequeño	Menú infantil, porciones adecuadas	Opiniones sobre limpieza y presentación	Prefiere restaurantes ya probados
Esteban	Prefiere opciones saludables y ligeras	Filtros de comida saludable	Información nutricional cuando está disponible	Evita comidas pesadas
Valeria	Se guía por tendencias y estética	Instagram, fotos llamativas en la app	Tendencias populares en redes	Elige lugares aptos para Instagram
Pablo	Valora la rapidez y fiabilidad	Calificación, distancia	Opiniones confiables	Pide de los mismos lugares siempre
Victoria	Busca lo más conveniente según su ubicación	Lugares cercanos a la universidad	Opiniones de estudiantes	Filtra por ubicación y tiempo disponible
Daniela	Le gusta repetir pedidos que le gustaron	Pedidos anteriores, sugerencias similares	Recomendaciones basadas en historial	Valora calidad sobre precio

Definición del problema: Definir

Notamos que un gran problema que tienen los usuarios que entrevistamos es el tiempo que demoran en escoger que quieren comer. Entonces decidimos crear una solución para reducir el tiempo que un consumidor tarda en decidir qué comer.

Idear

Para resolver la problemática pensamos realizar un software de pedidos de comida, que utiliza un algoritmo de recomendación. Pensamos que funcionará, ya que el algoritmo reduce el tiempo de búsqueda al predecir posibles intereses.

Conocer al usuario

Con las entrevistas y el método “Design Thinking” esperamos **descubrir las variables más importantes** que toman en cuenta las personas para decidir qué comer.

Nuestra **hipótesis** es que los siguientes aspectos serán predominantes:

- **Precio**
- **Tipo de comida**

Base de datos

Como el objetivo del proyecto es conocer las ventajas de los grafos, utilizaremos **filtrado por relación: Collaborative-Filtering**.

Hoy en día estos algoritmos usan la mayor cantidad de datos que puedan tener una relación en la toma de decisiones. Sin embargo, nos concentraremos en los siguientes **tipos de relaciones** (las cuales consideramos las **más importantes**):

- **Personas**
 - Similitud Restaurante (50% o más de los restaurantes favoritos es igual)
 - Similitud Pedidos (50% o más de los pedidos anteriores es igual)
- **Restaurantes**
 - Tipo(restaurante con tipos de comida iguales)
 - Precio(rango de precios similar)
- **Personas & Restaurantes**
 - Pedido

Por lo que los **nodos** de la base de datos serán las **personas y restaurantes**.
Habrà una base de datos específica para restaurantes con sus relaciones y una con personas y restaurantes

Por otro lado, las aristas **similitud no** contarán con **dirección**, ya que solo tomamos en cuenta la perspectiva del consumidor y no de los restaurantes, además de que es recíproca entre clientes.

App

Por ahora el prototipo será una aplicación sencilla de consola. Con la cual probaremos la eficiencia del algoritmo. Además de un prototipo de interfaz que estaremos enseñando a voluntarios para recibir feedback. Una vez el algoritmo y interfaz gráfica estén definidos, procederemos a crear el app, la cual prevemos que sea una web-app. Sin embargo, nos abstenemos a decidirlo, ya que según el feedback y consejos que recibamos tomaremos la decisión final.

Testeando la app con usuarios