UNIVERSIDAD DEL VALLE DE GUATEMALA

Colegio Universitario



**Sistema Recomendación**

**Proyecto2**

José André Estrada Contreras - 23659

Angel Esteban Esquit Hernández - 23221

Javier Eduardo España Pacheco - 23361

Algoritmos y Estructuras de datos

Guatemala, 2024

**Investigación**

Un sistema de recomendación es una subclase de sistemas de filtrado de información que buscan predecir la "calificación" o "preferencia" que un usuario daría a un artículo. En esencia, son como cajas negras que analizan un conjunto de usuarios y muestran los elementos que un usuario individual podría disfrutar.

Diferencia entre sistemas de recomendación offline y online:

En el mundo real, las personas que nos rodean pueden considerarse como sistemas de recomendación. Por ejemplo, amigos y familiares pueden sugerir tiendas, estilos de ropa o películas basadas en nuestro gusto personal. En contraste, los sistemas de recomendación en línea, como los de Facebook, Netflix, LinkedIn y Amazon, utilizan algoritmos para ofrecer recomendaciones personalizadas basadas en datos de usuario, como historial de navegación, preferencias y comportamiento en línea.

Tipos de sistemas de recomendación:

* Filtrado colaborativo: Basado en el análisis de comportamientos, actividades o preferencias de los usuarios para predecir sus gustos a partir de la similitud con otros usuarios.
* Filtrado basado en contenido: Utiliza descripciones de los elementos y perfiles de preferencia del usuario para recomendar elementos similares a los que el usuario ha disfrutado previamente.
* Sistemas híbridos: Combina enfoques colaborativos y basados en contenido para mejorar la precisión y superar problemas comunes como el inicio en frío y la escasez de datos.

Bases de datos basadas en grafos

Las bases de datos basadas en grafos son sistemas de gestión de datos que almacenan y manipulan información utilizando estructuras de grafo. En un grafo, los datos se representan como nodos (también llamados vértices) que están conectados entre sí mediante relaciones (también llamadas aristas). Cada nodo puede contener propiedades o atributos que describen la entidad que representa, mientras que las relaciones pueden tener también propiedades que especifican la naturaleza de la conexión entre los nodos.

Este enfoque de modelado de datos es particularmente útil para representar y analizar relaciones complejas entre entidades en un sistema. Las bases de datos basadas en grafos son especialmente eficientes para consultas que implican recorrer y analizar las conexiones entre nodos, lo que las hace ideales para casos de uso como redes sociales, sistemas de recomendación, análisis de redes, detección de fraudes y rutas de navegación.

Algunas características comunes de las bases de datos basadas en grafos incluyen la capacidad de almacenar y consultar grandes volúmenes de datos relacionales, la flexibilidad para modelar relaciones complejas y la eficiencia en consultas que implican navegación a través de múltiples niveles de conexión en la estructura del grafo.

Ejemplos de sistemas que utilizan bases de datos basadas en grafos:

* Neo4j es una de las bases de datos de grafos más populares y ampliamente utilizadas. Se utiliza en una variedad de aplicaciones, incluidos los sistemas de recomendación. Su capacidad para representar relaciones complejas entre datos lo hace ideal para modelar interacciones entre usuarios, elementos recomendados y otros factores relevantes en los sistemas de recomendación.
* Amazon Neptune es un servicio de base de datos de grafos completamente administrado disponible en Amazon Web Services (AWS). Es compatible con varios modelos de datos de grafos y ofrece un alto rendimiento para consultas y transacciones en conjuntos de datos de gran escala. Se puede utilizar para implementar sistemas de recomendación basados en grafos con alta disponibilidad y escalabilidad.
* Microsoft Azure Cosmos DB es un servicio de base de datos multimodal en la nube que admite varios modelos de datos, incluidos los grafos. Ofrece funcionalidades avanzadas para consultas y análisis de datos basados en grafos, lo que lo convierte en una opción viable para implementar sistemas de recomendación que requieran el modelado de relaciones complejas.
* TigerGraph es una plataforma de base de datos y análisis de grafos que se centra en ofrecer un rendimiento y una escalabilidad excepcionales. Se utiliza en aplicaciones que requieren análisis de red, como la recomendación de productos en línea, la detección de fraudes y la gestión de riesgos. Su capacidad para manejar grandes volúmenes de datos y realizar consultas complejas lo convierte en una opción a considerar para sistemas de recomendación basados en grafos.
* Apache Giraph es un sistema de procesamiento de grafos distribuido diseñado para grandes conjuntos de datos. Se utiliza en casos de uso que involucran análisis de redes sociales, recomendaciones personalizadas y modelado de relaciones entre entidades. Su arquitectura distribuida y escalable lo hacen adecuado para implementaciones de sistemas de recomendación a gran escala.

Algoritmos de similitud de conjuntos:

**Coeficiente de Dice**:

El coeficiente de Dice, también conocido como coeficiente de Sørensen-Dice, es similar al coeficiente de Jaccard pero con una ligera variación en la fórmula.

Se utiliza comúnmente en la comparación de muestras biológicas y en la minería de texto para medir la similitud entre dos cadenas de caracteres.

La fórmula del coeficiente de Dice es: Dice(A,B) =

**Coeficiente de Simpson**:

El coeficiente de Simpson es otra medida de similitud entre dos conjuntos, que se define como el tamaño de la intersección dividido por el mínimo de los tamaños de los conjuntos.

Se utiliza en ecología para medir la similitud entre comunidades biológicas y en análisis de diversidad para comparar la riqueza de especies.

La fórmula del coeficiente de Simpson es: Simpson(A,B) =

​**Coeficiente de Tversky**:

El coeficiente de Tversky es una medida generalizada de similitud entre dos conjuntos que permite ajustar el peso dado a los falsos positivos y falsos negativos en la comparación.

Se utiliza en diversas aplicaciones donde se requiere un ajuste fino de la sensibilidad de la similitud, como en la comparación de imágenes médicas y en la identificación de patrones en grandes conjuntos de datos.

La fórmula del coeficiente de Tversky es: Tversky(A,B)=

​**Distancia de Hamming**:

La distancia de Hamming es una medida de la diferencia entre dos cadenas de igual longitud.

Se utiliza en la detección y corrección de errores en la transmisión de datos, en la bioinformática para comparar secuencias genéticas y en la teoría de la información para medir la entropía.

La distancia de Hamming cuenta el número de posiciones en las que los símbolos correspondientes en las dos cadenas son diferentes.

**Distancia de Jaccard**:

La distancia de Jaccard es una medida de disimilitud entre dos conjuntos, que se define como 1 menos el coeficiente de Jaccard.

Se utiliza en problemas donde la presencia o ausencia de elementos en los conjuntos es importante pero no su orden, como en la comparación de documentos y la detección de plagio.

La fórmula de la distancia de Jaccard es: JaccardDistance(A,B) = 1− Jaccard(A,B).

Razones por las que se escogió el algoritmo Jaccard:

* **Flexibilidad y generalidad**: El coeficiente de Jaccard es una medida de similitud que no está limitada por la naturaleza específica de los elementos en los conjuntos. Puede aplicarse a conjuntos de elementos de cualquier tipo, ya sean palabras en un documento, productos en una tienda en línea o usuarios en una red social. Esta flexibilidad lo hace muy versátil y adaptable a una amplia gama de problemas.
* **Sensibilidad a la cardinalidad**: A diferencia de otras medidas de similitud que se ven afectadas por el tamaño absoluto de los conjuntos, el coeficiente de Jaccard es relativamente insensible a la cardinalidad de los conjuntos. Esto significa que funciona bien para conjuntos de diferentes tamaños, lo cual es común en muchos escenarios del mundo real donde los conjuntos pueden variar en tamaño de manera significativa.
* **Interpretación intuitiva**: El coeficiente de Jaccard se interpreta fácilmente. Una similitud de 1 indica conjuntos idénticos, mientras que 0 indica conjuntos completamente diferentes. Esta interpretación intuitiva facilita la comunicación de resultados y su comprensión por parte de los usuarios finales.
* **Eficiencia computacional**: El cálculo del coeficiente de Jaccard es simple y eficiente computacionalmente, lo que lo hace adecuado para aplicaciones donde se requiere procesamiento rápido de grandes volúmenes de datos, como en motores de recomendación, análisis de redes sociales y búsqueda de información.
* **Adaptabilidad a diferentes contextos**: El coeficiente de Jaccard se puede adaptar fácilmente a diferentes contextos y necesidades mediante la modificación de los elementos considerados en los conjuntos o ajustando los criterios de similitud según sea necesario. Esto lo hace altamente adaptable a diferentes escenarios y requisitos específicos del problema.

Encuesta

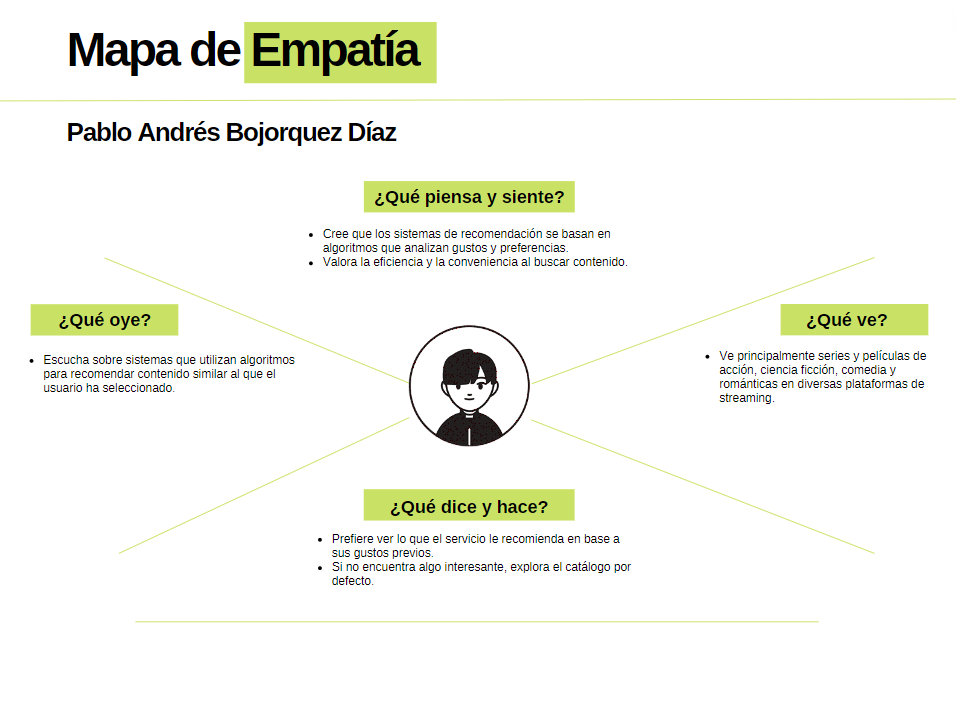
<https://docs.google.com/forms/d/1YT75Ze0EZ6cGuwywNaHJk0uQJVz5DZJ9MQ3u4BKhUWU/edit?hl=ES&hl=ES&hl=ES&no_redirect=true>

Respuestas

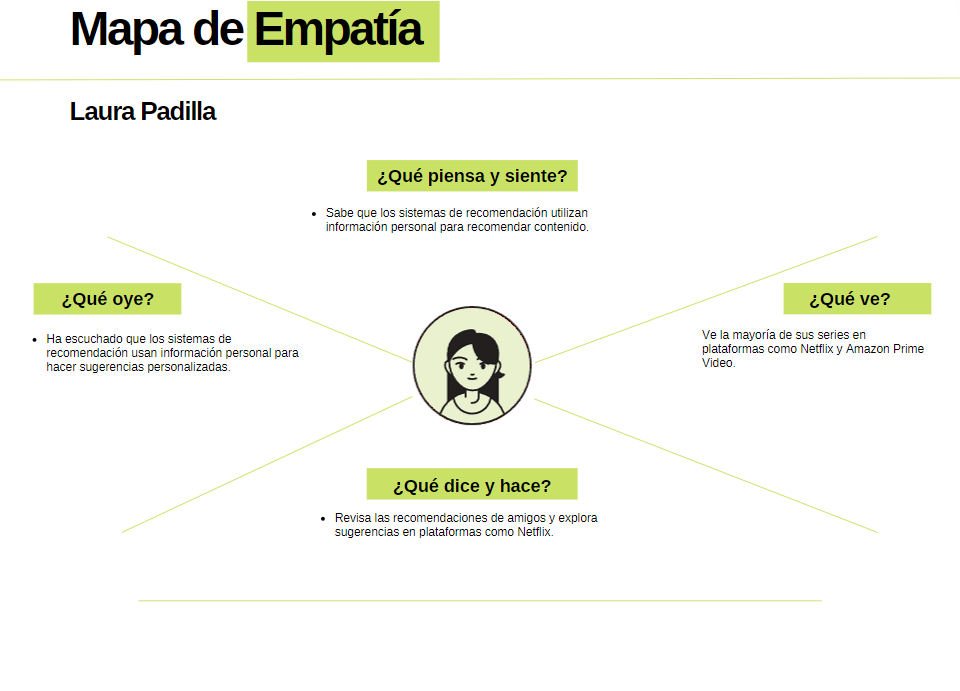
<https://docs.google.com/spreadsheets/d/1dEwZjuEPo8P5X9eExnWT4Z4OqBM5E-iVPX_51ibMwWQ/edit?usp=sharing>

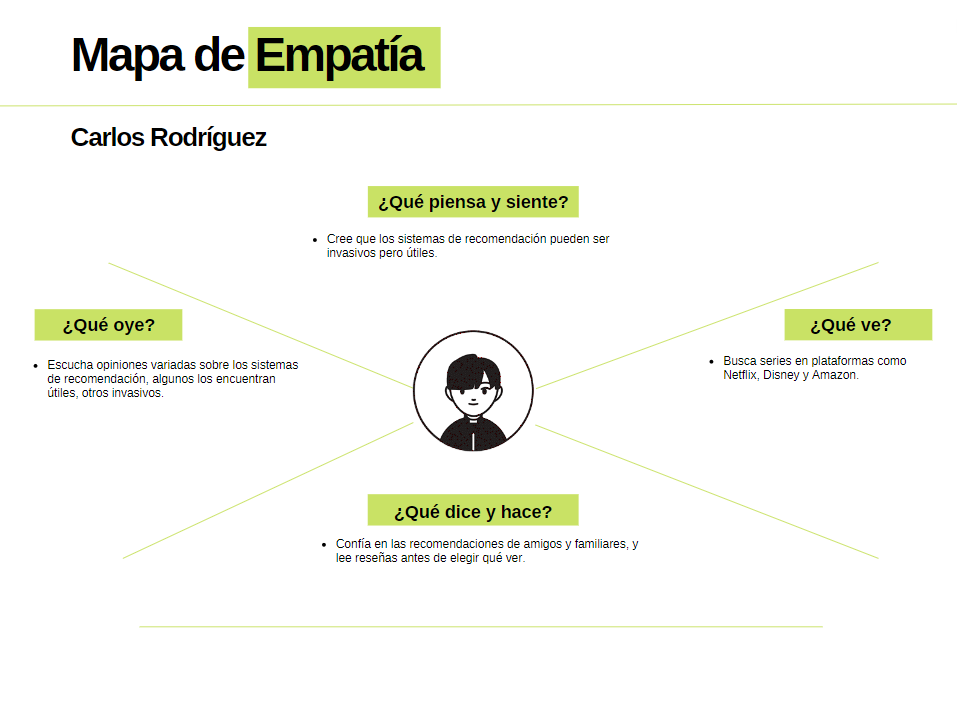
Mapas de Empatía

<https://www.canva.com/design/DAGDY13eV20/FzE6U_pRS6YTu0Ty-Tl36g/edit?utm_content=DAGDY13eV20&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>









Pseudocódigo

**Definición de los datos en el grafo:**

Cada serie de televisión se representa como un nodo en el grafo.

Cada serie tiene atributos como géneros, actores y director.

Las relaciones entre los nodos indican la similitud entre las series en función de estos atributos.

**Obtención de vecinos:**

Dada una serie de televisión, se obtienen sus vecinos en el grafo. Estos son otras series conectadas por relaciones.

**Cálculo de la similitud:**

Para cada par de series (la serie original y un vecino), se calcula la similitud total.

La similitud se calcula por separado para géneros, actores y directores.

Para los géneros y actores, se utiliza la similitud de Jaccard, que compara la intersección y la unión de los conjuntos.

Para los directores, se compara directamente si son iguales o no.

**Promedio de las similitudes:**

Se promedian las similitudes obtenidas para los géneros, actores y directores para obtener la similitud total entre las dos series.

**Resultados:**

Se obtiene un resultado de similitud entre 0 y 1 para cada par de series, donde 0 indica ninguna similitud y 1 indica máxima similitud.

Estos resultados pueden utilizarse para recomendar series similares a una serie dada.

function calcularSimilitud(serieA, serieB):

similitudGeneros = similitudJaccard(serieA.generos, serieB.generos)

similitudActores = similitudJaccard(serieA.actores, serieB.actores)

similitudDirectores = (serieA.director == serieB.director) ? 1 : 0

similitudTotal = (similitudGeneros + similitudActores + similitudDirectores) / 3

retornar similitudTotal

function similitudJaccard(conjuntoA, conjuntoB):

interseccion = tamañoInterseccion(conjuntoA, conjuntoB)

union = tamañoUnion(conjuntoA, conjuntoB)

similitud = interseccion / union

retornar similitud

function tamañoInterseccion(conjuntoA, conjuntoB):

elementosComunes = 0

para cada elemento en conjuntoA:

si elemento está en conjuntoB:

elementosComunes = elementosComunes + 1

retornar elementosComunes

function tamañoUnion(conjuntoA, conjuntoB):

tamañoUnion = tamaño(conjuntoA) + tamaño(conjuntoB) - tamañoInterseccion(conjuntoA, conjuntoB)

retornar tamañoUnion

**Grafo inicial sobre series de televisión:**

Diagrama

Descripción generada automáticamente

Base de datos:

Cada nodo representa una entidad, ya sea una serie de televisión o una persona. Por ejemplo, hay nodos para las series "Stranger Things", "Black Mirror", "The Crown" y "Dark", así como nodos para personas como "Millie Bobby Brown", "Finn Wolfhard", "Charlie Brooker", etc.

Ejemplo de la estructura de los nodos:

CREATE (Nombredelnodo:Nombredeltipodenodo {atributosdelnodo})

Creación de nodos para los géneros de las series

CREATE (Comedia:Genero {nombre:'Comedia'})

CREATE (Terror:Genero {nombre:'Terror'})

CREATE (Drama:Genero {nombre:'Drama'})

CREATE (Misterio:Genero {nombre:'Misterio'})

Creación de los nodos para los actores de las series

CREATE (Millie:Persona {name:'Millie Bobby Brown', born:2004})

CREATE (Finn:Persona {name:'Finn Wolfhard', born:2002})

CREATE (David:Persona {name:'David Harbour', born:1975})

CREATE (Winona:Persona {name:'Winona Ryder', born:1971})

CREATE (Olivia:Persona {name:'Olivia Colman', born:1974})

CREATE (Tobias:Persona {name:'Tobias Menzies', born:1974})

CREATE (Helena:Persona {name:'Helena Bonham Carter', born:1966})

CREATE (Louis:Persona {name:'Louis Hofmann', born:1997})

CREATE (Lisa:Persona {name:'Lisa Vicari', born:1997})

Creación de los nodos para los directores/creadores de las series

CREATE (DufferBros:Persona {name:'Duffer Brothers', born:1984})

CREATE (ShawnLevy:Persona {name:'Shawn Levy', born:1968})

CREATE (Charlie:Persona {name:'Charlie Brooker', born:1971})

CREATE (Peter:Persona {name:'Peter Morgan', born:1963})

CREATE (Jantje:Persona {name:'Jantje Friese', born:1977})

CREATE (Baran:Persona {name:'Baran bo Odar', born:1978})

Creación de los nodos de las series y sus conexiones con los otros nodos

CREATE (StrangerThings:Series {title:'Stranger Things', released:2016, tagline:'Upside Down'})

CREATE (StrangerThings)-[:PERTENECE\_A]->(Drama)

CREATE (StrangerThings)-[:ACTED\_IN {roles:['Eleven']}]->(Millie),

(StrangerThings)-[:ACTED\_IN {roles:['Mike']}]->(Finn),

(StrangerThings)-[:ACTED\_IN {roles:['Hopper']}]->(David),

(StrangerThings)-[:ACTED\_IN {roles:['Joyce']}]->(Winona),

(StrangerThings)-[:DIRECTED]->(DufferBros),

(StrangerThings)-[:DIRECTED]->(ShawnLevy)

CREATE (BlackMirror:Series {title:'Black Mirror', released:2011, tagline:'Technology Nightmare'})

CREATE (BlackMirror)-[:PERTENECE\_A]->(Misterio)

CREATE (BlackMirror)-[:CREATED\_BY]->(Charlie)

CREATE (TheCrown:Series {title:'The Crown', released:2016, tagline:'Royal Drama'})

CREATE (TheCrown)-[:PERTENECE\_A]->(Drama)

CREATE (TheCrown)-[:ACTED\_IN {roles:['Queen Elizabeth II']}]->(Olivia),

(TheCrown)-[:ACTED\_IN {roles:['Prince Philip']}]->(Tobias),

(TheCrown)-[:ACTED\_IN {roles:['Princess Margaret']}]->(Helena),

(TheCrown)-[:CREATED\_BY]->(Peter)

CREATE (Dark:Series {title:'Dark', released:2017, tagline:'Time Travel Mystery'})

CREATE (Dark)-[:PERTENECE\_A]->(Misterio)

CREATE (Dark)-[:ACTED\_IN {roles:['Jonas']}]->(Louis),

(Dark)-[:ACTED\_IN {roles:['Martha']}]->(Lisa),

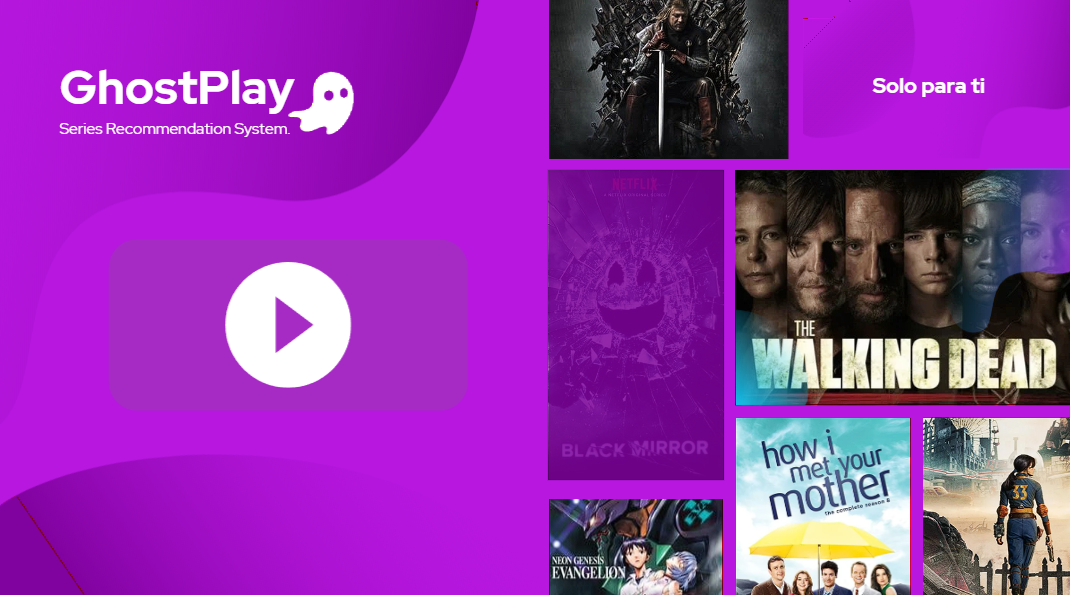
(Dark)-[:CREATED\_BY]->(Baran),

(Dark)-[:CREATED\_BY]->(Jantje)

;

Prototipo Interfaz Grafica

https://www.canva.com/design/DAGDZvESTOs/GnflKEWZMkg2F7JGv6mMOA/edit?utm\_content=DAGDZvESTOs&utm\_campaign=designshare&utm\_medium=link2&utm\_source=sharebutton



Interfaz de usuario gráfica

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

Referencias

Neo4j. (2024, March 18). *NEO4J Graph Database & Analytics | Graph Database Management System*. Graph Database & Analytics. https://neo4j.com/

Dataaspirant. (2016, May 30). An introduction to recommendation engines - Dataconomy. *Dataconomy*. https://dataconomy.com/2015/03/13/an-introduction-to-recommendation-engines/

*Amazon Neptune database*. (n.d.). [Video]. Amazon Web Services, Inc. https://aws.amazon.com/es/neptune/

Seesharprun. (n.d.). *Azure Cosmos DB*. Microsoft Learn. https://learn.microsoft.com/es-es/azure/cosmos-db/

TigerGraph. (2024, April 2). *Graph Analytics Platform | Graph Database | TigerGraph*. https://www.tigergraph.com/

*Giraph - Welcome to Apache Giraph!* (n.d.). https://giraph.apache.org/